# 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

802.11™


## IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

12 June 2007

**IEEE Std 802.11™-2007**
(Revision of
IEEE Std 802.11-1999 )

**IEEE Std 802.11™-2007**
(Revision of
IEEE Std 802.11-1999)

# 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

Sponsor

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

**Approved 8 March 2007**

**IEEE-SA Standards Board**

**Abstract:** This revision specifies technical corrections and clarifications to IEEE Std 802.11 for wireless local area networks (WLANS) as well as enhancements to the existing medium access control (MAC) and physical layer (PHY) functions. It also incorporates Amendments 1 through 8 including a corrigendum.
**Keywords:** 2.4 GHz, 4.9 GHz, 5 GHz, advanced encryption standard, AES, carrier sense multiple access/collision avoidance, CCMP, Counter mode with Cipher-block chaining Message authentication code Protocol, confidentiality, CSMA/CA, DFS, dynamic frequency selection, international roaming, LAN, local area network, MAC, medium access controller, PHY, physical layer, QoS, quality of service, radio frequency, RF, temporal key integrity protocol, TKIP, TPC, transmit power control, wireless LAN, WLAN

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

Copyright © 2007 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 12 June 2007. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Print: ISBN 0-7381-5655-8 SH95708
PDF: ISBN 0-7381-5656-6 SS95708

*No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.*

**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

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

# Introduction

This introduction is not part of IEEE Std 802.11-2007, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area network—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

This revision gives users, in one document, the IEEE 802.11 standard for wireless local area networks (WLANS) with all the amendments that have been published to date. The original standard was published in 1999 and reaffirmed in 2003. The following documents have been rolled into this revision and are, therefore, now retired along with the original, reaffirmed edition of IEEE Std 802.11:

— IEEE Std 802.11a™-1999 (Amendment 1)
— IEEE Std 802.11b™-1999 (Amendment 2)
— IEEE Std 802.11b-1999/Corrigendum 1-2001
— IEEE Std 802.11d™-2001 (Amendment 3)
— IEEE Std 802.11g™-2003 (Amendment 4)
— IEEE Std 802.11h™-2003 (Amendment 5)
— IEEE Std 802.11i™-2004 (Amendment 6)
— IEEE Std 802.11j™-2004 (Amendment 7)
— IEEE Std 802.11e™-2005 (Amendment 8)

In addition, this revision specifies technical corrections and clarifications to IEEE Std 802.11 as well as enhancements to the existing medium access control (MAC) and physical layer (PHY) functions. Such enhancements include improved data link security, codified vendor-specific extensions to the protocol, and incorporated interpretation requests.

## 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 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. A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applications desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standard Department.


Copyright © 2007 IEEE. All rights reserved.

## Participants

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

**Stuart J. Kerry**, *Chair*
**Al Petrick**, *Vice-Chair and Treasurer*
**Harry R. Worstell**, *Vice-Chair*
**Tim Godfrey**, *Secretary*
**Nanci Vogtli**, *Publicity Standing Committee*
**Teik-Kheong Tan**, *Chair, Wireless Next Generation Standing Committee*
**Terry L. Cole** and **Simon Barber**, *Technical Editors*

**Richard H. Paine**, *Chair, Task Group k*
**Bruce P. Kraemer**, *Chair, Task Group n*
**Sheung Li**, *Vice-Chair, Task Group n*
**Lee Armstrong**, *Chair, Task Group p*
**Clint Chaplin**, *Chair, Task Group r*
**Donald E. Eastlake III**, *Chair, Task Group s*
**Charles R. Wright**, *Chair, Task Group t*
**Stephen McCann**, *Chair, Task Group u*
**Pat R. Calhoun**, *Chair, Task Group v*
**Jesse Walker**, *Chair, Task Group w*
**Peter Ecclesine**, *Chair, Contention-Based Protocol Study Group*

When the IEEE 802.11 Working Group approved this revision, Task Group m had the following membership:

**Robert O'Hara**, *Chair*
**Terry L. Cole**, *Editor*

Bernard D. Aboba
Osama S. Aboul-Magd
Santosh P. Abraham
Tomoko Adachi
Jonathan R. Agre
Jon Adams
Carlos H. Aldana
Thomas Alexander
Areg Alimian
Keith Amann
Veera Anantha
Merwyn B. Andrade
Carl F. Andren
Scott Andrews
David C. Andrus
Hidenori Aoki
Tsuguhide Aoki
Michimasa Aramaki
Takashi Aramaki
Sirikiat Lek Ariyavisitakul
Lee R. Armstrong
Larry Arnett
Yusuke Asai
Arthur W. Astrin
Malik Audeh
Geert A. Awater
David Bagby
Michael Bahr
Dennis J. Baker
Ramanathan Balachander
Simon Barber
Richard N. Barnwell
John R. Barr
Kevin M. Barry
Charles R. Bartel
Burak H. Baysal
John L. Benko
Mathilde Benveniste
Don Berry
Nehru Bhandaru
Yogesh B. Bhatt
Bjorn A. Bjerke
Simon Black
Scott Blue
Jan Boer
Herve Bonneville
William M. Brasier
Alistair G. Buttar
Pat R. Calhoun
Nancy Cam-Winget
Necati Canpolat
Bill Carney
Pat Carson
Broady B. Cash
RongFeng Chang
Clint F. Chaplin
Amalavoyal Chari
James Chen
Jeng-Hong Chen
Shiuh Chen
Ye Chen
Yi-Ming Chen
Alexander L. Cheng
Hong Cheng
Greg L. Chesson
Aik Chindapol
Sunghyun Choi
Won-Joon Choi
Liwen Chu
Dong-Ming Chuang
Ken Clements
John T. Coffey
W. Steven Conner
Charles I. Cook
Kenneth Cook
Steven Crowley
Marc de Courville
Rolf J. De Vegt
Sabine Demel
Yoshiharu Doi
Brett L. Douglas
Baris B. Dundar
Chris Durand
Roger P. Durand
Sebastien Dure
Yaron Dycian
Donald E. Eastlake
Peter Ecclesine

Copyright © 2007 IEEE. All rights reserved.

Richard Eckard
Jonathan P. Edney
Bruce Edwards
John Egan
Stephen P. Emeott
Marc Emmelmann
Darwin Engwer
Joseph Epstein
Patrik Eriksson
Mustafa Eroz
Andrew X. Estrada
Christoph Euscher
Stefano M. Faccin
John C. Fakatselis
Lars P. Falk
Steve W. Fantaske
Michael Faulkner
Paul H. Feinberg
Alex Feldman
Matthew J. Fischer
Wayne K. Fisher
Michael D. Foegelle
Brian Ford
Guido Frederiks
Benoit Fremont
Takashi Fukagawa
Hiroshi Furukawa
James Gardner
Monisha Ghosh
James P. K. Gilb
Jeffrey M. Gilbert
Tim Godfrey
Sandesh Goel
Wataru Gohda
Sudheer Grandhi
Gordon P. Gray
Paul K. Gray
Larry Green
Daqing Gu
Srikanth Gumamdi
David Gurevich
Fred Haisch
Robert J. Hall
Neil N. Hamady
Seishi Hanaoka
Christopher J. Hansen
James J. Harford
Daniel N. Harkins
Brian D. Hart
Chris Hartman
Thomas Haslestad
Amer A. Hassan
Vann (William) Hasty
James P. Hauser
Yutaka Hayakawa
Shigenori Hayase
Kevin V. Hayes
Haixiang He
David J. Hedberg
Robert F. Heile
Gregory Scott Henderson
Eleanor Hepworth
Frans M. Hermodsson
Karl F. Heubaum
Odagiri Hideaki
Guido R. Hiertz
Garth D. Hillman
Christopher S. Hinsz
Michael M. Hoghooghi
Allen Hollister
Hooman Honary
William D. Horne
Henry Horng
Yungping A. Hsu
David Hunter
Muhammad Z. Ikram
Daichi Imamura
Yasuhiko Inoue
Kazuhito Ishida
Takashi Ishidoshiro
Takumi Ito
Lakshmi Iyer
Eric A. Jacobsen
Marc Jalfon
KyungHun Jang
Yuh-Ren Jauh
Ho-In J. Jeon
Taehyun Jeon
Jorjeta G. Jetcheva
Lusheng Ji
Yung-Yih Jian
Jari E. Jokela
VK Jones
Bobby Jose
Avinash Joshi
Tyan-Shu Jou
Carl W. Kain
Naveen K. Kakani
Srinivas Kandala
Shantanu Kangude
Jeyhan Karaoguz
Kevin J. Karcz
Pankaj R. Karnik
Assaf Y. Kasher
Masato Kato
Douglas Kavner
Patrick Kelly
Stuart J. Kerry
John W. Ketchum
Ryoji Kido
Tomohiro Kikuma
JinKyeong (Joseph) Kim
Joonsuk Kim
Yongsuk Kim
Youngsoo Kim
Ziv Kimhi
Shigeo Kishimoto
Guenter Kleindl
Jarkko Kneckt
Kiyotaka Kobayashi
Mark M. Kobayashi
Keiichiro Koga
Andrei Kojukhov
Thomas Kolze
George Kondylis
Rahul Kopikar
John M. Kowalski
Bruce P. Kraemer
Jan P. Kruys
Thomas Kuehnel
Rajneesh Kumar
Takushi Kunihiro
Ted Kuo
Thomas M. Kurihara
Denis Kuwahara
Bo Kvarnstrom
Joe Kwak
Edwin Kwon
Paul Lambert
David S. Landeta
Jeremy A. Landt
Joseph P. Lauer
David J. Leach
Dongjun Lee
Insun Lee
Myung J. Lee
Sung-Won Lee
Tae-Jin Lee
Martin Lefkowitz
Uriel Lemberger
Joseph Levy
Scott Leyonhjelm
Jia-Ru Li
Pen Li
Sheung Li
Yuan Li
Haixiang Liang
Jie Liang
Wei Lih Lim
Huashih A. Lin
Albert Liu
Changwen Liu
Der-Zheng Liu
Ed W. Liu
Hang Liu
I-Ru Liu
Jason Liu
Xiaoyu Liu
Yong Liu
Peter Loc
Peter M. Lojko
Hui-Ling Lou
Phil MacKenzie
Doug K. Makishima
Majid M. Malek
Jouni K. Malinen
Mahalingam Mani
Kevin Mankin
William Marshall
Art Martin
Tomoyuki Matsumoto
Yoichi Matsumoto
Ryoko Matsuo
Sudheer Matta
Thomas A. Maufer
Stephen McCann
Kelly P. McClellan
William J. McFarland
Timothy McGovern
Darren P. McNamara


Copyright © 2007 IEEE. All rights reserved.

Justin P. McNew
Irina Medvedev
Pratik Mehta
Mark G. Merrill
Klaus Meyer
Arnaud Meylan
Morgan H. Miki
Robert R. Miller
Seungwook Min
Cimarron Mittelsteadt
Fanny Mlinarsky
Andreas F. Molisch
Michael Montemurro
Rondal J. Moore
Rajendra T. Moorti
Mike Moreton
Yuichi Morioka
Steven A. Morley
Patrick Mourot
Markus D. Muck
Syed Aon Mujtaba
Peter A. Murphy
Peter Murray
Andrew Myles
Yukimasa Nagai
Tetsuya Nakamura
Seigo Nakao
Hiroyuki Nakase
Bhaskar Nallapureddy
Seung Hoon Nam
Sanjiv Nanda
Partha Narasimhan
Slobodan Nedic
Paul D. Newton
Chiu Ngo
Qiang Ni
Gunnar Nitsche
Huaning Niu
Erwin Noble
Richard H. Noens
Ivan F. Oakes
Knut T. Odman
Masakatsu Ogawa
Hiroshi Oguma
Eric J. Ojard
Chandra S. Olson
Timothy S. Olson
Jon O'Nan
Job Oostveen
Satoshi Oyama
Sebnem Z. Ozer
Richard H. Paine
Michael J. Paljug
Stephen Palm
Ashish Pandharipande
Paul W. Panish
Thomas Pare
Jong Ae Park
Steve Parker
Kourosh Parsa
Yaron Peleg
Eldad Perahia
James E. Petranovich
Al Petrick
Fahd Pirzada
Joe Pitarresi
Subbu Ponnuswamy
Neeraj Poojary
Stephen P. Pope
James D. Portaro
Henry Ptasinski
Aleksandar Purkovic
Emily H. Qi
Luke Qian
Jim E. Raab
Ali Raissinia
Ajay Rajkumar
Taori Rakesh
Sridhar Ramesh
Noman Rangwala
Gregg Rasor
Stephen G. Rayment
Ivan Reede
Stanley A. Reible
Joe A. Repice
Edward Reuss
Maximilian Riegel
Eilon Riess
Carlos A. Rios
Randy Roebuck
Justinian Rosca
Jon W. Rosdahl
Michael Rude
Marian X. Rudolf
Bahareh Sadeghi
Emek Sadot
Yousuf Saifullah
Kazuyuki Sakoda
Atul Salhotra
Hemanth Sampath
Anil Sanwalka
Octavian V. Sarca
Toshiyuki Sashihara
Ambatipudi R. Sastry
Monica Saxena
Vincenzo Scarpa
Richard N. Schnacke
Steven D. Schnier
Erik Schylander
Michael Seals
Joe Sensendorf
Vikram Seth
Ariel Sharon
Stephen J. Shellhammer
Ian Sherlock
Matthew J. Sherman
Ming Sheu
Shusaku Shimada
Takashi Shono
William M. Shvodian
D. J. Shyy
Thomas M. Siep
Floyd Simpson
Massimiliano Siti
Roger R. Skidmore
Matt Smith
Kapil Sood
Amjad Soomro
Robert T. Soranno
Ranga Srinivasan
Robert Stacey
R. J. Stancavage
Dorothy Stanley
Martin J. Staszak
William K. Steck
Adrian P. Stephens
Fabrice Stevens
Carl R. Stevenson
Victor J. Stolpman
Guenael T. Strutt
Tsutomu Sugawara
Sumei Sun
Shravan K. Surineni
Hirokazu Tagiri
Eiji Takagi
Masahiro Takagi
Seiichiro Takahashi
Mineo Takai
Daisuke Takeda
Sonal Tambe
Pek-Yew Tan
Teik-Kheong Tan
Hideki Tanaka
Yasuhiro Tanaka
Jeffrey Tao
Clifford Tavares
Stephan Ten Brink
Jerry Thrasher
Eric T. Tokubo
Alexander Tolpin
James D. Tomcik
Timothy P. Towell
Jason Trachewsky
Solomon B. Trainin
Jean Tsao
Rodger Tseng
Chih C. Tsien
Tom Tsoulogiannis
David Tung
Sandra L. Turner
Mike E. Tzamaloukas
Yusuke Uchida
Stefano Valle
Niels Van Erven
Richard van Leeuwen
Richard D. J. Van Nee
Bart Van Poucke
Nico J. van Waes
Allert van Zelst
Fabian Varas
Dmitri Varsanofiev
Dalton T. Victor
Bert Visscher
George A. Vlantis
Nanci Vogtli
Tim Wakeley
Jesse R. Walker
Brad N. Wallace
Vivek Wandile
Huaiyuan Wang

Copyright © 2007 IEEE. All rights reserved.

Christopher G. Ware
Craig D. Warren
Fujio Watanabe
Mati A. Wax
Mark Webster
Bryan Wells
Jim Wendt
Menzo M. Wentink
Filip Weytjens
Stephen R. Whitesell
Richard Williams
James M. Wilson
Jack H. Winters
Jeffrey J. Wojtiuk
Jin Kue Wong
Timothy G. Wong
Marcus Wong
James Woodyatt
Harry R. Worstell
Charles R. Wright
Gang Wu
Ariton Xhafa
Bo Xia
Akiyoshi Yagi
Katsuhiko Yamada
Takeshi Yamamoto
Tomoya Yamaura
Lily Yang
Raziq Yaqub
Huanchun Ye
James Yee
Chi-Hsiang Yeh
Chris Young
Heejung Yu
Hon M. Yung
Erol K. Yurtkuran
Artur Zaks
Eldad Zeira
Jinyun Zhang
Chunhui Zhu
Jeffrey C. Zhu
Juan-Carlos Zuniga
Johnny Zweig
James Zyren

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

Ahmed Abdelhalim
Osama S. Aboulmagd
Tomoko Adachi
Toru Aihara
Keith Amann
Butch Anton
Charles L. Barest
John R. Barr
Hugh Barrass
Roger Berg
Parag D. Bhatt
Gennaro Boggia
Matthew K. Burnburg
Alistair G. Buttar
William A. Byrd
James T. Carlo
Jay Catelli
Clint F. Chaplin
Yi-Ming Chen
Yung-Mu Chen
Mr Aik Chindapol
Keith Chow
Tommy P. Cooper
Javier Del-Prado-Pavon
Kalyan R. Dharanipragada
Thomas J. Dineen
Surinder K. Dureja
Sourav K. Dutta
Peter Ecclesine
Darwin A. Engwer
John W. Fendrich
Michael A. Fischer
Wayne K. Fisher
Andre F. Fournier
Devon L. Gayle
Michael D. Geipel
Fernando Genkuong
Theodore Georgantas
Ian C. Gifford
James P. Gilb
Nikhil Goel
Sergiu R. Goma
Randall C. Groves
Robert M. Grow
C. G. Guy
Siamack Haghighi
David E. Halasz
Samudra E. Haque
Karl F. Heubaum
Shui H. Heung
Werner Hoelzl
Dennis Horwitz
Yasuhiko Inoue
Sergiu A. Iordanescu
Atsushi Ito
Peeya Iwagoshi
Raj Jain
David Johnston
Bobby Jose
Srinivas Kandala
Junghong Kao
Efthymios G. Karabetsos
Kevin J. Karcz
Stuart J. Kerry
Brian G. Kiernan
Yongbum Kim
Patrick W. Kinney
Gunter Kleindl
Thomas M. Kurihara
Jeremy A. Landt
Juan L. Lazaro
Solomon Lee
John Lemon
Daniel G. Levesque
Joseph S. Levy
Jan-Ray Liao
William Lumpkins
G. L. Luri
Kevin D. Marquess
George J. Miao
William J. Mitchell
Jose Morales
Mike Moreton
Ronald G. Murias
Andrew F. Myles
M. Narayanan
Michael S. Newman
Erwin R. Noble
Richard H. Noens
Satoshi Obara
Robert O'hara
Stephen R. Palm
Glenn W. Parsons
Paul W. Piggin
Subburajan Ponnuswamy
Vikram Punj
Maximilian Riegel
Philip T. Robinson
Stephen C. Schwarm
Perry L. Schwartz
William M. Shvodian
Floyd D. Simpson
Jung Je Son
Amjad A. Soomro
Manikantan Srinivasan
Dorothy V. Stanley
Thomas E. Starai
Adrian P. Stephens
Mark A. Sturza
Masahiro Takagi
Sandra L. Turner
Mark-Rene Uchida
Harry R. Worstell
Forrest D. Wright
Oren Yuen
Janusz Zalewski


Copyright © 2007 IEEE. All rights reserved.

When the IEEE-SA Standards Board approved this revision on 8 March 2007, it had the following membership:

**Steve M. Mills,** *Chair*
**Robert M. Grow,** *Vice Chair*
**Don Wright,** *Past Chair*
**Judith Gorman,** *Secretary*

Richard DeBlasio
Julian Forster*
Alex Gelman
William R. Goldbach
Arnold M. Greenspan
Joanna N. Guenin
Kenneth S. Hanus
William B. Hopf
Richard H. Hulett
Hermann Koch
Joseph L. Koepfinger*
John Kulick
David J. Law
Glenn Parsons
Ronald C. Petersen
Tom A. Prevost
Narayanan Ramachandran
Greg Ratta
Robby Robson
Anne-Marie Sahazizian
Virginia C. Sulzberger*
Malcolm V. Thaden
Richard L. Townsend
Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Alan H. Cookson, *NIST Representative*
Virginia C. Sulzberger, *Member/TAB Representative*

Michelle D. Turner
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Copyright © 2007 IEEE. All rights reserved.

# Contents


1. Overview ..... 1

1.1 Scope ..... 1
1.2 Purpose ..... 1

2. Normative references ..... 3

3. Definitions ..... 5

4. Abbreviations and acronyms ..... 17

5. General description ..... 23

5.1 General description of the architecture ..... 23
5.1.1 How WLAN systems are different ..... 23
5.1.1.1 Destination address does not equal destination location ..... 23
5.1.1.2 Media impact on design and performance ..... 23
5.1.1.3 The impact of handling mobile STAs ..... 23
5.1.1.4 Interaction with other IEEE 802® layers ..... 24
5.1.1.5 Interaction with non-IEEE-802 protocols ..... 24
5.2 Components of the IEEE 802.11 architecture ..... 24
5.2.1 The independent BSS (IBSS) as an ad hoc network ..... 25
5.2.2 STA membership in a BSS is dynamic ..... 25
5.2.3 Distribution system (DS) concepts ..... 25
5.2.3.1 Extended service set (ESS): The large coverage network ..... 26
5.2.3.2 RSNA ..... 27
5.2.4 Area concepts ..... 27
5.2.5 Integration with wired LANs ..... 29
5.2.6 QoS BSS: The QoS network ..... 30
5.3 Logical service interfaces ..... 31
5.3.1 SS ..... 31
5.3.2 DSS ..... 32
5.4 Overview of the services ..... 32
5.4.1 Distribution of messages within a DS ..... 33
5.4.1.1 Distribution ..... 33
5.4.1.2 Integration ..... 34
5.4.1.3 QoS traffic scheduling ..... 34
5.4.2 Services that support the distribution service ..... 34
5.4.2.1 Mobility types ..... 34
5.4.2.2 Association ..... 35
5.4.2.3 Reassociation ..... 35
5.4.2.4 Disassociation ..... 36
5.4.3 Access control and data confidentiality services ..... 36
5.4.3.1 Authentication ..... 36
5.4.3.2 Deauthentication ..... 37
5.4.3.3 Data confidentiality ..... 38
5.4.3.4 Key management ..... 38
5.4.3.5 Data origin authenticity ..... 38
5.4.3.6 Replay detection ..... 39
5.4.4 Spectrum management services ..... 39
5.4.4.1 TPC ..... 39



Copyright © 2007 IEEE. All rights reserved.

5.4.4.2 DFS ... 39
5.4.5 Traffic differentiation and QoS support ... 39
5.4.6 Support for higher layer timer synchronization ... 40
5.5 Multiple logical address spaces ... 40
5.6 Differences between ESS and IBSS LANs ... 41
5.7 Reference model ... 42
5.8 IEEE Std 802.11 and IEEE Std 802.1X-2004 ... 42
5.8.1 IEEE 802.11 usage of IEEE Std 802.1X-2004 ... 43
5.8.2 Infrastructure functional model overview ... 43
5.8.2.1 AKM operations with AS ... 43
5.8.2.2 Operations with PSK ... 46
5.8.2.3 Disassociation ... 46
5.8.3 IBSS functional model description ... 46
5.8.3.1 Key usage ... 46
5.8.3.2 Sample IBSS 4-Way Handshakes ... 46
5.8.3.3 IBSS IEEE 802.1X example ... 48
5.8.4 Authenticator-to-AS protocol ... 49
5.8.5 PMKSA caching ... 49

6. MAC service definition ... 51

6.1 Overview of MAC services ... 51
6.1.1 Data service ... 51
6.1.1.1 Determination of UP ... 51
6.1.1.2 Interpretation of priority parameter in MAC service primitives ... 52
6.1.1.3 Interpretation of service class parameter in MAC service primitives in a STA ... 52
6.1.2 Security services ... 53
6.1.3 MSDU ordering ... 53
6.1.4 MSDU format ... 54
6.1.5 MAC data service architecture ... 54
6.2 Detailed service specification ... 55
6.2.1 MAC data services ... 55
6.2.1.1 MA-UNITDATA.request ... 55
6.2.1.2 MA-UNITDATA.indication ... 56
6.2.1.3 MA-UNITDATA.confirm ... 57

7. Frame formats ... 59

7.1 MAC frame formats ... 59
7.1.1 Conventions ... 59
7.1.2 General frame format ... 60
7.1.3 Frame fields ... 60
7.1.3.1 Frame Control field ... 60
7.1.3.2 Duration/ID field ... 64
7.1.3.3 Address fields ... 65
7.1.3.4 Sequence Control field ... 66
7.1.3.5 QoS Control field ... 67
7.1.3.6 Frame Body field ... 70
7.1.3.7 FCS field ... 70
7.1.4 Duration/ID field in data and management frames ... 71
7.2 Format of individual frame types ... 71
7.2.1 Control frames ... 71
7.2.1.1 RTS frame format ... 72

Copyright © 2007 IEEE. All rights reserved.

7.2.1.2 CTS frame format ... 73
7.2.1.3 ACK frame format ... 74
7.2.1.4 PS-Poll frame format ... 74
7.2.1.5 CF-End frame format ... 74
7.2.1.6 CF-End+CF-Ack frame format ... 75
7.2.1.7 Block Ack Request (BlockAckReq) frame format ... 75
7.2.1.8 Block Ack (BlockAck) frame format ... 76
7.2.2 Data frames ... 77
7.2.3 Management frames ... 79
7.2.3.1 Beacon frame format ... 80
7.2.3.2 IBSS ATIM frame format ... 81
7.2.3.3 Disassociation frame format ... 81
7.2.3.4 Association Request frame format ... 82
7.2.3.5 Association Response frame format ... 82
7.2.3.6 Reassociation Request frame format ... 83
7.2.3.7 Reassociation Response frame format ... 83
7.2.3.8 Probe Request frame format ... 84
7.2.3.9 Probe Response frame format ... 84
7.2.3.10 Authentication frame format ... 86
7.2.3.11 Deauthentication ... 86
7.2.3.12 Action frame format ... 87
7.3 Management frame body components ... 87
7.3.1 Fields that are not information elements ... 87
7.3.1.1 Authentication Algorithm Number field ... 87
7.3.1.2 Authentication Transaction Sequence Number field ... 87
7.3.1.3 Beacon Interval field ... 88
7.3.1.4 Capability Information field ... 88
7.3.1.5 Current AP Address field ... 91
7.3.1.6 Listen Interval field ... 91
7.3.1.7 Reason Code field ... 92
7.3.1.8 AID field ... 93
7.3.1.9 Status Code field ... 93
7.3.1.10 Timestamp field ... 95
7.3.1.11 Action field ... 95
7.3.1.12 Dialog Token field ... 96
7.3.1.13 DLS Timeout Value field ... 96
7.3.1.14 Block Ack Parameter Set field ... 97
7.3.1.15 Block Ack Timeout Value field ... 97
7.3.1.16 DELBA Parameter Set field ... 98
7.3.1.17 QoS Info field ... 98
7.3.2 Information elements ... 99
7.3.2.1 SSID element ... 101
7.3.2.2 Supported Rates element ... 101
7.3.2.3 FH Parameter Set element ... 102
7.3.2.4 DS Parameter Set element ... 102
7.3.2.5 CF Parameter Set element ... 103
7.3.2.6 TIM ... 103
7.3.2.7 IBSS Parameter Set element ... 104
7.3.2.8 Challenge Text element ... 105
7.3.2.9 Country information element ... 105
7.3.2.10 Hopping Pattern Parameters information element ... 107
7.3.2.11 Hopping Pattern Table information element ... 108
7.3.2.12 Request information element ... 109
7.3.2.13 ERP Information element ... 109


Copyright © 2007 IEEE. All rights reserved.

7.3.2.14 Extended Supported Rates element ..... 111
7.3.2.15 Power Constraint element ..... 112
7.3.2.16 Power Capability element ..... 112
7.3.2.17 TPC Request element ..... 113
7.3.2.18 TPC Report element ..... 113
7.3.2.19 Supported Channels element ..... 114
7.3.2.20 Channel Switch Announcement element ..... 114
7.3.2.21 Measurement Request element ..... 115
7.3.2.22 Measurement Report element ..... 118
7.3.2.23 Quiet element ..... 122
7.3.2.24 IBSS DFS element ..... 123
7.3.2.25 RSN information element ..... 123
7.3.2.26 Vendor Specific information element ..... 128
7.3.2.27 Extended Capabilities information element ..... 128
7.3.2.28 BSS Load element ..... 129
7.3.2.29 EDCA Parameter Set element ..... 130
7.3.2.30 TSPEC element ..... 132
7.3.2.31 TCLAS element ..... 136
7.3.2.32 TS Delay element ..... 138
7.3.2.33 TCLAS Processing element ..... 138
7.3.2.34 Schedule element ..... 139
7.3.2.35 QoS Capability element ..... 140
7.4 Action frame format details ..... 140
7.4.1 Spectrum management action details ..... 140
7.4.1.1 Measurement Request frame format ..... 141
7.4.1.2 Measurement Report frame format ..... 141
7.4.1.3 TPC Request frame format ..... 142
7.4.1.4 TPC Report frame format ..... 142
7.4.1.5 Channel Switch Announcement frame format ..... 143
7.4.2 QoS Action frame details ..... 143
7.4.2.1 ADDTS Request frame format ..... 143
7.4.2.2 ADDTS Response frame format ..... 144
7.4.2.3 DELTS frame format ..... 145
7.4.2.4 Schedule frame format ..... 146
7.4.3 DLS Action frame details ..... 146
7.4.3.1 DLS Request frame format ..... 146
7.4.3.2 DLS Response frame format ..... 147
7.4.3.3 DLS Teardown frame format ..... 148
7.4.4 Block Ack Action frame details ..... 149
7.4.4.1 ADDBA Request frame format ..... 149
7.4.4.2 ADDBA Response frame format ..... 150
7.4.4.3 DELBA frame format ..... 150
7.4.5 Vendor-specific action details ..... 151
7.5 Frame usage ..... 151

8. Security ..... 155

8.1 Framework ..... 155
8.1.1 Security methods ..... 155
8.1.2 RSNA equipment and RSNA capabilities ..... 155
8.1.3 RSNA establishment ..... 155
8.1.4 RSNA PeerKey Support ..... 156
8.1.5 RSNA assumptions and constraints ..... 157
8.2 Pre-RSNA security methods ..... 157

Copyright © 2007 IEEE. All rights reserved.

8.2.1 Wired equivalent privacy (WEP) ... 158
8.2.1.1 WEP overview ... 158
8.2.1.2 WEP MPDU format ... 158
8.2.1.3 WEP state ... 158
8.2.1.4 WEP procedures ... 159
8.2.2 Pre-RSNA authentication ... 161
8.2.2.1 Overview ... 161
8.2.2.2 Open System authentication ... 161
8.2.2.3 Shared Key authentication ... 162
8.3 RSNA data confidentiality protocols ... 165
8.3.1 Overview ... 165
8.3.2 Temporal Key Integrity Protocol (TKIP) ... 165
8.3.2.1 TKIP overview ... 165
8.3.2.2 TKIP MPDU formats ... 168
8.3.2.3 TKIP MIC ... 169
8.3.2.4 TKIP countermeasures procedures ... 171
8.3.2.5 TKIP mixing function ... 175
8.3.2.6 TKIP replay protection procedures ... 179
8.3.3 CTR with CBC-MAC Protocol (CCMP) ... 179
8.3.3.1 CCMP overview ... 179
8.3.3.2 CCMP MPDU format ... 180
8.3.3.3 CCMP cryptographic encapsulation ... 181
8.3.3.4 CCMP decapsulation ... 184
8.4 RSNA security association management ... 185
8.4.1 Security associations ... 185
8.4.1.1 Security association definitions ... 185
8.4.1.2 Security association life cycle ... 187
8.4.2 RSNA selection ... 189
8.4.3 RSNA policy selection in an ESS ... 189
8.4.3.1 TSN policy selection in an ESS ... 190
8.4.4 RSNA policy selection in an IBSS ... 190
8.4.4.1 TSN policy selection in an IBSS ... 191
8.4.5 RSN management of the IEEE 802.1X Controlled Port ... 191
8.4.6 RSNA authentication in an ESS ... 192
8.4.6.1 Preauthentication and RSNA key management ... 192
8.4.6.2 Cached PMKSAs and RSNA key management ... 193
8.4.7 RSNA authentication in an IBSS ... 193
8.4.8 RSNA key management in an ESS ... 195
8.4.9 RSNA key management in an IBSS ... 195
8.4.10 RSNA security association termination ... 196
8.5 Keys and key distribution ... 196
8.5.1 Key hierarchy ... 196
8.5.1.1 PRF ... 197
8.5.1.2 Pairwise key hierarchy ... 198
8.5.1.3 Group key hierarchy ... 200
8.5.1.4 PeerKey key hierarchy ... 201
8.5.2 EAPOL-Key frames ... 202
8.5.2.1 EAPOL-Key frame notation ... 210
8.5.3 4-Way Handshake ... 211
8.5.3.1 4-Way Handshake Message 1 ... 211
8.5.3.2 4-Way Handshake Message 2 ... 213
8.5.3.3 4-Way Handshake Message 3 ... 214
8.5.3.4 4-Way Handshake Message 4 ... 215
8.5.3.5 4-Way Handshake implementation considerations ... 216


Copyright © 2007 IEEE. All rights reserved.

8.5.3.6 Sample 4-Way Handshake ..... 217
8.5.3.7 4-Way Handshake analysis ..... 218
8.5.4 Group Key Handshake ..... 220
8.5.4.1 Group Key Handshake Message 1 ..... 220
8.5.4.2 Group Key Handshake Message 2 ..... 221
8.5.4.3 Group Key Handshake implementation considerations ..... 222
8.5.4.4 Sample Group Key Handshake ..... 222
8.5.5 RSNA Supplicant key management state machine ..... 223
8.5.5.1 Supplicant state machine states ..... 223
8.5.5.2 Supplicant state machine variables ..... 224
8.5.5.3 Supplicant state machine procedures ..... 224
8.5.5.4 Supplicant PeerKey state machine states ..... 227
8.5.5.5 Supplicant PeerKey state machine variables ..... 227
8.5.6 RSNA Authenticator key management state machine ..... 229
8.5.6.1 Authenticator state machine states ..... 229
8.5.6.2 Authenticator state machine variables ..... 233
8.5.6.3 Authenticator state machine procedures ..... 234
8.5.7 Nonce generation ..... 234
8.5.8 PeerKey Handshake ..... 235
8.5.8.1 SMK Handshake ..... 236
8.5.8.2 PeerKey setup and handshake error conditions ..... 241
8.5.8.3 STKSA rekeying ..... 241
8.5.8.4 Error Reporting ..... 242
8.6 Mapping EAPOL keys to IEEE 802.11 keys ..... 244
8.6.1 Mapping PTK to TKIP keys ..... 244
8.6.2 Mapping GTK to TKIP keys ..... 244
8.6.3 Mapping PTK to CCMP keys ..... 244
8.6.4 Mapping GTK to CCMP keys ..... 244
8.6.5 Mapping GTK to WEP-40 keys ..... 244
8.6.6 Mapping GTK to WEP-104 keys ..... 244
8.7 Per-frame pseudo-code ..... 245
8.7.1 WEP frame pseudo-code ..... 245
8.7.2 RSNA frame pseudo-code ..... 246
8.7.2.1 Per-MSDU Tx pseudo-code ..... 246
8.7.2.2 Per-MPDU Tx pseudo-code ..... 247
8.7.2.3 Per-MPDU Rx pseudo-code ..... 248
8.7.2.4 Per-MSDU Rx pseudo-code ..... 249

9. MAC sublayer functional description ..... 251

9.1 MAC architecture ..... 251
9.1.1 DCF ..... 251
9.1.2 PCF ..... 252
9.1.3 Hybrid coordination function (HCF) ..... 252
9.1.3.1 HCF contention-based channel access (EDCA) ..... 252
9.1.3.2 HCF controlled channel access (HCCA) ..... 254
9.1.4 Combined use of DCF, PCF, and HCF ..... 254
9.1.5 Fragmentation/defragmentation overview ..... 255
9.1.6 MAC data service ..... 255
9.2 DCF ..... 256
9.2.1 CS mechanism ..... 257
9.2.2 MAC-Level acknowledgments ..... 257
9.2.3 IFS ..... 258
9.2.3.1 SIFS ..... 258

Copyright © 2007 IEEE. All rights reserved.

9.2.3.2 PIFS ... 259
9.2.3.3 DIFS ... 259
9.2.3.4 AIFS ... 259
9.2.3.5 EIFS ... 259
9.2.4 Random backoff time ... 260
9.2.5 DCF access procedure ... 261
9.2.5.1 Basic access ... 261
9.2.5.2 Backoff procedure for DCF ... 262
9.2.5.3 Recovery procedures and retransmit limits ... 263
9.2.5.4 Setting and resetting the NAV ... 264
9.2.5.5 Control of the channel ... 264
9.2.5.6 RTS/CTS usage with fragmentation ... 266
9.2.5.7 CTS procedure ... 267
9.2.6 Individually addressed MPDU transfer procedure ... 267
9.2.7 Broadcast and multicast MPDU transfer procedure ... 268
9.2.8 ACK procedure ... 268
9.2.9 Duplicate detection and recovery ... 269
9.2.10 DCF timing relations ... 270
9.2.11 NAV distribution ... 271
9.2.12 Determination of PLME aCWmin characteristics ... 271
9.3 PCF ... 271
9.3.1 CFP structure and timing ... 272
9.3.2 PCF access procedure ... 274
9.3.2.1 Fundamental access ... 274
9.3.2.2 NAV operation during the CFP ... 274
9.3.3 PCF transfer procedure ... 275
9.3.3.1 PCF transfers when the PC STA is transmitter or recipient ... 275
9.3.3.2 Operation with overlapping point-coordinated BSSs ... 277
9.3.3.3 CFPMaxDuration limit ... 277
9.3.3.4 CF usage rules ... 277
9.3.4 CF polling list ... 278
9.3.4.1 Polling list processing ... 278
9.3.4.2 Polling list update procedure ... 278
9.4 Fragmentation ... 279
9.5 Defragmentation ... 279
9.6 Multirate support ... 280
9.6.1 Modulation classes ... 281
9.7 MSDU transmission restrictions ... 282
9.8 Operation across regulatory domains ... 282
9.8.1 Operation upon entering a regulatory domain ... 283
9.8.2 Support for FH PHYs ... 283
9.8.2.1 Determination of hopping patterns ... 283
9.9 HCF ... 286
9.9.1 HCF contention-based channel access (EDCA) ... 286
9.9.1.1 Reference implementation ... 286
9.9.1.2 EDCA TXOPs ... 287
9.9.1.3 Obtaining an EDCA TXOP ... 287
9.9.1.4 Multiple frame transmission in an EDCA TXOP ... 289
9.9.1.5 EDCA backoff procedure ... 290
9.9.1.6 Retransmit procedures ... 291
9.9.2 HCCA ... 292
9.9.2.1 HCCA procedure ... 292
9.9.2.2 TXOP structure and timing ... 295
9.9.2.3 HCCA transfer rules ... 296


Copyright © 2007 IEEE. All rights reserved.

9.9.3 Admission Control at the HC ... 298
9.9.3.1 Contention-based admission control procedures ... 298
9.9.3.2 Controlled-access admission control ... 300
9.10 Block Acknowledgment (Block Ack) ... 302
9.10.1 Introduction ... 302
9.10.2 Setup and modification of the Block Ack parameters ... 303
9.10.3 Data and acknowledgment transfer ... 303
9.10.4 Receive buffer operation ... 306
9.10.5 Teardown of the Block Ack mechanism ... 306
9.11 No Acknowledgment (No Ack) ... 306
9.12 Frame exchange sequences ... 306
9.13 Protection mechanism for non-ERP receivers ... 310

10. Layer management ... 313

10.1 Overview of management model ... 313
10.2 Generic management primitives ... 314
10.3 MLME SAP interface ... 314
10.3.1 Power management ... 315
10.3.1.1 MLME-POWERMGT.request ... 315
10.3.1.2 MLME-POWERMGT.confirm ... 316
10.3.2 Scan ... 317
10.3.2.1 MLME-SCAN.request ... 317
10.3.2.2 MLME-SCAN.confirm ... 318
10.3.3 Synchronization ... 320
10.3.3.1 MLME-JOIN.request ... 320
10.3.3.2 MLME-JOIN.confirm ... 321
10.3.4 Authenticate ... 322
10.3.4.1 MLME-AUTHENTICATE.request ... 322
10.3.4.2 MLME-AUTHENTICATE.confirm ... 323
10.3.4.3 MLME-AUTHENTICATE.indication ... 324
10.3.4.4 MLME-AUTHENTICATE.response ... 325
10.3.5 Deauthenticate ... 326
10.3.5.1 MLME-DEAUTHENTICATE.request ... 326
10.3.5.2 MLME-DEAUTHENTICATE.confirm ... 327
10.3.5.3 MLME-DEAUTHENTICATE.indication ... 328
10.3.6 Associate ... 329
10.3.6.1 MLME-ASSOCIATE.request ... 329
10.3.6.2 MLME-ASSOCIATE.confirm ... 330
10.3.6.3 MLME-ASSOCIATE.indication ... 331
10.3.6.4 MLME-ASSOCIATE.response ... 332
10.3.7 Reassociate ... 333
10.3.7.1 MLME-REASSOCIATE.request ... 333
10.3.7.2 MLME-REASSOCIATE.confirm ... 334
10.3.7.3 MLME-REASSOCIATE.indication ... 335
10.3.7.4 MLME-REASSOCIATE.response ... 336
10.3.8 Disassociate ... 337
10.3.8.1 MLME-DISASSOCIATE.request ... 337
10.3.8.2 MLME-DISASSOCIATE.confirm ... 338
10.3.8.3 MLME-DISASSOCIATE.indication ... 339
10.3.9 Reset ... 340
10.3.9.1 MLME-RESET.request ... 340
10.3.9.2 MLME-RESET.confirm ... 341
10.3.10 Start ... 342

Copyright © 2007 IEEE. All rights reserved.

10.3.10.1 MLME-START.request ..... 342
10.3.10.2 MLME-START.confirm ..... 344
10.3.11 Spectrum management protocol layer model ..... 345
10.3.12 Measurement request ..... 350
10.3.12.1 MLME-MREQUEST.request ..... 350
10.3.12.2 MLME-MREQUEST.confirm ..... 351
10.3.12.3 MLME-MREQUEST.indication ..... 352
10.3.13 Channel measurement ..... 353
10.3.13.1 MLME-MEASURE.request ..... 353
10.3.13.2 MLME-MEASURE.confirm ..... 354
10.3.14 Measurement report ..... 355
10.3.14.1 MLME-MREPORT.request ..... 355
10.3.14.2 MLME-MREPORT.confirm ..... 356
10.3.14.3 MLME-MREPORT.indication ..... 357
10.3.15 Channel switch ..... 358
10.3.15.1 MLME-CHANNELSWITCH.request ..... 358
10.3.15.2 MLME-CHANNELSWITCH.confirm ..... 359
10.3.15.3 MLME-CHANNELSWITCH.indication ..... 360
10.3.15.4 MLME-CHANNELSWITCH.response ..... 361
10.3.16 TPC request ..... 362
10.3.16.1 MLME-TPCADAPT.request ..... 362
10.3.16.2 MLME-TPCADAPT.confirm ..... 363
10.3.17 SetKeys ..... 364
10.3.17.1 MLME-SETKEYS.request ..... 364
10.3.17.2 MLME-SETKEYS.confirm ..... 365
10.3.18 DeleteKeys ..... 366
10.3.18.1 MLME-DELETEKEYS.request ..... 366
10.3.18.2 MLME-DELETEKEYS.confirm ..... 367
10.3.19 MIC (Michael) failure event ..... 368
10.3.19.1 MLME-MICHAELMICFAILURE.indication ..... 368
10.3.20 EAPOL ..... 369
10.3.20.1 MLME-EAPOL.request ..... 369
10.3.20.2 MLME-EAPOL.confirm ..... 370
10.3.21 MLME-PeerKeySTART ..... 371
10.3.21.1 MLME- PeerKeySTART.request ..... 371
10.3.22 SetProtection ..... 372
10.3.22.1 MLME-SETPROTECTION.request ..... 372
10.3.22.2 MLME-SETPROTECTION.confirm ..... 373
10.3.23 MLME-PROTECTEDFRAMEDROPPED ..... 374
10.3.23.1 MLME- PROTECTEDFRAMEDROPPED.indication ..... 374
10.3.24 TS management interface ..... 375
10.3.24.1 MLME-ADDTS.request ..... 375
10.3.24.2 MLME-ADDTS.confirm ..... 377
10.3.24.3 MLME-ADDTS.indication ..... 379
10.3.24.4 MLME-ADDTS.response ..... 381
10.3.24.5 MLME-DELTS.request ..... 383
10.3.24.6 MLME-DELTS.confirm ..... 384
10.3.24.7 MLME-DELTS.indication ..... 385
10.3.25 Management of direct links ..... 386
10.3.25.1 MLME-DLS.request ..... 386
10.3.25.2 MLME-DLS.confirm ..... 387
10.3.25.3 MLME-DLS.indication ..... 388
10.3.25.4 MLME-DLSTeardown.request ..... 389
10.3.25.5 MLME-DLSTeardown.confirm ..... 390


Copyright © 2007 IEEE. All rights reserved.

10.3.25.6 MLME-DLSTeardown.indication ..... 391
10.3.26 Higher layer synchronization support ..... 392
10.3.26.1 MLME-HL-SYNC.request ..... 392
10.3.26.2 MLME-HL-SYNC.confirm ..... 393
10.3.26.3 MLME-HL-SYNC.indication ..... 394
10.3.27 Block Ack ..... 395
10.3.27.1 MLME-ADDBA.request ..... 395
10.3.27.2 MLME-ADDBA.confirm ..... 397
10.3.27.3 MLME-ADDBA.indication ..... 398
10.3.27.4 MLME-ADDBA.response ..... 399
10.3.27.5 MLME-DELBA.request ..... 400
10.3.27.6 MLME-DELBA.confirm ..... 401
10.3.27.7 MLME-DELBA.indication ..... 402
10.3.28 Schedule element management ..... 403
10.3.28.1 MLME-SCHEDULE.request ..... 403
10.3.28.2 MLME-SCHEDULE.confirm ..... 404
10.3.28.3 MLME-SCHEDULE.indication ..... 405
10.3.29 Vendor-specific action ..... 406
10.3.29.1 MLME-VSPECIFIC.request ..... 406
10.3.29.2 MLME-VSPECIFIC.confirm ..... 407
10.3.29.3 MLME-VSPECIFIC.indication ..... 408
10.4 PLME SAP interface ..... 409
10.4.1 PLME-RESET.request ..... 409
10.4.1.1 Function ..... 409
10.4.1.2 Semantics of the service primitive ..... 409
10.4.1.3 When generated ..... 409
10.4.1.4 Effect of receipt ..... 409
10.4.2 PLME-CHARACTERISTICS.request ..... 410
10.4.2.1 Function ..... 410
10.4.2.2 Semantics of the service primitive ..... 410
10.4.2.3 When generated ..... 410
10.4.2.4 Effect of receipt ..... 410
10.4.3 PLME-CHARACTERISTICS.confirm ..... 411
10.4.3.1 Function ..... 411
10.4.3.2 Semantics of the service primitive ..... 411
10.4.3.3 When generated ..... 412
10.4.3.4 Effect of receipt ..... 412
10.4.4 PLME-DSSSTESTMODE.request ..... 413
10.4.4.1 Function ..... 413
10.4.4.2 Semantics of the service primitive ..... 413
10.4.4.3 When generated ..... 414
10.4.4.4 Effect of receipt ..... 414
10.4.5 PLME-DSSSTESTOUTPUT.request ..... 415
10.4.5.1 Function ..... 415
10.4.5.2 Semantics of the service primitive ..... 415
10.4.5.3 When generated ..... 415
10.4.5.4 Effect of receipt ..... 415
10.4.6 PLME-TXTIME.request ..... 416
10.4.6.1 Function ..... 416
10.4.6.2 Semantics of the service primitive ..... 416
10.4.6.3 When generated ..... 416
10.4.6.4 Effect of receipt ..... 416
10.4.7 PLME-TXTIME.confirm ..... 417
10.4.7.1 Function ..... 417

Copyright © 2007 IEEE. All rights reserved.

10.4.7.2 Semantics of the service primitive ..... 417
10.4.7.3 When generated ..... 417
10.4.7.4 Effect of receipt ..... 417

11. MLME ..... 419

11.1 Synchronization ..... 419
11.1.1 Basic approach ..... 419
11.1.1.1 TSF for infrastructure networks ..... 419
11.1.1.2 TSF for an IBSS ..... 419
11.1.2 Maintaining synchronization ..... 419
11.1.2.1 Beacon generation in infrastructure networks ..... 419
11.1.2.2 Beacon generation in an IBSS ..... 420
11.1.2.3 Beacon reception ..... 421
11.1.2.4 TSF timer accuracy ..... 421
11.1.3 Acquiring synchronization, scanning ..... 421
11.1.3.1 Passive scanning ..... 422
11.1.3.2 Active scanning ..... 422
11.1.3.3 Initializing a BSS ..... 423
11.1.3.4 Synchronizing with a BSS ..... 424
11.1.4 Adjusting STA timers ..... 424
11.1.5 Timing synchronization for FH PHYs ..... 424
11.2 Power management ..... 425
11.2.1 Power management in an infrastructure network ..... 425
11.2.1.1 STA Power Management modes ..... 425
11.2.1.2 AP TIM transmissions ..... 426
11.2.1.3 TIM types ..... 426
11.2.1.4 Power management with APSD ..... 427
11.2.1.5 AP operation during the CP ..... 429
11.2.1.6 AP operation during the CFP ..... 431
11.2.1.7 Receive operation for STAs in PS mode during the CP ..... 432
11.2.1.8 Receive operation for STAs in PS mode during the CFP ..... 432
11.2.1.9 Receive operation for non-AP STAs using APSD ..... 433
11.2.1.10 STAs operating in the Active mode ..... 433
11.2.1.11 AP aging function ..... 433
11.2.2 Power management in an IBSS ..... 433
11.2.2.1 Basic approach ..... 433
11.2.2.2 Initialization of power management within an IBSS ..... 435
11.2.2.3 STA power state transitions ..... 435
11.2.2.4 ATIM and frame transmission ..... 435
11.3 STA authentication and association ..... 436
11.3.1 Authentication and deauthentication ..... 438
11.3.1.1 Authentication—originating STA ..... 438
11.3.1.2 Authentication—destination STA ..... 439
11.3.1.3 Deauthentication—originating STA ..... 439
11.3.1.4 Deauthentication—destination STA ..... 439
11.3.2 Association, reassociation, and disassociation ..... 440
11.3.2.1 STA association procedures ..... 440
11.3.2.2 AP association procedures ..... 440
11.3.2.3 STA reassociation procedures ..... 441
11.3.2.4 AP reassociation procedures ..... 441
11.3.2.5 STA disassociation procedures ..... 442
11.3.2.6 Non-AP STA disassociation receipt procedure ..... 442
11.3.2.7 AP disassociation initiation procedure ..... 443


Copyright © 2007 IEEE. All rights reserved.

11.3.2.8 AP disassociation receipt procedure ..... 443
11.4 TS operation ..... 443
11.4.1 Introduction ..... 443
11.4.2 TSPEC construction ..... 444
11.4.3 TS lifecycle ..... 444
11.4.4 TS setup ..... 445
11.4.5 Failed TS setup ..... 447
11.4.6 Data transfer ..... 447
11.4.7 TS deletion ..... 448
11.4.8 TS timeout ..... 449
11.4.9 TS suspension ..... 450
11.4.10 TS Reinstatement ..... 450
11.5 Block Ack operation ..... 450
11.5.1 Setup and modification of the Block Ack parameters ..... 450
11.5.1.1 Procedure at the originator ..... 450
11.5.1.2 Procedure at the recipient ..... 452
11.5.2 Teardown of the Block Ack mechanism ..... 452
11.5.2.1 Procedure at the initiator of the Block Ack teardown ..... 453
11.5.2.2 Procedure at the recipient of the DELBA frame ..... 453
11.5.3 Error recovery upon a peer failure ..... 453
11.6 Higher layer timer synchronization ..... 455
11.6.1 Introduction ..... 455
11.6.2 Procedure at the STA ..... 456
11.7 DLS operation ..... 456
11.7.1 DLS procedures ..... 457
11.7.1.1 Setup procedure at the QoS STA ..... 457
11.7.1.2 Setup procedure at the AP ..... 458
11.7.2 Data transfer after setup ..... 458
11.7.3 DLS teardown ..... 459
11.7.3.1 STA-initiated DLS teardown procedure ..... 459
11.7.3.2 Teardown procedure at the AP ..... 460
11.7.3.3 AP-initiated DLS teardown procedure ..... 460
11.7.4 Error recovery upon a peer failure ..... 460
11.7.5 Secure DLS operation ..... 461
11.8 TPC procedures ..... 461
11.8.1 Association based on transmit power capability ..... 462
11.8.2 Specification of regulatory and local maximum transmit power levels ..... 462
11.8.3 Selection of a transmit power ..... 463
11.8.4 Adaptation of the transmit power ..... 463
11.9 DFS procedures ..... 463
11.9.1 Association based on supported channels ..... 464
11.9.2 Quieting channels for testing ..... 464
11.9.3 Testing channels for radars ..... 465
11.9.4 Discontinuing operations after detecting radars ..... 465
11.9.5 Detecting radars ..... 465
11.9.6 Requesting and reporting of measurements ..... 465
11.9.7 Selecting and advertising a new channel ..... 466
11.9.7.1 Selecting and advertising a new channel in an infrastructure BSS ..... 466
11.9.7.2 Selecting and advertising a new channel in an IBSS ..... 467

12. PHY service specification ..... 469

12.1 Scope ..... 469
12.2 PHY functions ..... 469

Copyright © 2007 IEEE. All rights reserved.

12.3 Detailed PHY service specifications ... 469
12.3.1 Scope and field of application ... 469
12.3.2 Overview of the service ... 469
12.3.3 Overview of interactions ... 469
12.3.4 Basic service and options ... 469
12.3.4.1 PHY-SAP peer-to-peer service primitives ... 470
12.3.4.2 PHY-SAP sublayer-to-sublayer service primitives ... 470
12.3.4.3 PHY-SAP service primitives parameters ... 470
12.3.4.4 Vector descriptions ... 471
12.3.5 PHY-SAP detailed service specification ... 472
12.3.5.1 PHY-DATA.request ... 472
12.3.5.2 PHY-DATA.indication ... 473
12.3.5.3 PHY-DATA.confirm ... 474
12.3.5.4 PHY-TXSTART.request ... 475
12.3.5.5 PHY-TXSTART.confirm ... 476
12.3.5.6 PHY-TXEND.request ... 477
12.3.5.7 PHY-TXEND.confirm ... 478
12.3.5.8 PHY-CCARESET.request ... 479
12.3.5.9 PHY-CCARESET.confirm ... 480
12.3.5.10 PHY-CCA.indication ... 481
12.3.5.11 PHY-RXSTART.indication ... 482
12.3.5.12 PHY-RXEND.indication ... 483

13. PHY management ... 485

14. Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz industrial, scientific, and medical (ISM) band ... 487

14.1 Overview ... 487
14.1.1 Overview of FHSS PHY ... 487
14.1.2 FHSS PHY functions ... 487
14.1.2.1 PLCP sublayer ... 487
14.1.2.2 PLME ... 487
14.1.2.3 PMD sublayer ... 487
14.1.3 Service specification method and notation ... 487
14.2 FHSS PHY-specific service parameter lists ... 488
14.2.1 Overview ... 488
14.2.2 TXVECTOR parameters ... 488
14.2.2.1 TXVECTOR LENGTH ... 488
14.2.2.2 TXVECTOR DATARATE ... 488
14.2.3 RXVECTOR parameters ... 488
14.2.3.1 TRXVECTOR LENGTH ... 489
14.2.3.2 RXVECTOR RSSI ... 489
14.3 FHSS PLCP sublayer ... 489
14.3.1 Overview ... 489
14.3.1.1 State diagram notation ... 489
14.3.2 PLCP frame format ... 490
14.3.2.1 PLCP Preamble field ... 491
14.3.2.2 PLCP Header field ... 491
14.3.2.3 PLCP data whitener ... 492
14.3.3 PLCP state machines ... 493
14.3.3.1 Transmit PLCP ... 493
14.3.3.2 CS/CCA procedure ... 496
14.3.3.3 Receive PLCP ... 499


Copyright © 2007 IEEE. All rights reserved.

14.4 PLME SAP layer management ..... 502
14.4.1 Overview ..... 502
14.4.2 FH PHY specific MLME procedures ..... 502
14.4.2.1 Overview ..... 502
14.4.2.2 FH synchronization ..... 502
14.4.3 FH PLME state machines ..... 502
14.4.3.1 Overview ..... 502
14.4.3.2 PLME state machine ..... 502
14.4.3.3 PLME management primitives ..... 504
14.5 FHSS PMD sublayer services ..... 505
14.5.1 Scope and field of application ..... 505
14.5.2 Overview of services ..... 505
14.5.3 Overview of interactions ..... 505
14.5.4 Basic service and options ..... 505
14.5.4.1 PMD_SAP peer-to-peer service primitives ..... 505
14.5.4.2 PMD_SAP sublayer-to-sublayer service primitives ..... 506
14.5.4.3 PMD_SAP service primitives parameters ..... 506
14.5.5 PMD_SAP detailed service specification ..... 507
14.5.5.1 PMD_DATA.request ..... 507
14.5.5.2 PMD_DATA.indicate ..... 508
14.5.5.3 PMD_TXRX.request ..... 509
14.5.5.4 PMD_PA_RAMP.request ..... 510
14.5.5.5 PMD_ANTSEL.request ..... 511
14.5.5.6 PMD_TXPWRLVL.request ..... 512
14.5.5.7 PMD_FREQ.request ..... 513
14.5.5.8 PMD_RSSI.indicate ..... 514
14.5.5.9 PMD_PWRMGMT.request ..... 515
14.6 FHSS PMD sublayer, 1.0 Mb/s ..... 516
14.6.1 1 Mb/s PMD operating specifications, general ..... 516
14.6.2 Regulatory requirements ..... 516
14.6.3 Operating frequency range ..... 516
14.6.4 Number of operating channels ..... 516
14.6.5 Operating channel center frequency ..... 517
14.6.6 Occupied channel bandwidth ..... 519
14.6.7 Minimum hop rate ..... 519
14.6.8 Hop sequences ..... 519
14.6.9 Unwanted emissions ..... 522
14.6.10 Modulation ..... 522
14.6.11 Channel data rate ..... 523
14.6.12 Channel switching/settling time ..... 523
14.6.13 Receive to transmit switch time ..... 523
14.6.14 PMD transmit specifications ..... 523
14.6.14.1 Nominal transmit power ..... 523
14.6.14.2 Transmit power levels ..... 524
14.6.14.3 Transmit power level control ..... 524
14.6.14.4 Transmit spectrum shape ..... 524
14.6.14.5 Transmit center frequency tolerance ..... 524
14.6.14.6 Transmitter ramp periods ..... 524
14.6.15 PMD receiver specifications ..... 525
14.6.15.1 Input signal range ..... 525
14.6.15.2 Receive center frequency acceptance range ..... 525
14.6.15.3 CCA power threshold ..... 525
14.6.15.4 Receiver sensitivity ..... 525
14.6.15.5 Intermodulation ..... 525

Copyright © 2007 IEEE. All rights reserved.

14.6.15.6 Desensitization (Dp) ..... 526
14.6.15.7 Receiver radiation ..... 526
14.6.16 Operating temperature range ..... 526
14.7 FHSS PMD sublayer, 2.0 Mb/s ..... 526
14.7.1 Overview ..... 526
14.7.2 4GFSK modulation ..... 526
14.7.2.1 Frame structure for HS FHSS PHY ..... 528
14.7.3 Channel data rate ..... 528
14.7.3.1 Input dynamic range ..... 528
14.7.3.2 Receiver sensitivity ..... 528
14.7.3.3 IMp ..... 528
14.7.3.4 Dp ..... 528
14.8 FHSS PHY MIB ..... 529
14.8.1 FH PHY attributes ..... 529
14.8.2 FH PHY attribute definitions ..... 530
14.8.2.1 dot11PHYType ..... 530
14.8.2.2 dot11RegDomainsSupported ..... 531
14.8.2.3 dot11CurrentRegDomain ..... 531
14.8.2.4 dot11TempType ..... 531
14.8.2.5 dot11CurrentPowerState ..... 531
14.8.2.6 dot11SupportedDataRatesTX ..... 531
14.8.2.7 dot11SupportedDataRatesRX ..... 532
14.8.2.8 aMPDUMaxLength ..... 532
14.8.2.9 dot11SupportedTxAntennas ..... 532
14.8.2.10 dot11CurrentTxAntenna ..... 533
14.8.2.11 dot11SupportedRxAntenna ..... 533
14.8.2.12 dot11DiversitySupport ..... 533
14.8.2.13 dot11DiversitySelectionRx ..... 533
14.8.2.14 dot11NumberSupportedPowerLevels ..... 534
14.8.2.15 dot11TxPowerLevel1-8 ..... 534
14.8.2.16 dot11CurrentTxPowerLevel ..... 534
14.8.2.17 dot11HopTime ..... 534
14.8.2.18 dot11CurrentChannelNumber ..... 535
14.8.2.19 dot11MaxDwellTime ..... 535
14.8.2.20 dot11CurrentSet ..... 535
14.8.2.21 dot11CurrentPattern ..... 535
14.8.2.22 dot11CurrentIndex ..... 535
14.8.2.23 dot11CurrentPowerState ..... 535
14.9 FH PHY characteristics ..... 536

15. DSSS PHY specification for the 2.4 GHz band designated for ISM applications ..... 537

15.1 Overview ..... 537
15.1.1 Scope ..... 537
15.1.2 DSSS PHY functions ..... 537
15.1.2.1 PLCP sublayer ..... 537
15.1.2.2 PMD sublayer ..... 537
15.1.2.3 PLME ..... 537
15.1.3 Service specification method and notation ..... 537
15.2 DSSS PLCP sublayer ..... 538
15.2.1 Overview ..... 538
15.2.2 PLCP frame format ..... 538
15.2.3 PLCP field definitions ..... 538
15.2.3.1 PLCP SYNC field ..... 538


Copyright © 2007 IEEE. All rights reserved.

15.2.3.2 PLCP SFD ... 538
15.2.3.3 PLCP IEEE 802.11 SIGNAL field ... 539
15.2.3.4 PLCP IEEE 802.11 SERVICE field ... 539
15.2.3.5 PLCP LENGTH field ... 539
15.2.3.6 PLCP CRC field ... 539
15.2.4 PLCP/DSSS PHY data scrambler and descrambler ... 541
15.2.5 PLCP data modulation and modulation rate change ... 541
15.2.6 Transmit PLCP ... 541
15.2.7 Receive PLCP ... 542
15.3 DSSS PLME ... 545
15.3.1 PLME_SAP sublayer management primitives ... 545
15.3.2 DSSS PHY MIB ... 546
15.3.3 DS PHY characteristics ... 547
15.4 DSSS PMD sublayer ... 547
15.4.1 Scope and field of application ... 547
15.4.2 Overview of service ... 548
15.4.3 Overview of interactions ... 548
15.4.4 Basic service and options ... 548
15.4.4.1 PMD_SAP peer-to-peer service primitives ... 548
15.4.4.2 PMD_SAP peer-to-peer service primitive parameters ... 549
15.4.4.3 PMD_SAP sublayer-to-sublayer service primitives ... 549
15.4.4.4 PMD_SAP service primitive parameters ... 550
15.4.5 PMD_SAP detailed service specification ... 551
15.4.5.1 PMD_DATA.request ... 551
15.4.5.2 PMD_DATA.indicate ... 552
15.4.5.3 PMD_TXSTART.request ... 553
15.4.5.4 PMD_TXEND.request ... 554
15.4.5.5 PMD_ANTSEL.request ... 555
15.4.5.6 PMD_ANTSEL.indicate ... 556
15.4.5.7 PMD_TXPWRLVL.request ... 557
15.4.5.8 PMD_RATE.request ... 558
15.4.5.9 PMD_RATE.indicate ... 559
15.4.5.10 PMD_RSSI.indicate ... 560
15.4.5.11 PMD_SQ.indicate ... 561
15.4.5.12 PMD_CS.indicate ... 562
15.4.5.13 PMD_ED.indicate ... 563
15.4.5.14 PMD_ED.request ... 564
15.4.5.15 PHY-CCA.indicate ... 565
15.4.6 PMD operating specifications, general ... 566
15.4.6.1 Operating frequency range ... 566
15.4.6.2 Number of operating channels ... 566
15.4.6.3 Spreading sequence ... 567
15.4.6.4 Modulation and channel data rates ... 567
15.4.6.5 Transmit and receive in-band and out-of-band spurious emissions ... 567
15.4.6.6 TX-to-RX turnaround time ... 567
15.4.6.7 RX-to-TX turnaround time ... 568
15.4.6.8 Slot time ... 568
15.4.6.9 Transmit and receive antenna port impedance ... 568
15.4.6.10 Transmit and receive operating temperature range ... 568
15.4.7 PMD transmit specifications ... 568
15.4.7.1 Transmit power levels ... 568
15.4.7.2 Minimum transmitted power level ... 568
15.4.7.3 Transmit power level control ... 568
15.4.7.4 Transmit spectrum mask ... 568

Copyright © 2007 IEEE. All rights reserved.

15.4.7.5 Transmit center frequency tolerance ... 569
15.4.7.6 Chip clock frequency tolerance ... 569
15.4.7.7 Transmit power-on and power-down ramp ... 569
15.4.7.8 RF carrier suppression ... 570
15.4.7.9 Transmit modulation accuracy ... 570
15.4.8 PMD receiver specifications ... 572
15.4.8.1 Receiver minimum input level sensitivity ... 572
15.4.8.2 Receiver maximum input level ... 572
15.4.8.3 Receiver adjacent channel rejection ... 572
15.4.8.4 CCA ... 573

16. Infrared (IR) PHY specification ... 575

16.1 Overview ... 575
16.1.1 Scope ... 576
16.1.2 IR PHY functions ... 576
16.1.2.1 PLCP sublayer ... 576
16.1.2.2 PMD sublayer ... 576
16.1.2.3 PLME ... 576
16.1.3 Service specification method and notation ... 576
16.2 IR PLCP sublayer ... 577
16.2.1 Overview ... 577
16.2.2 PLCP frame format ... 577
16.2.3 PLCP modulation and rate change ... 577
16.2.4 PLCP field definitions ... 578
16.2.4.1 PLCP SYNC field ... 578
16.2.4.2 PLCP SFD field ... 578
16.2.4.3 PLCP DR field ... 578
16.2.4.4 PLCP DCLA field ... 578
16.2.4.5 PLCP LENGTH field ... 578
16.2.4.6 PLCP CRC field ... 579
16.2.4.7 PSDU field ... 579
16.2.5 PLCPs ... 579
16.2.5.1 Transmit PLCP ... 579
16.2.5.2 Receive PLCP ... 579
16.2.5.3 CCA procedure ... 580
16.2.5.4 PMD_SAP peer-to-peer service primitive parameters ... 580
16.3 IR PMD sublayer ... 581
16.3.1 Overview ... 581
16.3.2 PMD operating specifications, general ... 581
16.3.2.1 Modulation and channel data rates ... 581
16.3.2.2 Octet partition and PPM symbol generation procedure ... 582
16.3.2.3 Operating environment ... 582
16.3.2.4 Operating temperature range ... 582
16.3.3 PMD transmit specifications ... 583
16.3.3.1 Transmitted peak optical power ... 583
16.3.3.2 Basic pulse shape and parameters ... 583
16.3.3.3 Emitter radiation pattern mask ... 583
16.3.3.4 Optical emitter peak wavelength ... 586
16.3.3.5 Transmit spectrum mask ... 586
16.3.4 PMD receiver specifications ... 586
16.3.4.1 Receiver sensitivity ... 586
16.3.4.2 Receiver dynamic range ... 586
16.3.4.3 Receiver field of view (FOV) ... 586


Copyright © 2007 IEEE. All rights reserved.

16.3.5 ED, CS, and CCA definitions ..... 587
16.3.5.1 ED signal ..... 587
16.3.5.2 CS signal ..... 587
16.3.5.3 CCA ..... 587
16.3.5.4 CHNL_ID ..... 588
16.4 PHY attributes ..... 588

17. Orthogonal frequency division multiplexing (OFDM) PHY specification for the 5 GHz band ..... 591

17.1 Introduction ..... 591
17.1.1 Scope ..... 591
17.1.2 OFDM PHY functions ..... 591
17.1.2.1 PLCP sublayer ..... 591
17.1.2.2 PMD sublayer ..... 592
17.1.2.3 PLME ..... 592
17.1.2.4 Service specification method ..... 592
17.2 OFDM PHY specific service parameter list ..... 592
17.2.1 Introduction ..... 592
17.2.2 TXVECTOR parameters ..... 592
17.2.2.1 TXVECTOR LENGTH ..... 592
17.2.2.2 TXVECTOR DATARATE ..... 592
17.2.2.3 TXVECTOR SERVICE ..... 593
17.2.2.4 TXVECTOR TXPWR_LEVEL ..... 593
17.2.3 RXVECTOR parameters ..... 593
17.2.3.1 RXVECTOR LENGTH ..... 594
17.2.3.2 RXVECTOR RSSI ..... 594
17.2.3.3 DATARATE ..... 594
17.2.3.4 SERVICE ..... 594
17.3 OFDM PLCP sublayer ..... 594
17.3.1 Introduction ..... 594
17.3.2 PLCP frame format ..... 595
17.3.2.1 Overview of the PPDU encoding process ..... 595
17.3.2.2 Modulation-dependent parameters ..... 596
17.3.2.3 Timing related parameters ..... 597
17.3.2.4 Mathematical conventions in the signal descriptions ..... 598
17.3.2.5 Discrete time implementation considerations ..... 599
17.3.3 PLCP preamble (SYNC) ..... 600
17.3.4 SIGNAL field ..... 601
17.3.4.1 RATE field ..... 602
17.3.4.2 PLCP LENGTH field ..... 602
17.3.4.3 Parity (P), Reserved (R), and SIGNAL TAIL fields ..... 603
17.3.5 DATA field ..... 603
17.3.5.1 SERVICE field ..... 603
17.3.5.2 PPDU TAIL field ..... 603
17.3.5.3 Pad bits (PAD) ..... 603
17.3.5.4 PLCP DATA scrambler and descrambler ..... 604
17.3.5.5 Convolutional encoder ..... 604
17.3.5.6 Data interleaving ..... 605
17.3.5.7 Subcarrier modulation mapping ..... 607
17.3.5.8 Pilot subcarriers ..... 610
17.3.5.9 OFDM modulation ..... 610
17.3.6 CCA ..... 611
17.3.7 PLCP data modulation and modulation rate change ..... 611
17.3.8 PMD operating specifications (general) ..... 612

Copyright © 2007 IEEE. All rights reserved.

17.3.8.1 Outline description ... 612
17.3.8.2 Regulatory requirements ... 613
17.3.8.3 Operating channel frequencies ... 613
17.3.8.4 Transmit and receive in-band and out-of-band spurious emissions ... 614
17.3.8.5 TX RF delay ... 614
17.3.8.6 Slot time ... 614
17.3.8.7 Transmit and receive antenna port impedance ... 614
17.3.8.8 Transmit and receive operating temperature range ... 614
17.3.9 PMD transmit specifications ... 614
17.3.9.1 Transmit power levels ... 614
17.3.9.2 Transmit spectrum mask ... 614
17.3.9.3 Transmission spurious ... 614
17.3.9.4 Transmit center frequency tolerance ... 615
17.3.9.5 Symbol clock frequency tolerance ... 615
17.3.9.6 Modulation accuracy ... 615
17.3.9.7 Transmit modulation accuracy test ... 616
17.3.10 PMD receiver specifications ... 617
17.3.10.1 Receiver minimum input sensitivity ... 617
17.3.10.2 Adjacent channel rejection ... 618
17.3.10.3 Nonadjacent channel rejection ... 618
17.3.10.4 Receiver maximum input level ... 618
17.3.10.5 CCA sensitivity ... 618
17.3.11 Transmit PLCP ... 618
17.3.12 Receive PLCP ... 621
17.4 OFDM PLME ... 622
17.4.1 PLME_SAP sublayer management primitives ... 622
17.4.2 OFDM PHY MIB ... 622
17.4.3 OFDM TXTIME calculation ... 625
17.4.4 OFDM PHY characteristics ... 625
17.5 OFDM PMD sublayer ... 627
17.5.1 Scope and field of application ... 627
17.5.2 Overview of service ... 627
17.5.3 Overview of interactions ... 627
17.5.4 Basic service and options ... 627
17.5.4.1 PMD_SAP peer-to-peer service primitives ... 627
17.5.4.2 PMD_SAP sublayer-to-sublayer service primitives ... 628
17.5.4.3 PMD_SAP service primitive parameters ... 628
17.5.5 PMD_SAP detailed service specification ... 629
17.5.5.1 PMD_DATA.request ... 629
17.5.5.2 PMD_DATA.indicate ... 630
17.5.5.3 PMD_TXSTART.request ... 631
17.5.5.4 PMD_TXEND.request ... 632
17.5.5.5 PMD_TXPWRLVL.request ... 633
17.5.5.6 PMD_RATE.request ... 634
17.5.5.7 PMD_RSSI.indicate ... 635

18. High Rate direct sequence spread spectrum (HR/DSSS) PHY specification ... 637

18.1 Overview ... 637
18.1.1 Scope ... 637
18.1.2 High Rate PHY functions ... 637
18.1.2.1 PLCP sublayer ... 638
18.1.2.2 PMD sublayer ... 638
18.1.2.3 PLME ... 638


Copyright © 2007 IEEE. All rights reserved.

18.1.3 Service specification method and notation ..... 638
18.2 High Rate PLCP sublayer ..... 638
18.2.1 Overview ..... 638
18.2.2 PPDU format ..... 638
18.2.2.1 Long PPDU format ..... 639
18.2.2.2 Short PPDU format ..... 639
18.2.3 PPDU field definitions ..... 639
18.2.3.1 Long PLCP SYNC field ..... 640
18.2.3.2 Long PLCP SFD ..... 640
18.2.3.3 Long PLCP SIGNAL field ..... 640
18.2.3.4 Long PLCP SERVICE field ..... 641
18.2.3.5 Long PLCP LENGTH field ..... 641
18.2.3.6 PLCP CRC (CRC-16) field ..... 643
18.2.3.7 Long PLCP data modulation and modulation rate change ..... 645
18.2.3.8 Short PLCP synchronization (shortSYNC) ..... 645
18.2.3.9 Short PLCP SFD field (shortSFD) ..... 645
18.2.3.10 Short PLCP SIGNAL field (shortSIGNAL) ..... 646
18.2.3.11 Short PLCP SERVICE field (shortSERVICE) ..... 646
18.2.3.12 Short PLCP LENGTH field (shortLENGTH) ..... 646
18.2.3.13 Short CRC-16 field (shortCRC) ..... 646
18.2.3.14 Short PLCP data modulation and modulation rate change ..... 646
18.2.4 PLCP/High Rate PHY data scrambler and descrambler ..... 646
18.2.5 Transmit PLCP ..... 647
18.2.6 Receive PLCP ..... 649
18.3 High Rate PLME ..... 652
18.3.1 PLME_SAP sublayer management primitives ..... 652
18.3.2 High Rate PHY MIB ..... 653
18.3.3 DS PHY characteristics ..... 654
18.3.4 High Rate TXTIME calculation ..... 655
18.3.5 Vector descriptions ..... 655
18.4 High Rate PMD sublayer ..... 656
18.4.1 Scope and field of application ..... 656
18.4.2 Overview of service ..... 656
18.4.3 Overview of interactions ..... 656
18.4.4 Basic service and options ..... 656
18.4.4.1 PMD_SAP peer-to-peer service primitives ..... 657
18.4.4.2 PMD_SAP sublayer-to-sublayer service primitives ..... 657
18.4.5 PMD_SAP detailed service specification ..... 658
18.4.5.1 PMD_DATA.request ..... 658
18.4.5.2 PMD_DATA.indicate ..... 659
18.4.5.3 PMD_MODULATION.request ..... 660
18.4.5.4 PMD_PREAMBLE.request ..... 661
18.4.5.5 PMD_PREAMBLE.indicate ..... 662
18.4.5.6 PMD_TXSTART.request ..... 663
18.4.5.7 PMD_TXEND.request ..... 664
18.4.5.8 PMD_ANTSEL.request ..... 665
18.4.5.9 PMD_TXPWRLVL.request ..... 666
18.4.5.10 PMD_RATE.request ..... 667
18.4.5.11 PMD_RSSI.indicate ..... 668
18.4.5.12 PMD_SQ.indicate ..... 669
18.4.5.13 PMD_CS.indicate ..... 670
18.4.5.14 PMD_ED.indicate ..... 671
18.4.5.15 PMD_ED.request ..... 672
18.4.6 PMD operating specifications, general ..... 673

Copyright © 2007 IEEE. All rights reserved.

18.4.6.1 Operating frequency range ... 673
18.4.6.2 Number of operating channels ... 673
18.4.6.3 Modulation and channel data rates ... 674
18.4.6.4 Spreading sequence and modulation for 1 Mb/s and 2 Mb/s ... 674
18.4.6.5 Spreading sequences and modulation for CCK modulation at 5.5 Mb/s and 11 Mb/s ... 674
18.4.6.6 DSSS/PBCC data modulation and modulation rate (optional) ... 677
18.4.6.7 Channel Agility (optional) ... 679
18.4.6.8 Transmit and receive in-band and out-of-band spurious emissions ... 682
18.4.6.9 TX-to-RX turnaround time ... 682
18.4.6.10 RX-to-TX turnaround time ... 682
18.4.6.11 Slot time ... 682
18.4.6.12 Channel switching/settling time ... 682
18.4.6.13 Transmit and receive antenna port impedance ... 682
18.4.6.14 Transmit and receive operating temperature range ... 682
18.4.7 PMD transmit specifications ... 682
18.4.7.1 Transmit power levels ... 682
18.4.7.2 Transmit power level control ... 683
18.4.7.3 Transmit spectrum mask ... 683
18.4.7.4 Transmit center frequency tolerance ... 683
18.4.7.5 Chip clock frequency tolerance ... 683
18.4.7.6 Transmit power-on and power-down ramp ... 684
18.4.7.7 RF carrier suppression ... 684
18.4.7.8 Transmit modulation accuracy ... 685
18.4.8 PMD receiver specifications ... 687
18.4.8.1 Receiver minimum input level sensitivity ... 687
18.4.8.2 Receiver maximum input level ... 687
18.4.8.3 Receiver adjacent channel rejection ... 687
18.4.8.4 CCA ... 687

19. ERP specification ... 689

19.1 Overview ... 689
19.1.1 Introduction ... 689
19.1.2 Operational modes ... 689
19.1.3 Scope ... 690
19.1.4 ERP functions ... 690
19.2 PHY-specific service parameter list ... 691
19.3 Extended Rate PLCP sublayer ... 692
19.3.1 Introduction ... 692
19.3.2 PPDU format ... 692
19.3.2.1 Long preamble PPDU format ... 693
19.3.2.2 Short preamble PPDU format ... 694
19.3.2.3 ERP-OFDM PPDU format ... 695
19.3.2.4 DSSS-OFDM long preamble PPDU format ... 695
19.3.2.5 Short DSSS-OFDM PLCP PPDU format ... 697
19.3.3 PLCP data modulation and rate change ... 697
19.3.3.1 Long and short preamble formats ... 697
19.3.3.2 ERP-PBCC 22 Mb/s and 33 Mb/s formats ... 698
19.3.3.3 ERP-OFDM format ... 700
19.3.3.4 Long and short DSSS-OFDM PLCP format ... 700
19.3.4 PLCP transmit procedure ... 701
19.3.5 CCA ... 701
19.3.6 PLCP receive procedure ... 701


Copyright © 2007 IEEE. All rights reserved.

19.4 ERP PMD operating specifications (general) ..... 702
19.4.1 Regulatory requirements ..... 702
19.4.2 Operating channel frequencies ..... 702
19.4.3 Transmit and receive in-band and out-of-band spurious emissions ..... 702
19.4.4 Slot time ..... 702
19.4.5 SIFS value ..... 702
19.4.6 CCA performance ..... 702
19.4.7 PMD transmit specifications ..... 703
19.4.7.1 Transmit power levels ..... 703
19.4.7.2 Transmit center frequency tolerance ..... 703
19.4.7.3 Symbol clock frequency tolerance ..... 703
19.5 ERP operation specifications ..... 703
19.5.1 Receiver minimum input level sensitivity ..... 703
19.5.2 Adjacent channel rejection ..... 704
19.5.3 Receive maximum input level capability ..... 704
19.5.4 Transmit spectral mask ..... 704
19.6 ERP-PBCC operation specifications ..... 704
19.6.1 Receiver minimum input level sensitivity ..... 704
19.6.2 Receiver adjacent channel rejection ..... 704
19.7 DSSS-OFDM operation specifications ..... 705
19.7.1 Overview ..... 705
19.7.2 Single carrier to multicarrier transition requirements ..... 705
19.7.2.1 Spectral binding requirement ..... 706
19.7.2.2 Sample-power matching requirement ..... 711
19.7.2.3 Transition time alignment ..... 712
19.7.2.4 Single carrier termination ..... 713
19.7.2.5 Transition carrier frequency requirement ..... 713
19.7.2.6 Transition carrier phase requirement ..... 714
19.7.2.7 Transmit modulation accuracy requirement ..... 715
19.8 ERP PLME ..... 715
19.8.1 PLME SAP ..... 715
19.8.2 MIB ..... 715
19.8.3 TXTIME ..... 717
19.8.3.1 ERP-OFDM TXTIME calculations ..... 717
19.8.3.2 ERP-PBCC TXTIME calculations ..... 718
19.8.3.3 DSSS-OFDM TXTIME calculations ..... 718
19.8.4 ERP-OFDM PLCP PSDU definition ..... 719
19.9 Extended rate PMD sublayer ..... 720
19.9.1 Scope and field of application ..... 720
19.9.2 Overview of service ..... 720
19.9.3 Overview of Interactions ..... 720
19.9.4 Basic service and options ..... 720
19.9.4.1 PMD_SAP peer-to-peer service primitives ..... 720
19.9.4.2 PMD_SAP sublayer-to-sublayer service primitives ..... 720
19.9.4.3 PMD_SAP service primitive parameters ..... 720
19.9.5 PMD_SAP detailed service specification ..... 722
19.9.5.1 PMD_DATA.request ..... 723
19.9.5.2 PMD_DATA.indicate ..... 723
19.9.5.3 PMD_MODULATION.request ..... 723
19.9.5.4 PMD_PREAMBLE.request ..... 723
19.9.5.5 PMD_TXSTART.request ..... 723
19.9.5.6 PMD_TXEND.request ..... 723
19.9.5.7 PMD_ANTSEL.request ..... 723
19.9.5.8 PMD_TXPRWLVL.request ..... 723

Copyright © 2007 IEEE. All rights reserved.

19.9.5.9 PMD_RATE.request ... 723
19.9.5.10 PMD_RSSI.indicate ... 723
19.9.5.11 PMD_SQ.indicate ... 723
19.9.5.12 PMD_CS.indicate ... 724
19.9.5.13 PMD_ED.indicate ... 724

Annex A (normative) Protocol Implementation Conformance Statement (PICS) proforma ... 725

A.1 Introduction ... 725
A.2 Abbreviations and special symbols ... 725
A.2.1 Symbols for Status column ... 725
A.2.2 General abbreviations for Item and Support columns ... 725
A.3 Instructions for completing the PICS proforma ... 726
A.3.1 General structure of the PICS proforma ... 726
A.3.2 Additional information ... 726
A.3.3 Exception information ... 726
A.3.4 Conditional status ... 727
A.4 PICS proforma—IEEE Std 802.11-2007 ... 728
A.4.1 Implementation identification ... 728
A.4.2 Protocol summary ... 728
A.4.4 MAC protocol ... 729
A.4.3 IUT configuration ... 729
A.4.5 Frequency hopping (FH) PHY functions ... 737
A.4.6 Direct sequence PHY functions ... 739
A.4.7 IR baseband PHY functions ... 743
A.4.8 OFDM PHY functions ... 746
A.4.9 High Rate, direct sequence PHY functions ... 755
A.4.10 Regulatory Domain Extensions ... 760
A.4.11 ERP functions ... 761
A.4.12 Spectrum management extensions ... 764
A.4.13 Regulatory classes extensions ... 766
A.4.14 QoS base functionality ... 766
A.4.15 QoS enhanced distributed channel access (EDCA) ... 767
A.4.16 QoS hybrid coordination function (HCF) controlled channel access (HCCA) ... 767

Annex B (informative) Hopping sequences ... 769

Annex C (informative) Formal description of a subset of MAC operation ... 783

C.1 Introduction to the MAC formal description ... 786
C.1.1 Fundamental assumptions ... 786
C.1.2 Notation conventions ... 786
C.1.3 Modeling techniques ... 787
C.2 Data type and operator definitions for the MAC state machines ... 788
C.3 State machines for MAC STAs ... 836
C.4 State machines for MAC AP ... 913

Annex D (normative) ASN.1 encoding of the MAC and PHY MIB ... 985

Annex E (reserved for future use) ... 1075

Annex F (informative) High Rate PHY/FH interoperability ... 1077

F.1 Additional CCA recommendations ... 1077


Copyright © 2007 IEEE. All rights reserved.

Annex G (informative) An example of encoding a frame for OFDM PHY ..... 1079

G.1 Introduction ..... 1079
G.2 The message ..... 1079
G.3 Generation of the preamble ..... 1080
G.3.1 Generation of the short sequences ..... 1080
G.3.2 Generation of the long sequences ..... 1083
G.4 Generation of the SIGNAL field ..... 1086
G.4.1 SIGNAL field bit assignment ..... 1086
G.4.2 Coding the SIGNAL field bits ..... 1086
G.4.3 Interleaving the SIGNAL field bits ..... 1087
G.4.4 SIGNAL field frequency domain ..... 1087
G.4.5 SIGNAL field time domain ..... 1088
G.5 Generating the DATA bits ..... 1090
G.5.1 Delineating, SERVICE field prepending, and zero padding ..... 1090
G.5.2 Scrambling ..... 1091
G.6 Generating the first DATA symbol ..... 1094
G.6.1 Coding the DATA bits ..... 1094
G.6.2 Interleaving the DATA bits ..... 1095
G.6.3 Mapping into symbols ..... 1098
G.7 Generating the additional DATA symbols ..... 1099
G.8 The entire packet ..... 1100

Annex H (informative) RSNA reference implementations and test vectors ..... 1107

H.1 TKIP temporal key mixing function reference implementation and test vector ..... 1107
H.1.1 Test vectors ..... 1117
H.2 Michael reference implementation and test vectors ..... 1119
H.2.1 Michael test vectors ..... 1119
H.2.2 Sample code for Michael ..... 1120
H.3 PRF reference implementation and test vectors ..... 1127
H.3.1 PRF reference code ..... 1127
H.3.2 PRF test vectors ..... 1128
H.4 Suggested pass-phrase-to-PSK mapping ..... 1128
H.4.1 Introduction ..... 1128
H.4.2 Reference implementation ..... 1129
H.4.3 Test vectors ..... 1130
H.5 Suggestions for random number generation ..... 1131
H.5.1 Software sampling ..... 1131
H.5.2 Hardware-assisted solution ..... 1132
H.6 Additional test vectors ..... 1133
H.6.1 Notation ..... 1133
H.6.2 WEP cryptographic encapsulation ..... 1134
H.6.3 TKIP test vector ..... 1135
H.6.4 CCMP test vector ..... 1136
H.6.5 PRF test vectors ..... 1136
H.7 Key hierarchy test vectors ..... 1138
H.7.1 Pairwise key derivation ..... 1138

Annex I (informative) Regulatory classes ..... 1141

I.1 External regulatory references ..... 1141
I.2 Radio performance specifications ..... 1143
I.2.1 Transmit and receive in-band and out-of-band spurious emissions ..... 1143

Copyright © 2007 IEEE. All rights reserved.

I.2.2 Transmit power levels ... 1143
I.2.3 Transmit spectrum mask ... 1144

Annex J (normative) Country information element and regulatory classes ... 1147

Annex K (informative) Admission control ... 1151

K.1 Example use of TSPEC for admission control ... 1151
K.2 Recommended practices for contention-based admission control ... 1152
K.2.1 Use of ACM (admission control mandatory) subfield ... 1152
K.2.2 Deriving medium time ... 1152
K.3 Guidelines and reference design for sample scheduler and admission control unit ... 1153
K.3.1 Guidelines for deriving service schedule parameters ... 1153
K.3.2 TSPEC construction ... 1153
K.3.3 Reference design for sample scheduler and admission control unit ... 1155

Annex L (informative) An example of encoding a TIM virtual bit map ... 1159

L.1 Introduction ... 1159
L.2 Examples ... 1159
L.3 Sample C code ... 1160

Annex M (informative) Integration function ... 1165

M.1 Introduction ... 1165
M.2 Ethernet V2.0/IEEE 802.3 LAN integration function ... 1165
M.3 Example ... 1165
M.4 Integration service versus bridging ... 1167

Annex N (informative) AP functional description ... 1169

N.1 Introduction ... 1169
N.2 Terminology ... 1169
N.3 Primary ACM_STA functions ... 1173
N.4 Primary AP functions ... 1173
N.5 Primary DS functions ... 1175
N.6 Primary portal function ... 1175
N.7 AU example ... 1175

Annex O (informative) DS SAP specification ... 1177

O.1 Introduction ... 1177
O.2 SAP primitives ... 1178
O.2.1 MSDU transfer ... 1178
O.2.2 Mapping updates ... 1180

Annex P (informative) Bibliography ... 1181

P.1 General ... 1181
P.2 Specification and description language (SDL) documentation ... 1183


Copyright © 2007 IEEE. All rights reserved.

# List of figures

Figure 5-1—BSSs ..... 25
Figure 5-2—DSs and APs ..... 26
Figure 5-3—ESS ..... 27
Figure 5-4—A representative signal intensity map ..... 28
Figure 5-5—Collocated coverage areas ..... 29
Figure 5-6—Connecting to other IEEE 802 LANs ..... 29
Figure 5-7—Complete IEEE 802.11 architecture ..... 32
Figure 5-8—IEEE 802.11 architecture (again) ..... 41
Figure 5-9—Logical architecture of an IBSS ..... 41
Figure 5-10—Portion of the ISO/IEC basic reference model covered in this standard ..... 42
Figure 5-11—Establishing the IEEE 802.11 association ..... 44
Figure 5-12—IEEE 802.1X EAP authentication ..... 44
Figure 5-13—Establishing pairwise and group keys ..... 45
Figure 5-14—Delivery of subsequent group keys ..... 45
Figure 5-15—Sample 4-Way Handshakes in an IBSS ..... 47
Figure 5-16—Example using IEEE 802.1X authentication ..... 48
Figure 6-1—MAC data plane architecture ..... 55
Figure 7-1—MAC frame format ..... 60
Figure 7-2—Frame Control field ..... 60
Figure 7-3—Sequence Control field ..... 66
Figure 7-4—QoS AP PS Buffer State subfield ..... 69
Figure 7-5—Frame Control field subfield values within control frames ..... 72
Figure 7-6—RTS frame ..... 72
Figure 7-7—CTS frame ..... 73
Figure 7-8—ACK frame ..... 74
Figure 7-9—PS-Poll frame ..... 74
Figure 7-10—CF-End frame ..... 74
Figure 7-11—CF-End+CF-Ack frame ..... 75
Figure 7-12—BlockAckReq frame ..... 75
Figure 7-13—BAR Control field ..... 75
Figure 7-14—Block Ack Starting Sequence Control field ..... 76
Figure 7-15—BlockAck frame ..... 76
Figure 7-16—BA Control field ..... 76
Figure 7-17—Data frame ..... 77
Figure 7-18—Management frame format ..... 79
Figure 7-19—Authentication Algorithm Number field ..... 87
Figure 7-20—Authentication Transaction Sequence Number field ..... 87
Figure 7-21—Beacon Interval field ..... 88
Figure 7-22—Capability Information field ..... 88
Figure 7-23—Current AP Address field ..... 91
Figure 7-24—Listen Interval field ..... 91
Figure 7-25—Reason Code field ..... 92
Figure 7-26—AID field ..... 93
Figure 7-27—Status Code field ..... 93
Figure 7-28—Timestamp field ..... 95
Figure 7-29—Action field ..... 95
Figure 7-30—Dialog Token fixed field ..... 96
Figure 7-31—DLS Timeout Value fixed field ..... 96
Figure 7-32—Block Ack Parameter Set fixed field ..... 97
Figure 7-33—Block Ack Timeout Value fixed field ..... 97
Figure 7-34—DELBA Parameters fixed field ..... 98

Copyright © 2007 IEEE. All rights reserved.

Figure 7-35—QoS Info field when sent by an AP ..... 98
Figure 7-36—QoS Info field when set by a non-AP STA ..... 98
Figure 7-37—Element format ..... 99
Figure 7-38—SSID element format ..... 101
Figure 7-39—Supported rates element format ..... 102
Figure 7-40—FH Parameter Set element format ..... 102
Figure 7-41—DS Parameter Set element format ..... 103
Figure 7-42—CF Parameter Set element format ..... 103
Figure 7-43—TIM element format ..... 103
Figure 7-44—IBSS Parameter Set element format ..... 104
Figure 7-45—Challenge Text element format ..... 105
Figure 7-46—Country information element format ..... 105
Figure 7-47—Hopping Pattern Parameters information element ..... 107
Figure 7-48—Hopping Pattern Table information element ..... 108
Figure 7-49—Request information element ..... 109
Figure 7-50—ERP Information element ..... 111
Figure 7-51—Extended Supported Rates element format ..... 111
Figure 7-52—Power Constraint element format ..... 112
Figure 7-53—Power Capability element format ..... 112
Figure 7-54—TPC Request element format ..... 113
Figure 7-55—TPC Report element format ..... 113
Figure 7-56—Supported Channels element format ..... 114
Figure 7-57—Channel Switch Announcement element format ..... 114
Figure 7-58—Measurement Request element format ..... 115
Figure 7-59—Measurement Request Mode field ..... 115
Figure 7-60—Measurement Request field format for a basic request ..... 117
Figure 7-61—Measurement Request field format for a CCA request ..... 117
Figure 7-62—Measurement Request field format for a RPI histogram request ..... 118
Figure 7-63—Measurement Report element format ..... 118
Figure 7-64—Measurement Report Mode field ..... 118
Figure 7-65—Measurement Report field format for a basic report ..... 119
Figure 7-66—Map field format ..... 120
Figure 7-67—Measurement Report field format for a CCA report ..... 120
Figure 7-68—Measurement Report field format for an RPI histogram report ..... 121
Figure 7-69—Quiet element format ..... 122
Figure 7-70—IBSS DFS element format ..... 123
Figure 7-71—Channel Map field format ..... 123
Figure 7-72—RSN information element format ..... 123
Figure 7-73—Suite selector format ..... 125
Figure 7-74—RSN Capabilities field format ..... 127
Figure 7-75—Vendor Specific information element format ..... 128
Figure 7-76—Extended Capabilities element format ..... 129
Figure 7-77—BSS Load element format ..... 129
Figure 7-78—EDCA Parameter Set element ..... 130
Figure 7-79—AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record field format ..... 130
Figure 7-80—ACI/AIFSN field ..... 130
Figure 7-81—ECWmin and ECWmax fields ..... 131
Figure 7-82—TSPEC element format ..... 132
Figure 7-83—TS Info field ..... 132
Figure 7-84—Nominal MSDU Size field ..... 134
Figure 7-85—TCLAS element format ..... 136
Figure 7-86—Frame Classifier field ..... 136
Figure 7-87—Frame Classifier field of Classifier Type 0 ..... 137
Figure 7-88—Frame Classifier field of Classifier Type 1 for traffic over IPv4 ..... 137


Copyright © 2007 IEEE. All rights reserved.

Figure 7-89—Frame Classifier field of Classifier Type 1 for traffic over IPv6 ..... 137
Figure 7-90—Frame Classifier field of Classifier Type 2 ..... 138
Figure 7-91—TS Delay element ..... 138
Figure 7-92—TCLAS Processing element ..... 138
Figure 7-93—Schedule element ..... 139
Figure 7-94—Schedule Info field ..... 139
Figure 7-95—QoS Capability element format ..... 140
Figure 7-96—Measurement Request frame body format ..... 141
Figure 7-97—Measurement Report frame body format ..... 141
Figure 7-98—TPC Request frame body format ..... 142
Figure 7-99—TPC Report frame body format ..... 142
Figure 7-100—Channel Switch Announcement frame body format ..... 143
Figure 7-101—Vendor Specific Action frame format ..... 151
Figure 8-1—Construction of expanded WEP MPDU ..... 158
Figure 8-2—WEP encapsulation block diagram ..... 160
Figure 8-3—WEP decapsulation block diagram ..... 160
Figure 8-4—TKIP encapsulation block diagram ..... 166
Figure 8-5—TKIP decapsulation block diagram ..... 167
Figure 8-6—Construction of expanded TKIP MPDU ..... 168
Figure 8-7—TKIP MIC relation to IEEE 802.11 processing (informative) ..... 169
Figure 8-8—TKIP MIC processing format ..... 170
Figure 8-9—Michael message processing ..... 171
Figure 8-10—Michael block function ..... 171
Figure 8-11—Authenticator MIC countermeasures ..... 173
Figure 8-12—Supplicant MIC countermeasures ..... 174
Figure 8-13—Phase 1 key mixing ..... 177
Figure 8-14—Phase 2 key mixing ..... 178
Figure 8-15—Expanded CCMP MPDU ..... 180
Figure 8-16—CCMP encapsulation block diagram ..... 181
Figure 8-17—AAD construction ..... 182
Figure 8-18—Nonce construction ..... 183
Figure 8-19—CCMP decapsulation block diagram ..... 184
Figure 8-20—Pairwise key hierarchy ..... 198
Figure 8-21—Group key hierarchy (informative) ..... 200
Figure 8-22—PeerKey hierarchy ..... 201
Figure 8-23—EAPOL-Key frame ..... 203
Figure 8-24—Key Information bit layout ..... 203
Figure 8-25—KDE format ..... 206
Figure 8-26—GTK KDE format ..... 207
Figure 8-27—MAC address KDE format ..... 207
Figure 8-28—PMKID KDE format ..... 208
Figure 8-29—SMK KDE format ..... 208
Figure 8-30—Nonce KDE format ..... 208
Figure 8-31—Lifetime KDE format ..... 208
Figure 8-32—Error KDE format ..... 208
Figure 8-33—Sample 4-Way Handshake ..... 218
Figure 8-34—Sample Group Key Handshake ..... 222
Figure 8-35—RSNA Supplicant key management state machine ..... 223
Figure 8-36—PeerKey Handshake Supplicant key management state machine ..... 228
Figure 8-37—Authenticator state machines, part 1 ..... 230
Figure 8-38—Authenticator state machines, part 2 ..... 231
Figure 8-39—Authenticator state machines, part 3 ..... 231
Figure 8-40—Authenticator state machines, part 4 ..... 232
Figure 9-1—MAC architecture ..... 251

Copyright © 2007 IEEE. All rights reserved.

Figure 9-2—Fragmentation ..... 255
Figure 9-3—Some IFS relationships ..... 258
Figure 9-4—Example of exponential increase of CW ..... 260
Figure 9-5—Basic access method ..... 261
Figure 9-6—Backoff procedure ..... 262
Figure 9-7—RTS/CTS/data/ACK and NAV setting ..... 264
Figure 9-8—Transmission of a multiple-fragment MSDU using SIFS ..... 265
Figure 9-9—RTS/CTS with fragmented MSDU ..... 266
Figure 9-10—RTS/CTS with transmitter priority and missed acknowledgment ..... 267
Figure 9-11—Individually addressed data/ACK MPDU ..... 268
Figure 9-12—DCF timing relationships ..... 270
Figure 9-13—CFP/CP alternation ..... 272
Figure 9-14—Beacon frames and CFPs ..... 273
Figure 9-15—Example of delayed beacon and foreshortened CFP ..... 273
Figure 9-16—Example of PCF frame transfer ..... 275
Figure 9-17—Reference implementation model ..... 287
Figure 9-18—EDCA mechanism timing relationships ..... 289
Figure 9-19—CAP/CFP/CP periods ..... 293
Figure 9-20—Polled TXOP ..... 295
Figure 9-21—Message sequence chart for Block Ack mechanism: (a) setup, (b) data and acknowledgment transfer and (c) tear down ..... 302
Figure 9-22—A typical Block Ack sequence when immediate policy is used ..... 304
Figure 9-23—A typical BlockAck sequence when delayed policy is used ..... 305
Figure 10-1—GET and SET operations ..... 313
Figure 10-2—Layer management model ..... 345
Figure 10-3—Measurement request—accepted ..... 346
Figure 10-4—Measurement request—rejected ..... 347
Figure 10-5—TPC adaptation ..... 348
Figure 10-6—Channel switch ..... 349
Figure 11-1—Beacon transmission on a busy network ..... 420
Figure 11-2—Beacon transmission in an IBSS ..... 420
Figure 11-3—Probe response ..... 423
Figure 11-4—Infrastructure power management operation (no PCF operating) ..... 427
Figure 11-5—Power management in an IBSS—basic operation ..... 434
Figure 11-6—Relationship between state variables and services ..... 437
Figure 11-7—TS lifecycle ..... 445
Figure 11-8—TS setup ..... 445
Figure 11-9—Failed TS setup detected within non-AP STA MAC ..... 447
Figure 11-10—TS deletion ..... 449
Figure 11-11—TS timeout ..... 451
Figure 11-12—Block Ack setup ..... 452
Figure 11-13—Block Ack deletion ..... 453
Figure 11-14—Error recovery by the receiver upon a peer failure ..... 454
Figure 11-15—The four steps involved in direct-link handshake ..... 456
Figure 11-16—DLS message flow ..... 457
Figure 11-17—STA-initiated DLS teardown message flow ..... 459
Figure 14-1—State diagram notation example ..... 489
Figure 14-2—PLCP frame format ..... 490
Figure 14-3—Frame synchronous scrambler/descrambler ..... 492
Figure 14-4—PLCP data whitener format ..... 493
Figure 14-5—PLCP top-level state diagram ..... 493
Figure 14-6—Transmit state machine ..... 494
Figure 14-7—Data whitener encoding procedure ..... 495
Figure 14-8—Transmit state timing ..... 497


Copyright © 2007 IEEE. All rights reserved.

Figure 14-9—CS/CCA state machine ..... 498
Figure 14-10—CS/CCA state timing ..... 500
Figure 14-11—Receive state machine ..... 501
Figure 14-12—Data whitener decoding procedure ..... 501
Figure 14-13—Receive timing ..... 503
Figure 14-14—PLME state machine ..... 504
Figure 14-15—PMD layer reference model ..... 505
Figure 14-16—Transmit modulation mask ..... 523
Figure 14-17—4GFSK transmit modulation ..... 527
Figure 15-1—PLCP frame format ..... 538
Figure 15-2—CRC-16 implementation ..... 540
Figure 15-3—Example CRC calculation ..... 540
Figure 15-4—Data scrambler ..... 541
Figure 15-5—Data descrambler ..... 541
Figure 15-6—Transmit PLCP ..... 542
Figure 15-7—PLCP transmit state machine ..... 543
Figure 15-8—Receive PLCP ..... 543
Figure 15-9—PLCP receive state machine ..... 545
Figure 15-10—PMD layer reference model ..... 548
Figure 15-11—Transmit spectrum mask ..... 569
Figure 15-12—Transmit power-on ramp ..... 569
Figure 15-13—Transmit power-down ramp ..... 570
Figure 15-14—Modulation accuracy measurement example ..... 570
Figure 15-15—Chip clock alignment with baseband eye pattern ..... 571
Figure 16-1—PPDU frame format ..... 577
Figure 16-2—Basic pulse shape ..... 583
Figure 16-3—Emitter radiation pattern Mask 1 ..... 584
Figure 16-4—Emitter radiation pattern Mask 2 ..... 585
Figure 16-5—Mask 2 device orientation drawing ..... 585
Figure 16-6—Transmit spectrum mask ..... 586
Figure 17-1—PPDU frame format ..... 595
Figure 17-2—Illustration of OFDM frame with cyclic extension and windowing for (a) single reception or (b) two receptions of the FFT period ..... 599
Figure 17-3—Inputs and outputs of inverse Fourier transform ..... 600
Figure 17-4—OFDM training structure ..... 600
Figure 17-5—SIGNAL field bit assignment ..... 602
Figure 17-6—SERVICE field bit assignment ..... 603
Figure 17-7—Data scrambler ..... 604
Figure 17-8—Convolutional encoder (k = 7) ..... 605
Figure 17-9—Example of the bit-stealing and bit-insertion procedure (r = 3/4, 2/3) ..... 606
Figure 17-10—BPSK, QPSK, 16-QAM, and 64-QAM constellation bit encoding ..... 608
Figure 17-11—Subcarrier frequency allocation ..... 611
Figure 17-12—Transmitter and receiver block diagram for the OFDM PHY ..... 612
Figure 17-13—Constellation error ..... 617
Figure 17-14—Transmit PLCP ..... 619
Figure 17-15—PLCP transmit state machine ..... 620
Figure 17-16—Receive PLCP ..... 621
Figure 17-17—PLCP receive state machine ..... 623
Figure 17-18—PMD layer reference model ..... 627
Figure 18-1—Long PPDU format ..... 639
Figure 18-2—Short PPDU format ..... 640
Figure 18-3—CRC-16 implementation ..... 644
Figure 18-4—Example of CRC calculation ..... 645
Figure 18-5—Data scrambler ..... 646

Copyright © 2007 IEEE. All rights reserved.

Figure 18-6—Data descrambler ..... 647
Figure 18-7—Transmit PLCP ..... 648
Figure 18-8—PLCP transmit state machine ..... 649
Figure 18-9—Receive PLCP ..... 650
Figure 18-10—PLCP receive state machine ..... 652
Figure 18-11—Layer reference model ..... 656
Figure 18-12—PBCC modulator scheme ..... 677
Figure 18-13—PBCC convolutional encoder ..... 677
Figure 18-14—Cover code mapping ..... 678
Figure 18-15—China and North American channel selection—nonoverlapping ..... 679
Figure 18-16—China and North American channel selection—overlapping ..... 680
Figure 18-17—European channel selection—nonoverlapping ..... 680
Figure 18-18—European channel selection—overlapping ..... 680
Figure 18-19—Transmit spectrum mask ..... 683
Figure 18-20—Transmit power-on ramp ..... 684
Figure 18-21—Transmit power-down ramp ..... 684
Figure 18-22—Modulation accuracy measurement example ..... 685
Figure 18-23—Chip clock alignment with baseband eye pattern ..... 686
Figure 19-1—Long preamble PPDU format for DSSS-OFDM ..... 696
Figure 19-2—Short preamble PPDU format for DSSS-OFDM ..... 697
Figure 19-3—22/33 Mb/s ERP-PBCC convolutional encoder ..... 698
Figure 19-4—ERP-PBCC-22 and ERP-PBCC-33 cover code mapping ..... 699
Figure 19-5—33 Mb/s clock switching ..... 699
Figure 19-6—DSSS-OFDM PSDU ..... 700
Figure 19-7—Single carrier to multicarrier transition definition ..... 706
Figure 19-8—Linear distortions common to the single carrier and multicarrier signal segments ..... 707
Figure 19-9—Spectral shaping achieved by OFDM symbol onset and termination shaping ..... 708
Figure 19-10—Subcarrier spectrums for rectangular windowing and Clause 17 suggested windowing ..... 708
Figure 19-11—Foundational brickwall filter ..... 709
Figure 19-12—Continuous time Hanning window ..... 710
Figure 19-13—Specified pulse ..... 710
Figure 19-14—Single carrier frequency response ..... 711
Figure 19-15—Comparing signal power ..... 711
Figure 19-16—Aligning the 11 MHz and 20 MHz clocks ..... 712
Figure 19-17—Single carrier to OFDM time alignment ..... 712
Figure 19-18—Single carrier termination requirement ..... 713
Figure 19-19—Carrier frequency coherency shall be maintained ..... 713
Figure 19-20—The phase of the first OFDM segment symbol is established by the last Barker symbol ..... 714
Figure 19-21—BPSK and QPSK signaling with the I/Q channels maximally energized ..... 714
Figure H.1—Randomness generating circuit ..... 1133
Figure I.1—Transmit spectrum mask ..... 1144
Figure I.2—Transmit spectrum masks for the U.S. 4.9 GHz public safety band ..... 1145
Figure K.1—Schedule for stream from STA i ..... 1156
Figure K.2—Schedule for streams from STAs i to k ..... 1156
Figure K.3—Reallocation of TXOPs when a stream is dropped ..... 1157
Figure L.1—Virtual bitmap example #1 ..... 1159
Figure L.2—Virtual bitmap example #2 ..... 1159
Figure L.3—Virtual bitmap example #3 ..... 1160
Figure N.1—Very high level UML use case diagram for the AP ..... 1170
Figure N.2—Very high level UML use case diagram for the WLAN system ..... 1170
Figure N.3—High-level UML use case diagram for the WLAN system ..... 1171
Figure N.4—High-level UML entity diagram for the WLAN system ..... 1172
Figure N.5—AP UML composition diagram (alternate syntax) ..... 1173
Figure N.6—High-level UML use case diagram for the AP ..... 1174


Copyright © 2007 IEEE. All rights reserved.

Figure O.1—Location of the DS SAP ..... 1177

Copyright © 2007 IEEE. All rights reserved.


Samsung Ex. 1044, Page 43 of 1232

# List of tables

Table 7-1—Valid type and subtype combinations ..... 61
Table 7-2—To/From DS combinations in data frames ..... 62
Table 7-3—Duration/ID field encoding ..... 64
Table 7-4—QoS Control field ..... 67
Table 7-5—TID subfield ..... 67
Table 7-6—Ack Policy subfield in QoS Control field of QoS data frames ..... 68
Table 7-7—Address field contents ..... 77
Table 7-8—Beacon frame body ..... 80
Table 7-9—Disassociation frame body ..... 82
Table 7-10—Association Request frame body ..... 82
Table 7-11—Association Response frame body ..... 82
Table 7-12—Reassociation Request frame body ..... 83
Table 7-13—Reassociation Response frame body ..... 84
Table 7-14—Probe Request frame body ..... 84
Table 7-15—Probe Response frame body ..... 85
Table 7-16—Authentication frame body ..... 86
Table 7-17—Presence of challenge text information element ..... 86
Table 7-18—Deauthentication frame body ..... 86
Table 7-19—Action frame body ..... 87
Table 7-20—STA usage of QoS, CF-Pollable, and CF-Poll Request ..... 88
Table 7-21—AP usage of QoS, CF-Pollable, and CF-Poll Request ..... 89
Table 7-22—Reason codes ..... 92
Table 7-23—Status codes ..... 94
Table 7-24—Category values ..... 96
Table 7-25—Settings of the Max SP Length subfield ..... 99
Table 7-26—Element IDs ..... 100
Table 7-27—Coverage Class field parameters ..... 106
Table 7-28—Summary of use of Enable, Request, and Report bits ..... 116
Table 7-29—Measurement Type definitions for measurement requests ..... 116
Table 7-30—Measurement Type definitions for measurement reports ..... 119
Table 7-31—RPI definitions for an RPI histogram report ..... 122
Table 7-32—Cipher suite selectors ..... 125
Table 7-33—Cipher suite usage ..... 126
Table 7-34—AKM suite selectors ..... 126
Table 7-35—PTKSA/GTKSA/STKSA replay counters usage ..... 128
Table 7-36—ACI-to-AC coding ..... 131
Table 7-37—Default EDCA Parameter Set element parameter values ..... 131
Table 7-38—Direction subfield encoding ..... 132
Table 7-39—Access Policy subfield ..... 133
Table 7-40—TS Info Ack Policy subfield encoding ..... 133
Table 7-41—Setting of Schedule subfield ..... 134
Table 7-42—Frame classifier type ..... 136
Table 7-43—Encoding of Processing subfield ..... 139
Table 7-44—Spectrum management Action Value field values ..... 140
Table 7-45—QoS Action field values ..... 143
Table 7-46—ADDTS Request frame body ..... 143
Table 7-47—ADDTS Response frame body ..... 144
Table 7-48—DELTS frame body ..... 145
Table 7-49—Schedule frame body ..... 146
Table 7-50—DLS Action field values ..... 146
Table 7-51—DLS Request frame body ..... 146


Copyright © 2007 IEEE. All rights reserved.

Table 7-52—DLS Response frame body ..... 147
Table 7-53—DLS Teardown frame body ..... 148
Table 7-54—Block Ack Action field values ..... 149
Table 7-55—ADDBA Request frame body ..... 149
Table 7-56—ADDBA Response frame body ..... 150
Table 7-57—DELBA frame body ..... 150
Table 7-58—Frame subtype usage by BSS type, MAC entity type, and coordination function ..... 151
Table 8-1—AAD length ..... 182
Table 8-2—Cipher suite key lengths ..... 205
Table 8-3—Key RSC field ..... 206
Table 8-4—KDE ..... 206
Table 8-5—MUI values ..... 208
Table 8-6—SMK error types ..... 209
Table 9-1—UP-to-AC mappings ..... 253
Table 9-2—Modulation classes ..... 281
Table 9-3—HCC family – N = 11; Family indices (SEQ) 1 through 10 ..... 284
Table 9-4—EHCC family – Code length = 9, N = 11; Family Indices (SEQ) 1 through 9 ..... 285
Table 9-5—EHCC family – Code length = 8, N = 11; Family indices (SEQ) 1 through 8 ..... 285
Table 9-6—Attributes applicable to frame exchange sequence definition ..... 307
Table 10-1—Supported TS Management primitives ..... 375
Table 11-1—Power Management modes ..... 426
Table 11-2—Encoding of ResultCode to Status Code field value ..... 446
Table 11-3—Encoding of ReasonCode to Reason Code field value for DELTS ..... 448
Table 11-4—Encoding of ResultCode to Status Code field value ..... 452
Table 11-5—Encoding of ReasonCode to Reason Code field value for DELBA ..... 453
Table 11-6—Mapping of Status Code field value to ResultCode ..... 458
Table 11-7—Encoding of ReasonCode to Reason Code field value for DLS teardown ..... 459
Table 11-8—Allowed measurement requests ..... 465
Table 12-1—PHY-SAP peer-to-peer service primitives ..... 470
Table 12-2—PHY-SAP sublayer-to-sublayer service primitives ..... 470
Table 12-3—PHY-SAP service primitive parameters ..... 470
Table 12-4—Vector descriptions ..... 471
Table 14-1—TXVECTOR parameters ..... 488
Table 14-2—RXVECTOR parameters ..... 489
Table 14-3—PSF bit descriptions ..... 491
Table 14-4—PLCP field bit descriptions ..... 495
Table 14-5—PMD_SAP peer-to-peer service primitives ..... 505
Table 14-6—PMD_SAP sublayer-to-sublayer service primitives ..... 506
Table 14-7—List of parameters for PMD primitives ..... 506
Table 14-8—Transmit power levels ..... 512
Table 14-9—Operating frequency range ..... 516
Table 14-10—Number of operating channels ..... 517
Table 14-11—Requirements in China, North America and Europe (excluding Spain and France; values specified in GHz) ..... 517
Table 14-12—Requirements in Japan (values specified in GHz) ..... 518
Table 14-13—Requirements in Spain (values specified in GHz) ..... 518
Table 14-14—Requirements in France (values specified in GHz) ..... 519
Table 14-15—Base-Hopping sequence b(i) for China, North America and most of Europe ..... 520
Table 14-16—Base-Hopping sequence b(i) for Spain ..... 520
Table 14-17—Base-Hopping sequence b(i) for France ..... 521
Table 14-18—Symbol encoding into carrier deviation (1 Mb/s, 2GFSK) ..... 522
Table 14-19—1 Mb/s Dp ..... 526
Table 14-20—Symbol encoding into carrier deviation ..... 527
Table 14-21—2 Mb/s Dp ..... 529

Copyright © 2007 IEEE. All rights reserved.

Table 14-22—FHSS PHY attributes ..... 529
Table 14-23—Regulatory domain codes ..... 531
Table 14-24—Supported data rate codes (dot11SupportedDataRatesTX) ..... 532
Table 14-25—Supported data rate codes (dot11SupportedDataRatesRX) ..... 532
Table 14-26—Number of transmit antennas ..... 532
Table 14-27—Number of receive antennas ..... 533
Table 14-28—Diversity support codes ..... 533
Table 14-29—Diversity select antenna codes ..... 534
Table 14-30—Transmit power levels ..... 534
Table 14-31—FH PHY characteristics ..... 536
Table 15-1—MIB attribute default values/ranges ..... 546
Table 15-2—DS PHY characteristics ..... 547
Table 15-3—PMD_SAP peer-to-peer service primitives ..... 548
Table 15-4—DSSS PMD_SAP peer-to-peer service primitives ..... 549
Table 15-5—PMD_SAP sublayer-to-sublayer service primitives ..... 549
Table 15-6—List of parameters for the PMD primitives ..... 550
Table 15-7—DSSS PHY frequency channel plan ..... 566
Table 15-8—1 Mb/s DBPSK encoding table ..... 567
Table 15-9—2 Mb/s DQPSK encoding table ..... 567
Table 16-1—IR PMD_SAP peer-to-peer service primitives ..... 580
Table 16-2—Sixteen-PPM basic rate mapping ..... 581
Table 16-3—Four-PPM enhanced rate mapping ..... 582
Table 16-4—Peak optical power as a function of emitter radiation pattern mask ..... 583
Table 16-5—Definition of the emitter radiation pattern Mask 1 ..... 584
Table 16-6—Definition of emitter radiation pattern Mask 2 ..... 584
Table 16-7—Definition of the receiver FOV ..... 587
Table 16-8—IR PHY MIB attributes ..... 588
Table 16-9—IR PHY characteristics ..... 589
Table 17-1—TXVECTOR parameters ..... 593
Table 17-2—RXVECTOR parameters ..... 593
Table 17-3—Modulation-dependent parameters ..... 597
Table 17-4—Timing-related parameters ..... 597
Table 17-5—Contents of the SIGNAL field ..... 602
Table 17-6—Modulation-dependent normalization factor $K_{MOD}$ ..... 607
Table 17-7—BPSK encoding table ..... 609
Table 17-8—QPSK encoding table ..... 609
Table 17-9—16-QAM encoding table ..... 609
Table 17-10—64-QAM encoding table ..... 609
Table 17-11—Major parameters of the OFDM PHY ..... 612
Table 17-12—Allowed relative constellation error versus data rate ..... 615
Table 17-13—Receiver performance requirements ..... 617
Table 17-14—MIB attribute default values/ranges ..... 623
Table 17-15—OFDM PHY characteristics ..... 626
Table 17-16—PMD_SAP peer-to-peer service primitives ..... 628
Table 17-17—PMD_SAP sublayer-to-sublayer service primitives ..... 628
Table 17-18—List of parameters for the PMD primitives ..... 628
Table 18-1—SERVICE field definitions ..... 641
Table 18-2—Example of LENGTH calculations for CCK ..... 642
Table 18-3—Example of LENGTH calculations for PBCC ..... 643
Table 18-4—MIB attribute default values/ranges ..... 653
Table 18-5—High Rate PHY characteristics ..... 654
Table 18-6—Parameter vectors ..... 655
Table 18-7—PMD_SAP peer-to-peer service primitives ..... 657
Table 18-8—PMD_SAP sublayer-to-sublayer service primitives ..... 657


Copyright © 2007 IEEE. All rights reserved.

Table 18-9—High Rate PHY frequency channel plan ..... 673
Table 18-10—1 Mb/s DBPSK encoding table ..... 674
Table 18-11—2 Mb/s DQPSK encoding table ..... 674
Table 18-12—DQPSK encoding table ..... 676
Table 18-13—5.5 Mb/s CCK encoding table ..... 676
Table 18-14—QPSK encoding table ..... 676
Table 18-15—China and North American operating channels ..... 680
Table 18-16—European operating channels (except France and Spain) ..... 680
Table 18-17—China and North American Set 1 hop patterns ..... 681
Table 18-18—European Set 1 hop patterns (except France and Spain) ..... 681
Table 19-1—TXVECTOR parameters ..... 691
Table 19-2—RXVECTOR parameters ..... 692
Table 19-3—SERVICE field bit definitions ..... 693
Table 19-4—Example of LENGTH calculations for ERP-PBCC-22 ..... 694
Table 19-5—CCA parameters ..... 703
Table 19-6—MIB attribute default values/ranges ..... 715
Table 19-7—ERP characteristics ..... 719
Table 19-8—PMD_SAP peer-to-peer services ..... 720
Table 19-9—PMD_SAP sublayer-to-sublayer services ..... 721
Table 19-10—List of parameters for the PMD primitives ..... 721
Table G.1—The message ..... 1080
Table G.2—Frequency domain representation of the short sequences ..... 1081
Table G.3—One period of IFFT of the short sequences ..... 1081
Table G.4—Time domain representation of the short sequence ..... 1082
Table G.5—Frequency domain representation of the long sequences ..... 1084
Table G.6—Time domain representation of the long sequence ..... 1084
Table G.7—Bit assignment for SIGNAL field ..... 1086
Table G.8—SIGNAL field bits after encoding ..... 1086
Table G.9—SIGNAL field bits after interleaving ..... 1087
Table G.10—Frequency domain representation of SIGNAL field ..... 1087
Table G.11—Frequency domain representation of SIGNAL field with pilots inserted ..... 1088
Table G.12—Time domain representation of SIGNAL field ..... 1089
Table G.13—First 144 DATA bits ..... 1090
Table G.14—Last 144 DATA bits ..... 1091
Table G.15—Scrambling sequence for seed 1011101 ..... 1092
Table G.16—First 144 bits after scrambling ..... 1092
Table G.17—Last 144 bits after scrambling ..... 1093
Table G.18—Coded bits of first DATA symbol ..... 1094
Table G.19—First permutation ..... 1095
Table G.20—Second permutation ..... 1096
Table G.21—Interleaved bits of first DATA symbol ..... 1097
Table G.22—Frequency domain of first DATA symbol ..... 1098
Table G.23—Polarity of the pilot subcarriers ..... 1099
Table G.24—The entire packet ..... 1100
Table H.1—Test vectors for block function ..... 1119
Table H.2—Test vectors for Michael ..... 1119
Table H.3—Notation example ..... 1133
Table H.4—Sample plaintext MPDU ..... 1134
Table H.5—ARC4 encryption ..... 1134
Table H.6—Expanded MPDU after WEP encapsulation ..... 1134
Table H.7—Sample TKIP parameters ..... 1135
Table H.8—Sample plaintext and ciphertext MPDUs, using parameter from Table H.7 ..... 1135
Table H.9—RSN PRF Test Vector 1 ..... 1136
Table H.10—RSN PRF Test Vector 2 ..... 1137

Copyright © 2007 IEEE. All rights reserved.

Table H.11—RSN PRF Test Vector 3 ..... 1137
Table H.12—RSN PRF Test Vector 4 ..... 1137
Table H.13—Sample values for pairwise key derivations ..... 1138
Table H.14—Sample derived CCMP temporal key (TK) ..... 1138
Table H.15—Sample derived PTK ..... 1139
Table I.1—Regulatory requirement list ..... 1141
Table I.2—Emissions limits sets ..... 1142
Table I.3—Behavior limits sets ..... 1142
Table I.4—Transmit power level by regulatory domain ..... 1143
Table I.5—U.S. public safety transmit power levels by regulatory domain ..... 1143
Table I.6—Japanese transmit power levels by regulatory domain ..... 1144
Table J.1—Regulatory classes for 4.9 GHz and 5 GHz bands in the United States ..... 1148
Table J.2—Regulatory classes for 5 GHz bands in Europe ..... 1148
Table J.3—Regulatory classes for 4.9 GHz and 5 GHz bands in Japan ..... 1149
Table K.1—Admissible TSPECs ..... 1151
Table M.1—IEEE 802.11 integration service STT ..... 1165
Table M.2—Ethernet/IEEE 802.3 to IEEE 802.11 translation ..... 1166
Table M.3—IEEE 802.11 to Ethernet/IEEE 802.3 translation ..... 1166


Copyright © 2007 IEEE. All rights reserved.

# 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

## 1. Overview

### 1.1 Scope

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

### 1.2 Purpose

The purpose of this standard is to provide wireless connectivity to automatic machinery, equipment, or STAs that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area. This standard also offers regulatory bodies a means of standardizing access to one or more frequency bands for the purpose of local area communication.

Specifically, this standard

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

— Defines the MAC procedures to support the asynchronous MAC service data unit (MSDU) delivery services.

— Defines several PHY signaling techniques and interface functions that are controlled by the IEEE 802.11 MAC.

— Permits the operation of an IEEE 802.11-conformant device within a wireless local area network (WLAN) that may coexist with multiple overlapping IEEE 802.11 WLANs.

— Describes the requirements and procedures to provide data confidentiality of user information being transferred over the wireless medium (WM) and authentication of IEEE 802.11-conformant devices.

— Defines mechanisms for dynamic frequency selection (DFS) and transmit power control (TPC) that may be used to satisfy regulatory requirements for operation in the 5 GHz band. The regulations and conformance tests are listed in Clause 2.

Copyright © 2007 IEEE. All rights reserved.

— Defines the MAC procedures to support local area network (LAN) applications with quality of service (QoS) requirements, including the transport of voice, audio, and video.


Copyright © 2007 IEEE. All rights reserved.

## 2. Normative references

The following referenced documents are indispensable for the application of this standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ETSI EN 301 893, Broadband Radio Access Networks (BRAN); 5 GHz high performance RLAN; Part 2: Harmonized EN covering essential requirements of article 3.2 of the R&TTE Directive.[1]

FIPS PUB 180-1-1995, Secure Hash Standard.[2]

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

IEEE Std 802®-2001, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[3, 4]

IEEE Std 802.1X™-2004, IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control.[5]

IEEE Std C95.1™-1991 (Reaff 1997), IEEE Standard Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz.

IETF RFC 1321, The MD5 Message-Digest Algorithm, April 1992 (status: informational).[6]

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

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

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

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

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

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

---

[1]ETSI documents are available from ETSI, 650 Route des Lucioles, F-06921 Sopia Antipolis Cedex, France (http://www.etsi.org).

[2]FIPS publications are available from the National Technical Information Service (NTIS), U.S. Dept. of Commerce, 5285 Port Royal Road, Springfield, VA 22161 (http://www.ntis.org/).

[3]"IEEE" and "802" are registered trademarks in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc.

[4]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://www.standards.ieee.org/).

[5]The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Inc.

[6]Internet RFCs are available from the Internet Engineering Task Force at http://www.ietf.org/.

[7]ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). They are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 51 of 1232

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

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

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

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

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

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

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

ISO/IEC Technical Report 11802-5:1997(E), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 5: Medium Access Control (MAC) Bridging of Ethernet V2.0 in Local Area Networks (previously known as IEEE Std 802.1H-1997 [B13][8]).

ISO/IEC 14977:1996, Information technology—Information technology. Syntactic Metalanguage. Extended BNF.

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

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.

ITU Radio Regulations, volumes 1–4.[9]

ITU-T Recommendation V.41 (11/88), Code-independent error-control system.

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

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

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

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

[9]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).


Copyright © 2007 IEEE. All rights reserved.

## 3. Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms* [B11] should be referenced for terms not defined in this clause.

**3.1 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 (STA) to contend for the channel in order to transmit medium access control (MAC) service data units (MSDUs) with certain priorities.

**3.2 access control:** The prevention of unauthorized usage of resources.

**3.3 access point (AP):** Any entity that has station (STA) functionality and provides access to the distribution services, via the wireless medium (WM) for associated STAs.

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

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

**3.6 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.

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

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

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

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

**3.11 Authentication Server (AS):** An entity that provides an authentication service to an Authenticator. This service determines, from the credentials provided by the Supplicant, whether the Supplicant is authorized to access the services provided by the Authenticator. (IEEE Std 802.1X-2004[10])

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

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

**3.14 authorized:** To be explicitly allowed.

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

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

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 53 of 1232

**3.16 basic service set (BSS):** A set of stations (STAs) that have successfully synchronized using the JOIN service primitives[11] and one STA that has used the START primitive. Membership in a BSS does not imply that wireless communication with all other members of the BSS is possible.

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

**3.18 broadcast address:** A unique multicast address that specifies all stations (STAs).

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

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

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

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

**3.23 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).

**3.24 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.

**3.25 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.

**3.26 coordination function:** The logical function that determines when a station (STA) operating within a basic service set (BSS) is permitted to transmit 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) BSS will have one DCF and one HCF.

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

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

NOTE—See IETF RFC 3610.[12]

[11]Description of these primitives can be found in 10.3.3.

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


Copyright © 2007 IEEE. All rights reserved.

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

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

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

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

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

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

**3.35 directed frame:** *See:* **unicast frame**.

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

**3.37 disassociation service:** The service that removes an existing association.

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

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

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

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

**3.42 distribution system service (DSS):** The set of services provided by the distribution system (DS) that enable the medium access control (MAC) to transport MAC service data units (MSDUs) between stations (STAs) that are not in direct communication with each other over a single instance of the wireless medium (WM). These services include transport of MSDUs between the access points (APs) of basic service sets (BSSs) within an extended service set (ESS), transport of MSDUs between portals and BSSs within an ESS, and transport of MSDUs between STAs in the same BSS in cases where the MSDU has a multicast or broadcast destination address or where the destination is an individual address and the STA is associated with an AP. DSSs are provided between pairs of IEEE 802.11 MACs.

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

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

Copyright © 2007 IEEE. All rights reserved.

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

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

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

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

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

**3.50 encapsulation:** The process of generating an protected frame by encapsulating plaintext data.

**3.51 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 (STAs) in a QoS basic service set (BSS). This access mechanism is also used by the QoS access point (AP) and operates concurrently with hybrid coordination function (HCF) controlled channel access (HCCA).

**3.52 enhanced distributed channel access function (EDCAF):** A logical function in a quality of service (QoS) station (STA) 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.

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

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

**3.55 4-Way Handshake:** A pairwise key management protocol defined by this standard. This handshake confirms mutual possession of a pairwise master key (PMK) by two parties and distributes a group temporal key (GTK).

**3.56 4-Way station-to-station link (STSL) transient key (STK) Handshake:** A key management protocol between two parties that confirms mutual possession of an STSL master key (SMK) and distributes an STK.

**3.57 fragmentation:** The process of segmenting 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. These processes are described in 5.8.1.9 of ISO/IEC 7498-1:1994.

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

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


Copyright © 2007 IEEE. All rights reserved.

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

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

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

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

**3.64 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

**3.65 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 (STAs) with prioritized and parameterized QoS access to the wireless medium (WM), while continuing to support non-QoS STAs 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 STAs may use during both the contention period (CP) and the contention-free period (CFP).

**3.66 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 (STAs). The HC is collocated with a QoS access point (AP).

**3.67 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 (STAs) for downlink unicast, uplink, and direct-link transmissions.

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

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

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

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

**3.72 integration service:** The service that enables delivery of medium access control (MAC) service data units (MSDUs) between the distribution system (DS) and a non-IEEE-802.11 local area network (LAN) (via a portal).

Copyright © 2007 IEEE. All rights reserved.

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

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

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

**3.76 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).

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

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

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

**3.80 master session key (MSK):** Keying material that is derived between the Extensible Authentication Protocol (EAP) peer and exported by the EAP method to the Authentication Server (AS). This key is at least 64 octets in length.

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

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

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

**3.84 message integrity code (MIC):** A value generated by a cryptographic function. If the input data are changed, a new value cannot be correctly computed without knowledge of the cryptographic key(s) used by the cryptographic function. This is traditionally called a *message authentication code* (MAC), but the acronym MAC is already reserved for another meaning in this standard.

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

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

**3.87 multicast:** When applied to a medium access control (MAC) service data unit (MSDU), it is an MSDU with a multicast address as the destination address (DA). When applied to a MAC protocol data unit (MPDU) or control frame, it is an MPDU or control frame with a multicast address as the receiver address (RA).

**3.88 multicast address:** A medium access control (MAC) address that has the group bit set.

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


Copyright © 2007 IEEE. All rights reserved.

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

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

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

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

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

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

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

**3.97 pairwise master key (PMK):** The highest order key used within this standard. The PMK may be derived from a key generated by an Extensible Authentication Protocol (EAP) method or may be obtained directly from a preshared key (PSK).

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

**3.99 pairwise transient key (PTK):** A value that is derived from the pairwise master key (PMK), Authenticator address (AA), Supplicant address (SPA), Authenticator nonce (ANonce), and Supplicant nonce (SNonce) using the pseudo-random function (PRF) and that is split up into as many as five keys, i.e., temporal encryption key, two temporal message integrity code (MIC) keys, EAPOL-Key encryption key (KEK), EAPOL-Key confirmation key (KCK).

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

**3.101 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.

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

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

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 59 of 1232

**3.104 per-frame encryption key:** A unique encryption key constructed for each medium access control (MAC) protocol data unit (MPDU), employed by some IEEE 802.11 security protocols.

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

**3.106 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.

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

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

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

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

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

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

**3.113 preshared key (PSK):** A static key that is distributed to the units in the system by a method outside the scope of this standard, always by some out-of-band means.

**3.114 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.

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

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

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

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

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


Copyright © 2007 IEEE. All rights reserved.

**3.120 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.

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

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

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

**3.124 receive power:** Mean power measured at the antenna connector.

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

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

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

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

**3.129 robust-security-network-association- (RSNA-) enabled equipment:** A station (STA) when it is RSNA-capable and dot11RSNAEnabled is set to TRUE.

**3.130 robust security network association (RSNA) key management:** Key management that includes the 4-Way Handshake, the Group Key Handshake, and the PeerKey Handshake.

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

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

**3.133 selector:** An item specifying a list constituent in an IEEE 802.11 Management Message information element.

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

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

Copyright © 2007 IEEE. All rights reserved.

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

**3.136 station (STA):** Any device that contains an IEEE 802.11-conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM).

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

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

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

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

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

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

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

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

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

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

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

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

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


Copyright © 2007 IEEE. All rights reserved.

**3.150 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).

**3.151 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 (AP), 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 (STA) with multiple streams. However, such classification is beyond the scope of this standard.

**3.152 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; eight identify TCs, and the other eight identify parameterized TSs. The TID is assigned to an MSDU in the layers above the MAC.

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

**3.154 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).

**3.155 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.

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

**3.157 transmission opportunity (TXOP):** An interval of time when a particular quality of service (QoS) station (STA) 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 STA by successfully contending for the channel or assigned by the hybrid coordinator (HC).

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

**3.159 transmit power:** The effective isotropic radiated power (EIRP) when referring to the operation of a 5 GHz IEEE 802.11 orthogonal frequency division multiplexing (OFDM) physical layer (PHY) in a country where so regulated.

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

Copyright © 2007 IEEE. All rights reserved.

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

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

**3.163 unicast:** When applied to a medium access control (MAC) service data unit (MSDU), it is an MSDU with a single recipient address as the destination address (DA). When applied to a MAC protocol data unit (MPDU) or control frame, it is an MPDU or control frame with a single recipient address as the receiver address (RA).

**3.164 unicast address:** A medium access control (MAC) address that does not have the group bit set. *Syn:* **directed address**, **individual address**.

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

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

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

**3.168 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.

**3.169 wired equivalent privacy (WEP):** A deprecated cryptographic data confidentiality algorithm specified by IEEE Std 802.11 that may be used to provide data confidentiality that is subjectively equivalent to the data confidentiality of a wired local area network (LAN) medium that does not employ cryptographic techniques to enhance data confidentiality.

**3.170 wireless distribution system (WDS):** A mechanism for wireless communication using a four address frame format specified in this standard. This standard describes such a frame format, but does not describe how such a mechanism or frame format would be used.

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

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


Copyright © 2007 IEEE. All rights reserved.

## 4. Abbreviations and acronyms

| | |
|---|---|
| AA | Authenticator address |
| AAD | additional authentication data |
| AC | access category |
| ACI | access category index |
| ACK | acknowledgment |
| ACM | admission control mandatory |
| ACU | admission control unit |
| ADDBA | add Block Acknowledgment |
| ADDTS | add traffic stream |
| AES | advanced encryption standard |
| AID | association identifier |
| AIFS | arbitration interframe space |
| AIFSN | arbitration interframe space number |
| AKM | authentication and key management |
| AKMP | Authentication and Key Management Protocol |
| ANonce | Authenticator nonce |
| AP | access point |
| APSD | automatic power-save delivery |
| ARP | Address Resolution Protocol |
| AS | Authentication Server |
| ASN.1 | Abstract Syntax Notation One |
| ATIM | announcement traffic indication message |
| BA | Block Acknowledgment |
| BAR | Block Acknowledgment request |
| BCC | binary convolutional code |
| BPSK | binary phase shift keying |
| BSA | basic service area |
| BSS | basic service set |
| BSSID | basic service set identification |
| BT | bit time |
| CAP | controlled access phase |
| CBC | cipher-block chaining |
| CBC-MAC | cipher-block chaining message authentication code |
| CCA | clear channel assessment |
| CCK | complementary code keying |
| CCM | CTR with CBC-MAC |
| CCMP | CTR with CBC-MAC Protocol |
| CF | contention free |
| CFP | contention-free period |
| C-MPDU | coded MPDU |
| C-PSDU | coded PSDU |
| CP | contention period |
| CRC | cyclic redundancy code |
| CS | carrier sense |

Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| CSMA/CA | carrier sense multiple access with collision avoidance |
| CTR | counter mode |
| CTS | clear to send |
| CW | contention window |
| DA | destination address |
| DBPSK | differential binary phase shift keying |
| DCF | distributed coordination function |
| DCLA | dc level adjustment |
| DELBA | delete Block Acknowledgment |
| DELTS | delete traffic stream |
| DFS | dynamic frequency selection |
| DIFS | distributed (coordination function) interframe space |
| DLL | data link layer |
| DLS | direct-link setup |
| Dp | desensitization |
| DQPSK | differential quadrature phase shift keying |
| DR | data rate |
| DS | distribution system |
| DSCP | differentiated services code point |
| DSM | distribution system medium |
| DSS | distribution system service |
| DSSDU | distribution system service data unit |
| DSSS | direct sequence spread spectrum |
| DSSS-OFDM | PHYs using DSSS-OFDM modulation under 19.7 rules |
| DTIM | delivery traffic indication message |
| EAP | Extensible Authentication Protocol (IETF RFC 3748-2004 [B26]) |
| EAPOL | Extensible Authentication Protocol over LANs (IEEE Std 802.1X-2004) |
| ED | energy detection |
| EDCA | enhanced distributed channel access |
| EDCAF | enhanced distributed channel access function |
| EHCC | extended hyperbolic congruence code |
| EIFS | extended interframe space |
| EIRP | equivalent isotropically radiated power |
| EOSP | end of service period |
| ERP | extended rate PHY conforming to Clause 19 |
| ERP-CCK | PHYs using CCK modulation under Clause 19 rules |
| ERP-DSSS | PHYs using DSSS modulation under Clause 19 rules |
| ERP-DSSS/CCK | PHYs using DSSS or CCK modulation under Clause 19 rules |
| ERP-OFDM | PHYs using OFDM modulation under 19.5 rules |
| ERP-PBCC | PHYs using extended rate PBCC modulation under 19.6 rules |
| ESA | extended service area |
| ESS | extended service set |
| EVM | error vector magnitude |
| FC | frame control |
| FCS | frame check sequence |
| FER | frame error ratio |


Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| FFT | Fast Fourier Transform |
| FH | frequency hopping |
| FHSS | frequency-hopping spread spectrum |
| FIFO | first in first out |
| FOV | field of view |
| GFSK | Gaussian frequency shift key or keying |
| GMK | group master key |
| GNonce | group nonce |
| GTK | group temporal key |
| GTKSA | group temporal key security association |
| GI | guard interval |
| HC | hybrid coordinator |
| HCC | hyperbolic congruence code |
| HCCA | HCF controlled channel access |
| HCF | hybrid coordination function |
| HEC | header error check |
| HEMM | HCCA, EDCA mixed mode |
| HIPERLAN | high-performance radio local area network |
| HR/DSSS | High Rate direct sequence spread spectrum using the long preamble and header |
| HR/DSSS/short | High Rate direct sequence spread spectrum using the optional short preamble and header mode |
| HR/DSSS/PBCC | High Rate direct sequence spread spectrum using the optional packet binary convolutional coding mode and the long preamble and header |
| HR/DSSS/PBCC/short | High Rate direct sequence spread spectrum using the optional packet binary convolutional coding mode and the optional short preamble and header |
| IBSS | independent basic service set |
| ICMP | Internet Control Message Protocol |
| ICV | integrity check value |
| IFFT | inverse Fast Fourier Transform |
| IFS | interframe space |
| IMp | intermodulation protection |
| INonce | initiator nonce |
| IR | infrared |
| IrDA | infrared data association |
| ISM | industrial, scientific, and medical |
| IUT | implementation under test |
| IV | initialization vector |
| KCK | EAPOL-Key confirmation key |
| KDE | key data encapsulation |
| KEK | EAPOL-Key encryption key |
| LAN | local area network |
| LED | light-emitting diode |
| LFSR | linear feedback shift register |
| LLC | logical link control |
| LME | layer management entity |
| LRC | long retry count |
| LSB | least significant bit |

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 67 of 1232

| | |
|---|---|
| MAC | medium access control |
| MAC_I | initiator mac address |
| MAC_P | peer mac address |
| MIB | management information base |
| MIC | message integrity code |
| MLME | MAC sublayer management entity |
| MMPDU | MAC management protocol data unit |
| MPDU | MAC protocol data unit |
| MSB | most significant bit |
| MSDU | MAC service data unit |
| MSK | master session key |
| MUI | message unique identifier |
| N/A | not applicable |
| NAV | network allocation vector |
| NonERP | nonextended rate PHY conforming to Clause 15 or Clause 18, but not to Clause 19 |
| NTP | Network Time Protocol (IETF RFC 1305-1992 [B16]) |
| OFDM | orthogonal frequency division multiplexing |
| OSI | Open System Interconnection |
| OUI | organizationally unique identifier |
| PAE | port access entity (IEEE Std 802.1X-2004) |
| PBCC | packet binary convolutional code |
| PC | point coordinator |
| PCF | point coordination function |
| PDU | protocol data unit |
| PER | packet error rate |
| PHY | physical layer |
| PHYCS | PHY carrier sense |
| PHYED | PHY energy detection |
| PICS | protocol implementation conformance statement |
| PIFS | point (coordination function) interframe space |
| PLCP | physical layer convergence procedure |
| PLME | physical layer management entity |
| PLW | PSDU length word |
| PMD | physical medium dependent |
| PMK | pairwise master key |
| PMKID | pairwise master key identifier |
| PMKSA | pairwise master key security association |
| PN | packet number |
| PN | pseudonoise (code sequence) |
| PNonce | peer nonce |
| PPDU | PLCP protocol data unit |
| PPM | pulse position modulation |
| PRF | pseudo-random function |
| PRN | pseudo-random number |
| PRNG | pseudo-random number generator |
| PS | power save (mode) |


Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| PSDU | PLCP service data unit |
| PSF | PLCP Signaling field |
| PSK | preshared key |
| PTK | pairwise transient key |
| PTKSA | pairwise transient key security association |
| QAM | quadrature amplitude modulation |
| QLRC | QoS long retry counter |
| QoS | quality of service |
| QPSK | quadrature phase shift keying |
| QSRC | QoS short retry counter |
| RA | receiver address or receiving station address |
| RADIUS | remote authentication dial-in user service (IETF RFC 2865-2000 [B23]) |
| RF | radio frequency |
| RLAN | radio local area network |
| RPI | receive power indicator |
| RSC | receive sequence counter |
| RSN | robust security network |
| RSNA | robust security network association |
| RSSI | receive signal strength indicator |
| RTS | request to send |
| RX | receive or receiver |
| SA | source address |
| SAP | service access point |
| S-APSD | scheduled automatic power-save delivery |
| SDL | specification and description language |
| SDU | service data unit |
| SFD | start frame delimiter |
| SKCK | STSL key confirmation key |
| SKEK | STSL key encryption key |
| SI | service interval |
| SIFS | short interframe space |
| SLRC | station long retry count |
| SME | station management entity |
| SMK | STSL master key |
| SMKSA | STSL master key security association |
| SMT | station management |
| SNAP | Sub-Network Access Protocol |
| SNonce | Supplicant nonce |
| SP | service period |
| SPA | Supplicant address |
| SQ | signal quality (PN code correlation strength) |
| SRC | short retry count |
| SS | station service |
| SSID | service set identifier |
| SSRC | station short retry count |
| STA | station |

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 69 of 1232

| | |
|---|---|
| STK | STSL transient key |
| STKSA | STSL transient key security association |
| STSL | station-to-station link |
| STT | selective translation table |
| SYNC | synchronization |
| TA | transmitter address or transmitting station address |
| TBTT | target beacon transmission time |
| TC | traffic category |
| TCLAS | traffic classification |
| TID | traffic identifier |
| TIM | traffic indication map |
| TKIP | Temporal Key Integrity Protocol |
| TPC | transmit power control |
| TS | traffic stream |
| TSC | TKIP sequence counter |
| TSF | timing synchronization function |
| TSID | traffic stream identifier |
| TSN | transition security network |
| TSPEC | traffic specification |
| TTAK | TKIP-mixed transmit address and key |
| TU | time unit |
| TX | transmit or transmitter |
| TXE | transmit enable |
| TXOP | transmission opportunity |
| U-APSD | unscheduled automatic power-save delivery |
| UCT | unconditional transition |
| U-NII | unlicensed national information infrastructure |
| UP | user priority |
| WLAN | wireless local area network |
| WDS | wireless distribution system |
| WEP | wired equivalent privacy |
| WM | wireless medium |


Copyright © 2007 IEEE. All rights reserved.

# 5. General description

## 5.1 General description of the architecture

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

### 5.1.1 How WLAN systems are different

Wireless networks have fundamental characteristics that make them significantly different from traditional wired LANs. Some countries impose specific requirements for radio equipment in addition to those specified in this standard. This standard does not provide information to meet these country-specific radio regulations.

#### 5.1.1.1 Destination address does not equal destination location

In wired LANs, an address is equivalent to a physical location. This is implicitly assumed in the design of wired LANs. In IEEE Std 802.11, the addressable unit is a station (STA). The STA is a message destination, but not (in general) a fixed location.

#### 5.1.1.2 Media impact on design and performance

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

a) Use a medium that has neither absolute nor readily observable boundaries outside of which STAs with conformant PHY transceivers are known to be unable to receive network frames

b) Are unprotected from other signals that may be sharing the medium

c) Communicate over a medium significantly less reliable than wired PHYs

d) Have dynamic topologies

e) Lack full connectivity, and therefore the assumption normally made that every STA can hear every other STA is invalid (i.e., STAs may be “hidden” from each other)

f) Have time-varying and asymmetric propagation properties

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

Because of limitations on wireless PHY ranges, WLANs intended to cover reasonable geographic distances may be built from basic coverage building blocks. 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.

#### 5.1.1.3 The impact of handling mobile STAs

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

Copyright © 2007 IEEE. All rights reserved.

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

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

#### 5.1.1.4 Interaction with other IEEE 802® layers

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

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

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.

#### 5.1.1.5 Interaction with non-IEEE-802 protocols

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

## 5.2 Components of the IEEE 802.11 architecture

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

The basic service set (BSS) is the basic building block of an IEEE 802.11 LAN. Figure 5-1 shows two BSSs, each of which has two STAs that are members of the BSS.

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


Copyright © 2007 IEEE. All rights reserved.

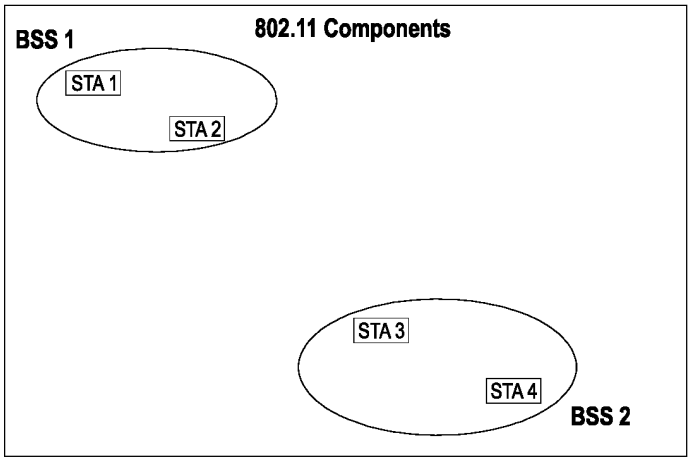



**Figure 5-1—BSSs**

### 5.2.1 The independent BSS (IBSS) as an ad hoc network

The IBSS is the most basic type of IEEE 802.11 LAN. A minimum IEEE 802.11 LAN may consist of only two STAs. Since the BSSs shown in Figure 5-1 are simple and lack other components (contrast this with Figure 5-2), the two can be taken to be represenative of two IBSSs.

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

### 5.2.2 STA membership in a BSS is dynamic

A STA's membership in a BSS is dynamic (STAs turn on, turn off, come within range, and go out of range). To become a member of a BSS, a STA joins the BSS using the synchronization procedure described in 11.1.3.4. To access all the services of an infrastructure BSS, a STA shall become "associated." These associations are dynamic and involve the use of the distribution system service (DSS), which is described in 5.3.2.

### 5.2.3 Distribution system (DS) concepts

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

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

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

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

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

Copyright © 2007 IEEE. All rights reserved.

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

Figure 5-2 adds the DS, DSM and AP components to the IEEE 802.11 architecture picture.

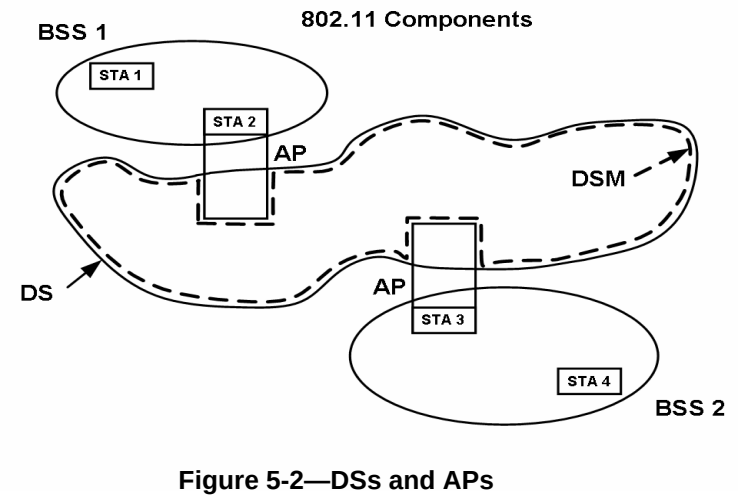



**Figure 5-2—DSs and APs**

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

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

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

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

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

Nothing is assumed by IEEE Std 802.11 about the relative physical locations of the BSSs in Figure 5-3.

All of the following are possible:

a) The BSSs may partially overlap. This is commonly used to arrange contiguous coverage within a physical volume.

b) The BSSs could be physically disjoint. Logically there is no limit to the distance between BSSs.

c) The BSSs may be physically collocated. This may be done to provide redundancy.

d) One (or more) IBSS or ESS networks may be physically present in the same space as one (or more) ESS networks. This may arise for a number of reasons. Some examples are when an ad hoc network is operating in a location that also has an ESS network, when physically overlapping IEEE 802.11 networks have been set up by different organizations, and when two or more different access and security policies are needed in the same location.


Copyright © 2007 IEEE. All rights reserved.

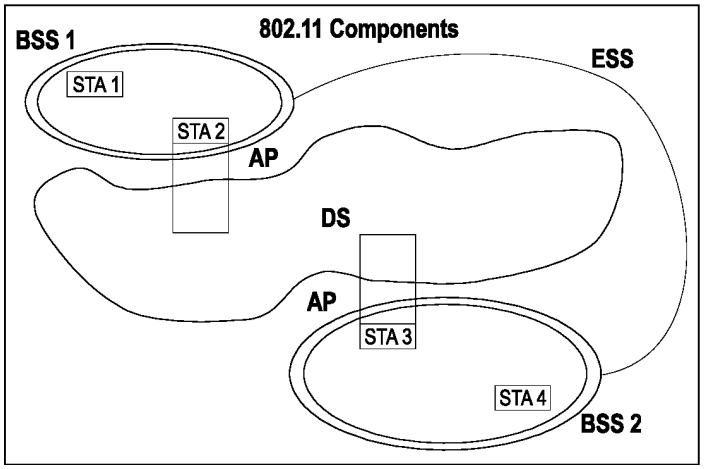



**Figure 5-3—ESS**

#### 5.2.3.2 RSNA

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

— Enhanced authentication mechanisms for STAs

— Key management algorithms

— Cryptographic key establishment

— An enhanced data cryptographic encapsulation mechanism, called Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP), and, optionally, Temporal Key Integrity Protocol (TKIP).

An RSNA relies on several components external to the IEEE 802.11 architecture.

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

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

### 5.2.4 Area concepts

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

Copyright © 2007 IEEE. All rights reserved.

Figure 5-4 shows a signal strength map for a simple square room with a standard metal desk and an open doorway. Figure 5-4 is a static snapshot; the propagation patterns change dynamically as STAs and objects in the environment move. In Figure 5-4 the dark (solid) blocks in the lower left are a metal desk and there is a doorway at the top right of the figure. The figure indicates relative differences in field strength with different intensities and indicates the variability of field strength even in a static environment. The difference between the greatest signal strength and the least signal strength in Figure 5-4 is 50 dB.

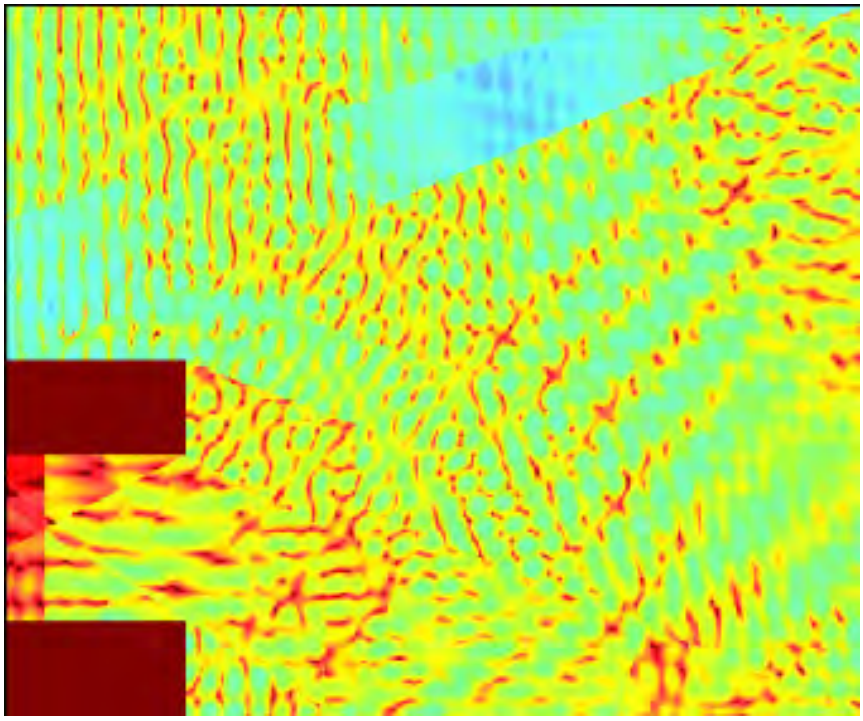


**Figure 5-4—A representative signal intensity map**

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

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

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


Copyright © 2007 IEEE. All rights reserved.

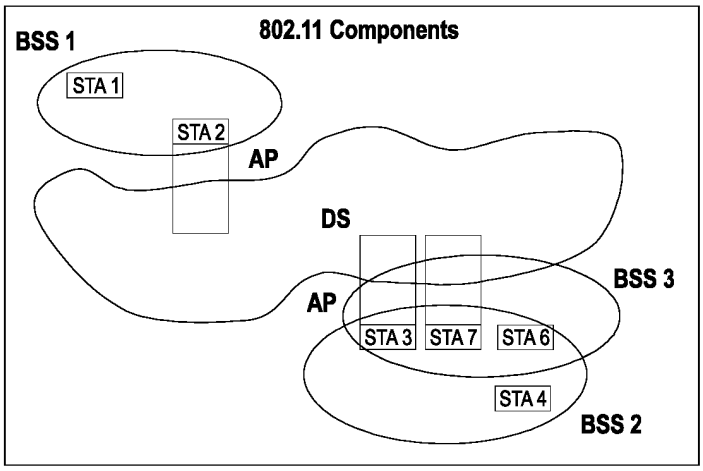



**Figure 5-5—Collocated coverage areas**

### 5.2.5 Integration with wired LANs

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

A portal is the logical point at which MSDUs from an integrated non-IEEE-802.11 LAN enter the IEEE 802.11 DS. For example, a portal is shown in Figure 5-6 connecting to a wired IEEE 802 LAN.

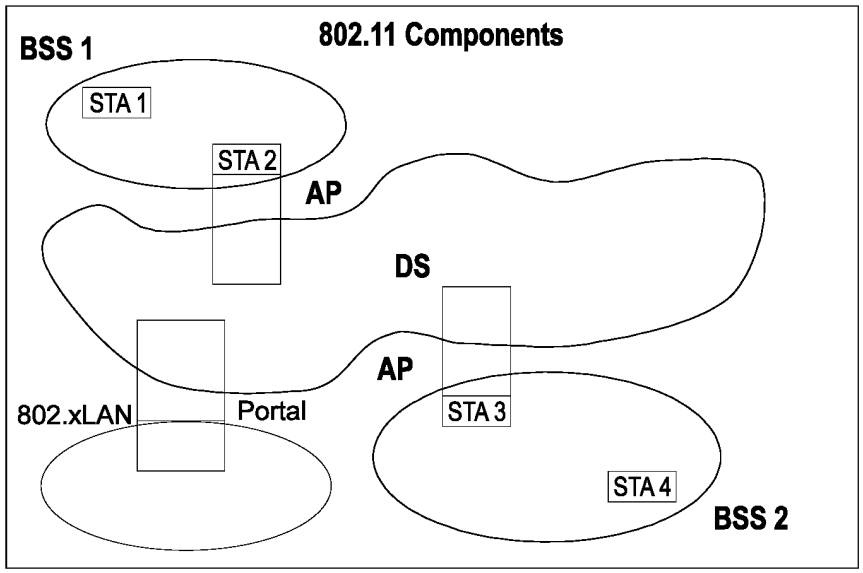



**Figure 5-6—Connecting to other IEEE 802 LANs**

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

Copyright © 2007 IEEE. All rights reserved.

### 5.2.6 QoS BSS: 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 STAs associated with a QoS access point in a QoS BSS. A subset of the QoS enhancements is available for use between STAs that are members of the same QoS IBSS. Because a QoS STA implements a superset of STA functionality, as defined in this standard, the STA may associate with a non-QoS access point in a non-QoS BSS, to provide non-QoS MAC data service when there is no QoS BSS with which to associate.

The enhancements that distinguish QoS STAs from non-QoS STAs and QoS APs from non-QoS APs are collectively termed the *QoS facility*. The quantity of certain, QoS-specific, mechanisms may vary among QoS implementations, as well as between QoS STAs and QoS APs, 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 QoS STA or QoS AP must implement those core QoS facilities necessary for its QoS functions to interoperate with other STAs in the BSS. 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 STAs separately from the core QoS facilities. A comprehensive statement on mandatory and optional functionalities is available in Annex A.

This standard 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) STA based on its requirements requests the HC for TXOPs, both for its own transmissions as well as for transmissions from the AP to itself.[13] The request is initiated by the station management entity (SME) of the non-AP STA. The HC, which is collocated at the AP, either accepts or rejects the request based on an admission control policy. If the request is accepted, the HC schedules TXOPs for both the AP and the non-AP STA. For transmissions from the non-AP STA, the HC polls the non-AP STA based on the parameters supplied by the non-AP STA at the time of its request. For transmissions to the non-AP STA, the AP directly obtains TXOPs from the collocated HC and delivers the queued frames to the non-AP STA, again based on the parameters supplied by the non-AP STA. 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.

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

A QoS STA associated in a non-QoS BSS shall act as a non-QoS STA.

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


Copyright © 2007 IEEE. All rights reserved.

## 5.3 Logical service interfaces

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

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

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

a) Authentication
b) Association
c) Deauthentication
d) Disassociation
e) Distribution
f) Integration
g) Data confidentiality
h) Reassociation
i) MSDU delivery
j) DFS
k) TPC
l) Higher layer timer synchronization (QoS facility only)
m) QoS traffic scheduling (QoS facility only)

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

### 5.3.1 SS

The service provided by STAs is known as the SS.

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

The SS is as follows:

a) Authentication
b) Deauthentication
c) Data confidentiality
d) MSDU delivery
e) DFS
f) TPC
g) Higher layer timer synchronization (QoS facility only)
h) QoS traffic scheduling (QoS facility only)

Copyright © 2007 IEEE. All rights reserved.

### 5.3.2 DSS

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

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

The services that comprise the DSS are as follows:

a) Association
b) Disassociation
c) Distribution
d) Integration
e) Reassociation
f) QoS traffic scheduling (QoS facility only)

DSSs are specified for use by MAC sublayer entities.

Figure 5-7 combines the components from previous figures with both types of services to show the complete IEEE 802.11 architecture.

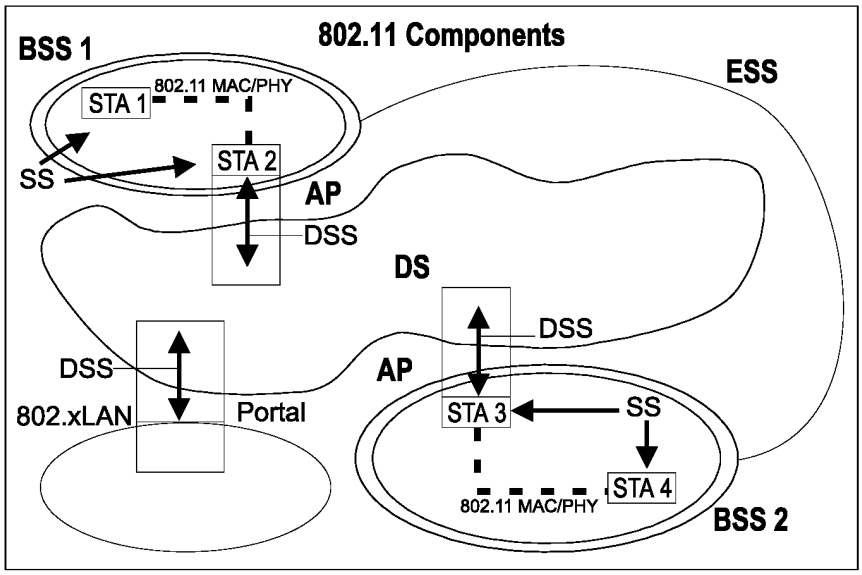



**Figure 5-7—Complete IEEE 802.11 architecture**

## 5.4 Overview of the services

There are many services specified by IEEE Std 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.


Copyright © 2007 IEEE. All rights reserved.

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

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

The IEEE 802.11 MAC sublayer uses three types of messages—*data*, *management*, and *control* (see Clause 7). The data messages are handled via the MAC data service path.

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

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

The examples here assume an ESS network environment. The differences between the ESS and the IBSS network environments are discussed separately in 5.6.

### 5.4.1 Distribution of messages within a DS

#### 5.4.1.1 Distribution

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

Refer to the ESS network in Figure 5-7 and consider a data message being sent from STA 1 to STA 4. The message is sent from STA 1 and received by STA 2 (the “input” AP). The AP gives the message to the distribution service of the DS. It is the job of the distribution service to deliver the message within the DS in such a way that it arrives at the appropriate DS destination for the intended recipient. In this example, the message is distributed to STA 3 (the “output” AP) and STA 3 accesses the WM to send the message to STA 4 (the intended destination).

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

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

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

While IEEE Std 802.11 does not specify DS implementations, it does recognize and support the use of the WM as the DSM. This is specifically supported by the IEEE 802.11 frame formats. (Refer to Clause 7 for details.)

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 81 of 1232

#### 5.4.1.2 Integration

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

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

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

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

#### 5.4.1.3 QoS traffic scheduling

QoS traffic scheduling provides intra-BSS QoS frame transfers under the HCF, using either contention-based or controlled channel access. At each TXOP, a traffic scheduling entity at the STA 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

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

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

#### 5.4.2.1 Mobility types

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

a) ***No-transition:*** In this type, two subclasses that are usually indistinguishable are identified:

   1) Static—no motion.

   2) Local movement—movement within the PHY range of the communicating STAs [i.e., movement within a basic service area (BSA)].

b) ***BSS-transition:*** This type is defined as a STA movement from one BSS in one ESS to another BSS within the same ESS.

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

The different association services support the different categories of mobility.


Copyright © 2007 IEEE. All rights reserved.

#### 5.4.2.2 Association

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

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

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

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

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

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

A STA learns what APs are present and what operational capabilities are available from each of those APs and then invokes the association service to establish an association. For details of how a STA learns about what APs are present, see 11.1.3.

#### 5.4.2.3 Reassociation

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

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

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

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 83 of 1232

#### 5.4.2.4 Disassociation

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

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

The disassociation service may be invoked by either party to an association (non-AP STA or AP). Disassociation is a notification, not a request. Disassociation cannot be refused by either party to the association.

APs may need to disassociate STAs to enable the AP to be removed from a network for service or for other reasons.

STAs shall attempt to disassociate when they leave a network. However, the MAC protocol does not depend on STAs invoking the disassociation service. (MAC management is designed to accommodate loss of communication with an associated STA.)

### 5.4.3 Access control and data confidentiality services

Two services are required for IEEE Std 802.11 to provide functionality equivalent to that which is inherent to wired LANs. The design of wired LANs assumes the physical attributes of wire. In particular, wired LAN design assumes the physically closed and controlled nature of wired media. The physically open medium nature of an IEEE 802.11 LAN violates those assumptions.

Two services are provided to bring the IEEE 802.11 functionality in line with wired LAN assumptions: authentication and data confidentiality. Authentication is used instead of the wired media physical connection. Data confidentiality is used to provide the confidential aspects of closed wired media.

In a WLAN that does not support RSNA, two services, authentication and data confidentiality, are defined. IEEE 802.11 authentication is used instead of the wired media physical connection. WEP encryption was defined to provide the data confidentiality aspects of closed wired media.

An RSNA uses the IEEE 802.1X authentication service along with TKIP and CCMP to provide access control. The IEEE 802.11 station management entity (SME) provides key management via an exchange of IEEE 802.1X EAPOL-Key frames. Data confidentiality and data integrity are provided by RSN key management together with the TKIP and CCMP.

#### 5.4.3.1 Authentication

IEEE 802.11 authentication operates at the link level between IEEE 802.11 STAs. IEEE Std 802.11 does not provide either end-to-end (message origin to message destination) or user-to-user authentication.

IEEE Std 802.11 attempts to control LAN access via the authentication service. IEEE 802.11 authentication is an SS. This service may be used by all STAs to establish their identity to STAs with which they communicate, in both ESS and IBSS networks. If a mutually acceptable level of authentication has not been established between two STAs, an association shall not be established.

IEEE Std 802.11 defines two authentication methods: Open System authentication and Shared Key authentication. Open System authentication admits any STA to the DS. Shared Key authentication relies on WEP to demonstrate knowledge of a WEP encryption key. The IEEE 802.11 authentication mechanism also allows definition of new authentication methods.


Copyright © 2007 IEEE. All rights reserved.

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

In an RSNA, IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port.

The Open System authentication algorithm is used in RSNs based on infrastructure BSS and IBSS, although Open System authentication is optional in an RSN based on an IBSS. RSNA disallows the use of Shared Key authentication.

Management information base (MIB) functions are provided in Annex D to support the standardized authentication schemes.

A STA may be authenticated with many other STAs at any given instant.

#### 5.4.3.1.1 Preauthentication

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

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

If the authentication is left until reassociation time, this may impact the speed with which a STA can reassociate between APs, limiting BSS-transition mobility performance. The use of preauthentication takes the authentication service overhead out of the time-critical reassociation process.

### 5.4.3.2 Deauthentication

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

In an ESS, because authentication is a prerequisite for association, the act of deauthentication shall cause the STA to be disassociated. The deauthentication service may be invoked by either authenticated party (non-AP STA or AP). Deauthentication is not a request; it is a notification. Deauthentication shall not be refused by either party. When an AP sends a deauthentication notice to an associated STA, the association shall also be terminated.

In an RSN ESS, Open System authentication is required. In an RSN ESS, deauthentication results in termination of any association for the deauthenticated STA. It also results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the pairwise transient key security association (PTKSA). The deauthentication notification is provided to IEEE Std 802.1X-2004 via the MAC layer.

In an RSNA, deauthentication also destroys any related PTKSA, group temporal key security association (GTKSA), station-to-station link (STSL) master key security association (SMKSA), and STSL transient key security association (STKSA) that exist in the STA and closes the associated IEEE 802.1X Controlled Port. If pairwise master key (PMK) caching is not enabled, deauthentication also destroys the pairwise master key security association (PMKSA) from which the deleted PTKSA was derived.

Copyright © 2007 IEEE. All rights reserved.

In an RSN IBSS, Open System authentication is optional, but a STA is required to recognize Deauthentication frames. Deauthentication results in the IEEE 802.1X Controlled Port for that STA being disabled and deletes the PTKSA.

#### 5.4.3.3 Data confidentiality

In a wired LAN, only those STAs physically connected to the wire can send or receive LAN traffic. With a wireless shared medium, there is no physical connection, and all STAs and certain other RF devices in or near the LAN may be able to send, receive, and/or interfere with the LAN traffic. Any IEEE 802.11-compliant STA can receive all like-PHY IEEE 802.11 traffic that is within range and can transmit to any other IEEE 802.11 STA within range. Thus, the connection of a single wireless link (without data confidentiality) to an existing wired LAN may seriously degrade the security level of the wired LAN.

To bring the security of the WLAN up to the level implicit in wired LAN design, IEEE Std 802.11 provides the ability to protect the contents of messages. This functionality is provided by the data confidentiality service. Data confidentiality is an SS.

IEEE Std 802.11 provides three cryptographic algorithms to protect data traffic: WEP, TKIP, and CCMP. WEP and TKIP are based on the ARC4[14] algorithm, and CCMP is based on the advanced encryption standard (AES). A means is provided for STAs to select the algorithm(s) to be used for a given association.

The default data confidentiality state for all IEEE 802.11 STAs is “in the clear.” If the data confidentiality service is not invoked, all messages shall be sent unprotected. If this policy is unacceptable to the sender, it shall not send data frames; and if the policy is unacceptable to the receiver, it shall discard any received data frames. Unprotected data frames received at a STA configured for mandatory data confidentiality, as well as protected data frames using a key not available at the receiving STA, are discarded without an indication to LLC (or without indication to distribution services in the case of “To DS” frames received at an AP). These frames are acknowledged on the WM [if received without frame check sequence (FCS) error] to avoid wasting WM bandwidth on retries of frames that are being discarded.

#### 5.4.3.4 Key management

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys. The procedures defined in this standard provide fresh keys by means of protocols called the 4-Way Handshake and Group Key Handshake.

#### 5.4.3.5 Data origin authenticity

The data origin authenticity mechanism defines a means by which a STA that receives a data frame can determine which STA transmitted the MAC protocol data unit (MPDU). This feature is required in an RSNA to prevent one STA from masquerading as a different STA. This mechanism is provided for STAs that use CCMP or TKIP.

Data origin authenticity is only applicable to unicast data frames. The protocols do not guarantee data origin authenticity for broadcast/multicast data frames, as this cannot be accomplished using symmetric keys and public key methods are too computationally expensive.

[14]Details of the ARC4 algorithm are available from RSA Security, Inc. Contact RSA RSA Security, 174 Middlesex Turnpike, Bedford, MA 01730 (http://www.rsasecurity.com/), for algorithm details and the uniform ARC4 license terms that RSA offers to anyone wishing to use ARC4 for the purpose of implementing the IEEE 802.11 WEP option. If necessary, contact the IEEE Standards Department Intellectual Property Rights Administrator for details on how to communicate with RSA.


Copyright © 2007 IEEE. All rights reserved.

#### 5.4.3.6 Replay detection

The replay detection mechanism defines a means by which a STA that receives a data frame from another STA can detect whether the data frame is an unauthorized retransmission. This mechanism is provided for STAs that use CCMP or TKIP.

### 5.4.4 Spectrum management services

Two services are required to satisfy requirements in some regulatory domains for operation in the 5 GHz band. These services are called transmit power control (TPC) and dynamic frequency selection (DFS).

#### 5.4.4.1 TPC

Radio regulations may require radio local area networks (RLANs) operating in the 5 GHz band to use transmitter power control, involving specification of a regulatory maximum transmit power and a mitigation requirement for each allowed channel, to reduce interference with satellite services. The TPC service is used to satisfy this regulatory requirement.

The TPC service provides for the following:

— Association of STAs with an AP in a BSS based on the STAs' power capability.

— Specification of regulatory and local maximum transmit power levels for the current channel.

— Selection of a transmit power for each transmission in a channel within constraints imposed by regulatory requirements.

— Adaptation of transmit power based on a range of information, including path loss and link margin estimates.

#### 5.4.4.2 DFS

Radio regulations may require RLANs operating in the 5 GHz band to implement a mechanism to avoid co-channel operation with radar systems and to ensure uniform utilization of available channels. The DFS service is used to satisfy these regulatory requirements.

The DFS service provides for the following:

— Association of STAs with an AP in a BSS based on the STAs' supported channels.

— Quieting the current channel so it can be tested for the presence of radar with less interference from other STAs.

— Testing channels for radar before using a channel and while operating in a channel.

— Discontinuing operations after detecting radar in the current channel to avoid interference with radar.

— Detecting radar in the current and other channels based on regulatory requirements.

— Requesting and reporting of measurements in the current and other channels.

— Selecting and advertising a new channel to assist the migration of a BSS or IBSS after radar is detected.

### 5.4.5 Traffic differentiation and QoS support

IEEE Std 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

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 87 of 1232

IEEE 802.11 specification with any other end-to-end QoS delivery mechanism like Resource Reservation Protocol (RSVP) are beyond the scope of this standard.

### 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 standard defines a MAC service that enables layers above the MAC to accurately synchronize application-dependent timers shared among STAs. 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 standard, they rely on an indication from each STA's MAC that is provided essentially simultaneously, via multicast, to the STAs. 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[15] on the air by STAs 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.

## 5.5 Multiple logical address spaces

Just as the IEEE 802.11 architecture allows for the possibility that the WM, DSM, and an integrated wired LAN may all be different physical media, it also allows for the possibility that each of these components may be operating within different address spaces. IEEE Std 802.11 only uses and specifies the use of the WM address space.

Each IEEE 802.11 PHY operates in a single medium—the WM. The IEEE 802.11 MAC operates in a single address space. MAC addresses are used on the WM in the IEEE 802.11 architecture. Therefore, it is unnecessary for the standard to explicitly specify that its addresses are "WM addresses." This is assumed throughout this standard.

IEEE Std 802.11 has chosen to use the IEEE 802 48-bit address space (see 7.1.3.3.1). Thus IEEE 802.11 addresses are compatible with the address space used by the IEEE 802 LAN family.

The IEEE 802.11 choice of address space implies that for many instantiations of the IEEE 802.11 architecture, the wired LAN MAC address space and the IEEE 802.11 MAC address space may be the same. In those situations where a DS that uses MAC level IEEE 802 addressing is appropriate, all three of the logical address spaces used within a system could be identical. While this is a common case, it is not the only combination allowed by the architecture. The IEEE 802.11 architecture allows for all three logical address spaces to be distinct.

A multiple address space example is one in which the DS implementation uses network layer addressing. In this case, the WM address space and the DS address space would be different.

Note that IEEE 802.11 STAs within a single ESS share the same address space, fulfilling the transparency requirement from the definition of the DS. The DSS uses this same address space, even in the case where the DSM uses a different address space.

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


Copyright © 2007 IEEE. All rights reserved.

The ability of the architecture to handle multiple logical media and address spaces is key to the ability of IEEE Std 802.11 to be independent of the DS implementation and to interface cleanly with network layer mobility approaches. The implementation of the DS is unspecified and is beyond the scope of this standard.

## 5.6 Differences between ESS and IBSS LANs

In 5.2.1 the concept of the IBSS LAN was introduced. It was noted that an IBSS is often used to support an ad hoc network. In an IBSS network, a STA communicates directly with one or more other STAs.

Consider the full IEEE 802.11 architecture as shown in Figure 5-8.

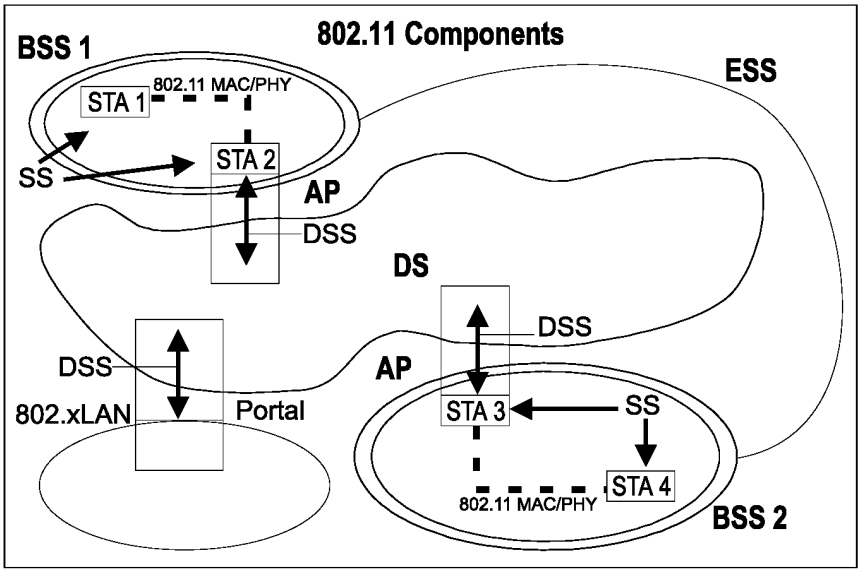



**Figure 5-8—IEEE 802.11 architecture (again)**

An IBSS consists of STAs that are directly connected. Thus there is (by definition) only one BSS. Further, because there is no physical DS, there cannot be a portal, an integrated wired LAN, or the DSS. The logical picture reduces to Figure 5-9.

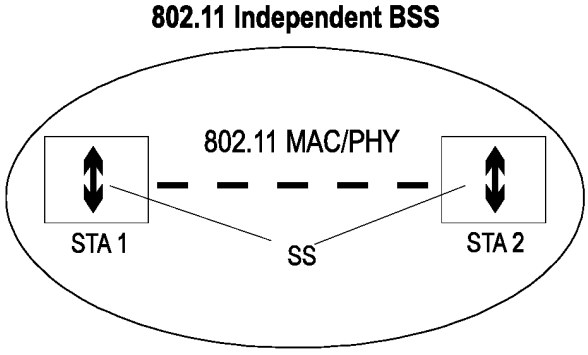



**Figure 5-9—Logical architecture of an IBSS**

Copyright © 2007 IEEE. All rights reserved.

Only the minimum two STAs are shown in Figure 5-9. An IBSS may have an arbitrary number of members. In an IBSS, only Class 1 and Class 2 frames are allowed because there is no DS in an IBSS.

The services that apply to an IBSS are the SSs. A QoS IBSS 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 QoS IBSS has no HC and does not support polled TXOP operation and setting up of TSPEC.

In an IBSS, each STA must enforce its own security policy. In an ESS, an AP can enforce a uniform security policy across all STAs.

## 5.7 Reference model

This standard presents the architectural view, emphasizing the separation of the system into two major parts: the MAC of the data link layer (DLL) and the PHY. These layers are intended to correspond closely to the lowest layers of the ISO/IEC basic reference model of Open Systems Interconnection (OSI) (ISO/IEC 7498-1: 1994). The layers and sublayers described in this standard are shown in Figure 5-10.

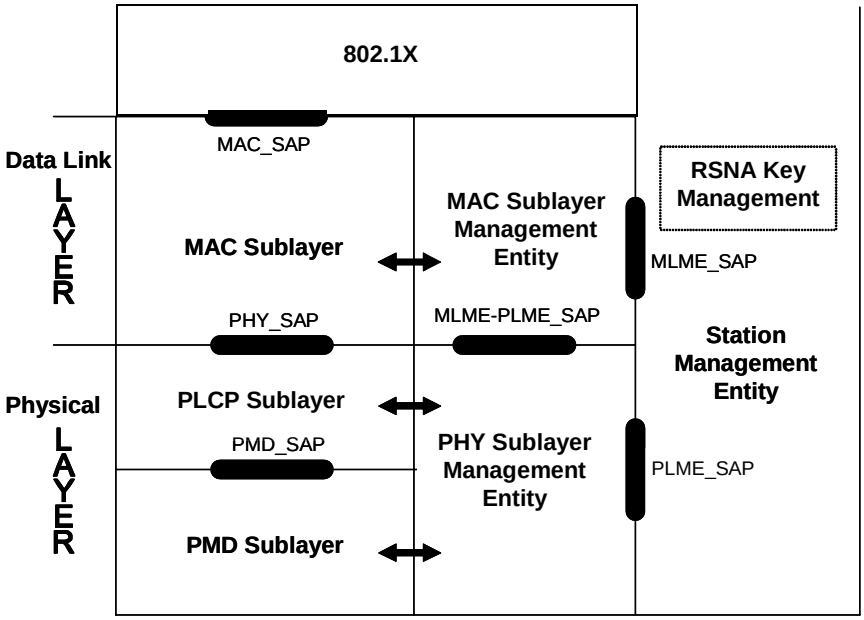



**Figure 5-10—Portion of the ISO/IEC basic reference model covered in this standard**

There is an interface between the IEEE 802.1X Supplicant/Authenticator and the SME not shown in Figure 5-10. This interface is described in IEEE Std 802.1X-2004.

## 5.8 IEEE Std 802.11 and IEEE Std 802.1X-2004

An RSNA relies on IEEE Std 802.1X-2004 to provide authentication services and uses the IEEE 802.11 key management scheme defined in 8.5. The IEEE 802.1X access control mechanisms apply to the association between a STA and an AP and to the relationship between the IBSS STA and STA peer. The AP's SME performs the Authenticator and, optionally, the Supplicant and AS roles. In an ESS, a non-AP STA's SME


Copyright © 2007 IEEE. All rights reserved.

performs the Supplicant role. In an IBSS, a STA's SME takes on both the Supplicant and Authenticator roles and may take on the AS role.

### 5.8.1 IEEE 802.11 usage of IEEE Std 802.1X-2004

IEEE Std 802.11 depends upon IEEE Std 802.1X-2004 to control the flow of MAC service data units (MSDUs) between the DS and STAs by use of the IEEE 802.1X Controlled/Uncontrolled Port model. IEEE 802.1X authentication frames are transmitted in IEEE 802.11 data frames and passed via the IEEE 802.1X Uncontrolled Port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between two STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. It is the responsibility of both the Supplicant and the Authenticator to implement port blocking. Each association between a pair of STAs creates a unique pair of IEEE 802.1X Ports, and authentication takes place relative to those ports alone.

IEEE Std 802.11 depends upon IEEE Std 802.1X-2004 and the 4-Way Handshake and Group Key Handshake, described in Clause 8, to establish and change cryptographic keys. Keys are established after authentication has completed. Keys may change for a variety of reasons, including expiration of an IEEE 802.1X authentication timer, key compromise, danger of compromise, or policy.

### 5.8.2 Infrastructure functional model overview

This subclause summarizes the system setup and operation of an RSN, in two cases: when an IEEE 802.1X AS is used and when a PSK is used. For an ESS, the AP includes an Authenticator, and each associated STA includes a Supplicant.

#### 5.8.2.1 AKM operations with AS

The following AKM operations are carried out when an IEEE 802.1X AS is used:

a) Prior to any use of IEEE Std 802.1X-2004, IEEE Std 802.11 assumes that the Authenticator and AS have established a secure channel. The security of the channel between the Authenticator and the AS is outside the scope of this standard.

   Authentication credentials must be distributed to the Supplicant and AS prior to association.

b) A STA discovers the AP's security policy through passively monitoring Beacon frames or through active probing (shown in Figure 5-11). If IEEE 802.1X authentication is used, the EAP authentication process starts when the AP's Authenticator sends the EAP-Request (shown in Figure 5-12) or the STA's Supplicant sends the EAPOL-Start message. EAP authentication frames pass between the Supplicant and AS via the Authenticator and Supplicant's Uncontrolled Ports. This is shown in Figure 5-12.

c) The Supplicant and AS authenticate each other and generate a PMK. The PMK is sent from the AS to the Authenticator over the secure channel. See Figure 5-12.

A 4-Way Handshake utilizing EAPOL-Key frames is initiated by the Authenticator to do the following:

— Confirm that a live peer holds the PMK.

— Confirm that the PMK is current.

— Derive a fresh pairwise transient key (PTK) from the PMK.

— Install the pairwise encryption and integrity keys into IEEE Std 802.11.

— Transport the group temporal key (GTK) and GTK sequence number from Authenticator to Supplicant and install the GTK and GTK sequence number in the STA and, if not already installed, in the AP.

— Confirm the cipher suite selection.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 91 of 1232

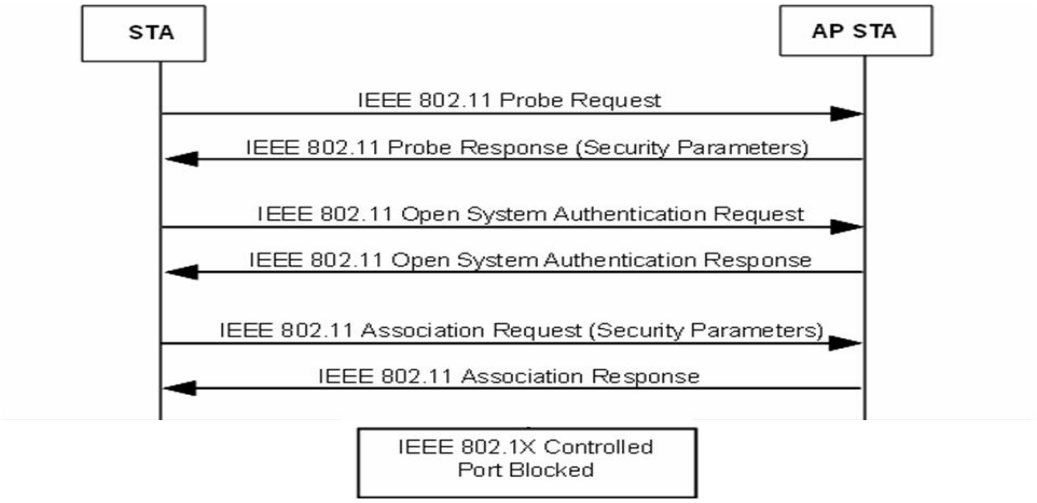



**Figure 5-11—Establishing the IEEE 802.11 association**

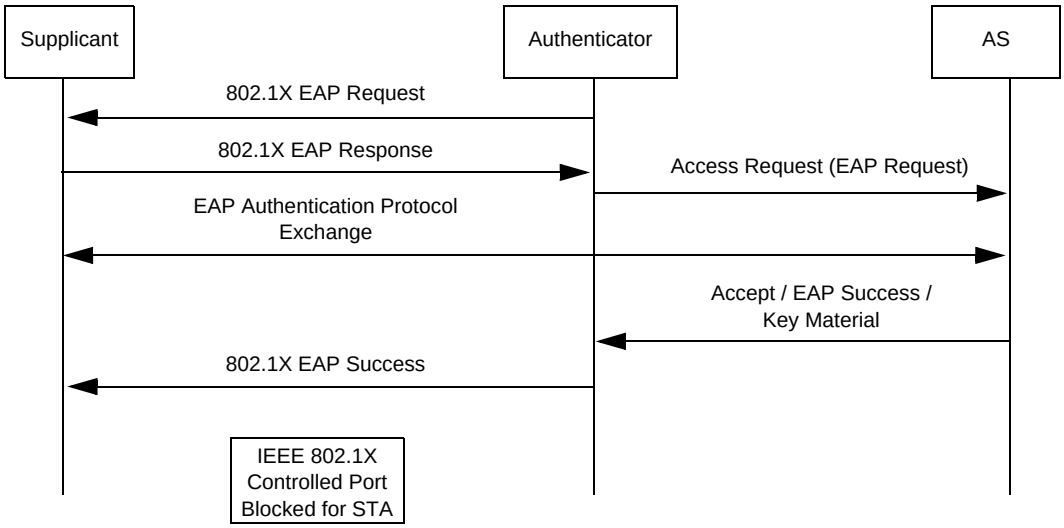



**Figure 5-12—IEEE 802.1X EAP authentication**

Installing the PTK, and where applicable the GTK keys, causes the MAC to encrypt and decrypt all subsequent MSDUs irrespective of their path through the controlled or uncontrolled ports.

Upon successful completion of the 4-Way Handshake, the Authenticator and Supplicant have authenticated each other; and the IEEE 802.1X Controlled Ports are unblocked to permit general data traffic. See Figure 5-13.


Copyright © 2007 IEEE. All rights reserved.

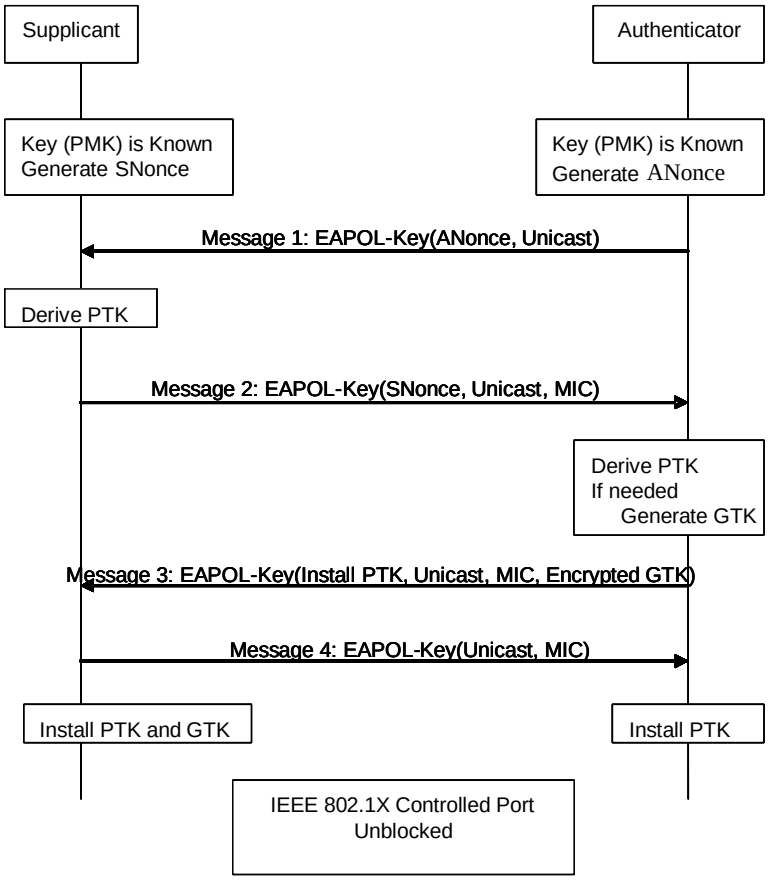



**Figure 5-13—Establishing pairwise and group keys**

If the Authenticator later changes the GTK, it sends the new GTK and GTK sequence number to the Supplicant using the Group Key Handshake to allow the Supplicant to continue to receive broadcast/ multicast messages and, optionally, to transmit and receive unicast frames. EAPOL-Key frames are used to carry out this exchange. See Figure 5-14.

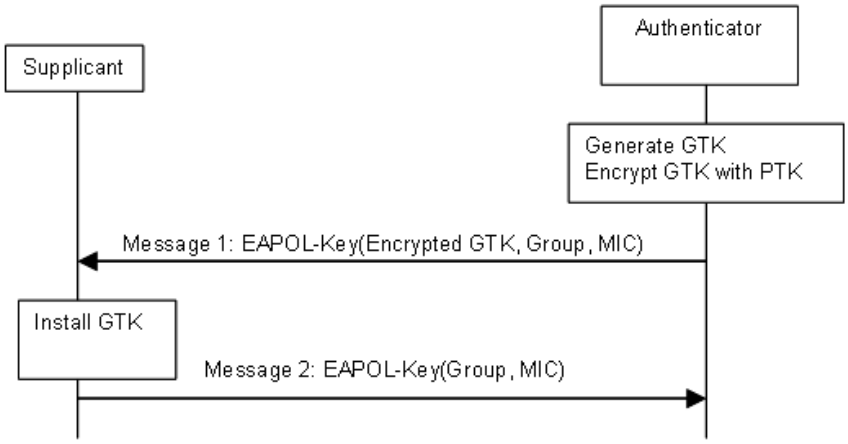



**Figure 5-14—Delivery of subsequent group keys**

Copyright © 2007 IEEE. All rights reserved.

#### 5.8.2.2 Operations with PSK

The following AKM operations are carried out when the PMK is a PSK:

— A STA discovers the AP's security policy through passively monitoring Beacon frames or through active probing (shown in Figure 5-11). A STA associates with an AP and negotiates a security policy. The PMK is the PSK.

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

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

#### 5.8.2.3 Disassociation

Disassociation initiated by either STA in an RSNA causes the deletion of the PTKSA at both ends and the deletion of the GTKSA in a non-AP STA. The controlled and uncontrolled ports created for this association will also be deleted.

### 5.8.3 IBSS functional model description

This subclause summarizes the system setup and operation of an RSNA in an IBSS. An IBSS RSNA is specified in 8.4.7.

#### 5.8.3.1 Key usage

In an IBSS, the unicast data frames between two STAs are protected with a pairwise key. The key is part of the PTK, which is derived during a 4-Way Handshake. In an IBSS, the 4-Way Handshak may follow IEEE 802.11 authentication of one STA to another. Such authentication may be used by the peer to cause deletion of the PTKSA and Block the Controlled Port thus reseting any previous handshake.

In an IBSS, the broadcast/multicast data frames are protected by a key, e.g., named B1, that is generated by the STA transmitting the broadcast/multicast frame. To allow other STAs to decrypt broadcast/multicast frames, B1 must be sent to all the other STAs in the IBSS. B1 is sent in an EAPOL-Key frame, encrypted under the EAPOL-Key encryption key (KEK) portion of the PTK, and protected from modification by the EAPOL-Key confirmation key (KCK) portion of the PTK.

In an IBSS, a STA's SME responds to Deauthentication frames from a STA by deleting the PTKSA associated with that STA.

#### 5.8.3.2 Sample IBSS 4-Way Handshakes

In this example (see Figure 5-15), there are three STAs: S1, S2, S3. The broadcast/multicast frames sent by S1 are protected by B1; similarly B2 for S2, and B3 for S3.

For STAs S2 and S3 to decrypt broadcast/multicast frames from S1, B1 must be sent to S2 and S3. This is done using the 4-Way Handshake initially and using the Group Key Handshake for GTK updates.

The 4-Way Handshake from S1 to S2 allows S1 to send broadcast/multicast frames to S2, but does not allow S2 to send broadcast/multicast frames to S1 because S2 has a different transmit GTK. Therefore, S2 needs to initiate a 4-Way Handshake to S1 to allow S1 to decrypt S2's broadcast/multicast frames. Similarly, S2 also needs to initiate a 4-Way Handshake to S3 to enable S3 to receive broadcast/multicast messages from S2.

In a similar manner S3 needs to complete the 4-Way Handshake with S1 and S2 to deliver B3 to S1 and S2.


Copyright © 2007 IEEE. All rights reserved.

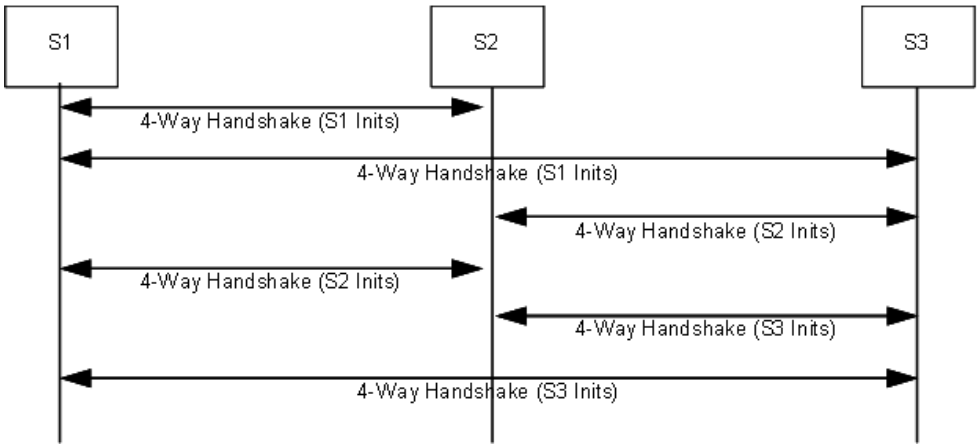



**Figure 5-15—Sample 4-Way Handshakes in an IBSS**

In this example, there are six 4-Way Handshakes. In general, *N* STA Supplicants require *N*(*N*–1) 4-Way Handshakes.

NOTE—In principle the KCK and KEK from a single 4-Way Handshake can be used for the Group Key Handshake in both directions, but using two 4-Way Handshakes means the Authenticator key state machine does not need to be different between IBSS and ESS.

The Group Key Handshake can be used to send the GTKs to the correct STAs. The 4-Way Handshake is used to derive the pairwise key and to send the initial GTK. Because in an IBSS there are two 4-Way Handshakes between any two STA Supplicants and Authenticators, the pairwise key used between any two STAs is from the 4-Way Handshake initiated by the STA Authenticator with the higher MAC address (see 8.5.1 for the notion of address comparison). The KCK and KEK used for a Group Key Handshake are the KCK and KEK derived by the 4-Way Handshake initiated by the same Authenticator that is initiating the Group Key Handshake.

In an IBSS, a secure link exists between two STAs when both 4-Way Handshakes have completed successfully. The Supplicant and Authenticator 4-Way Handshake state machines interact so the IEEE 802.1X variable portValid is not set until both 4-Way Handshakes complete.

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

a) The fourth STA receives a Beacon or Probe Response frame from a MAC address with which it has not completed a 4-Way Handshake.

b) A STA's SME receives a MLME-PROTECTEDFRAMEDROPPED.indication primitive from a MAC address with which it has not completed a 4-Way Handshake. This could be a multicast/ broadcast data frame transmitted by any of the STAs. If the SME wants to set up a security association to the peer STA, but does not know the security policy of the peer, it should send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

c) A STA's SME receives Message 1 of the 4-Way Handshake sent to a STA because the initiator received a broadcast data frame, Beacon frame, or Probe Response frame from that STA. If a STA received a 4-Way Handshake, wants to set up a security association to the peer STA, but does not

Copyright © 2007 IEEE. All rights reserved.

know the security policy of the peer, it should send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

#### 5.8.3.3 IBSS IEEE 802.1X example

When IEEE 802.1X authentication is used, each STA will need to include an IEEE 802.1X Authenticator and AS. A STA learns that a peer STA is RSNA-enabled and the peer's security policy (e.g., whether the AKMP is PSK or IEEE 802.1X authentication) from the Beacon or Probe Response frame.

Each STA's Supplicant will send an EAPOL-Start message to every other STA to which it wants to authenticate, and each STA's Authenticator will respond with the identity of the credential it wants to use.

The EAPOL-Start and EAP-Request/Identity messages are initiated when a protected data frame (indicated via a MLME-PROTECTEDFRAMEDROPPED.indication primitive), an IEEE 802.1X message, Beacon frame, or Probe Response frame is received from a MAC address with which the STA has not completed IEEE 802.1X authentication. If the SME wants to set up a security association to the peer STA, but does not know the security policy of the peer, it should send a Probe Request frame to the peer STA to find its security policy before setting up a security association to the peer STA.

Although Figure 5-16 shows the two IEEE 802.1X exchanges serialized, they may occur interleaved.

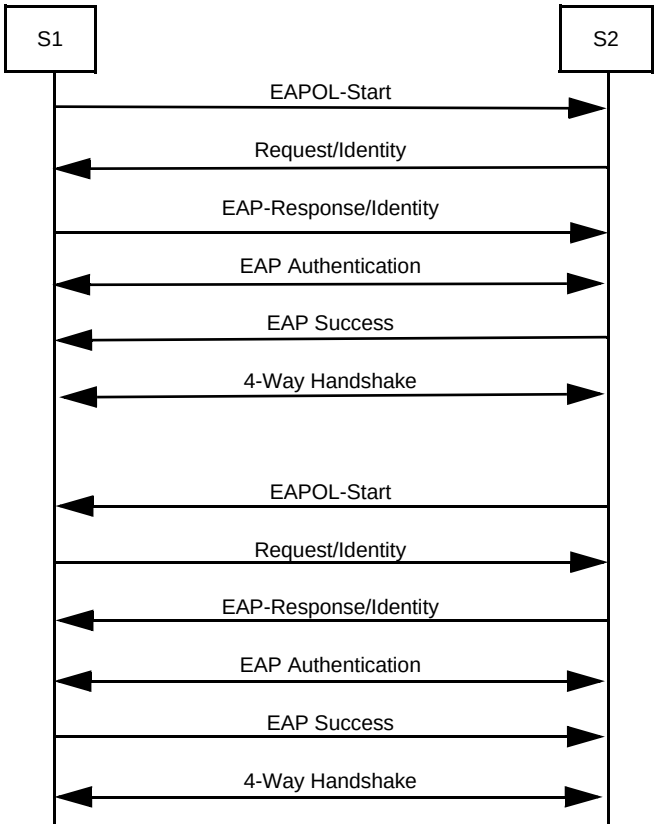



**Figure 5-16—Example using IEEE 802.1X authentication**


Copyright © 2007 IEEE. All rights reserved.

### 5.8.4 Authenticator-to-AS protocol

The Authenticator-to-AS authentication definition is out of the scope of this standard, but, to provide security assurances, the protocol must support the following functions:

a) Mutual authentication between the Authenticator and AS

b) A channel for the Supplicant/AS authentication

c) The ability to pass the generated key from the AS to the Authenticator in a manner that provides authentication of the key source, ensures integrity of the key transfer, and preserves data confidentiality of the key from all other parties

Suitable protocols include, but are not limited to, remote authentication dial-in user service (RADIUS) (IETF RFC 2865-2000 [B23]) and Diameter (IETF RFC 3588-2003 [B25]).

### 5.8.5 PMKSA caching

The Authenticator and Supplicant may cache PMKSAs, which include the IEEE 802.1X state. A PMKSA can be deleted from the cache for any reason and at any time.

The STA may supply a list of PMK or PSK key identifiers in the (Re)Association Request frame. Each key identifier names a PMKSA; the PMKSA may contain a single PMK. The Authenticator specifies the selected PMK or PSK key identifier in Message 1 of the 4-Way Handshake. The selection of the key identifiers to be included within the (Re)Association Request frame and Message 1 of the 4-Way Handshake is out of the scope of this standard.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 97 of 1232

Copyright © 2007 IEEE. All rights reserved.

# 6. MAC service definition

## 6.1 Overview of MAC services

### 6.1.1 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 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 data service provided by the MAC. Due to the characteristics of the WM, broadcast and multicast MSDUs may experience a lower QoS, compared to that of unicast MSDUs. All STAs will support the data service, but only QoS STAs in a QoS BSS 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 non-QoS STAs, there are two service classes within the 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 QoS STA: 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 QoS STAs either associated in a BSS or having membership in an IBSS, 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 BSS with some STAs supporting the QoS facility and others not supporting the QoS facility, in delivering an MSDU to a non-QoS STA, the QoS STA uses the access category (AC) corresponding to the UP of the MSDU.

#### 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 AP sent by other STAs in the BSS

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

a) Non-QoS subtypes, in which case the AP 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 AP 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 AP 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.

QoS APs deliver the UP with the received MSDUs to the DS.

Copyright © 2007 IEEE. All rights reserved.

#### 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 STAs 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 STAs 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 shall be sent with priority parameter set to 0.

The non-integer values of the priority parameter are allowed at all non-QoS STAs. The use of priority value of ContentionFree is deprecated at QoS STAs. The integer values of the priority parameter (i.e., TID) are supported only at QoS STAs that are either associated in a QoS BSS or members of a QoS IBSS. A range of 0 through 15 is supported by QoS STAs associated in a QoS BSS; whereas a range of 0 through 7 is supported by QoS STAs that are members of a QoS IBSS. If a QoS STA is associated in a non-QoS BSS, the STA is functioning as a non-QoS STA, so the priority value is always Contention or ContentionFree.

At QoS STAs associated in a QoS BSS, 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 AP. If a PC is not present, MSDUs with a priority of ContentionFree shall be delivered using an UP of 0. At STAs associated in a non-QoS BSS, all MSDUs with an integer priority are considered equivalent to MSDUs with a priority of Contention.

If a STA is associated in a QoS BSS, 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 STA receives in non-QoS data frames are reported to LLC with a priority of Contention, if they are received during the CP, or ContentionFree, if they are received during the CFP.

#### 6.1.1.3 Interpretation of service class parameter in MAC service primitives in a STA

In QoS STAs, 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 QoS STA 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 STA is an AP 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 QoS STA, the MSDU is transmitted using a non-QoS data frame.


Copyright © 2007 IEEE. All rights reserved.

When a QoS data frame is received from another STA, 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 a STA, 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 STA.

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

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

### 6.1.2 Security services

Security services in IEEE Std 802.11 are provided by the authentication service and the TKIP and CCMP mechanisms. The scope of the security services provided is limited to station-to-station data exchange. The data confidentiality service offered by an IEEE 802.11 TKIP and CCMP implementation is the protection of the MSDU. For the purposes of this standard, TKIP and CCMP are viewed as logical services located within the MAC sublayer as shown in the reference model, Figure 5-10 (in 5.7). Actual implementations of the TKIP and CCMP services are transparent to the LLC and other layers above the MAC sublayer.

The security services provided by TKIP and CCMP in IEEE Std 802.11 are as follows:

a) Data Confidentiality;

b) Authentication; and

c) Access control in conjunction with layer management.

During the authentication exchange, both parties exchange authentication information as described in Clause 8.

The MAC sublayer security services provided by TKIP and CCMP rely on information from nonlayer-2 management or system entities. Management entities communicate information to TKIP and CCMP through a set of MAC sublayer management entity (MLME) interfaces and MIB attributes; in particular, the decision tree for TKIP and CCMP defined in 8.7 is driven by MIB attributes.

The use of WEP for confidentiality, authentication, or access control is deprecated. The WEP algorithm is unsuitable for the purposes of this standard.

### 6.1.3 MSDU ordering

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

In a non-QoS STA, the 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 data service cannot tolerate this possible reordering, the optional StrictlyOrdered service

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 101 of 1232

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 QoS STAs operating in a BSS, 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.

STAs operating in a non-QoS BSS shall follow the reordering rules as defined for a non-QoS STA.

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 MSDU format

This standard is part of the IEEE 802 family of LAN standards, and as such all MSDUs are LLC PDUs as defined in ISO/IEC 8802-2: 1998. In order to achieve interoperability, implementers are recommended to apply the procedures described in ISO/IEC Technical Report 11802-5:1997(E) (previously known as IEEE Std 802.1H-1997 [B13]), along with a selective translation table (STT) that handles a few specific network protocols, with specific attention to the operations required when passing MSDUs to or from LANs or operating system components that use the Ethernet frame format. Note that such translations may be required in a non-AP STA.

### 6.1.5 MAC data service architecture

The MAC data plane architecture (i.e., processes that involve transport of all or part of an MSDU) is shown in Figure 6-1. During transmission, an MSDU goes through some or all of the following processes: frame delivery deferral during power save mode, sequence number assignment, fragmentation, encryption, integrity protection, and frame formatting. IEEE Std 802.1X-2004 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 6-1.

During reception, a received data frame goes through processes of MPDU header and 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 and if the MSDU does not represent an IEEE 802.1X frame. TKIP and CCMP MPDU frame order enforcement occurs after decryption, but prior to MSDU defragmentation; therefore, defragmentation will fail if MPDUs arrive out of order.


Copyright © 2007 IEEE. All rights reserved.

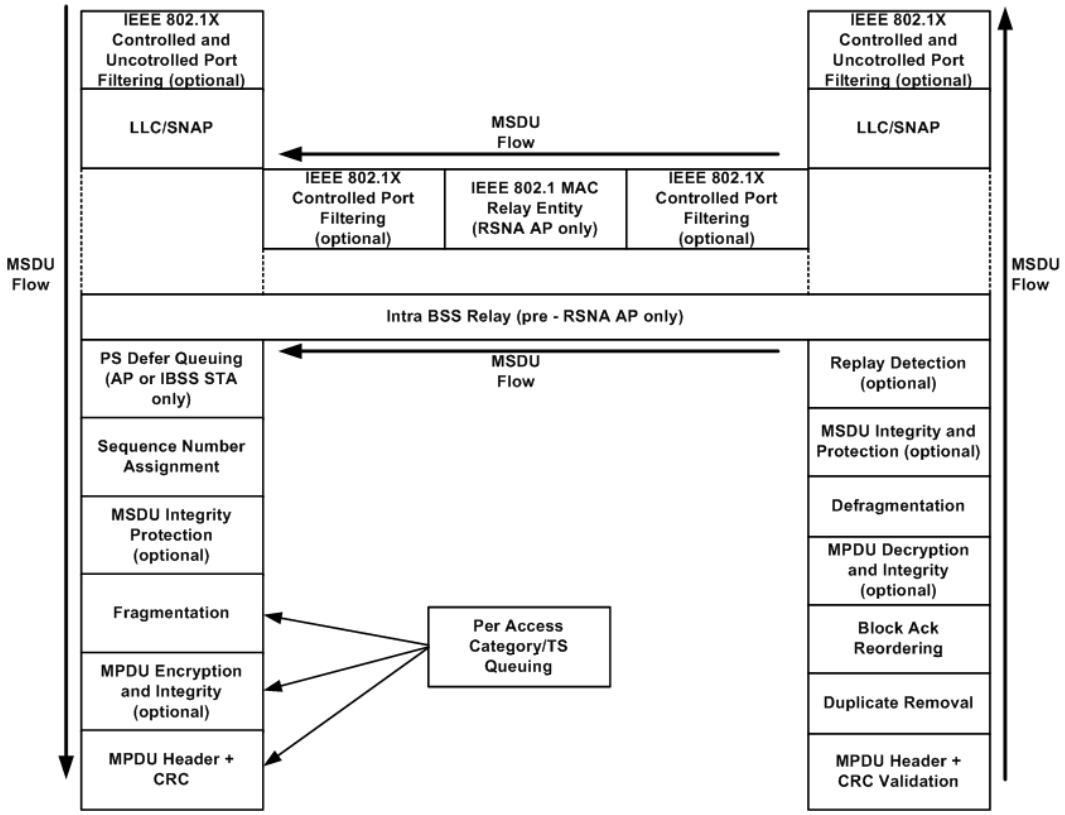



**Figure 6-1—MAC data plane architecture**

## 6.2 Detailed service specification

### 6.2.1 MAC data services

The IEEE 802.11 MAC supports the following service primitives as defined in ISO/IEC 8802-2: 1998:

— MA-UNITDATA.request

— MA-UNITDATA.indication

— MA-UNITDATA.confirm

The LLC definitions of the primitives and specific parameter value restrictions imposed by IEEE Std 802.11 are given in 6.2.1.1 through 6.2.1.3.

#### 6.2.1.1 MA-UNITDATA.request

##### 6.2.1.1.1 Function

This primitive requests a transfer of an MSDU from a local LLC sublayer entity to a single peer LLC sublayer entity, or multiple peer LLC sublayer entities in the case of group addresses.

##### 6.2.1.1.2 Semantics of the service primitive

The parameters of the primitive are as follows:

Copyright © 2007 IEEE. All rights reserved.

MA-UNITDATA.request (
source address,
destination address,
routing information,
data,
priority,
service class
)

The source address (SA) parameter specifies an individual MAC sublayer address of the sublayer entity from which the MSDU is being transferred.

The destination address (DA) parameter specifies either an individual or a group MAC sublayer entity address.

The routing information parameter specifies the route desired for the data transfer (a null value indicates source routing is not to be used). For IEEE Std 802.11, the routing information parameter must be null.

The data parameter specifies the MSDU to be transmitted by the MAC sublayer entity. For IEEE Std 802.11, the length of the MSDU must be less than or equal to 2304 octets.

The priority parameter specifies the priority desired for the data unit transfer. 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. The allowed values of service class are described in 6.1.1.3 and 6.1.3.

#### 6.2.1.1.3 When generated

This primitive is generated by the LLC sublayer entity when an MSDU is to be transferred to a peer LLC sublayer entity or entities.

#### 6.2.1.1.4 Effect of receipt

On receipt of this primitive, 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.confirm 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 appends all MAC specified fields (including DA, SA, FCS, and all fields that are unique to IEEE Std 802.11), 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.confirm primitive with transmission status set to Successful.

### 6.2.1.2 MA-UNITDATA.indication

#### 6.2.1.2.1 Function

This primitive defines the transfer of an MSDU from the MAC sublayer entity to the LLC sublayer entity, or entities in the case of group addresses. In the absence of error, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated MA-UNITDATA.request primitive.


Copyright © 2007 IEEE. All rights reserved.

#### 6.2.1.2.2 Semantics of the service primitive

The parameters of the primitive are as follows:

```
MA-UNITDATA.indication(
                        source address,
                        destination address,
                        routing information,
                        data,
                        reception status,
                        priority,
                        service class
                        )
```

The SA parameter is an individual address as specified by the SA field of the incoming frame.

The DA parameter is either an individual or a group address as specified by the DA field of the incoming frame.

The routing information parameter specifies the route that was used for the data transfer. IEEE Std 802.11 shall always set this field to null.

The data parameter specifies the MSDU as received by the local MAC entity.

The reception status parameter indicates the success or failure of the received frame for those frames that IEEE Std 802.11 reports via a MA-UNITDATA.indication primitive. This MAC always reports "success" 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. 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. The allowed values of service class are described in 6.1.1.3 and 6.1.3.

#### 6.2.1.2.3 When generated

The MA-UNITDATA.indication primitive is passed from the MAC sublayer entity to the LLC sublayer entity or entities to indicate the arrival of a frame at the local MAC sublayer entity. Frames are reported only if they are validly formatted at the MAC sublayer, received without error, received with valid (or null) security and integrity information, and their destination address designates the local MAC sublayer entity.

#### 6.2.1.2.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent on the content of the MSDU.

### 6.2.1.3 MA-UNITDATA.confirm

#### 6.2.1.3.1 Function

This primitive has local significance and provides the LLC sublayer with status information for the corresponding preceding MA-UNITDATA.request primitive.

#### 6.2.1.3.2 Semantics of the service primitive

The parameters of the primitive are as follows:

Copyright © 2007 IEEE. All rights reserved.

MA-UNITDATA.confirm(
source address,
destination address,
transmission status,
provided priority,
provided service class
)

The SA parameter is an individual MAC sublayer entity address as specified in the associated MA-UNITDATA.request primitive.

The DA parameter is either an individual or group MAC sublayer entity address as specified in the associated MA-UNITDATA.request primitive.

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

a) Successful.
b) Undeliverable (excessive data length).
c) Undeliverable (non-null source routing).
d) Undeliverable: unsupported priority (for priorities other than Contention or ContentionFree at a non-QoS STA; or for priorities other than Contention, ContentionFree, or an integer between and including 0 and 15 at a QoS STA).
e) Undeliverable: unsupported service class (for service classes other than ReorderableMulticast or StrictlyOrdered for non-QoS STAs and service classes other than QoSAck or QoSNoAck for QoS STAs).
f) Unavailable priority (for ContentionFree when no PC or HC is available, or an integer between and including 1 and 15 at a STA that is associated in a non-QoS BSS, or an integer between and including 8 and 15 at a STA that is a member of an IBSS, in which case the MSDU is transmitted with a provided priority of Contention).
g) Undeliverable: unavailable service class (for StrictlyOrdered service when the STA's power management mode is other than "active" for non-QoS STAs; QoS STAs do not return this value as they do not provide the StrictlyOrdered service).
h) Undeliverable (no BSS available).
i) Undeliverable (cannot encrypt with a null key).

The provided priority parameter specifies the priority that was used for the associated data unit transfer (Contention, 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 non-QoS STAs, the value of this parameter is ReorderableMulticast or StrictlyOrdered. In QoS STAs, it is QoSAck or QoSNoAck.

#### 6.2.1.3.3 When generated

The MA-UNITDATA.confirm primitive is passed from the MAC sublayer entity to the LLC sublayer entity to indicate the status of the service provided for the corresponding MA-UNITDATA.request primitive.

#### 6.2.1.3.4 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent upon the type of operation employed by the LLC sublayer entity.


Copyright © 2007 IEEE. All rights reserved.

# 7. Frame formats

The format of the MAC frames is specified in this clause. A STA shall be able properly to construct a subset of the frames specified in this clause for transmission and to decode a (potentially different) subset of the frames specified in this clause upon validation following reception. 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

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 *FCS*, which contains an IEEE 32-bit CRC.

### 7.1.1 Conventions

The MPDUs or frames in the MAC sublayer are described as a sequence of fields in specific order. Each figure in Clause 7 depicts the fields/subfields as they appear in the MAC frame and in the order in which they are passed to the physical layer convergence procedure (PLCP), from left to right.

In figures, all bits within fields are numbered, from 0 to *k*, where the length of the field is $k + 1$ bits. The octet boundaries within a field can be obtained by taking the bit numbers of the field modulo 8. Octets within numeric fields that are longer than a single octet are depicted in increasing order of significance, from lowest numbered bit to highest numbered bit. The octets in fields longer than a single octet are sent to the PLCP in order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

Any field containing a CRC is an exception to this convention and is transmitted commencing with the coefficient of the highest-order term.

MAC addresses are assigned as ordered sequences of bits. The Individual/Group bit is always transferred first and is bit 0 of the first octet.

Values specified in decimal are coded in natural binary unless otherwise stated. The values in Table 7-1 are in binary, with the bit assignments shown in the table. Values in other tables are shown in decimal notation.

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.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 107 of 1232

— *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.

Reserved fields and subfields are set to 0 upon transmission and are ignored upon reception.

### 7.1.2 General frame format

The MAC frame format comprises a set of fields that occur in a fixed order in all frames. Figure 7-1 depicts the general MAC frame format. The first three fields (Frame Control, Duration/ID, and Address 1) and the last field (FCS) in Figure 7-1 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 type 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.

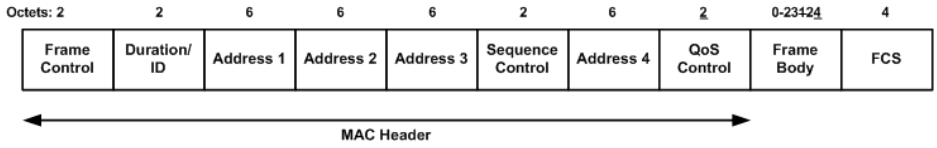



**Figure 7-1—MAC frame format**

### 7.1.3 Frame fields

#### 7.1.3.1 Frame Control field

The Frame Control field consists of the following subfields: Protocol Version, Type, Subtype, To DS, From DS, More Fragments, Retry, Power Management, More Data, Protected Frame, and Order. The format of the Frame Control field is illustrated in Figure 7-2.

| B0 B1 | B2 B3 | B4 B7 | B8 | B9 | B10 | B11 | B12 | B13 | B14 | B15 |
|---|---|---|---|---|---|---|---|---|---|---|
| Protocol Version | Type | Subtype | To DS | From DS | More Frag | Retry | Pwr Mgt | More Data | Protected Frame | Order |
| Bits : 2 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 7-2—Frame Control field**


Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.1.1 Protocol Version field

The Protocol Version field is 2 bits in length and is invariant in size and placement across all revisions of this standard. For this standard, the value of the protocol version is 0. All other values are reserved. The revision level will be incremented only when a fundamental incompatibility exists between a new revision and the prior edition of the standard. A MAC entity that receives a frame with a higher revision level than it supports shall discard the frame without indication to the sending STA or to LLC.

#### 7.1.3.1.2 Type and Subtype fields

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 has several defined subtypes. In data frames, the most significant bit (MSB) of the Subtype field, b7, is defined as the QoS subfield. Table 7-1 defines the valid combinations of type and subtype. (The numeric values in Table 7-1 are shown in binary.)

**Table 7-1—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 |
| 00 | Management | 1001 | ATIM |
| 00 | Management | 1010 | Disassociation |
| 00 | Management | 1011 | Authentication |
| 00 | Management | 1100 | Deauthentication |
| 00 | Management | 1101 | Action |
| 00 | Management | 1110–1111 | Reserved |
| 01 | Control | 0000–0111 | Reserved |
| 01 | Control | 1000 | Block Ack Request (BlockAckReq) |
| 01 | Control | 1001 | Block Ack (BlockAck) |
| 01 | Control | 1010 | PS-Poll |
| 01 | Control | 1011 | RTS |
| 01 | Control | 1100 | CTS |
| 01 | Control | 1101 | ACK |
| 01 | Control | 1110 | CF-End |
| 01 | Control | 1111 | CF-End + CF-Ack |

Copyright © 2007 IEEE. All rights reserved.

**Table 7-1—Valid type and subtype combinations *(continued)***

| Type value b3 b2 | Type description | Subtype value b7 b6 b5 b4 | Subtype description |
|---|---|---|---|
| 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 (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 | QoS Data |
| 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 that 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 and From DS fields

The meaning of the combinations of values for the To DS and From DS fields are shown in Table 7-2.

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

| To DS and From DS values | Meaning |
|---|---|
| To DS = 0<br>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 STA to another non-AP STA within the same BSS, as well as all management and control frames. |
| To DS = 1<br>From DS = 0 | A data frame destined for the DS or being sent by a STA associated with an AP to the Port Access Entity in that AP. |
| To DS = 0<br>From DS = 1 | A data frame exiting the DS or being sent by the Port Access Entity in an AP. |
| To DS = 1<br>From DS = 1 | A data frame using the four-address format. This standard does not define procedures for using this combination of field values. |


Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.1.4 More Fragments field

The More Fragments field is 1 bit in length and is set to 1 in all data or management type frames that have another fragment of the current MSDU or current MMPDU to follow. It is set to 0 in all other frames.

#### 7.1.3.1.5 Retry field

The Retry field is 1 bit in length and is set to 1 in any data or management type frame that is a retransmission of an earlier frame. It is set to 0 in all other frames. A receiving STA uses this indication to aid in the process of eliminating duplicate frames.

#### 7.1.3.1.6 Power Management field

The Power Management field is 1 bit in length and is used to indicate the power management mode of a STA. The value of this field remains constant in each frame from a particular STA within a frame exchange sequence defined in 9.12. The value indicates the mode in which the STA will be after the successful completion of the frame exchange sequence.

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

#### 7.1.3.1.7 More Data field

The More Data field is 1 bit in length and is used to indicate to a STA in PS mode that more MSDUs or MMPDUs are buffered for that STA at the AP. The More Data field is valid in directed data or management type frames transmitted by an AP to a STA in PS mode. A value of 1 indicates that at least one additional buffered MSDU or MMPDU is present for the same STA.

The More Data field may be set to 1 in directed data type frames transmitted by a CF-Pollable STA to the PC in response to a CF-Poll to indicate that the STA has at least one additional buffered MSDU available for transmission in response to a subsequent CF-Poll.

The More Data field is set to 0 in all other directed frames.

The More Data field is set to 1 in broadcast/multicast frames transmitted by the AP when additional broadcast/multicast MSDUs or MMPDUs remain to be transmitted by the AP during this beacon interval. The More Data field is set to 0 in broadcast/multicast frames transmitted by the AP when no more broadcast/ multicast MSDUs or MMPDUs remain to be transmitted by the AP during this beacon interval and in all broadcast/multicast frames transmitted by non-AP STAs.

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

#### 7.1.3.1.8 Protected Frame field

The Protected Frame field is 1 bit in length. The Protected Frame field is set to 1 if the Frame Body field contains information that has been processed by a cryptographic encapsulation algorithm. The Protected Frame field is set to 1 only within data frames and within management frames of subtype Authentication. The Protected Frame field is set to 0 in all other frames. When the Protected Frame field is set to 1, the Frame Body field is protected utilizing the cryptographic encapsulation algorithm and expanded as defined in Clause 8. The Protected Frame field is set to 0 in Data frames of subtype Null Function, CF-ACK (no data), CF-Poll (no data), and CF-ACK+CF-Poll (no data) (see 8.3.2.2 and 8.3.3.1, which show that the frame body must be 1 octet or longer to apply the encapsulation).

Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.1.9 Order field

The Order field is 1 bit in length and is set to 1 in any non-QoS data 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 QoS STAs set this subfield to 0.

### 7.1.3.2 Duration/ID field

The Duration/ID field is 16 bits in length. The contents of this field 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 frames of subtype PS-Poll, the Duration/ID field carries the association identifier (AID) of the STA that transmitted the frame in the 14 least significant bits (LSB), 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 non-QoS STAs, during the CFP, the Duration/ID field is set to a fixed value of 32 768.

c) In all other frames sent by non-QoS STAs and control frames sent by QoS STAs, 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 QoS STAs, the Duration/ID field contains a duration value as defined for each frame type in 7.1.4.

When 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 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 7-3.

**Table 7-3—Duration/ID field encoding**

| Bits 0–13 | Bit 14 | Bit 15 | Usage |
|---|---|---|---|
| 0–32 767 | | 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 |
| 1–2007 | 1 | 1 | AID in PS-Poll frames |
| 2008–16 383 | 1 | 1 | Reserved |


Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.3 Address fields

There are four address fields in the MAC frame format. These fields are used to indicate the basic service set identification (BSSID), source address (SA), destination address (DA), transmitting STA address (TA), and receiving STA address (RA). Certain frames may not contain some of the address fields.

Certain address field usage is specified by the relative position of the address field (1–4) within the MAC header, independent of the type of address present in that field. For example, receiver address matching is always performed on the contents of the Address 1 field in received frames, and the receiver address of CTS and ACK frames is always obtained from the Address 2 field in the corresponding RTS frame, or from the frame being acknowledged.

##### 7.1.3.3.1 Address representation

Each Address field contains a 48-bit address as defined in 5.2 of IEEE Std 802-1990.

##### 7.1.3.3.2 Address designation

A MAC sublayer address is one of the following two types:

a) *Individual address.* The address assigned to a particular STA on the network.

b) *Group address.* A multidestination address, which may be in use by one or more STAs on a given network. The two kinds of group addresses are as follows:

   1) *Multicast-group address.* An address associated by higher level convention with a group of logically related STAs.

   2) *Broadcast address.* A distinguished, predefined multicast address that always denotes the set of all STAs on a given LAN. All ones are interpreted to be the broadcast address. This group is predefined for each communication medium to consist of all STAs actively connected to that medium; it is used to broadcast to all the active STAs on that medium.

The address space is also partitioned into locally administered and universal (globally administered) addresses. The nature of a body and the procedures by which it administers these universal (globally administered) addresses is beyond the scope of this standard. See IEEE Std 802-1990 for more information.

##### 7.1.3.3.3 BSSID field

The BSSID field is a 48-bit field of the same format as an IEEE 802 MAC address. This field uniquely identifies each BSS. The value of this field, in an infrastructure BSS, is the MAC address currently in use by the STA in the AP of the BSS.

The value of this field in an IBSS is a locally administered IEEE MAC address formed from a 46-bit random number generated according to the procedure defined in 11.1.3. The individual/group bit of the address is set to 0. The universal/local bit of the address is set to 1. This mechanism is used to provide a high probability of selecting a unique BSSID.

The value of all 1s is used to indicate the wildcard BSSID. A wildcard BSSID shall not be used in the BSSID field except for management frames of subtype probe request.

##### 7.1.3.3.4 DA field

The DA field contains an IEEE MAC individual or group address that identifies the MAC entity or entities intended as the final recipient(s) of the MSDU (or fragment thereof) contained in the frame body field.

Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.3.5 SA field

The SA field contains an IEEE MAC individual address that identifies the MAC entity from which the transfer of the MSDU (or fragment thereof) contained in the frame body field was initiated. The individual/group bit is always transmitted as a zero in the source address.

#### 7.1.3.3.6 RA field

The RA field contains an IEEE MAC individual or group address that identifies the intended immediate recipient STA(s), on the WM, for the information contained in the frame body field.

#### 7.1.3.3.7 TA field

The TA field contains an IEEE MAC individual address that identifies the STA that has transmitted, onto the WM, the MPDU contained in the frame body field. The Individual/Group bit is always transmitted as a zero in the transmitter address.

### 7.1.3.4 Sequence Control field

The Sequence Control field is 16 bits in length and consists of two subfields, the Sequence Number and the Fragment Number. The format of the Sequence Control field is illustrated in Figure 7-3. Sequence Control field is not present in control frames.

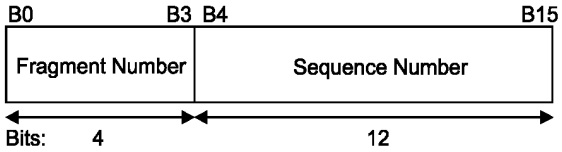



**Figure 7-3—Sequence Control field**

#### 7.1.3.4.1 Sequence Number field

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.

Non-QoS STAs, as well as QoS STAs operating as non-QoS STAs because they are in a non-QoS BSS or non-QoS IBSS, assign sequence numbers, to management frames and data frames (QoS subfield of the Subtype field is set to 0), from a single modulo-4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU.

QoS STAs associated in a QoS BSS 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 QoS STAs 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 a copy of the sequence number assigned to that MSDU or MMPDU. The sequence number remains constant in all retransmissions of an MSDU, MMPDU, or fragment thereof.


Copyright © 2007 IEEE. All rights reserved.

#### 7.1.3.4.2 Fragment Number field

The Fragment Number field is a 4-bit field indicating the number of each fragment of an MSDU or MMPDU. The fragment number is set to 0 in the first or only fragment of an MSDU or MMPDU and is incremented by one for each successive fragment of that MSDU or MMPDU. The fragment number remains constant in all retransmissions of the fragment.

### 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 of the Subtype field 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 STA) 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 7-4.

**Table 7-4—QoS Control field**

| Applicable frame (sub) types | Bits 0–3 | Bit 4 | Bits 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 | AP PS Buffer State |
| QoS data frames sent by non-AP STAs | 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 7.3.2.30) and is shown in Table 7-5. Additional information on the interpretation of the contents of this field appears in 6.1.1.2 .

**Table 7-5—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

Copyright © 2007 IEEE. All rights reserved.

poll is intended. The requirement to respond to that TID is nonbinding, and a STA 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 acknowledgment policy that is followed upon the delivery of the MPDU. The interpretation of these 2 bits is given in Table 7-6.

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

| Bits in QoS Control field | | Meaning |
|---|---|---|
| Bit 5 | Bit 6 | |
| 0 | 0 | Normal Ack.<br>The addressed recipient returns an ACK or QoS +CF-Ack frame after a short interframe space (SIFS) period, according to the procedures defined in 9.2.8, 9.3.3, and 9.9.2.3.<br>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<br>The addressed recipient takes no action upon receipt of the frame. More details are provided in 9.11.<br>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. |
| 0 | 1 | No explicit acknowledgment.<br>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.<br>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<br>The addressed recipient takes no action upon the receipt of the frame except for recording the state. The recipient can expect a<br>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 BSS. In QoS data frames with subtypes that include CF-Poll, the addressed STA is granted a TXOP that begins a SIFS


Copyright © 2007 IEEE. All rights reserved.

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 an AP 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 STA sending this frame. The Queue Size subfield is present in QoS data frames sent by STAs associated in a BSS with bit 4 of the QoS Control field set to 1. The AP may use information contained in the Queue Size subfield to determine the TXOP duration assigned to non-AP STA.

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 STA (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 STAs associated in a BSS with bit 4 of the QoS Control field set to 0. The AP 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 STA should account for this when attempting to determine whether a given transmission fits within a specified TXOP duration.

#### 7.1.3.5.7 AP PS Buffer State subfield

The AP PS Buffer State subfield, defined in Figure 7-4, is an 8-bit field that indicates the PS buffer state at the AP for a non-AP STA. The AP PS Buffer State subfield is further subdivided into three subfields: Buffer State Indicated, Highest-Priority Buffered AC, and AP Buffered Load.

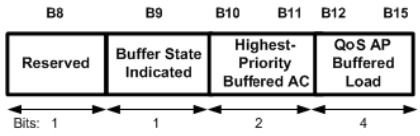



**Figure 7-4—QoS AP PS Buffer State subfield**

Copyright © 2007 IEEE. All rights reserved.

The Buffered State Indicated subfield is 1 bit in length and is used to indicate whether the AP PS buffer state is specified. A value of 1 indicates that the AP 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 AP, excluding the MSDU of the present frame.

The AP 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 QoS AP (excluding the MSDU of the present QoS data frame). An AP Buffered Load field value of 15 indicates that the buffer size is greater than 57 344 octets. An AP 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 AP Buffered Load subfield are reserved; and the values of these subfields are either unspecified or unknown.

#### 7.1.3.6 Frame Body field

The Frame Body is a variable length field that contains information specific to individual frame types and subtypes. The minimum frame body is 0 octets. The maximum length frame body is defined by the maximum length (MSDU + ICV + IV), where integrity check value (ICV) and initialization vector (IV) are the WEP fields defined in 8.2.1.

#### 7.1.3.7 FCS field

The FCS field is a 32-bit field containing a 32-bit CRC. The FCS is calculated over all the fields of the MAC header and the Frame Body field. These are referred to as the *calculation fields.*

The FCS is calculated using the following standard generator polynomial of degree 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The FCS is the ones complement of the sum (modulo 2) of the following:

a) The remainder of $x^k \times (x^{31} + x^{30} + x^{29} + \ldots + x^2 + x + 1)$ divided (modulo 2) by $G(x)$, where $k$ is the number of bits in the calculation fields, and

b) The remainder after multiplication of the contents (treated as a polynomial) of the calculation fields by $x^{32}$ and then division by $G(x)$.

The FCS field is transmitted commencing with the coefficient of the highest-order term.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the calculation fields by the generator polynomial $G(x)$. The ones complement of this remainder is transmitted, with the highest-order bit first, as the FCS field.

At the receiver, the initial remainder is preset to all ones and the serial incoming bits of the calculation fields and FCS, when divided by $G(x)$, results in the absence of transmission errors, in a unique nonzero remainder value. The unique remainder value is the polynomial:

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$$


Copyright © 2007 IEEE. All rights reserved.

### 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 QoS STAs 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 (dot11EDCAQAP-TableTXOPLimit for the AP) 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

In the following descriptions, “immediately previous” frame means a frame whose reception concluded within the SIFS interval preceding the start of the current frame.

Copyright © 2007 IEEE. All rights reserved.

The subfields within the Frame Control field of control frames are set as illustrated in Figure 7-5.

| B0 | | | | | | | | | | B15 |
|---|---|---|---|---|---|---|---|---|---|---|
| Protocol Version | Type | Subtype | To DS | From DS | More Frag | Retry | Pwr Mgt | More Data | Protected Frame | Order |
| Protocol Version | Control | Subtype | 0 | 0 | 0 | 0 | Pwr Mgt | 0 | 0 | 0 |
| Bits : 2 | 2 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 7-5—Frame Control field subfield values within control frames**

### 7.2.1.1 RTS frame format

The frame format for the RTS frame is as defined in Figure 7-6.

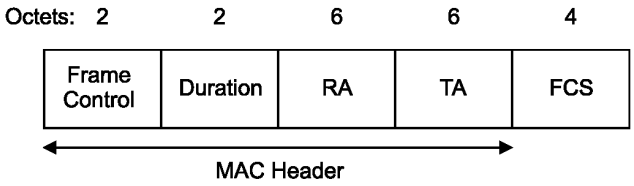



**Figure 7-6—RTS frame**

The RA field of the RTS frame is the address of the STA, on the WM, that is the intended immediate recipient of the pending directed data or management frame.

The TA field is the address of the STA transmitting the RTS frame.

For all RTS frames sent by non-QoS STAs, the duration value is the time, in microseconds, required to transmit the pending data or management frame, plus one 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 STAs 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.


Copyright © 2007 IEEE. All rights reserved.

#### 7.2.1.2 CTS frame format

The frame format for the CTS frame is as defined in Figure 7-7.

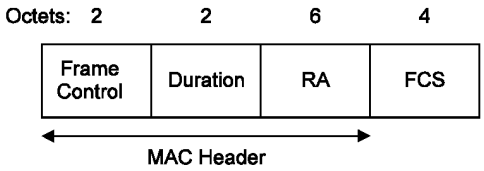



**Figure 7-7—CTS frame**

When the CTS frame follows an RTS frame, the RA field of the CTS frame is copied from the TA field of the immediately previous RTS frame to which the CTS is a response. When the CTS is the first frame in a frame exchange, the RA field is set to the MAC address of the transmitter.

For all CTS frames sent in response to RTS frames, the 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 a non-QoS STA, if 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 two SIFS intervals plus one ACK frame. At a non-QoS STA, if 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 STAs 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.

Copyright © 2007 IEEE. All rights reserved.

#### 7.2.1.3 ACK frame format

The frame format for the ACK frame is as defined in Figure 7-8.

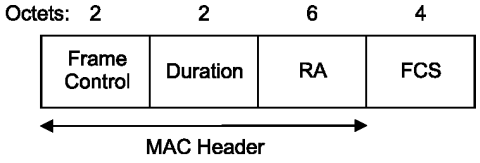



**Figure 7-8—ACK frame**

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 non-QoS STAs, if 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. In all other ACK frames, the duration value is the value obtained from the Duration/ID field of the immediately previous data, 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.

#### 7.2.1.4 PS-Poll frame format

The frame format for the PS-Poll frame is as defined in Figure 7-9.

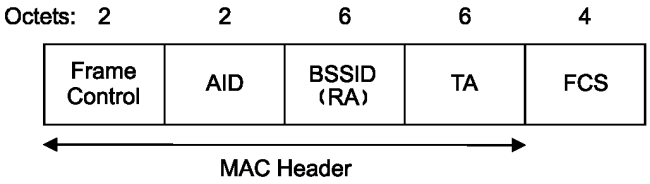



**Figure 7-9—PS-Poll frame**

The BSSID is the address of the STA contained in the AP. The TA field is the address of the STA transmitting the frame. The AID is the value assigned to the STA transmitting the frame by the AP in the association response frame that established that STA's current association.

The AID value always has its two MSBs each set to 1.

#### 7.2.1.5 CF-End frame format

The frame format for the CF-End frame is as defined in Figure 7-10.

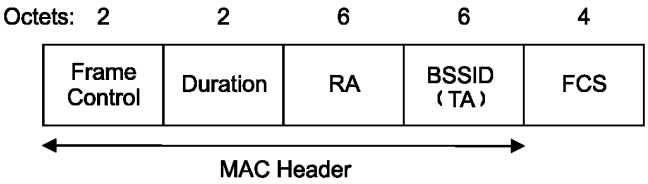



**Figure 7-10—CF-End frame**


Copyright © 2007 IEEE. All rights reserved.

The BSSID field is the address of the STA contained in the AP. The RA field is the broadcast group address.

The Duration field is set to 0.

#### 7.2.1.6 CF-End+CF-Ack frame format

The frame format for the CF-End+CF-Ack frame is as defined in Figure 7-11.

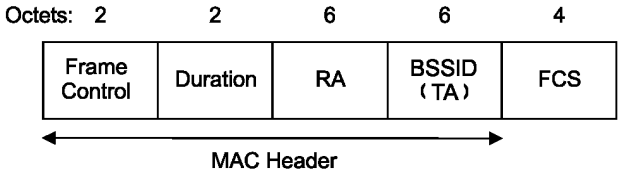



**Figure 7-11—CF-End+CF-Ack frame**

The BSSID field is the address of the STA contained in the AP. The RA field is the broadcast group address.

The Duration field is set to 0.

#### 7.2.1.7 Block Ack Request (BlockAckReq) frame format

The frame format of the BlockAckReq frame is defined in Figure 7-12.

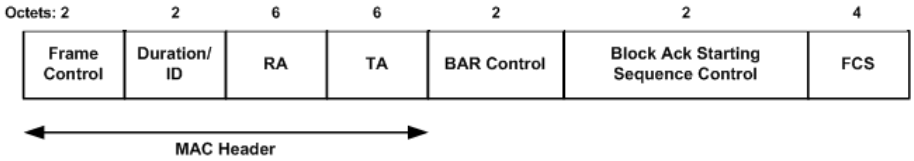



**Figure 7-12—BlockAckReq frame**

The Duration/ID field value is greater than or equal to the time[16], 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 STA.

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

The BAR Control field is shown in Figure 7-13.

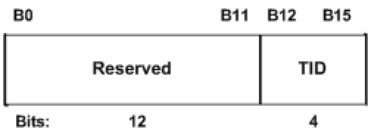



**Figure 7-13—BAR Control field**

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

Copyright © 2007 IEEE. All rights reserved.

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 7-14. 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.

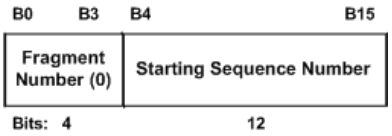



**Figure 7-14—Block Ack Starting Sequence Control field**

### 7.2.1.8 Block Ack (BlockAck) frame format

The frame format of the BlockAck frame is defined in Figure 7-15.

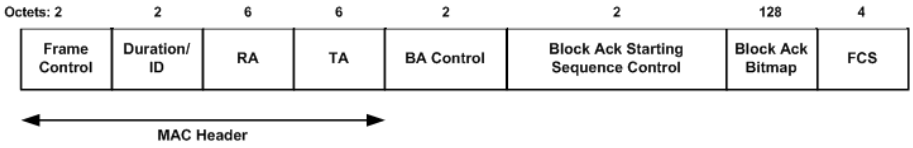



**Figure 7-15—BlockAck frame**

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[17] 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 STA that requested the Block Ack.

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

The BA Control field defined in Figure 7-16 consists of the TID subfield.

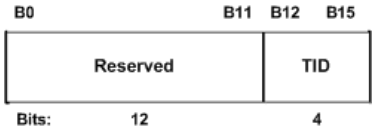



**Figure 7-16—BA Control field**

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

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


Copyright © 2007 IEEE. All rights reserved.

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

The frame format for a data frame is dependent on the QoS subfield of the Subtype field and is as defined in Figure 7-17.

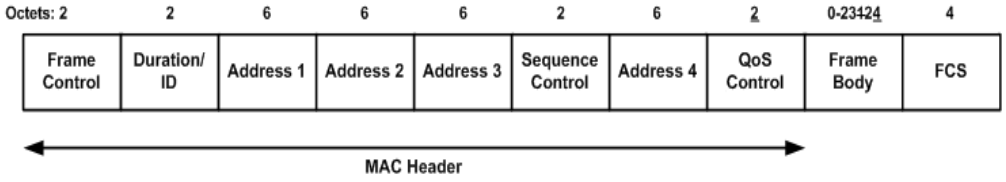



**Figure 7-17—Data frame**

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 QoS STA always uses QoS data frames for data transmissions to other QoS STAs. A QoS STA uses frames with the QoS subfield of the Subtype field set to 0 for data transmissions to non-QoS STAs. A non-QoS STA 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 STA knows that all STAs in a BSS have QoS capability, in which case the transmitting STAs use 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 BSS that are members of the multicast group have QoS capability, in which case STAs 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 fields in the Frame Control field and are defined in Table 7-7. 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 7-7—Address field contents**

| To DS | From DS | Address 1 | Address 2 | Address 3 | Address 4 |
|---|---|---|---|---|---|
| 0 | 0 | RA = DA | TA = SA | BSSID | N/A |
| 0 | 1 | RA = DA | TA = BSSID | SA | N/A |
| 1 | 0 | RA = BSSID | TA = SA | DA | N/A |
| 1 | 1 | RA | TA | DA | SA |

Copyright © 2007 IEEE. All rights reserved.

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 from a STA in the BSS of which the receiving STA is a member.

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

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

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

The RA field is the unicast address of the STA that is the immediate intended receiver of the frame or the multicast or broadcast address of the STAs that are the immediate intended receivers of the frame.

The TA field is the address of the STA 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 (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 PCF, the Duration/ID field is set to 32 768. Within all data frames sent by the QoS STA, the Duration/ID field contains a duration value as defined in 7.1.4. Within all data frames sent during the CP by non-QoS STAs, 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.


Copyright © 2007 IEEE. All rights reserved.

— 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 (without regard for the RA, DA, and/or BSSID address values that may be present in these frames) to update their NAV settings as appropriate under the coordination function rules.

### 7.2.3 Management frames

The frame format for a management frame is independent of frame subtype and is as defined in Figure 7-18.

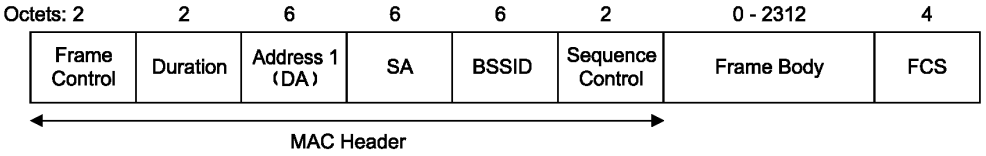



**Figure 7-18—Management frame format**

A STA uses the contents of the Address 1 (DA) field to perform the address matching for receive decisions. In the case where the Address 1 (DA) field contains a group address and the frame type is other than Beacon, the BSSID also is validated to ensure that the broadcast or multicast originated from a STA in the BSS of which the receiving STA is a member. If the frame type is Beacon, other address matching rules apply, as specified in 11.1.2.3. Frames of type Probe Request with a group address in the Address 1 field are processed as described in 11.1.3.2.1.

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 wildcard BSSID as defined in the procedures specified in 11.1.3.

The DA field is the destination of the frame.

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

Within all management 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 QoS STA, the Duration field contains a duration value as defined in 7.1.4. Within all management frames sent during the CP by non-QoS STAs, 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.

Copyright © 2007 IEEE. All rights reserved.

— 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 set to 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 fields followed by the information elements defined for each management frame subtype. All fields and information elements are mandatory unless stated otherwise and appear in the specified, relative order. STAs that encounter an element ID they do not 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. A STA receiving a vendor-specific IE that it does not support shall ignore the vendor-specific IE. Unused element ID codes are reserved.

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

#### 7.2.3.1 Beacon frame format

The frame body of a management frame of subtype Beacon contains the information shown in Table 7-8. 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 and FH Pattern Table elements. If both FH Parameters and FH Pattern Table elements are sent, they shall describe the same hopping pattern. Note that the information described in FH Parameters and FH Pattern Table elements also may be contained in the Probe Response frame.

**Table 7-8—Beacon frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Timestamp | |
| 2 | Beacon interval | |
| 3 | Capability | |
| 4 | Service Set Identifier (SSID) | |
| 5 | Supported rates | |
| 6 | Frequency-Hopping (FH) Parameter Set | The FH Parameter Set information element is present within Beacon frames generated by STAs using FH PHYs. |
| 7 | DS Parameter Set | The DS Parameter Set information element is present within Beacon frames generated by STAs using Clause 15, Clause 18, and Clause 19 PHYs. |
| 8 | CF Parameter Set | The CF Parameter Set information element is present only within Beacon frames generated by APs supporting a PCF. |
| 9 | IBSS Parameter Set | The IBSS Parameter Set information element is present only within Beacon frames generated by STAs in an IBSS. |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-8—Beacon frame body *(continued)***

| Order | Information | Notes |
|---|---|---|
| 10 | Traffic indication map (TIM) | The TIM information element is present only within Beacon frames generated by APs. |
| 11 | Country | The Country information element shall be present when dot11MultiDomainCapabilityEnabled is true or dot11SpectrumManagementRequired is true. |
| 12 | FH Parameters | FH Parameters as specified in 7.3.2.10 may be included if dot11MultiDomainCapabilityEnabled is true. |
| 13 | FH Pattern Table | FH Pattern Table information as specified in 7.3.2.11 may be included if dot11MultiDomainCapabilityEnabled is true. |
| 14 | Power Constraint | Power Constraint element shall be present if dot11SpectrumManagementRequired is true. |
| 15 | Channel Switch Announcement | Channel Switch Announcement element may be present if dot11SpectrumManagementRequired is true. |
| 16 | Quiet | Quiet element may be present if dot11SpectrumManagementRequired is true. |
| 17 | IBSS DFS | IBSS DFS element shall be present if dot11SpectrumManagementRequired is true in an IBSS. |
| 18 | TPC Report | TPC Report element shall be present if dot11SpectrumManagementRequired is true. |
| 19 | ERP Information | The ERP Information element is present within Beacon frames generated by STAs using extended rate PHYs (ERPs) defined in Clause 19 and is optionally present in other cases. |
| 20 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 21 | RSN | The RSN information element shall be present within Beacon frames generated by STAs that have dot11RSNAEnabled set to TRUE. |
| 22 | BSS Load | The BSS Load element is present when dot11QosOption-Implemented and dot11QBSSLoadImplemented are both true. |
| 23 | EDCA Parameter Set | 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 element is present when dot11QosOption-Implemented is true and EDCA Parameter Set element is not present. |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

#### 7.2.3.2 IBSS ATIM frame format

The frame body of a management frame of subtype ATIM is null.

#### 7.2.3.3 Disassociation frame format

The frame body of a management frame of subtype Disassociation contains the information shown in Table 7-9.

Copyright © 2007 IEEE. All rights reserved.

**Table 7-9—Disassociation frame body**

| Order | Information |
|---|---|
| 1 | Reason code |
| 2 | One or more vendor-specific information elements may appear in this frame. |

### 7.2.3.4 Association Request frame format

The frame body of a management frame of subtype Association Request contains the information shown in Table 7-10.

**Table 7-10—Association Request frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Capability | |
| 2 | Listen interval | |
| 3 | SSID | |
| 4 | Supported rates | |
| 5 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 6 | Power Capability | The Power Capability element shall be present if dot11SpectrumManagementRequired is true. |
| 7 | Supported Channels | The Supported Channels element shall be present if dot11SpectrumManagementRequired is true. |
| 8 | RSN | The RSN information element is only present within Association Request frames generated by STAs that have dot11RSNAEnabled set to TRUE. |
| 9 | QoS Capability | The QoS Capability element is present when dot11QosOption-Implemented is true. |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

### 7.2.3.5 Association Response frame format

The frame body of a management frame of subtype Association Response contains the information shown in Table 7-11.

**Table 7-11—Association Response frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Capability | |
| 2 | Status code | |
| 3 | AID | |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-11—Association Response frame body *(continued)***

| Order | Information | Notes |
|---|---|---|
| 4 | Supported rates | |
| 5 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 6 | EDCA Parameter Set | |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

### 7.2.3.6 Reassociation Request frame format

The frame body of a management frame of subtype Reassociation Request contains the information shown in Table 7-12.

**Table 7-12—Reassociation Request frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Capability | |
| 2 | Listen interval | |
| 3 | Current AP address | |
| 4 | SSID | |
| 5 | Supported rates | |
| 6 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 7 | Power Capability | The Power Capability element shall be present if dot11SpectrumManagementRequired is true. |
| 8 | Supported Channels | The Supported Channels element shall be present if dot11SpectrumManagementRequired is true. |
| 9 | RSN | The RSN information element is only present within Reassociation Request frames generated by STAs that have dot11RSNAEnabled set to TRUE. |
| 10 | QoS Capability | The QoS Capability element is present when dot11QosOption-Implemented is true. |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

### 7.2.3.7 Reassociation Response frame format

The frame body of a management frame of subtype Reassociation Response contains the information shown in Table 7-13.

Copyright © 2007 IEEE. All rights reserved.

**Table 7-13—Reassociation Response frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Capability | |
| 2 | Status code | |
| 3 | AID | |
| 4 | Supported rates | |
| 5 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 6 | EDCA Parameter Set | |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

### 7.2.3.8 Probe Request frame format

The frame body of a management frame of subtype Probe Request contains the information shown in Table 7-14. If the dot11MultiDomainCapabilityEnabled attribute is true, a STA may include a Request information element in the Probe Request frame. The format of the Request information element is specified in 7.3.2.12.

**Table 7-14—Probe Request frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | SSID | |
| 2 | Supported rates | |
| 3 | Request information | May be included if dot11MultiDomainCapabilityEnabled is true. |
| 4 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

### 7.2.3.9 Probe Response frame format

The frame body of a management frame of subtype Probe Response contains the information shown in Table 7-15. If the dot11MultiDomainCapabilityEnabled attribute is true, the Probe Response frame contains a Country information element and all information elements identified by the Requested Element IDs of a Request information element. Note that the information returned as a result of a Probe Request frame with a Request information element may include the FH parameters and/or the FH Pattern Table possibly replicating optional elements identified by orders 12 and 13.

A STA shall return only the information elements that it supports. In an improperly formed Request information element, a STA may ignore the first information element requested that is not ordered properly and all subsequent information elements requested. In the probe response frame, the STA shall return the requested information elements in the same order as requested in the Request information element.


Copyright © 2007 IEEE. All rights reserved.

**Table 7-15—Probe Response frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Timestamp | |
| 2 | Beacon interval | |
| 3 | Capability | |
| 4 | SSID | |
| 5 | Supported rates | |
| 6 | FH Parameter Set | The FH Parameter Set information element is present within Probe Response frames generated by STAs using FH PHYs. |
| 7 | DS Parameter Set | The DS Parameter Set information element is present within Probe Response frames generated by STAs using Clause 15, Clause 18, and Clause 19 PHYs. |
| 8 | CF Parameter Set | The CF Parameter Set information element is present only within Probe Response frames generated by APs supporting a PCF. |
| 9 | IBSS Parameter Set | The IBSS Parameter Set information element is present only within Probe Response frames generated by STAs in an IBSS. |
| 10 | Country | Included if dot11MultiDomainCapabilityEnabled or dot11SpectrumManagementRequired is true. |
| 11 | FH Parameters | FH Parameters, as specified in 7.3.2.10, may be included if dot11MultiDomainCapabilityEnabled is true. |
| 12 | FH Pattern Table | FH Pattern Table information, as specified in 7.3.2.11, may be included if dot11MultiDomainCapabilityEnabled is true. |
| 13 | Power Constraint | Shall be included if dot11SpectrumManagementRequired is true. |
| 14 | Channel Switch Announcement | May be included if dot11SpectrumManagementRequired is true. |
| 15 | Quiet | May be included if dot11SpectrumManagementRequired is true. |
| 16 | IBSS DFS | Shall be included if dot11SpectrumManagementRequired is true in an IBSS. |
| 17 | TPC Report | Shall be included if dot11SpectrumManagementRequired is true. |
| 18 | ERP Information | The ERP Information element is present within Probe Response frames generated by STAs using ERPs and is optionally present in other cases. |
| 19 | Extended Supported Rates | The Extended Supported Rates element is present whenever there are more than eight supported rates, and it is optional otherwise. |
| 20 | RSN | The RSN information element is only present within Probe Response frames generated by STAs that have dot11RSNA-Enabled set to TRUE. |
| 21 | BSS Load | The BSS Load element is present when dot11QosOption-Implemented and dot11QBSSLoadImplemented are both true. |
| 22 | EDCA Parameter Set | The EDCA Parameter Set element is present when dot11QosOptionImplemented is true. |
| Last–*1* | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements, except the Requested Information elements. |
| Last–*n* | Requested information elements | Elements requested by the Request information element of the Probe Request frame. |

Copyright © 2007 IEEE. All rights reserved.

### 7.2.3.10 Authentication frame format

The frame body of a management frame of subtype Authentication contains the information shown in Table 7-16. Only Authentication frames with the authentication algorithm set to Open System authentication may be used within an RSNA. RSNA STAs shall not associate if shared authentication was invoked prior to RSN association.

**Table 7-16—Authentication frame body**

| Order | Information | Notes |
|---|---|---|
| 1 | Authentication algorithm number | |
| 2 | Authentication transaction sequence number | |
| 3 | Status code | The status code information is reserved in certain Authentication frames as defined in Table 7-17. |
| 4 | Challenge text | The challenge text information is present only in certain Authentication frames as defined in Table 7-17. |
| Last | Vendor Specific | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

**Table 7-17—Presence of challenge text information element**

| Authentication algorithm | Authentication transaction sequence no. | Status code | Challenge text |
|---|---|---|---|
| Open System | 1 | Reserved | Not present |
| Open System | 2 | Status | Not present |
| Shared Key | 1 | Reserved | Not present |
| Shared Key | 2 | Status | Present |
| Shared Key | 3 | Reserved | Present |
| Shared Key | 4 | Status | Not present |

### 7.2.3.11 Deauthentication

The frame body of a management frame of subtype Deauthentication contains the information shown in Table 7-18.

**Table 7-18—Deauthentication frame body**

| Order | Information |
|---|---|
| 1 | Reason code |
| Last | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |


Copyright © 2007 IEEE. All rights reserved.

#### 7.2.3.12 Action frame format

The frame body of a management frame of subtype Action contains the information shown in Table 7-19.

**Table 7-19—Action frame body**

| Order | Information |
|---|---|
| 1 | Action |
| Last | One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements. |

## 7.3 Management frame body components

### 7.3.1 Fields that are not information elements

#### 7.3.1.1 Authentication Algorithm Number field

The Authentication Algorithm Number field indicates a single authentication algorithm. The length of the Authentication Algorithm Number field is 2 octets. The Authentication Algorithm Number field is illustrated in Figure 7-19. The following values are defined for authentication algorithm number:

Authentication algorithm number = 0: Open System

Authentication algorithm number = 1: Shared Key

All other values of authentication number are reserved.

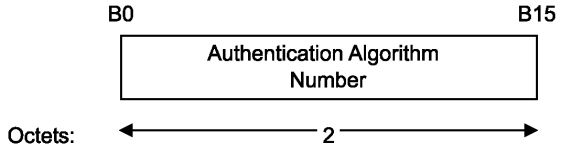



**Figure 7-19—Authentication Algorithm Number field**

#### 7.3.1.2 Authentication Transaction Sequence Number field

The Authentication Transaction Sequence Number field indicates the current state of progress through a multistep transaction. The length of the Authentication Transaction Sequence Number field is 2 octets. The Authentication Transaction Sequence Number field is illustrated in Figure 7-20.

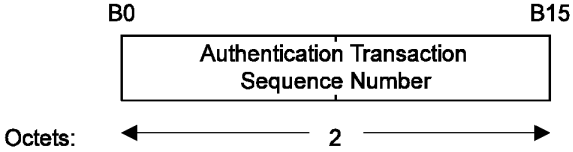



**Figure 7-20—Authentication Transaction Sequence Number field**

Copyright © 2007 IEEE. All rights reserved.

#### 7.3.1.3 Beacon Interval field

The Beacon Interval field represents the number of time units (TUs) between target beacon transmission times (TBTTs). The length of the Beacon Interval field is 2 octets. The Beacon Interval field is illustrated in Figure 7-21.

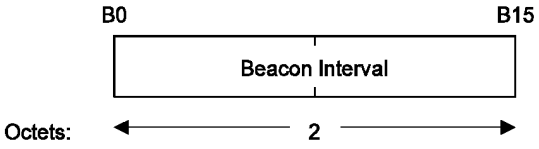



**Figure 7-21—Beacon Interval field**

#### 7.3.1.4 Capability Information field

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 format of the Capability Information field is defined in Figure 7-22. No subfield is supplied for ERP as a STA supports ERP operation if it includes all of the Clause 19 mandatory rates in its supported rate set.

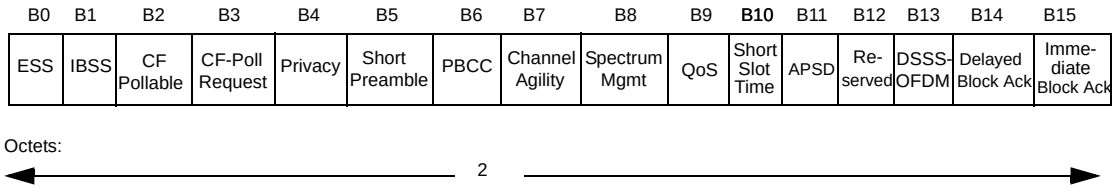



**Figure 7-22—Capability Information field**

Each Capability Information subfield is interpreted according to the management frame subtype, 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 7-20.

**Table 7-20—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 |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-20—STA usage of QoS, CF-Pollable, and CF-Poll Request *(continued)***

| QoS | CF-Pollable | CF-Poll request | Meaning |
|---|---|---|---|
| 1 | 0 | 0 | QoS STA requesting association in a QoS BSS |
| 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 management frames according to Table 7-21. A non-QoS AP 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 7-21—AP usage of QoS, CF-Pollable, and CF-Poll Request**

| QoS | CF-Pollable | CF-Poll Request | Meaning |
|---|---|---|---|
| 0 | 0 | 0 | No PC at non-QoS AP |
| 0 | 0 | 1 | PC at non-QoS AP for delivery only (no polling) |
| 0 | 1 | 0 | PC at non-QoS AP for delivery and polling |
| 0 | 1 | 1 | Reserved |
| 1 | 0 | 0 | QoS AP (HC) does not use CFP for delivery of unicast data frames |
| 1 | 0 | 1 | QoS AP (HC) uses CFP for delivery, but does not send CF-Polls to non-QoS STAs |
| 1 | 1 | 0 | QoS AP (HC) uses CFP for delivery, and sends CF-Polls to non-QoS STAs |
| 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 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.

STAs 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 frames exchanged within the IBSS. If data confidentiality is not required, STAs in an IBSS 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.

Copyright © 2007 IEEE. All rights reserved.

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 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 MMPDUs transmitted within the BSS.

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

STAs shall set the Short Preamble subfield to 1 in transmitted Association Request and Reassociation Request 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.

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 frames to indicate that the packet binary convolutional code (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.

STAs shall set the PBCC subfield to 1 in transmitted Association Request, Reassociation Request, DLS Request, and DLS Response frames when the MIB attribute dot11PBCCOption-Implemented is true. Otherwise, STAs shall set the PBCC subfield to 0.

Bit 7 of the Capabilities Information field shall be used to indicate Channel Agility capability 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 dot11Qos-OptionImplemented is true and set it to 0 otherwise.

STAs shall set the Short Slot Time subfield to 1 in transmitted Association Request, Reassociation Request, DLS Request, and DLS Response MMPDUs when the MIB attribute dot11ShortSlotTime-OptionImplemented 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 dot11ShortSlotTime-OptionImplemented 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.


Copyright © 2007 IEEE. All rights reserved.

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

APs 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.

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

STAs 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.

STAs 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.

Unused bits of the Capability Information field are reserved.

#### 7.3.1.5 Current AP Address field

The Current AP Address field is the MAC address of the AP with which the STA is currently associated. The length of the Current AP Address field is 6 octets. The Current AP Address field is illustrated in Figure 7-23.

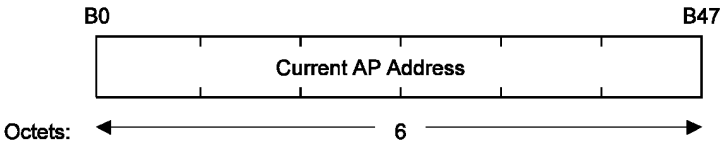



**Figure 7-23—Current AP Address field**

#### 7.3.1.6 Listen Interval field

The Listen Interval field is used to indicate to the AP how often a STA in power save mode wakes to listen to Beacon management frames. The value of this parameter is the STA's Listen Interval parameter of the MLME-ASSOCIATE.request primitive and is expressed in units of Beacon Interval. The length of the Listen Interval field is 2 octets. The Listen Interval field is illustrated in Figure 7-24.

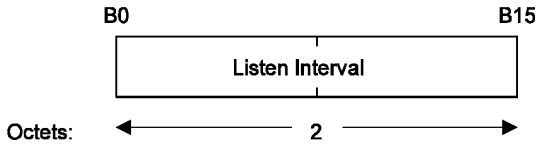



**Figure 7-24—Listen Interval field**

An AP may use the Listen Interval information in determining the lifetime of frames that it buffers for a STA.

Copyright © 2007 IEEE. All rights reserved.

### 7.3.1.7 Reason Code field

This Reason Code field is used to indicate the reason that an unsolicited notification management frame of type Disassociation, Deauthentication, DELTS, DELBA, or DLS Teardown was generated. The length of the Reason Code field is 2 octets. The Reason Code field is illustrated in Figure 7-25.

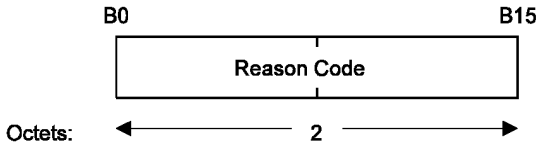



**Figure 7-25—Reason Code field**

The reason codes are defined in Table 7-22.

**Table 7-22—Reason codes**

| Reason code | Meaning |
|---|---|
| 0 | Reserved |
| 1 | Unspecified reason |
| 2 | Previous authentication no longer valid |
| 3 | Deauthenticated because sending STA is leaving (or has left) IBSS or ESS |
| 4 | Disassociated due to inactivity |
| 5 | Disassociated because AP is unable to handle all currently associated STAs |
| 6 | Class 2 frame received from nonauthenticated STA |
| 7 | Class 3 frame received from nonassociated STA |
| 8 | Disassociated because sending STA is leaving (or has left) BSS |
| 9 | STA requesting (re)association is not authenticated with responding STA |
| 10 | Disassociated because the information in the Power Capability element is unacceptable |
| 11 | Disassociated because the information in the Supported Channels element is unacceptable |
| 12 | Reserved |
| 13 | Invalid information element, i.e., an information element defined in this standard for which the content does not meet the specifications in Clause 7 |
| 14 | Message integrity code (MIC) failure |
| 15 | 4-Way Handshake timeout |
| 16 | Group Key Handshake timeout |
| 17 | Information element in 4-Way Handshake different from (Re)Association Request/Probe Response/Beacon frame |
| 18 | Invalid group cipher |
| 19 | Invalid pairwise cipher |
| 20 | Invalid AKMP |
| 21 | Unsupported RSN information element version |
| 22 | Invalid RSN information element capabilities |
| 23 | IEEE 802.1X authentication failed |
| 24 | Cipher suite rejected because of the security policy |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-22—Reason codes *(continued)***

| Reason code | Meaning |
|---|---|
| 25–31 | Reserved |
| 32 | Disassociated for unspecified, QoS-related reason |
| 33 | Disassociated because QoS AP lacks sufficient bandwidth for this QoS STA |
| 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 STA is transmitting outside the limits of its TXOPs |
| 36 | Requested from peer STA as the STA is leaving the BSS (or resetting) |
| 37 | Requested from peer STA as it does not want to use the mechanism |
| 38 | Requested from peer STA as the STA received frames using the mechanism for which a setup is required |
| 39 | Requested from peer STA due to timeout |
| 45 | Peer STA does not support the requested cipher suite |
| 46–65 535 | Reserved |

### 7.3.1.8 AID field

The AID field is a value assigned by an AP during association that represents the 16-bit ID of a STA. The length of the AID field is 2 octets. The AID field is illustrated in Figure 7-26.

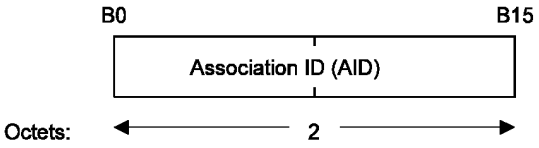



**Figure 7-26—AID field**

The value assigned as the AID is in the range 1–2007 and is placed in the 14 LSBs of the AID field, with the two MSBs of the AID field each set to 1 (see 7.1.3.2).

### 7.3.1.9 Status Code field

The Status Code field is used in a response management frame to indicate the success or failure of a requested operation. The length of the Status Code field is 2 octets. The Status Code field is illustrated in Figure 7-27.

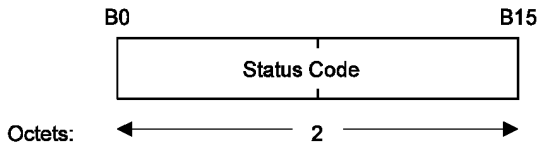



**Figure 7-27—Status Code field**

Copyright © 2007 IEEE. All rights reserved.

If an operation is successful, then the status code is set to 0. If an operation results in failure, the status code indicates a failure cause. The failure cause codes are defined in Table 7-23.

**Table 7-23—Status codes**

| Status code | Meaning |
|---|---|
| 0 | Successful |
| 1 | Unspecified failure |
| 2–9 | Reserved |
| 10 | Cannot support all requested capabilities in the Capability Information field |
| 11 | Reassociation denied due to inability to confirm that association exists |
| 12 | Association denied due to reason outside the scope of this standard |
| 13 | Responding STA does not support the specified authentication algorithm |
| 14 | Received an Authentication frame with authentication transaction sequence number out of expected sequence |
| 15 | Authentication rejected because of challenge failure |
| 16 | Authentication rejected due to timeout waiting for next frame in sequence |
| 17 | Association denied because AP is unable to handle additional associated STAs |
| 18 | Association denied due to requesting STA not supporting all of the data rates in the BSSBasicRateSet parameter |
| 19 | Association denied due to requesting STA not supporting the short preamble option |
| 20 | Association denied due to requesting STA not supporting the PBCC modulation option |
| 21 | Association denied due to requesting STA not supporting the Channel Agility option |
| 22 | Association request rejected because Spectrum Management capability is required |
| 23 | Association request rejected because the information in the Power Capability element is unacceptable |
| 24 | Association request rejected because the information in the Supported Channels element is unacceptable |
| 25 | Association denied due to requesting STA not supporting the Short Slot Time option |
| 26 | Association denied due to requesting STA not supporting the DSSS-OFDM option |
| 27–31 | Reserved |
| 32 | Unspecified, QoS-related failure |
| 33 | Association denied because QoS AP has insufficient bandwidth to handle another QoS STA |
| 34 | Association denied due to excessive frame loss rates and/or poor conditions on current operating channel |
| 35 | Association (with QoS BSS) denied because the requesting STA does not support the QoS facility |
| 36 | Reserved |
| 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 STA may attempt to set another TS with the suggested changes to the TSPEC |
| 40 | Invalid information element, i.e., an information element defined in this standard for which the content does not meet the specifications in Clause 7 |
| 41 | Invalid group cipher |
| 42 | Invalid pairwise cipher |
| 43 | Invalid AKMP |
| 44 | Unsupported RSN information element version |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-23—Status codes *(continued)***

| Status code | Meaning |
|---|---|
| 45 | Invalid RSN information element capabilities |
| 46 | Cipher suite rejected because of security policy |
| 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 BSS |
| 50 | The Destination STA is not a QoS STA |
| 51 | Association denied because the ListenInterval is too large |
| 52–65 535 | Reserved |

### 7.3.1.10 Timestamp field

This field represents the value of the timing synchronization function (TSF) timer (see 11.1) of a frame's source. The length of the Timestamp field is 8 octets. The Timestamp field is illustrated in Figure 7-28.

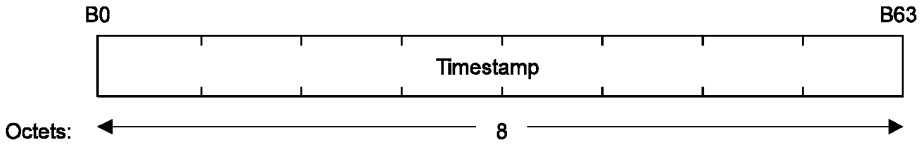



**Figure 7-28—Timestamp field**

### 7.3.1.11 Action field

The Action field provides a mechanism for specifying extended management actions. The format of the Action field is shown in Figure 7-29.

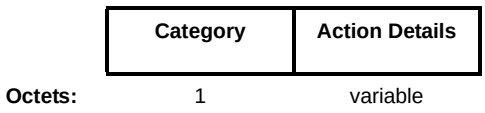



**Figure 7-29—Action field**

The Category field is set to one of the nonreserved values shown in Table 7-24. 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 to 0, then the STA shall return the Action frame to the source without change except that the MSB of the Category field is set to 1.

The Action Details field contains the details of the action. The details of the actions allowed in each category are described in the appropriate subclause referenced in Table 7-24.

Copyright © 2007 IEEE. All rights reserved.

**Table 7-24—Category values**

| Code | 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 |
| 4–126 | Reserved | — |
| 127 | Vendor-specific | 7.4.5 |
| 128–255 | Error | — |

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

### 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 7-30. 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.

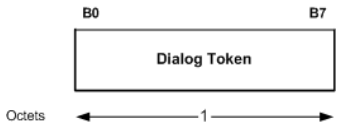



**Figure 7-30—Dialog Token fixed field**

### 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 7-31.

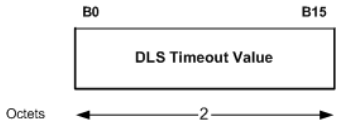



**Figure 7-31—DLS Timeout Value fixed field**

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.


Copyright © 2007 IEEE. All rights reserved.

### 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 7-32.

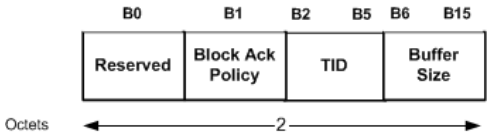



**Figure 7-32—Block Ack Parameter Set fixed field**

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.[18]

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 7-33.

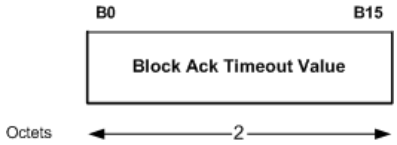



**Figure 7-33—Block Ack Timeout Value fixed field**

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.

[18]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.

Copyright © 2007 IEEE. All rights reserved.

### 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 7-34.

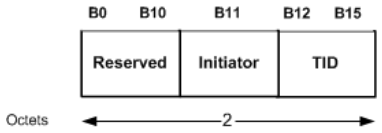



**Figure 7-34—DELBA Parameters fixed field**

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.

### 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 STA is an AP or a non-AP STA.

The format of the QoS Info field, when sent by the AP, is defined in Figure 7-35.

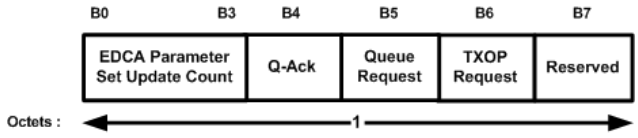



**Figure 7-35—QoS Info field when sent by an AP**

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

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

APs 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.

APs 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 STA, is defined in Figure 7-36.

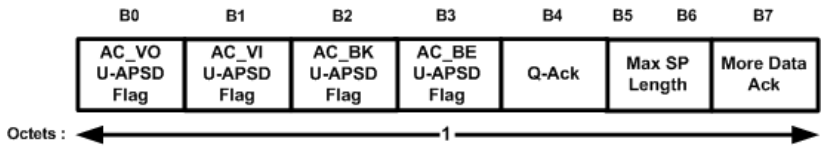



**Figure 7-36—QoS Info field when set by a non-AP STA**


Copyright © 2007 IEEE. All rights reserved.

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 STAs 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 AP may deliver to a non-AP STA during any SP triggered by the non-AP STA. 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 7-25.

**Table 7-25—Settings of the Max SP Length subfield**

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

Non-AP STAs 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 STAs set the More Data Ack subfield to 0 otherwise. For APs, the More Data Ack subfield is reserved.

### 7.3.2 Information elements

Elements are defined to have a common general format consisting of a 1 octet Element ID field, a 1 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined in this standard. The Length field specifies the number of octets in the Information field. See Figure 7-37.

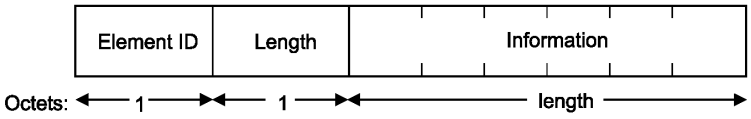



**Figure 7-37—Element format**

Copyright © 2007 IEEE. All rights reserved.

The set of valid elements is defined in Table 7-26.

**Table 7-26—Element IDs**

| Information element | Element ID | Length (in octets) |
|---|---|---|
| SSID (see 7.3.2.1) | 0 | 2 to 34 |
| Supported rates (see 7.3.2.2) | 1 | 3 to 10 |
| FH Parameter Set (see 7.3.2.3) | 2 | 7 |
| DS Parameter Set (see 7.3.2.4) | 3 | 3 |
| CF Parameter Set (see 7.3.2.5) | 4 | 8 |
| TIM (see 7.3.2.6) | 5 | 6 to 256 |
| IBSS Parameter Set (see 7.3.2.7) | 6 | 4 |
| Country (see 7.3.2.9) | 7 | 8 to 256 |
| Hopping Pattern Parameters (see 7.3.2.10) | 8 | 4 |
| Hopping Pattern Table (see 7.3.2.11) | 9 | 6 to 256 |
| Request (see 7.3.2.12) | 10 | 2 to 256 |
| BSS Load (see 7.3.2.28) | 11 | 7 |
| EDCA Parameter Set (see 7.3.2.29) | 12 | 20 |
| TSPEC (see 7.3.2.30) | 13 | 57 |
| TCLAS (see 7.3.2.31) | 14 | 2 to 257 |
| Schedule (see 7.3.2.34) | 15 | 16 |
| Challenge text (see 7.3.2.8) | 16 | 3 to 255 |
| Reserved | 17–31 | |
| Power Constraint (see 7.3.2.15) | 32 | 3 |
| Power Capability (see 7.3.2.16) | 33 | 4 |
| TPC Request (see 7.3.2.17) | 34 | 2 |
| TPC Report (see 7.3.2.18) | 35 | 4 |
| Supported Channels (see 7.3.2.19) | 36 | 4 to 256 |
| Channel Switch Announcement (see 7.3.2.20) | 37 | 5 |
| Measurement Request (see 7.3.2.21) | 38 | 5 to 16 |
| Measurement Report (see 7.3.2.22) | 39 | 5 to 24 |
| Quiet (see 7.3.2.23) | 40 | 8 |
| IBSS DFS (see 7.3.2.24) | 41 | 10 to 255 |
| ERP Information (see 7.3.2.13) | 42 | 3 |
| TS Delay (see 7.3.2.32) | 43 | 6 |
| TCLAS Processing (see 7.3.2.33) | 44 | 3 |
| Reserved | 45 | |
| QoS Capability (see 7.3.2.35) | 46 | 3 |
| Reserved | 47 | |
| RSN (see 7.3.2.25) | 48 | 36 to 256 |
| Reserved | 49 | |
| Extended Supported Rates (see 7.3.2.14) | 50 | 3 to 257 |
| Reserved | 51–126 | |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-26—Element IDs *(continued)***

| Information element | Element ID | Length (in octets) |
|---|---|---|
| Extended Capabilities | 127 | 2 to 257 |
| Reserved | 128–220 | |
| Vendor Specific (see 7.3.2.26) | 221 | 3 to 257 |
| Reserved | 222–255 | |

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.

### 7.3.2.1 SSID element

The SSID element indicates the identity of an ESS or IBSS. See Figure 7-38.

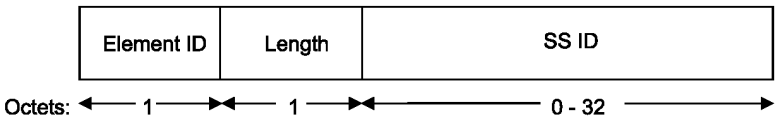



**Figure 7-38—SSID element format**

The length of the SSID information field is between 0 and 32 octets. A 0 length information field is used within Probe Request management frames to indicate the wildcard SSID.

### 7.3.2.2 Supported Rates element

The Supported Rates element specifies up to eight rates in the Operational-Rate-Set parameter, as described in the MLME-JOIN.request and MLME-START.request primitives. The information field is encoded as 1 to 8 octets, where each octet describes a single Supported Rate. If the number of rates in the Operational Rate Set exceeds eight, then an Extended Supported Rate element shall be generated to specify the remaining supported rates. The use of the Extended Supported Rates element is optional otherwise.

Within Beacon, Probe Response, Association Response, and Reassociation Response management frames, each Supported Rate contained in the BSSBasicRateSet parameter is encoded as an octet with the MSB (bit 7) set to 1, and bits 6 through 0 are set to the data rate, if necessary rounded up to the next 500kb/s, in units of 500 kb/s. For example, a 2.25 Mb/s rate contained in the BSSBasicRateSet parameter is encoded as X'85'. Rates not contained in the BSSBasicRateSet parameter are encoded with the MSB set to 0, and bits 6 through 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 10.4.4.2 (e.g., a 2 Mb/s rate not contained in the BSSBasicRateSet parameter is encoded as X'04'). The MSB of each Supported Rate octet in other management frame types is ignored by receiving STAs.

The Supported Rate information in Beacon and Probe Response management frames is delivered to the management entity in a STA via the BSSBasicRateSet parameter in the MLME-SCAN.confirm primitive. It is used by the management entity in a STA to avoid associating with a BSS if the STA cannot receive and transmit all the data rates in the BSSBasicRateSet parameter (see Figure 7-39).

Copyright © 2007 IEEE. All rights reserved.

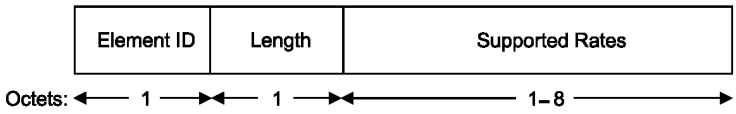



**Figure 7-39—Supported rates element format**

If the DSSS-OFDM bit is set to 1 in the transmitted Capability Information field of an MMPDU, then any supported rates transmitted in that frame that include rates that are common to both DSSS-OFDM and ERP-OFDM shall be interpreted by receiving and transmitting STA to indicate support for both DSSS-OFDM and ERP-OFDM at the indicated rate. However, if any of those rates are indicated as basic (a rate in the BSSBasicRateSet parameter), then the basic rate designation shall be interpreted by receiving and transmitting STA to apply only for the ERP-OFDM modulation and rate. If the PBCC bit is set to 1 in the transmitted capability field of an MMPDU, then any supported rates transmitted in that frame that include rates that are common to both PBCC and CCK shall be interpreted by receiving and transmitting STA to indicate support for both PBCC and CCK at the indicated rate. However, if any of those rates are indicated as basic, then the basic rate designation shall be interpreted by receiving and transmitting STA to apply only for the CCK modulation and rate. That is, if the rate is indicated as basic, the basic designation does not apply to DSSS-OFDM, PBCC, or ERP-PBCC.

### 7.3.2.3 FH Parameter Set element

The FH Parameter Set element contains the set of parameters necessary to allow synchronization for STAs using an FH PHY. The information field contains Dwell Time, Hop Set, Hop Pattern, and Hop Index parameters. The total length of the information field is 5 octets. See Figure 7-40.

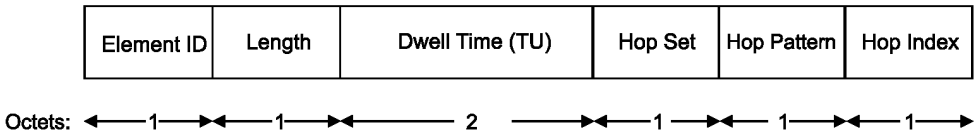



**Figure 7-40—FH Parameter Set element format**

The Dwell Time field is 2 octets in length and contains the dwell time in TU.

The Hop Set field identifies the current set (dot11CurrentSet) of hop patterns and is a single octet.

The Hop Pattern field identifies the current pattern (dot11CurrentPattern) within a set of hop patterns and is a single octet.

The Hop Index field selects the current index (dot11CurrentIndex) within a pattern and is a single octet.

The description of the attributes used in this subclause can be found in 14.8.2.

### 7.3.2.4 DS Parameter Set element

The DS Parameter Set element contains information to allow channel number identification for STAs using a DSSS PHY. The information field contains a single parameter containing the dot11CurrentChannel-Number (see 15.4.6.2 for values). The length of the dot11CurrentChannelNumber parameter is 1 octet. See Figure 7-41.


Copyright © 2007 IEEE. All rights reserved.

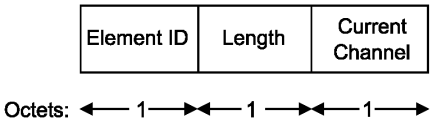



**Figure 7-41—DS Parameter Set element format**

### 7.3.2.5 CF Parameter Set element

The CF Parameter Set element contains the set of parameters necessary to support the PCF. The information field contains the CFPCount, CFPPeriod, CFPMaxDuration, and CFPDurRemaining fields. The total length of the information field is 6 octets. See Figure 7-42.

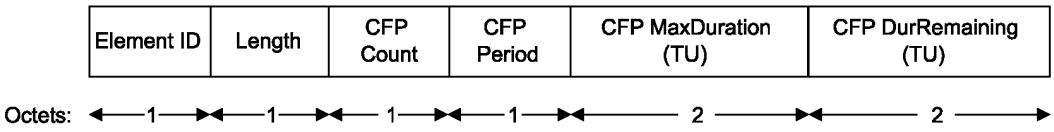



**Figure 7-42—CF Parameter Set element format**

CFPCount indicates how many delivery traffic indication messages (DTIMs) (including the current frame) appear before the next CFP start. A CFPCount of 0 indicates that the current DTIM marks the start of the CFP.

CFPPeriod indicates the number of DTIM intervals between the start of CFPs. The value is an integral number of DTIM intervals.

CFPMaxDuration indicates the maximum duration, in TU, of the CFP that may be generated by this PCF. This value is used by STAs to set their NAV at the TBTT of Beacon frames that begin CFPs.

CFPDurRemaining indicates the maximum time, in TU, remaining in the present CFP, and is set to 0 in CFP Parameter elements of Beacon frames transmitted during the CP. The value of CFPDurRemaining is referenced to the immediately previous TBTT. This value is used by all STAs to update their NAVs during CFPs.

### 7.3.2.6 TIM

The TIM element contains four fields: DTIM Count, DTIM Period, Bitmap Control, and Partial Virtual Bitmap. See Figure 7-43.

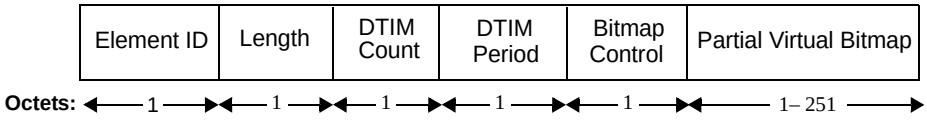



**Figure 7-43—TIM element format**

The Length field for this element indicates the length of the information field, which is constrained as described below.

The DTIM Count field indicates how many Beacon frames (including the current frame) appear before the next DTIM. A DTIM Count of 0 indicates that the current TIM is a DTIM. The DTIM count field is a single octet.

Copyright © 2007 IEEE. All rights reserved.

The DTIM Period field indicates the number of beacon intervals between successive DTIMs. If all TIMs are DTIMs, the DTIM Period field has the value 1. The DTIM Period value 0 is reserved. The DTIM period field is a single octet.

The Bitmap Control field is a single octet. Bit 0 of the field contains the Traffic Indicator bit associated with Association ID 0. This bit is set to 1 in TIM elements with a value of 0 in the DTIM Count field when one or more broadcast or multicast frames are buffered at the AP. The remaining 7 bits of the field form the Bitmap Offset.

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

The Partial Virtual Bitmap field consists of octets numbered $N1$ through $N2$ of the traffic indication virtual bitmap, where $N1$ is the largest even number such that bits numbered 1 through $(N1 \times 8) - 1$ in the bitmap are all 0 and $N2$ is the smallest number such that bits numbered $(N2 + 1) \times 8$ through 2007 in the bitmap are all 0. In this case, the Bitmap Offset subfield value contains the number $\lfloor N1/2 \rfloor$, and the Length field is set to $(N2 - N1) + 4$.

In the event that all bits other than bit 0 in the virtual bitmap are 0, the Partial Virtual Bitmap field is encoded as a single octet equal to 0, the Bitmap Offset subfield is 0, and the Length field is 4.

#### 7.3.2.7 IBSS Parameter Set element

The IBSS Parameter Set element contains the set of parameters necessary to support an IBSS. The information field contains the ATIM Window parameter. See Figure 7-44.

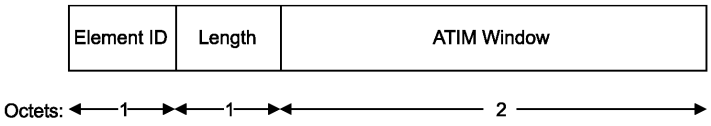



**Figure 7-44—IBSS Parameter Set element format**

The ATIM Window field is 2 octets in length and contains the ATIM Window length in TU.


Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.8 Challenge Text element

The Challenge Text element contains the challenge text within Authentication exchanges. The element information field length is dependent upon the authentication algorithm and the transaction sequence number as specified in 8.2.2.2. See Figure 7-45.

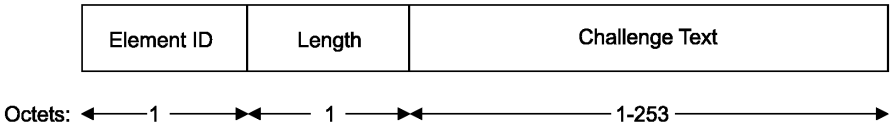



**Figure 7-45—Challenge Text element format**

### 7.3.2.9 Country information element

The Country information element contains the information required to allow a STA to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. The format of this information element shall be as shown in Figure 7-46.

| | Element ID | Length |
|---|---|---|
| Country String (Octet 1) | Country String (Octet 2) | Country String (Octet 3) |
| First Channel Number/ Regulatory Extension Identifier | Number of Channels/ Regulatory Class | Maximum Transmit Power Level/ Coverage Class |
| . . . | . . . | . . . |
| Pad (if needed) | | |

**Figure 7-46—Country information element format**

The element ID for this information element shall be 7. The length of the information element is variable, as the element may contain more than one triplet comprising the First Channel Number, Number of Channels, and Maximum Transmit Power Level fields and referred to as subband triplets. Alternatively, where dot11RegulatoryClassesRequired is true and the First Channel Number/Regulatory Extension Identifier octet has a positive integer value of 201 or greater, then that triplet comprises the Regulatory Extension Identifier, Regulatory Class, and Coverage Class fields. Together they are referred to as a regulatory triplet. The minimum length of the information element is 8 octets.

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

The First Channel Number/Regulatory Extension Identifier field shall be 1 octet in length. If the field has a positive integer value less than 201, then it shall contain a positive integer value that indicates the lowest channel number in the subband described in this information element. The group of channels described by each pair of the First Channel Number and the Number of Channels fields shall not have overlapping channel identifiers. [For example, the pairs (2,4) and (5,2) overlap and shall not be used together.] The First Channel Numbers shall be monotonically increasing where dot11RegulatoryClassesRequired is not true.

Where dot11RegulatoryClassesRequired is true, consecutive subband triplets following a regulatory triplet shall have monotonically increasing First Channel Number fields.

Copyright © 2007 IEEE. All rights reserved.

The Number of Channels field of the subelement shall be 1 octet in length.

The Maximum Transmit Power Level field is a signed number and shall be 1 octet in length. It shall indicate the maximum power, in dBm, allowed to be transmitted. As the method of measurement for maximum transmit power level differs by regulatory domain, the value in this field shall be interpreted according to the regulations applicable for the domain identified by the Country String.

A regulatory class is an index into a set of values for radio equipment sets of rules. The Regulatory Class field shall be 1 octet in length.

A coverage class is an index into a set of values for aAirPropagationTime. The Coverage Class field shall be 1 octet in length.

The Coverage Class field of the regulatory triplet specifies the aAirPropagationTime characteristic used in BSS operation, as shown in Table 7-27. The characteristic aAirPropagationTime describes variations in actual propagation time that are accounted for in a BSS and, together with maximum transmit power level, allow control of BSS diameter.

**Table 7-27—Coverage Class field parameters**

| Coverage class value | aAirPropagationTime (μs) |
|---|---|
| 0 | ≤ 1 |
| 1 | 3 |
| 2 | 6 |
| 3 | 9 |
| 4 | 12 |
| 5 | 15 |
| 6 | 18 |
| 7 | 21 |
| 8 | 24 |
| 9 | 27 |
| 10 | 30 |
| 11 | 33 |
| 12 | 36 |
| 13 | 39 |
| 14 | 42 |
| 15 | 45 |
| 16 | 48 |
| 17 | 51 |
| 18 | 54 |
| 19 | 57 |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-27—Coverage Class field parameters *(continued)***

| Coverage class value | aAirPropagationTime (μs) |
|---|---|
| 20 | 60 |
| 21 | 63 |
| 22 | 66 |
| 23 | 69 |
| 24 | 72 |
| 25 | 75 |
| 26 | 78 |
| 27 | 81 |
| 28 | 84 |
| 29 | 87 |
| 30 | 90 |
| 31 | 93 |
| 32–255 | — |

The Pad field is 0 or 1 octet in length. The length of the Country information element shall be evenly divisible by 2. The Pad shall be used to add a single octet to the element if the length is not evenly divisible by 2. The value of the Pad field shall be 0.

### 7.3.2.10 Hopping Pattern Parameters information element

The Hopping Pattern Parameters information element contains the information necessary to allow a STA to calculate the code family using the hyperbolic congruence code (HCC) and extended HCC (EHCC) algorithms. See 9.8.2.1 for a description of the HCC and EHCC algorithms. The format of this information element shall be as shown in Figure 7-47.




**Figure 7-47—Hopping Pattern Parameters information element**

The Element ID of this information element shall be 8. The length of this element is 4 octets.

The Prime Radix field of this element shall indicate the value to be used as the prime radix (*N*) in the HCC and EHCC algorithms. The value of this field shall be a positive integer. The size of this field is 1 octet.

The Number of Channels field of this element shall indicate the value to be used as the maximum for the family index (*a*) in the HCC and EHCC algorithms. The value of this field shall be a positive integer and shall not be less than the prime radix minus 3 (*N*–3). The size of this field is 1 octet.

Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.11 Hopping Pattern Table information element

The Hopping Pattern Table information element contains the information necessary for an FH implementation to be able to create the hopping sequences necessary to operate in the regulatory domain in which the information element was received. The format of the information element shall be as shown in Figure 7-48.

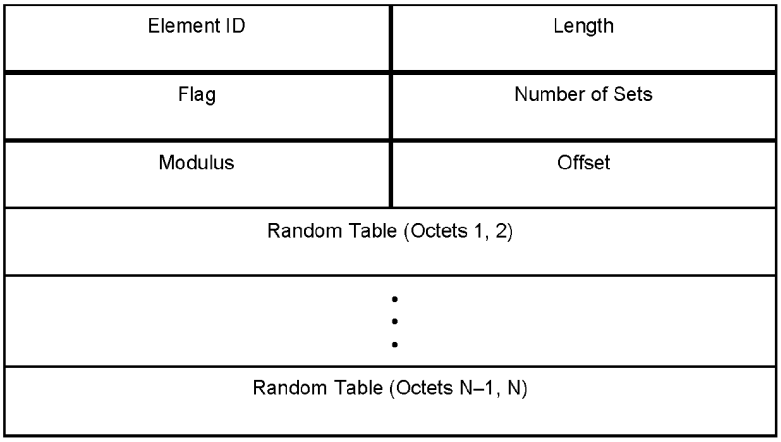



**Figure 7-48—Hopping Pattern Table information element**

The Element ID of this information element shall be 9. The information element is variable in length. The length of the information element is indicated by the Length field.

The Flag field indicates that a Random Table is present when the value is 1. When the flag value is 0, it indicates that a Random Table is not present and that the hop index method is to be used to determine the hopping sequence. The size of this field is 1 octet.

The Number of Sets field indicates the total number of sets within the hopping patterns. The size of this field is 1 octet.

The Modulus and Offset fields indicate the values to be used in the equations to create a hopping sequence from the Random Table information. The size of these fields are each 1 octet.

The Random Table field is a variable length field. It is a vector of single octet values that indicate the random sequence to be followed during a hopping sequence. The size of the Random Table field is found by subtracting 4 from the value of the Length field of this element.

Two equations are used to create a hopping sequence from the information in the Frequency Hopping information element and the Hopping Pattern Table information element. Equation (7-1), the Random Table Method, shall be used when the value of the Flag field of the Hopping Pattern Table information element is 1. Equation (7-2), the Hop Index Method, shall be used when the value of the Flag field of the Hopping Pattern Table information element is 0.

$$f_x(i) = [b(i) + x] \bmod (m) + q \quad (7\text{-}1)$$

$$f_x(i) = [(i-1) \times x] \bmod (m) + q \quad (7\text{-}2)$$

$$x = n \times p + s - 1 \quad (7\text{-}3)$$


Copyright © 2007 IEEE. All rights reserved.

where

*i* is the index.
*m* is the modulus.
*q* is the offset.
*n* is the number of sets.
*p* is the current pattern.
*s* is the current set number.

The values of *i*, *p*, and *s* are found in the Frequency Hopping information element. The values of *m*, *n*, and *q* are found in the Hopping Pattern Table information element.

#### 7.3.2.12 Request information element

This element is placed in a Probe Request frame to request that the responding STA include the requested information in the Probe Response frame. The format of the information element shall be as shown in Figure 7-49.

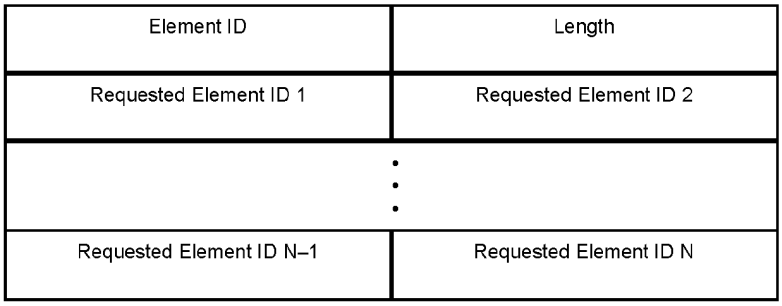



**Figure 7-49—Request information element**

The Element ID of this information element shall be 10. The information element is variable in length. The length of the information element is indicated in the Length field.

The Requested Element IDs are the list of elements that are to be included in the responding STA's Probe Response frame. The Requested Element IDs shall be listed in order of increasing element ID.

A STA shall return only those information elements that it supports. In an improperly formed Request information element, a STA may ignore the first information element requested that is not ordered properly and all subsequent information elements requested. In the probe response frame, the STA shall return the requested information elements in the same order requested in the Request information element of the probe request frame.

#### 7.3.2.13 ERP Information element

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

If one or more NonERP STAs are associated in the BSS, the Use_Protection bit shall be set to 1 in transmitted ERP Information elements.

Copyright © 2007 IEEE. All rights reserved.

In an IBSS, the setting of the Use_Protection bit is left to the STA. In an IBSS, there is no uniform concept of association; therefore, a typical algorithm for setting the Use_Protection bit will take into account the traffic pattern and history on the network. If a member of an IBSS detects one or more NonERP STAs that are members of the same IBSS or receives a Beacon from a member of the same IBSS with the Use_Protection bit set to 1, then the Use_Protection bit should be set to 1 in the ERP Information Element of transmitted Beacon and Probe Response frames.

The NonERP_Present bit shall be set to 1 when a NonERP STA is associated with the BSS. Examples of when the NonERP present bit may additionally be set to 1 include, but are not limited to, when

a) A NonERP infrastructure or independent BSS is overlapping (a NonERP BSS may be detected by the reception of a Beacon where the supported rates contain only Clause 15 or Clause 18 rates).

b) In an IBSS, if a Beacon frame is received from one of the IBSS participants where the supported rate set contains only Clause 15 or Clause 18 rates.

c) A management frame (excluding a Probe Request) is received where the supported rate set includes only Clause 15 or Clause 18 rates.

ERP APs and ERP STAs shall invoke the use of a protection mechanism after transmission or reception of the Use_Protection bit with a value of 1 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started. ERP APs and ERP STAs may additionally invoke protection mechanism use at other times. ERP APs and ERP STAs may disable protection mechanism use after transmission or reception of the Use_Protection bit with a value of 0 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started.

When there are no NonERP STAs associated with the BSS and the ERP Information Element sender's dot11ShortPreambleOptionImplemented MIB variable is set to true, then the Barker_Preamble_Mode bit may be set to 0. The Barker_Preamble_Mode bit shall be set to 1 by the ERP Information Element sender if one or more associated NonERP STAs are not short preamble capable as indicated in their Capability Information field, or if the ERP Information Element senders dot11ShortPreambleOptionImplemented MIB variable is set to false.

If a member of an IBSS detects one or more nonshort-preamble-capable STAs that are members of the same IBSS, then the Barker_Preamble_Mode bit should be set to 1 in the transmitted ERP Information Element.

ERP APs and ERP STAs shall use long preambles when transmitting Clause 15, Clause 18, and Clause 19 frames after transmission or reception of an ERP Information Element with a Barker_Preamble_Mode value of 1 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started, regardless of the value of the short preamble capability bit from the same received or transmitted MMPDU that contained the ERP Information Element. ERP APs and ERP STAs may additionally use long preambles when transmitting Clause 15, Clause 18, and Clause 19 frames at other times. ERP APs and ERP STAs may use short preambles when transmitting Clause 15, Clause 18, and Clause 19 frames after transmission or reception of an ERP Information Element with a Barker_Preamble_Mode value of 0 in an MMPDU to or from the BSS that the ERP AP or ERP STA has joined or started, regardless of the value of the short preamble capability bit from the same received or transmitted MMPDU. NonERP STAs and NonERP APs may also follow the rules given in this paragraph.

Recommended behavior for setting the Use_Protection bit is contained in 9.13.


Copyright © 2007 IEEE. All rights reserved.

The ERP Information element shall have the form shown in Figure 7-50.

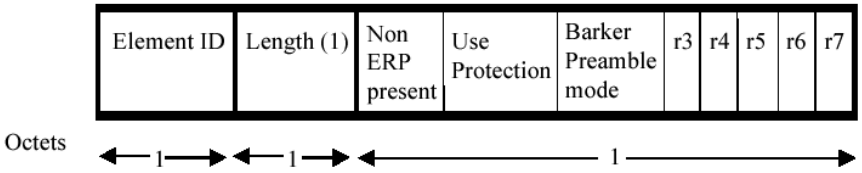



**Figure 7-50—ERP Information element**

Bits r3 through r7 are reserved, set to 0, and ignored on reception. Note that the length of this element is flexible and may be expanded in the future.

### 7.3.2.14 Extended Supported Rates element

The Extended Supported Rates element specifies the rates in the OperationalRateSet as described in the MLME_JOIN.request and MLME_START.request primitives that are not carried in the Supported Rates element. The information field is encoded as 1 to 255 octets where each octet describes a single supported rate.

Within Beacon, Probe Response, Association Response, and Reassociation Response management frames, each supported rate contained in the BSSBasicRateSet parameter, as defined in 10.3.10.1, is encoded as an octet with the MSB (bit 7) set to 1 and bits 6 through 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 10.4.4.2 (e.g., a 1 Mb/s rate contained in the BSSBasicRateSet parameter is encoded as X'82'). Rates not contained in the BSSBasicRateSet parameter are encoded with the MSB set to 0, and bits 6 through 0 are set to the appropriate value from the valid range column of the DATA_RATE row of the table in 10.4.4.2 (e.g., a 2 Mb/s rate not contained in the BSSBasicRateSet parameter is encoded as X'04'). The MSB of each octet in the Extended Supported Rate element in other management frame types is ignored by receiving STAs.

Extended Supported Rate information in Beacon and Probe Response management frames is used by STAs in order to avoid associating with a BSS if they do not support all the data rates in the BSSBasicRateSet parameter.

For STAs supporting eight or fewer data rates, this element is optional for inclusion in all of the frame types that include the supported rates element. For STAs supporting more than eight data rates, this element shall be included in all of the frame types that include the supported rates element.

The Extended Supported Rates element has the format shown in Figure 7-51.

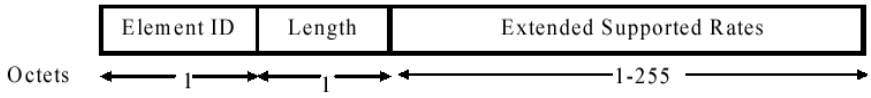



**Figure 7-51—Extended Supported Rates element format**

Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.15 Power Constraint element

The Power Constraint element contains the information necessary to allow a STA to determine the local maximum transmit power in the current channel. The format of the Power Constraint element is shown in Figure 7-52.

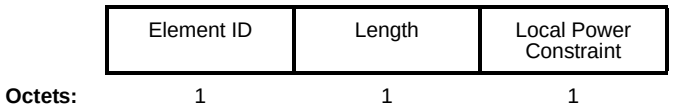



**Figure 7-52—Power Constraint element format**

The Length field shall be set to 1.

The Local Power Constraint field shall be set to a value that allows the mitigation requirements to be satisfied in the current channel. The field is coded as an unsigned integer in units of decibels relative to 1mW. The local maximum transmit power for a channel is thus defined as the maximum transmit power level specified for the channel in the Country element minus the local power constraint specified for the channel (from the MIB) in the Power Constraint element.

The Power Constraint element is included in Beacon frames, as described in 7.2.3.1, and Probe Response frames, as described in 7.2.3.9. The use of Power Constraint elements is described in 11.8.2.

### 7.3.2.16 Power Capability element

The Power Capability element specifies the minimum and maximum transmit powers with which a STA is capable of transmitting in the current channel. The format of the Power Capability element is shown in Figure 7-53.

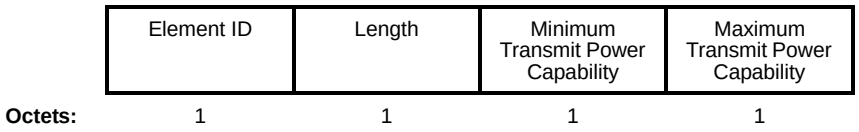



**Figure 7-53—Power Capability element format**

The Length field shall be set to 2.

The Minimum Transmit Power Capability field shall be set to the nominal minimum transmit power with which the STA is capable of transmitting in the current channel, with a tolerance ± 5 dB. The field is coded as a signed integer in units of decibels relative to 1 mW.

The Maximum Transmit Power Capability field shall be set to the nominal maximum transmit power with which the STA is capable of transmitting in the current channel, with a tolerance ± 5 dB. The field is coded as a signed integer in units of decibels relative to 1 mW.

The Power Capability element is included in Association Request frames, as described in 7.2.3.4, and Reassociation Request frames, as described in 7.2.3.6. The use of Power Capability elements is described in 11.8.1.


Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.17 TPC Request element

The TPC Request element contains a request for a STA to report transmit power and link margin information using a TPC Report element. The format of the TPC Request element is shown in Figure 7-54.

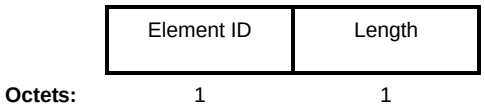



**Figure 7-54—TPC Request element format**

The Length field shall be set to 0.

The TPC Request element is included in TPC Request frames, as described in 7.4.1.3. The use of TPC Request elements and frames is described in 11.8.4.

### 7.3.2.18 TPC Report element

The TPC Report element contains transmit power and link margin information sent in response to a TPC Request element. A TPC Report element is included in a Beacon frame or Probe Response frame without a corresponding request. The format of the TPC Report element is shown in Figure 7-55.

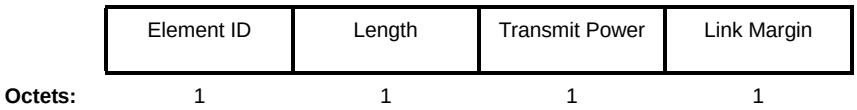



**Figure 7-55—TPC Report element format**

The Length field shall be set to 2.

The Transmit Power field shall be set to the transmit power used to transmit the frame containing the TPC Report element. The field is coded as a signed integer in units of decibels relative to 1 mW. The maximum tolerance for the transmit power value reported in the TPC Response element shall be ± 5 dB. This tolerance is defined as the difference, in decibels, between the reported power value and the actual EIRP of the STA (measured when transmitting 1500 octet frames).

The Link Margin field contains the link margin at the time and for the rate at which the frame containing the TPC Request element was received. The field is coded as a signed integer in units of decibels. The Link Margin field shall be set to 0 and shall be ignored when a TPC Report element is included in a Beacon frame or Probe Response frame. The measurement method of Link Margin is beyond the scope of this standard.

The TPC Report element is included in TPC Report frames, as described in 7.4.1.4; Beacon frames, as described in 7.2.3.1; and Probe Response frames, as described in 7.2.3.9. The use of TPC Report elements and frames is described in 11.8.4.

Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.19 Supported Channels element

The Supported Channels element contains a list of channel subbands (from those channels defined in 17.3.8.3.3) in which a STA is capable of operating. The format of the Supported Channels element is shown in Figure 7-56.

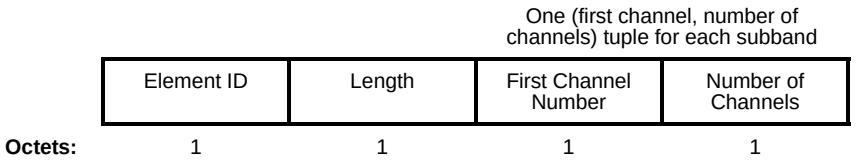



**Figure 7-56—Supported Channels element format**

The Length field is variable and depends on the number of subbands, defined by a First Channel Number–Number of Channels pair, that are included in the element.

The First Channel Number field shall be set to the first channel (as defined in 17.3.8.3.3) in a subband of supported channels.

The Number of Channels field shall be set to the number of channels in a subband of supported channels.

The Supported Channels element is included in Association Request frames, as described in 7.2.3.4, and Reassociation Request frames, as described in 7.2.3.6. The use of the Supported Channels element is described in 11.9.1 and 11.9.7.

### 7.3.2.20 Channel Switch Announcement element

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

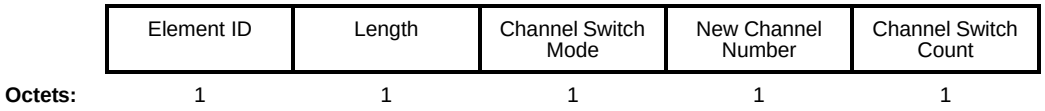



**Figure 7-57—Channel Switch Announcement element format**

The Length field shall be set to 3.

The Channel Switch Mode field indicates any restrictions on transmission until a channel switch. An AP in a BSS or a STA in an IBSS shall set the Channel Switch Mode field to either 0 or 1 on transmission. A Channel Switch Mode set to 1 means that the STA in a BSS to which the frame containing the element is addressed shall transmit no further frames within the BSS until the scheduled channel switch. A STA in an IBSS may treat a Channel Switch Mode field set to 1 as advisory. A Channel Switch Mode set to 0 does not impose any requirement on the receiving STA.

The New Channel Number field shall be set to the number of the channel to which the STA is moving (as defined in 17.3.8.3.3).


Copyright © 2007 IEEE. All rights reserved.

The Channel Switch Count field either shall be set to the number of TBTTs until the STA sending the Channel Switch Announcement element switches to the new channel or shall be set to 0. A value of 1 indicates that the switch shall occur immediately before the next TBTT. A value of 0 indicates that the switch shall occur at any time after the frame containing the element is transmitted.

The Channel Switch Announcement element is included in Channel Switch Announcement frames, as described in 7.4.1.5, and may be included in Beacon frames, as described in 7.2.3.1, and Probe Response frames, as described in 7.2.3.9. The use of Channel Switch Announcement elements and frames is described in 11.9.7.

### 7.3.2.21 Measurement Request element

The Measurement Request element contains a request that the receiving STA undertake the specified measurement action. The format of the Measurement Request element is shown in Figure 7-58.

| | Element ID | Length | Measurement Token | Measurement Request Mode (see Figure 7-59) | Measurement Type | Measurement Request |
|---|---|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | 1 | 1 | variable |

**Figure 7-58—Measurement Request element format**

| | Reserved (0) | Enable | Request | Report | Reserved (0) |
|---|---|---|---|---|---|
| **Bit:** | 0 | 1 | 2 | 3 | 4–7 |

**Figure 7-59—Measurement Request Mode field**

The Length field is variable and depends on the length of the Measurement Request field. The minimum value of the Length field is 3 (based on a minimum length for the Measurement Request field of 0 octets).

The Measurement Token shall be set to a nonzero number that is unique among the Measurement Request elements in a particular Measurement Request frame.

The Measurement Request Mode field (shown in Figure 7-59) is a bit field with the following bits defined:

— Enable bit (bit 1) indicates whether this element is used to request the destination STA to enable or disable the sending of measurement requests and autonomous measurement reports of a specified type to this STA. The Enable bit shall be set to 1 when the Request bit and Report bit are valid. The Enable bit shall be set to 0 when the Request bit and Report bit are invalid.

— Request bit (bit 2) indicates whether the STA receiving the request shall enable or disable measurement requests of the type specified in the Measurement Type field. The Request bit shall be set to 1 when enabling a measurement request. The Request bit shall be set to 0 when disabling a measurement request or when the Request bit is invalid (i.e., when the Enable bit is set to 0 or when the Measurement Type field contains a reserved measurement request type value).

— Report bit (bit 3) indicates whether the STA receiving the request shall enable or disable autonomous measurement reports of the type corresponding to the measurement report specified in the Measurement Type field. The Report bit shall be set to 1 when enabling an autonomous measurement report. The Report bit shall be set to 0 when disabling an autonomous measurement

Copyright © 2007 IEEE. All rights reserved.

report or when the Report bit is invalid (i.e., when the Enable bit is set to 0 or when the Measurement Type field contains a reserved measurement report type value).

— All other bits are reserved and shall be set to 0.

The use of the Enable, Request, and Report bits is also summarized in Table 7-28. See 11.9.6 for the description of how a STA shall handle requests to enable or disable measurement requests and autonomous reports.

**Table 7-28—Summary of use of Enable, Request, and Report bits**

| Bits | | | Meaning of bits |
|---|---|---|---|
| Enable | Request | Report | |
| 0 | 0 | 0 | When Enable bit is set to 0, Request and Report bits are invalid and shall be set to 0. |
| 0 | 0 | 1 | Not allowed. |
| 0 | 1 | 0 | Not allowed. |
| 0 | 1 | 1 | Not allowed. |
| 1 | 0 | 0 | The transmitting STA is requesting that it be sent neither measurement requests nor autonomous measurement reports of the types indicated in the Measurement Type field. |
| 1 | 1 | 0 | The transmitting STA is indicating it will accept measurement requests and requesting it not be sent autonomous measurement reports of the types indicated in the Measurement Type field. |
| 1 | 0 | 1 | The transmitting STA is requesting it not be sent measurement requests and indicating it will accept autonomous measurement reports of the types indicated in the Measurement Type field. |
| 1 | 1 | 1 | The transmitting STA is indicating it will accept measurement requests and autonomous measurement reports of the type indicated in the Measurement Type field. |

The Measurement Type field shall be set to a number that identifies a measurement request or a measurement report. The Measurement Types that have been allocated for measurement requests are shown in Table 7-29 and measurement reports are shown in Table 7-30 (in 7.3.2.22).

**Table 7-29—Measurement Type definitions for measurement requests**

| Name | Measurement Type |
|---|---|
| Basic request | 0 |
| Clear channel assessment (CCA) request | 1 |
| Receive power indication (RPI) histogram request | 2 |
| Reserved | 3–255 |


Copyright © 2007 IEEE. All rights reserved.

The Measurement Request field shall be null when the Enable bit is set to 1 and shall contain the specification of the measurement request, as described in 7.3.2.21.1 through 7.3.2.21.3, when the Enable bit is set to 0.

The Measurement Request element is included in a Measurement Request frame as described in 7.4.1.1. The use of Measurement Request elements and frames is described in 11.9.6.

### 7.3.2.21.1 Basic request

A Measurement Type in the Measurement Request element may indicate a basic request. The response to a basic request is a basic report. It is mandatory for a STA in a BSS to generate a basic report in response to a basic request if the request is received from the AP with which it is associated, except as specified in 11.9.6. The Measurement Request field corresponding to a basic request is shown in Figure 7-60.

| | Channel Number | Measurement Start Time | Measurement Duration |
|---|---|---|---|
| **Octets:** | 1 | 8 | 2 |

**Figure 7-60—Measurement Request field format for a basic request**

The Channel Number field shall be set to the channel number for which the measurement request applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF timer at the time (± 32 μs) at which the requested basic request measurement shall start. A value of 0 shall indicate it shall start immediately.

The Measurement Duration field shall be set to the duration of the requested measurement, expressed in TUs.

### 7.3.2.21.2 CCA request

A Measurement Type in the Measurement Request element may indicate a CCA request. A response to a CCA request is a CCA report. It is optional for a STA to generate a CCA report in response to a CCA Request. The Measurement Request field corresponding to a CCA request is shown in Figure 7-61.

| | Channel Number | Measurement Start Time | Measurement Duration |
|---|---|---|---|
| **Octets:** | 1 | 8 | 2 |

**Figure 7-61—Measurement Request field format for a CCA request**

The Channel Number field shall be set to the channel number for which the measurement request applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF at the time (± 32 μs) at which the requested CCA request measurement shall start. A value of 0 shall indicate it shall start immediately.

The Measurement Duration field shall be set to the duration of the requested measurement, expressed in TUs.

Copyright © 2007 IEEE. All rights reserved.

#### 7.3.2.21.3 RPI histogram request

A Measurement Type in the Measurement Request element may indicate an RPI histogram request. A response to an RPI histogram request is an RPI histogram report. It is optional for a STA to generate a RPI histogram report in response to a RPI histogram request. The Measurement Request field corresponding to an RPI histogram request is shown in Figure 7-62.

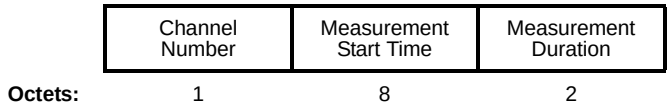



**Figure 7-62—Measurement Request field format for a RPI histogram request**

The Channel Number field shall be set to the channel number for which the measurement request applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF at the time (± 32 μs) at which the requested RPI histogram request measurement shall start. A value of 0 shall indicate it shall start immediately.

The Measurement Duration field shall be set to the duration of the requested measurement, expressed in TUs.

#### 7.3.2.22 Measurement Report element

The Measurement Report element contains a measurement report. The format of the Measurement Report element is shown in Figure 7-63.

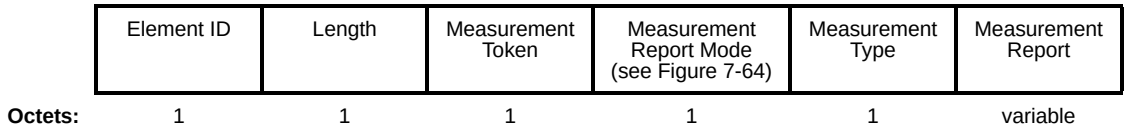



**Figure 7-63—Measurement Report element format**

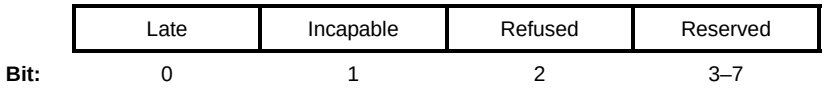



**Figure 7-64—Measurement Report Mode field**

The Length field is variable and depends on the length of the Measurement Report field. The minimum value of the Length field is 3.

The Measurement Token field shall be set to the Measurement Token in the corresponding Measurement Request element. If the Measurement Report element is being sent autonomously, then the Measurement Token shall be set to 0.


Copyright © 2007 IEEE. All rights reserved.

The Measurement Report Mode field (shown in Figure 7-64) is a bit field with the following bits defined:

— Late bit (bit 0) indicates whether this STA is unable to carry out a measurement request because it received the request after the requested measurement time. The Late bit shall be set to 1 to indicate the request was too late. The Late bit shall be set to 0 to indicate the request was received in time for the measurement to be executed.

— Incapable bit (bit 1) indicates whether this STA is incapable of generating a report of the type specified in the Measurement Type field that was previously requested by the destination STA of this Measurement Report element. The Incapable bit shall be set to 1 to indicate the STA is incapable. The Incapable bit shall be set to 0 to indicate the STA is capable or the report is autonomous.

— Refused bit (bit 2) indicates whether this STA is refusing to generate a report of the type specified in the Measurement Type field that was previously requested by the destination STA of this Measurement Report element. The Refused bit shall be set to 1 to indicate the STA is refusing. The Refused bit shall be set to 0 to indicate the STA is not refusing or the report is autonomous.

— All other bits are reserved and shall be set to 0.

The Measurement Type field shall be set to a number that identifies the measurement report. The Measurement Types that have been allocated are shown in Table 7-30.

**Table 7-30—Measurement Type definitions for measurement reports**

| Name | Measurement Type |
|---|---|
| Basic report | 0 |
| CCA report | 1 |
| RPI histogram report | 2 |
| Reserved | 3–255 |

The Measurement Report field shall be null when the Late bit is set to 1, the Incapable bit is set to 1, or the Refused bit is set to 1. Otherwise, it shall contain the specification of the measurement report, as described in 7.3.2.22.1 through 7.3.2.22.3.

The Measurement Report element is included in a Measurement Report frame as described in 7.4.1.2. The use of Measurement Report elements and frames is described in 11.9.6.

### 7.3.2.22.1 Basic report

A Measurement Type in the Measurement Report element may indicate a basic report. The format of the Measurement Report field corresponding to a basic report is shown in Figure 7-65. It is mandatory for a STA to support the generation of this report.

| | Channel Number | Measurement Start Time | Measurement Duration | Map (see Figure 7-66) |
|---|---|---|---|---|
| **Octets:** | 1 | 8 | 2 | 1 |

**Figure 7-65—Measurement Report field format for a basic report**

Copyright © 2007 IEEE. All rights reserved.

| | BSS | Orthogonal frequency division multiplexing (OFDM) Preamble | Unidentified Signal | Radar | Unmeasured | Reserved (0) |
|---|---|---|---|---|---|---|
| **Bit:** | 0 | 1 | 2 | 3 | 4 | 5-7 |

**Figure 7-66—Map field format**

The Channel Number field shall be set to the channel number to which the basic report applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF at the time (± 32 μs) at which the basic report measurement started.

The Measurement Duration field shall be set to the duration over which the basic report was measured, expressed in TUs.

The Map field is coded as a bit field, as shown in Figure 7-66, and shall contain the following bits:

— BSS bit, which shall be set to 1 when at least one valid MPDU was received in the channel during the measurement period from another BSS or IBSS. Otherwise, the BSS bit shall be set to 0.

— OFDM preamble bit, which shall be set to 1 when at least one sequence of short training symbols, as defined in 17.3.3, was detected in the channel during the measurement period without a subsequent valid Signal field (see 17.3.4). This may indicate the presence of an OFDM preamble, such as high-performance RLAN/2 (HIPERLAN/2). Otherwise, the OFDM preamble bit shall be set to 0.

— Unidentified Signal bit, which may be set to 1 when significant power is detected in the channel during the measurement period that cannot be characterized as radar, an OFDM preamble, or a valid MPDU. Otherwise, the Unidentified Signal bit shall be set to 0. The definition of significant power is implementation dependent.

— Radar bit, which shall be set to 1 when radar was detected operating in the channel during the measurement period. The algorithm to detect radar shall satisfy regulatory requirements and is outside the scope of this standard. Otherwise, the Radar bit shall be set to 0.

— Unmeasured bit, which shall be set to 1 when this channel has not been measured. Otherwise, the Unmeasured bit shall be set to 0. When the Unmeasured field is set to 1, all the other bit fields shall be set to 0.

### 7.3.2.22.2 CCA report

A Measurement Type in the Measurement Report element may indicate a CCA report. It is optional for a STA to support the generation of this report. The format of the Measurement Report field corresponding to a CCA report is shown in Figure 7-67.

| | Channel Number | Measurement Start Time | Measurement Duration | CCA Busy Fraction |
|---|---|---|---|---|
| **Octets:** | 1 | 8 | 2 | 1 |

**Figure 7-67—Measurement Report field format for a CCA report**


Copyright © 2007 IEEE. All rights reserved.

The Channel Number field shall contain the channel number to which the CCA report applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF at the time (± 32 μs) at which the CCA report measurement started.

The Measurement Duration field shall be set to the duration over which the CCA report was measured, expressed in TUs.

The CCA Busy Fraction field shall contain the fractional duration over which CCA indicated the channel was busy during the measurement duration. The resolution of the CCA busy measurement is in microseconds. The CCA Busy Fraction value is defined as Ceiling (255 * [Duration CCA indicated channel was busy (microseconds)] / (1024 * [Measurement duration (TUs)])).

#### 7.3.2.22.3 RPI histogram report

A Measurement Type in the Measurement Report element may indicate an RPI histogram report. It is optional for a STA to support the generation of this report. The format of the Measurement Report field corresponding to an RPI histogram report is shown in Figure 7-68.

| | Channel Number | Measurement Start Time | Measurement Duration |
|---|---|---|---|
| **Octets:** | 1 | 8 | 2 |

| | RPI 0 density | RPI 1 density | RPI 2 density | RPI 3 density | RPI 4 density | RPI 5 density | RPI 6 density | RPI 7 density |
|---|---|---|---|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 7-68—Measurement Report field format for an RPI histogram report**

The Channel Number field shall be set to the channel number to which the RPI histogram report applies (as defined in 17.3.8.3.3).

The Measurement Start Time field shall be set to the TSF at the time (± 32 μs) at which the RPI histogram report measurement started.

The Measurement Duration field shall be set to the duration over which the RPI histogram report was measured, expressed in TUs.

The RPI histogram report shall contain the RPI densities observed in the channel for the eight RPI levels defined in Table 7-31. To compute the RPI densities, the STA shall measure the received power level on the specified channel, as detected at the antenna connector, as a function of time over the measurement duration. The maximum tolerance of the received power measurements shall be ± 5 dB. Furthermore, the received signal power measurement should be a monotonic function of the actual power at the antenna. The time resolution of the received power measurements is in microseconds. The received power measurements are converted to a sequence of RPI values by quantizing the measurements according to Table 7-31. The RPI densities are then computed for each of the eight possible RPI values using Ceiling (255 * [Duration receiving at RPI value (microseconds) / (1024 * Measurement duration)]). The sum of the RPI densities will be approximately 255, but could be up to 262 because of rounding effects.

Copyright © 2007 IEEE. All rights reserved.

**Table 7-31—RPI definitions for an RPI histogram report**

| RPI | Power observed at the antenna (dBm) |
|---|---|
| 0 | Power ≤ –87 |
| 1 | –87 < Power ≤ –82 |
| 2 | –82 < Power ≤ –77 |
| 3 | –77 < Power ≤ –72 |
| 4 | –72 < Power ≤ –67 |
| 5 | –67 < Power ≤ –62 |
| 6 | –62 < Power ≤ –57 |
| 7 | –57 < Power |

The RPI histogram report provides an additional mechanism for a STA to gather information on the state of a channel from other STAs. The STA may use this information to assist in the choice of new channel, to help avoid false radar detections, and to assess the general level of interference present on a channel.

### 7.3.2.23 Quiet element

The Quiet element defines an interval during which no transmission shall occur in the current channel. This interval may be used to assist in making channel measurements without interference from other STAs in the BSS or IBSS. The format of the Quiet element is shown in Figure 7-69.

| | Element ID | Length | Quiet Count | Quiet Period | Quiet Duration | Quiet Offset |
|---|---|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | 1 | 2 | 2 |

**Figure 7-69—Quiet element format**

The Length field shall be set to 6.

The Quiet Count field shall be set to the number of TBTTs until the beacon interval during which the next quiet interval shall start. A value of 1 indicates the quiet interval shall start during the beacon interval starting at the next TBTT. A value of 0 is reserved.

The Quiet Period field shall be set to the number of beacon intervals between the start of regularly scheduled quiet intervals defined by this Quiet element. A value of 0 indicates that no periodic quiet interval is defined.

The Quiet Duration field shall be set to the duration of the quiet interval, expressed in TUs.

The Quiet Offset field shall be set to the offset of the start of the quiet interval from the TBTT specified by the Quiet Count field, expressed in TUs. The value of the Quiet Offset field shall be less than one beacon interval.

The Quiet element may be included in Beacon frames, as described in 7.2.3.1, and Probe Response frames, as described in 7.2.3.9. The use of Quiet elements is described in 11.9.2.


Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.24 IBSS DFS element

The IBSS DFS element contains information for DFS operation in an IBSS. The format of the IBSS DFS element is shown in Figure 7-70.

| | Element ID | Length | DFS Owner | DFS Recovery Interval | Channel Map (see Figure 7-71) |
|---|---|---|---|---|---|
| **Octets:** | 1 | 1 | 6 | 1 | 2*n* |

**Figure 7-70—IBSS DFS element format**

| | Channel Number | Map | *n* tuples, one for each supported channel |
|---|---|---|---|
| **Octets:** | 1 | 1 | |

**Figure 7-71—Channel Map field format**

The Length field is variable.

The DFS Owner field shall be set to the individual IEEE MAC address of the STA that is the currently known DFS Owner in the IBSS.

The DFS Recovery Interval field indicates the time interval that shall be used for DFS owner recovery, expressed as an integral number of beacon intervals. The DFS Recovery Interval value is static throughout the lifetime of the IBSS and is determined by the STA that starts the IBSS.

The Channel Map field shown in Figure 7-71 shall contain a Channel Number field and a Map field (see 7.3.2.22.1) for each channel supported by the STA transmitting the IBSS DFS element. Note that *n* in Figure 7-70 is the number of channels supported by the STA.

The IBSS DFS element may be included in Beacon frames, as described in 7.2.3.1, and Probe Response frames, as described in 7.2.3.9. The use of IBSS DFS elements is described in 11.9.7.2.

### 7.3.2.25 RSN information element

The RSN information element contains authentication and pairwise cipher suite selectors, a single group cipher suite selector, an RSN Capabilities field, the PMK identifier (PMKID) count, and PMKID list. See Figure 7-72. All STAs implementing RSNA shall support this element. The size of the RSN information element is limited by the size of an information element, which is 255 octets. Therefore, the number of pairwise cipher suites, AKM suites, and PMKIDs is limited.

| | Element ID | Length | Version | Group Cipher Suite | Pairwise Cipher Suite Count | Pairwise Cipher Suite List | AKM Suite Count | AKM Suite List | RSN Capabili-ties | PMKID Count | PMKI DList |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Octets:** | 1 | 1 | 2 | 4 | 2 | 4-*m* | 2 | 4-*n* | 2 | 2 | 16-*s* |

**Figure 7-72—RSN information element format**

Copyright © 2007 IEEE. All rights reserved.

In Figure 7-72, *m* denotes the pairwise cipher suite count, *n* the AKM suite count, and *s* is the PMKID count.

All fields use the bit convention from 7.1.1. The RSN information element shall contain up to and including the Version field. All fields after the Version field are optional. If any optional field is absent, then none of the subsequent fields shall be included.

Element ID shall be 48 decimal (30 hex).

Length gives the number of octets in the information field (field(s) following the Element ID and Length fields) of the information element.

The Version field indicates the version number of the RSNA protocol. The range of Version field values a STA supports shall be contiguous. Values 0 and 2 or higher of the Version field are reserved. RSN Version 1 is defined in this standard.

NOTE—The following represent sample information elements:

802.1X authentication, CCMP pairwise and group cipher suites (WEP-40, WEP-104, and TKIP not allowed):

30, // information element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
00 00 // No capabilities

802.1X authentication, CCMP pairwise and group cipher suites (WEP-40, WEP-104 and TKIP not allowed), preauthentication supported:

30, // information element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
01 00 // Preauthentication capabilities

802.1X authentication, Use GTK for pairwise cipher suite, WEP-40 group cipher suites, optional RSN Capabilities field omitted:

30, // information element id, 48 expressed as Hex value
12, // length in octets, 18 expressed as Hex value
01 00, // Version 1
00 0F AC 01, // WEP-40 as group cipher suite
01 00, // pairwise cipher suite count
00 0F AC 00, // Use group key as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication

802.1X authentication, Use CCMP for pairwise cipher suite, CCMP group cipher suites, preauthentication and a PMKID:

30, // information element id, 48 expressed as Hex value
26 // length in octets, 38 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // 802.1X authentication
01 00 // Preauthentication capabilities
01 00 // PMKID Count
01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 // PMKID


Copyright © 2007 IEEE. All rights reserved.

#### 7.3.2.25.1 Cipher suites

The Group Cipher Suite field contains the cipher suite selector used by the BSS to protect broadcast/ multicast traffic.

The Pairwise Cipher Suite Count field indicates the number of pairwise cipher suite selectors that are contained in the Pairwise Cipher Suite List field.

The Pairwise Cipher Suite List field contains a series of cipher suite selectors that indicate the pairwise cipher suites contained in the RSN information element.

A suite selector has the format shown in Figure 7-73.

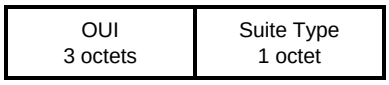



**Figure 7-73—Suite selector format**

The order of the organizationally unique indentifier (OUI) field shall follow the ordering convention for MAC addresses from 7.1.1.

Table 7-32 provides the cipher suite selectors defined by this standard.

**Table 7-32—Cipher suite selectors**

| OUI | Suite type | Meaning |
|---|---|---|
| 00-0F-AC | 0 | Use group cipher suite |
| 00-0F-AC | 1 | WEP-40 |
| 00-0F-AC | 2 | TKIP |
| 00-0F-AC | 3 | Reserved |
| 00-0F-AC | 4 | CCMP – default in an RSNA |
| 00-0F-AC | 5 | WEP-104 |
| 00-0F-AC | 6–255 | Reserved |
| Vendor OUI | Other | Vendor-specific |
| Other | Any | Reserved |

The cipher suite selector 00-0F-AC:4 (CCMP) shall be the default cipher suite value.

The cipher suite selectors 00-0F-AC:1 (WEP-40) and 00-0F-AC:5 (WEP-104) are only valid as a group cipher suite in a transition security network (TSN) to allow pre-RSNA devices to join the BSS.

Use of CCMP as the group cipher suite with TKIP as the pairwise cipher suite shall not be supported.

NOTE—If the STAs can support CCMP, then there is no need for a weaker data confidentiality protocol.

The cipher suite selector 00-0F-AC:0 (Use group cipher suite) is only valid as the pairwise cipher suite. An AP may specify the selector 00-0F-AC:0 (Use group cipher suite) for a pairwise cipher suite if it does not

Copyright © 2007 IEEE. All rights reserved.

support any pairwise cipher suites. If an AP specifies 00-0F-AC:0 (Use group cipher suite) as the pairwise cipher selection, this shall be the only pairwise cipher selection the AP advertises.

If CCMP is enabled, then the AP supports pairwise keys, and thus the suite selector 00-0F-AC:0 (Use group cipher suite) shall not be a valid option.

Table 7-33 indicates the circumstances under which each cipher suite shall be used.

**Table 7-33—Cipher suite usage**

| Cipher suite selector | GTK | PTK |
|---|---|---|
| Use group key | No | Yes |
| WEP-40 | Yes | No |
| WEP-104 | Yes | No |
| TKIP | Yes | Yes |
| CCMP | Yes | Yes |

#### 7.3.2.25.2 AKM suites

The AKM Suite Count field indicates the number of AKM suite selectors that are contained in the AKM Suite List field.

The AKM Suite List field contains a series of AKM suite selectors contained in the RSN information element. In an IBSS, only a single AKM suite selector may be specified because STAs in an IBSS must use the same AKM suite and because there is no mechanism to negotiate the AKMP in an IBSS (see 8.4.4).

Each AKM suite selector specifies an AKMP. Table 7-34 gives the AKM suite selectors defined by this standard.

**Table 7-34—AKM suite selectors**

| OUI | Suite type | Meaning | |
|---|---|---|---|
| | | Authentication type | Key management type |
| 00-0F-AC | 0 | Reserved | Reserved |
| 00-0F-AC | 1 | Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 8.4.6.2 – RSNA default | RSNA key management as defined in 8.5 or using PMKSA caching as defined in 8.4.6.2 – RSNA default |
| 00-0F-AC | 2 | PSK | RSNA key management as defined in 8.5, using PSK |
| 00-0F-AC | 3–255 | Reserved | Reserved |
| Vendor OUI | Any | Vendor-specific | Vendor-specific |
| Other | Any | Reserved | Reserved |


Copyright © 2007 IEEE. All rights reserved.

The AKM suite selector value 00-0F-AC:1 (Authentication negotiated over IEEE 802.1X) with (RSNA key management as defined in 8.5 or using PMKSA caching as defined in 8.4.6.2) shall be the assumed default when the AKM suite selector field is not supplied.

NOTE—The selector value 00-0F-AC:1 specifies only that IEEE Std 802.1X-2004 is used as the authentication transport. IEEE Std 802.1X-2004 selects the authentication mechanism.

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

NOTE—Selector values 00-0F-AC:1 and 00-0F-AC:2 can simultaneously be enabled by an Authenticator.

#### 7.3.2.25.3 RSN capabilities

The RSN Capabilities field indicates requested or advertised capabilities. The value of each of the RSN Capabilities fields shall be taken to be 0 if the RSN Capabilities field is not available in the RSN information element.

The length of the RSN Capabilities field is 2 octets. The format of the RSN Capabilities field is as illustrated in Figure 7-74 and described after the figure.

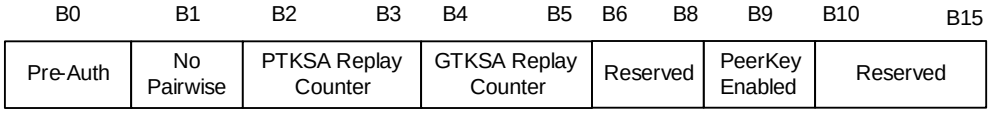



**Figure 7-74—RSN Capabilities field format**

— Bit 0: Pre-Authentication. An AP sets the Pre-Authentication subfield of the RSN Capabilities field to 1 to signal it supports preauthentication (see 8.4.6.1) and sets the subfield to 0 when it does not support preauthentication. A non-AP STA sets the Pre-Authentication subfield to 0.

— Bit 1: No Pairwise. If a STA can support WEP default key 0 simultaneously with a pairwise key (see 8.5.1), then the STA sets the No Pairwise subfield of the RSN Capabilities field to 0.

  If a STA cannot support WEP default key 0 simultaneously with a pairwise key (see 8.5.1), then the STA sets the No Pairwise subfield of the RSN Capabilities field to 1.

  The No Pairwise subfield describes a capability of a non-AP STA. STAs in an IBSS and APs set the No Pairwise subfield to 0.

  The No Pairwise subfield shall be set only in a TSN and when the pairwise cipher suite selected by the STA is TKIP.

— Bits 2–3: PTKSA Replay Counter. A STA sets the PTKSA Replay Counter subfield of the RSN Capabilities field to the value contained in dot11RSNAConfigNumberofPTKSAReplayCounters. The least significant bit (LSB) of dot11RSNAConfigNumberofPTKSAReplayCounters is put in bit 2. See 8.3.2.6 and 8.3.3.4.3. The meaning of the value in the PTKSA/GTKSA/STKSA Replay Counter subfield is defined in Table 7-35. The number of replay counters per STKSA is the same as the number of replay counters per PTKSA.

— Bits 4–5: GTKSA Replay Counter. A STA sets the GTKSA Replay Counter subfield of the RSN Capabilities field to the value contained in dot11RSNAConfigNumberofPTKSAReplayCounters. The LSB of dot11RSNAConfigNumberofGTKSAReplayCounters is put in bit 4. See 8.3.2.6 and 8.3.3.4.3. The meaning of the value in the GTKSA Replay Counter subfield is defined in Table 7-35.

— Bits 9: PeerKey Enabled. An AP STA sets the PeerKey Enabled subfield of the RSN Capabilities field to 1 to signal it supports PeerKey Handshake (see 8.5.8). This field is used by AP STA to describe its ability to support PeerKey Handshake.

— Bits 6–8 and 10–15: Reserved. The remaining subfields of the RSN Capabilities field are reserved and shall be set to 0 on transmission and ignored on reception.

Copyright © 2007 IEEE. All rights reserved.

**Table 7-35—PTKSA/GTKSA/STKSA replay counters usage**

| Replay counter value | Meaning |
|---|---|
| 0 | 1 replay counter per PTKSA/GTKSA/STKSA |
| 1 | 2 replay counters per PTKSA/GTKSA/STKSA |
| 2 | 4 replay counters per PTKSA/GTKSA/STKSA |
| 3 | 16 replay counters per PTKSA/GTKSA/STKSA |

#### 7.3.2.25.4 PMKID

The PMKID Count and List fields shall be used only in the RSN information element in the (Re)Association Request frame to an AP. The PMKID Count specifies the number of PMKIDs in the PMKID List field. The PMKID list contains 0 or more PMKIDs that the STA believes to be valid for the destination AP. The PMKID can refer to

a) A cached PMKSA that has been obtained through preauthentication with the target AP

b) A cached PMKSA from an EAP authentication

c) A PMKSA derived from a PSK for the target AP

See 8.5.1.2 for the construction of the PMKID.

NOTE—A STA can choose not to insert a PMKID in the PMKID List field if the STA does not want to use that PMKSA.

### 7.3.2.26 Vendor Specific information element

The Vendor Specific information element is used to carry information not defined in this standard within a single defined format, so that reserved information element IDs are not usurped for nonstandard purposes and so that interoperability is more easily achieved in the presence of nonstandard information. The information element is in the format shown in Figure 7-75 and requires that the first 3 octets of the information field contain the OUI of the entity that has defined the content of the particular Vendor Specific information element. The length of the information field ($n$) is $3 \le n \le 255$. The OUI field shall be a public OUI assigned by the IEEE. It is 3 octets in length. The length of the vendor-specific content is $n$–3 octets.

| | Element ID | Length | OUI | Vendor-specific content |
|---|---|---|---|---|
| **Octets:** | 1 | 1 | 3 | n-3 |

**Figure 7-75—Vendor Specific information element format**

Multiple Vendor Specific information elements may appear in a single frame. Each Vendor Specific information element can have a different OUI value. The number of Vendor Specific information elements that may appear in a frame is limited only by the maximum frame size.

### 7.3.2.27 Extended Capabilities information element

The Extended Capabilities information element carries information about the capabilities of an IEEE 802.11 STA, intended to augment the Capability Information field (CIF) when these bits are fully allocated. The format of this information element is shown in Figure 7-76.


Copyright © 2007 IEEE. All rights reserved.

| | Element ID | Length | Capabilities |
|---|---|---|---|
| **Octets:** | 1 | 1 | n |

**Figure 7-76—Extended Capabilities element format**

The element ID of this information element is 127.

The value of the Length field is equal to the number of octets in the Capabilities field.

The Capabilities field is a bit field indicating the capabilities being advertised by the STA transmitting the information element. There are no capabilities defined for this field in this revision of the standard.

### 7.3.2.28 BSS Load element

The BSS Load element contains information on the current STA population and traffic levels in the BSS. The element information format is defined in Figure 7-77. This element may be used by the non-AP STA for vendor-specific AP selection algorithm when roaming.

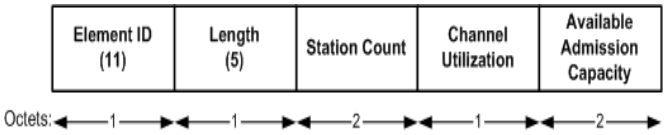



**Figure 7-77—BSS Load element format**

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

The Channel Utilization field is defined as the percentage of time, normalized to 255, the AP 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 dot11ChannelUtilization-BeaconIntervals 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 STAs to select an AP that is likely to accept future admission control requests, but it does not represent a guarantee that the HC will admit these requests.

Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.29 EDCA Parameter Set element

The EDCA Parameter Set element provides information needed by non-AP STAs for proper operation of the QoS facility during the CP. The format of the EDCA Parameter Set element is defined in Figure 7-78.

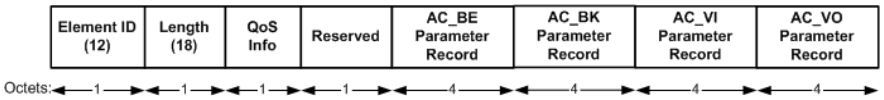



**Figure 7-78—EDCA Parameter Set element**

The EDCA Parameter Set element is used by the AP 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 STA 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 STAs 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 7-79.

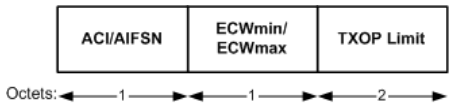



**Figure 7-79—AC_BE, AC_BK, AC_VI, and AC_VO Parameter Record field format**

The format of the ACI/AIFSN field is illustrated in Figure 7-80.

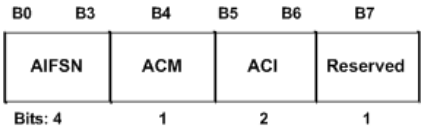



**Figure 7-80—ACI/AIFSN field**

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 7-36. 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 STA should defer before either invoking a backoff or starting a transmission. The minimum value for the AIFSN subfield is 2.


Copyright © 2007 IEEE. All rights reserved.

**Table 7-36—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 7-81.

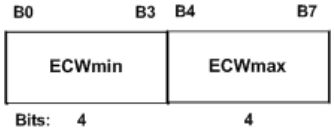



**Figure 7-81—ECWmin and ECWmax fields**

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 STAs for the parameters in the EDCA Parameter Set element are defined in Table 7-37.[19]

**Table 7-37—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 |

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

Copyright © 2007 IEEE. All rights reserved.

### 7.3.2.30 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 STA, for use by the HC and non-AP STA(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 7-82.

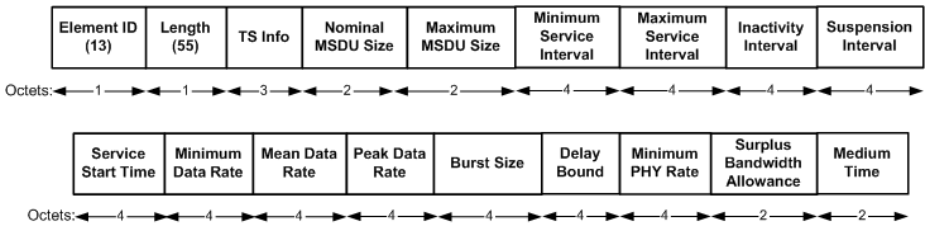



**Figure 7-82—TSPEC element format**

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 STAs 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 STA to any value that it deems appropriate, including leaving them unspecified.

The structure of the TS Info field is defined in Figure 7-83.




**Figure 7-83—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 a value that is a TSID. 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 STA, to which the TSPEC applies. The same TSID may be used for multiple TSs at different non-AP STAs. A non-AP STA 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 STA 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 7-38.

**Table 7-38—Direction subfield encoding**

| Bit 5 | Bit 6 | Usage |
|---|---|---|
| 0 | 0 | Uplink (MSDUs are sent from the non-AP STA to HC) |
| 1 | 0 | Downlink (MSDUs are sent from the HC to the non-AP STA) |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-38—Direction subfield encoding *(continued)***

| Bit 5 | Bit 6 | Usage |
|---|---|---|
| 0 | 1 | Direct link (MSDUs are sent from the non-AP STA to another non-AP STA) |
| 1 | 1 | Bidirectional link (equivalent to a downlink request plus an uplink request, each direction having the same parameters).<br>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 7-39.

**Table 7-39—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 STA to indicate that an aggregate schedule is required. It is set to 1 by the AP if an aggregate schedule is being provided to the non-AP STA. 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 7-40. 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 7-40—TS Info Ack Policy subfield encoding**

| Bit 14 | Bit 15 | Usage |
|---|---|---|
| 0 | 0 | Normal IEEE 802.11 acknowledgment<br>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. |

Copyright © 2007 IEEE. All rights reserved.

— 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 7-41. 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 AP shall set the aggregation bit to 1, indicating that an aggregate schedule is being provided to the non-AP STA.

**Table 7-41—Setting of Schedule subfield**

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

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

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 7-84. 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.

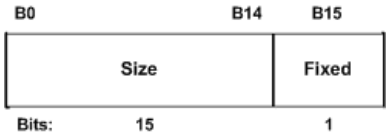



**Figure 7-84—Nominal MSDU Size field**

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 first scheduled SP starts. The service start time indicates to AP the time when a


Copyright © 2007 IEEE. All rights reserved.

non-AP STA first expects to be ready to send frames and a power-saving non-AP STA will be awake to receive frames. This may help the AP to schedule service so that the MSDUs encounter small delays in the MAC and help the power-saving non-AP STAs to reduce power consumption. The field represents the four lower order octets 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[20] 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.[21]

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.

[20]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 2212-1997 [B19], IETF RFC 2215-1997 [B20], and IETF RFC 3290-2002 [B24].

[21]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.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 183 of 1232

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.

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 STA, if this TSPEC was issued by that non-AP STA, 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 STA does not have specific requirements for these parameters if the TSPEC was issued by that non-AP STA or that the HC does not provide any specific values for these parameters if the TSPEC was issued by the HC.

#### 7.3.2.31 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 STAs 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 7-85.




**Figure 7-85—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 7-86.

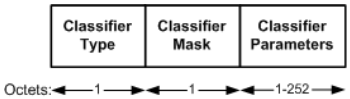



**Figure 7-86—Frame Classifier field**

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 7-42. Three classifier types are defined later in this subclause.

**Table 7-42—Frame classifier type**

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


Copyright © 2007 IEEE. All rights 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" [B5]). The Frame Classifier field for Classifier Type 0 is defined in Figure 7-87. It has a length of 16 octets.

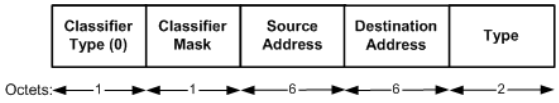



**Figure 7-87—Frame Classifier field of Classifier Type 0**

For Classifier Type 1, frame classifier is defined for both IPv4 and IPv6, shown in Figure 7-88 and Figure 7-89, 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 2474-1998 [B21]) and Protocol, or

  — In an IPv6 header: Flow Label.

The DSCP field shall 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 7-88—Frame Classifier field of Classifier Type 1 for traffic over IPv4**

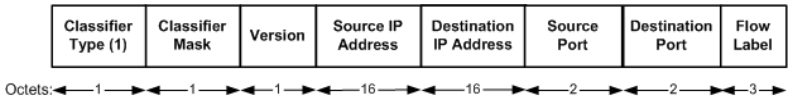



**Figure 7-89—Frame Classifier field of Classifier Type 1 for traffic over IPv6**

Copyright © 2007 IEEE. All rights reserved.

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 (see IEEE Std 802.1Q, 2003 Edition, [B14]). The Frame Classifier field for Classifier Type 2 is defined in Figure 7-90.

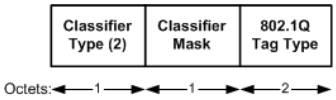



**Figure 7-90—Frame Classifier field of Classifier Type 2**

### 7.3.2.32 TS Delay element

The TS Delay element is used in the ADDTS Response frame transmitted by the HC to a non-AP STA and indicates the time after which the ADDTS may be retried. The TS Delay element is defined in Figure 7-91.

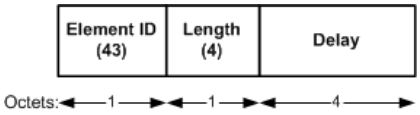



**Figure 7-91—TS Delay element**

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

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

### 7.3.2.33 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 7-92.

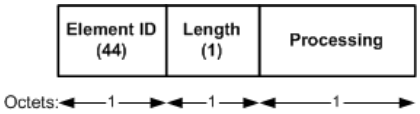



**Figure 7-92—TCLAS Processing element**


Copyright © 2007 IEEE. All rights reserved.

The Processing subfield is 1 octet long. The encoding of the Processing subfield is shown in Table 7-43.

**Table 7-43—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 shall not be any associated TCLAS elements. |
| 3–255 | Reserved |

### 7.3.2.34 Schedule element

The Schedule element is transmitted by the HC to a non-AP STA to announce the schedule that the HC/AP follows for admitted streams originating from or destined to that non-AP STA in the future. The information in this element may be used by the non-AP STA for power management, internal scheduling, or any other purpose. The element information format is shown in Figure 7-93.

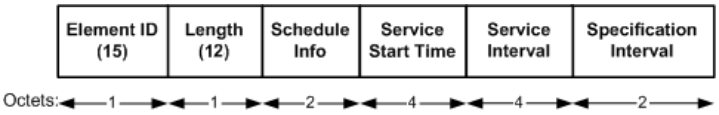



**Figure 7-93—Schedule element**

The Schedule Info field is shown in Figure 7-94.

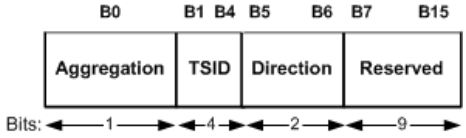



**Figure 7-94—Schedule Info field**

The Aggregation subfield is set to 1 if the schedule is an aggregate schedule for all TSIDs associated with the non-AP STA 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 7.3.2.30 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 octets of the TSF timer value at the start of the first SP. The AP uses this field to confirm or modify the service start time indicated in the TSPEC request.

Copyright © 2007 IEEE. All rights reserved.

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.

#### 7.3.2.35 QoS Capability element

The QoS Capability element contains a number of subfields that are used to advertise optional QoS capabilities at a QoS STA. 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 7-95.

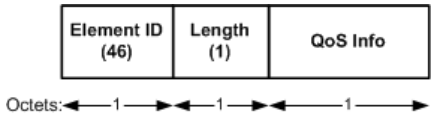



**Figure 7-95—QoS Capability element format**

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

### 7.4 Action frame format details

This subclause describes the Action frame formats, including the Action Details field, allowed in each of the action categories defined in Table 7-24 in 7.3.1.11.

#### 7.4.1 Spectrum management action details

Five Action frame formats are defined for spectrum management. An Action Value field, in the octet field immediately after the Category field, differentiates the five formats. The Action Value field values associated with each frame format are defined in Table 7-44.

**Table 7-44—Spectrum management Action Value field values**

| Action Value field value | Description |
|---|---|
| 0 | Measurement Request |
| 1 | Measurement Report |
| 2 | TPC Request |
| 3 | TPC Report |
| 4 | Channel Switch Announcement |
| 5–255 | Reserved |


Copyright © 2007 IEEE. All rights reserved.

### 7.4.1.1 Measurement Request frame format

The Measurement Request frame uses the Action frame body format and is transmitted by a STA requesting another STA to measure one or more channels. The format of the Measurement Request frame body is shown in Figure 7-96.

| | Category | Action Value | Dialog Token | Measurement Request Elements |
|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | variable |

**Figure 7-96—Measurement Request frame body format**

The Category field shall be set to 0 (representing spectrum management).

The Action Value field shall be set to 0 (representing a Measurement Request frame).

The Dialog Token field shall be set to a nonzero value chosen by the STA sending the measurement request to identify the request/report transaction.

The Measurement Request Elements field shall contain one or more of the Measurement Request elements described in 7.3.2.21. The number and length of the Measurement Request elements in a Measurement Request frame is limited by the maximum allowed MMPDU size.

### 7.4.1.2 Measurement Report frame format

The Measurement Report frame uses the Action frame body format and is transmitted by a STA in response to a Measurement Request frame or by a STA autonomously providing measurement information. The format of the Measurement Report frame body is shown in Figure 7-97.

| | Category | Action Value | Dialog Token | Measurement Report Elements |
|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | variable |

**Figure 7-97—Measurement Report frame body format**

The Category field shall be set to 0 (representing spectrum management).

The Action Value field shall be set to 1 (representing a Measurement Report frame).

The Dialog Token field shall be set to the value in any corresponding Measurement Request frame. If the Measurement Report frame is not being transmitted in response to a Measurement Request frame, then the Dialog token shall be set to 0.

The Measurement Report Elements field shall contain one or more of the Measurement Report elements described in 7.3.2.22. The number and length of the Measurement Report elements in a Measurement Report frame is limited by the maximum allowed MMPDU size.

Copyright © 2007 IEEE. All rights reserved.

### 7.4.1.3 TPC Request frame format

The TPC Request frame uses the Action frame body format and is transmitted by a STA requesting another STA for transmit power and link margin information. The format of the TPC Request frame body is shown in Figure 7-98.

| | Category | Action Value | Dialog Token | TPC Request element |
|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | 2 |

**Figure 7-98—TPC Request frame body format**

The Category field shall be set to 0 (representing spectrum management).

The Action Value field shall be set to 2 (representing a TPC Request frame).

The Dialog Token field shall be set to a nonzero value chosen by the STA sending the request to identify the transaction.

The TPC Request element shall be set as described in 7.3.2.17.

### 7.4.1.4 TPC Report frame format

The TPC Report frame uses the Action frame body format and is transmitted by a STA in response to a TPC Request frame. The format of the TPC Report frame body is shown in Figure 7-99.

| | Category | Action Value | Dialog Token | TPC Report element |
|---|---|---|---|---|
| **Octets:** | 1 | 1 | 1 | 4 |

**Figure 7-99—TPC Report frame body format**

The Category field shall be set to 0 (representing spectrum management).

The Action Value field shall be set to 3 (representing a TPC Report frame).

The Dialog Token field shall be set to the Dialog Token value in the corresponding TPC Request frame.

The TPC Report element shall be set as described 7.3.2.18.


Copyright © 2007 IEEE. All rights reserved.

#### 7.4.1.5 Channel Switch Announcement frame format

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

| | Category | Action Value | Channel Switch Announcement element |
|---|---|---|---|
| **Octets:** | 1 | 1 | 5 |

**Figure 7-100—Channel Switch Announcement frame body format**

The Category field shall be set to 0 (representing spectrum management).

The Action Value field shall be set to 4 (representing a Channel Switch Announcement frame).

The Channel Switch Announcement element shall be set as described 7.3.2.20.

### 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 7-45.

**Table 7-45—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 7-46.

**Table 7-46—ADDTS Request frame body**

| Order | Information |
|---|---|
| 1 | Category |
| 2 | Action |

Copyright © 2007 IEEE. All rights reserved.

**Table 7-46—ADDTS Request frame body *(continued)***

| Order | Information |
|---|---|
| 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 7.3.2.30, and the optional TCLAS element, defined in 7.3.2.31, 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 STA 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 7.3.2.33.

#### 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 7-47.

**Table 7-47—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 |


Copyright © 2007 IEEE. All rights reserved.

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 7.3.2.34, 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 7-48.

**Table 7-48—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 7.3.2.30.

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.

Copyright © 2007 IEEE. All rights reserved.

#### 7.4.2.4 Schedule frame format

The Schedule frame is transmitted by the HC to a non-AP STA to announce the schedule of delivery of data and polls. The frame body of the Schedule frame contains the information shown in Table 7-49.

**Table 7-49—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 7.3.2.34.

### 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 7-50.

**Table 7-50—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 7-51.

**Table 7-51—DLS Request frame body**

| Order | Information |
|---|---|
| 1 | Category |
| 2 | Action |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-51—DLS Request frame body *(continued)***

| Order | Information |
|---|---|
| 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 7-52.

**Table 7-52—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 |

Copyright © 2007 IEEE. All rights reserved.

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 7-53.

**Table 7-53—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.


Copyright © 2007 IEEE. All rights reserved.

### 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 7-54.

**Table 7-54—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 STA. The frame body of an ADDBA Request frame contains the information shown in Table 7-55.

**Table 7-55—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.

Copyright © 2007 IEEE. All rights reserved.

#### 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 7-56.

**Table 7-56—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 7-57.

**Table 7-57—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).


Copyright © 2007 IEEE. All rights reserved.

The DELBA Parameters field is defined in 7.3.1.16.

The Reason Code field is defined in 7.3.1.7.

### 7.4.5 Vendor-specific action details

The Vendor Specific Action frame is defined for vendor-specific signaling. The format of the Vendor Specific Action frame is shown in Figure 7-101. An OUI, in the octet field immediately after the Category field, differentiates the vendors.

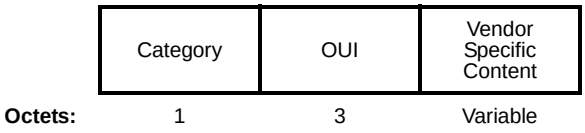



**Figure 7-101—Vendor Specific Action frame format**

The Category field is set to the value indicating the vendor-specific category, as specified in Table 7-24.

The OUI field is a public OUI assigned by the IEEE. It is 3 octets in length. It contains the OUI of the entity that has defined the content of the particular vendor-specific action.

The Vendor Specific Content contains vendor-specific field(s). The length of the Vendor Specific Content in a Vendor Specific Action frame is limited by the maximum allowed MMPDU size.

## 7.5 Frame usage

Table 7-58 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 7-58—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 | STA | STA | AP | STA | AP | STA | AP | STA | AP |
| (Re)Association Request | - - - | - - - | T | R | - - - | - - - | T | R | - - - | - - - |
| (Re)Association Response | - - - | - - - | R | T | R | T | R | T | R | T |
| 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 |

Copyright © 2007 IEEE. All rights reserved.

**Table 7-58—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 | STA | STA | AP | STA | AP | STA | AP | STA | AP |
| 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 |
| QoS (Data+)CF-Poll+CF-Ack | - - - | - - - | - - - | - - - | - - - | - - - | Rq, Rda | Tda, Tq | Rq, Rda | Tda, Tq |


Copyright © 2007 IEEE. All rights reserved.

**Table 7-58—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 | STA | STA | AP | STA | AP | STA | AP | STA | AP |
| 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 QoS STA 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 QoS STA 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 QoS STA 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.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.
Samsung Ex. 1044, Page 202 of 1232

# 8. Security

## 8.1 Framework

This standard defines two classes of security algorithms for IEEE 802.11 networks:

— Algorithms for creating and using an RSNA, called *RSNA algorithms*

— Pre-RSNA algorithms

NOTE—This standard does not prohibit STAs from simultaneously operating pre-RSNA and RSNA algorithms.

### 8.1.1 Security methods

Pre-RSNA security comprises the following algorithms:

— WEP, described in 8.2.1

— IEEE 802.11 entity authentication, described in 8.2.2

RSNA security comprises the following algorithms:

— TKIP, described in 8.3.2

— CCMP, described in 8.3.3

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

— Key management procedures, described in 8.5

### 8.1.2 RSNA equipment and RSNA capabilities

RSNA-capable equipment can create RSNAs. When dot11RSNAEnabled is true, RSNA-capable equipment shall include the RSN information element in Beacon, Probe Response, and (Re)Association Request frames and in Message 2 and Message 3 of the 4-Way Handshake. Pre-RSNA equipment is not capable of creating RSNAs.

### 8.1.3 RSNA establishment

A STA's SME establishes an RSNA in one of four ways:

a) When using IEEE 802.1X AKM in an ESS, an RSNA-capable STA's SME establishes an RSNA as follows:
   1) It identifies the AP as RSNA-capable from the AP's Beacon or Probe Response frames.
   2) It shall invoke Open System authentication.
   3) It negotiates cipher suites during the association process, as described in 8.4.2 and 8.4.3.
   4) It uses IEEE Std 802.1X-2004 to authenticate, as described in 8.4.6 and 8.4.7.
   5) It establishes temporal keys by executing a key management algorithm, using the protocol defined by 8.5.
   6) It programs the agreed-upon temporal keys and cipher suites into the MAC and invokes protection. See 8.3.2 and 8.3.3 for a description of the RSNA data protection mechanisms.

b) If an RSNA is based on a PSK in an ESS, the STA's SME establishes an RSNA as follows:
   1) It identifies the AP as RSNA-capable from the AP's Beacon or Probe Response frames.
   2) It shall invoke Open System authentication.
   3) It negotiates cipher suites during the association process, as described in 8.4.2 and 8.4.3.
   4) It establishes temporal keys by executing a key management algorithm, using the protocol defined by 8.5. It uses the PSK as the PMK.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 203 of 1232

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

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

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

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

2) It may optionally invoke Open System authentication.

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

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

d) An RSNA-capable STA's SME using IEEE 802.1X AKM in an IBSS establishes an RSNA as follows:

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

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

2) It may optionally invoke Open System authentication.

3) Each STA uses IEEE Std 802.1X-2004 to authenticate with the AS associated with the other STA's Authenticator, as described in 8.4.6 and 8.4.7. Hence two authentications are happening at the same time.

4) Each STA's SME establishes temporal keys by executing a key management algorithm, using the protocol defined in 8.5. Hence two such key management algorithms are happening in parallel between any two STA's Supplicants and Authenticators.

5) Both STAs use the agreed-upon temporal key portion of the PTK and pairwise cipher suite from one of the exchanges to protect the link. Each STA uses the GTK established by the exchange it initiated to protect the multicast and broadcast frames it transmits.

The time a security association takes to set up shall be less than the MIB variable dot11RSNAConfigSA-Timeout. The security association setup starts when initiated by the SME and completes when the MLME-SETPROTECTION.request primitive is invoked. The action the STA takes on the timeout is a policy decision. Some options include retrying the security association setup or trying another STA. This timeout allows recovery when one of the STAs setting up a security association fails to respond correctly to setting up the security association. It also allows recovery in IBSS when one of the two security associations fails because of a security association timeout.

### 8.1.4 RSNA PeerKey Support

The PeerKey protocol provides mutual authentication, session identification, and data confidentiality for a station-to-station connection. A PeerKey association, composed of an SMKSA and an STKSA, shall be allowed only within the context of an existing RSNA by both peers with a common AP. Both the initiator STA and the peer STA shall ensure that dot11RSNAEnabled is true before initiating the STSL master key (SMK) Handshake and STSL transient key (STK) Handshake and establishing their respective security associations.

A STSL session may chose to allow unprotected communication between STAs. In this case, the PeerKey protocol is not used.


Copyright © 2007 IEEE. All rights reserved.

### 8.1.5 RSNA assumptions and constraints

An RSNA assumes the following:

a) Each STA can generate cryptographic-quality random numbers. This assumption is fundamental, as cryptographic methods require a source of randomness. See H.6 for suggested hardware and software methods to achieve randomness suitable for this purpose.

b) When IEEE 802.1X authentication is used, the specific EAP method used performs mutual authentication. This assumption is intrinsic to the design of RSN in IEEE 802.11 LANs and cannot be removed without exposing both the STAs to man-in-the-middle attacks. EAP-MD5 is an example of an EAP method that does not meet this constraint (see IETF RFC 3748 [B26]). Furthermore, the use of EAP authentication methods where server and client credentials cannot be differentiated reduces the security of the method to that of a PSK due to the fact that malicious insiders can masquerade as servers and establish a man-in-the-middle attack.

   In particular, the mutual authentication requirement implies an unspecified prior enrollment process (e.g., a long-lived authentication key or establishment of trust through a third party such as a certification authority), as the STA must be able to identify the ESS or IBSS as a trustworthy entity and vice versa. The STA shares authentication credentials with the AS utilized by the selected AP or, in the case of PSK, the selected AP. The SSID provides an unprotected indication that the selected AP's authentication entity shares credentials with the STA. Only the successful completion of the IEEE 802.1X EAP or PSK authentication, after association, can validate any such indication that the AP is connected to an authorized network or service provider.

c) The mutual authentication method must be strong, meaning impersonation attacks are computationally infeasible when based on the information exposed by the authentication. This assumption is intrinsic to the design of RSN.

d) The AP and AS have a trustworthy channel between them that can be used to exchange cryptographic keys without exposure to any intermediate parties.

e) An IEEE 802.1X AS never exposes the common symmetric key to any party except the AP with which the STA is currently communicating. This is a very strong constraint. It implies that the AS itself is never compromised. It also implies that the IEEE 802.1X AS is embedded in the AP or that the AP is physically secure and the AS and the AP lie entirely within the same administrative domain. This assumption follows from the fact that if the AP and the AS are not co-located or do not share pairwise key encryption keys directly, then it is impossible to assure the mobile STA that its key, which is distributed by the AS to the AP, has not been compromised prior to use.

f) Similarly, a STA never shares with a third party a common symmetric key that it shares with a peer. Doing so destroys the utility of the key for detecting MPDU replay and forgery.

g) The STA's Supplicant and the Authenticator generate a different, fresh PTK for each session between the pair. This assumption is fundamental, as reuse of any PTK would enable compromise of all the data protected by that key.

h) The destination STA chosen by the transmitter is the correct destination. For example, Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP) are methods of determining the destination STA MAC address that are not secure from attacks by other members of the ESS. One of the possible solutions to this problem might be for the STA to send or receive only frames whose final DA or SA are the AP and for the AP to provide a network layer routing function. However, such solutions are outside the scope of this standard.

## 8.2 Pre-RSNA security methods

Except for Open System authentication, all pre-RSNA security mechanisms have been deprecated, as they fail to meet their security goals. New implementations should support pre-RSNA methods only to aid migration to RSNA methods.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 205 of 1232

### 8.2.1 Wired equivalent privacy (WEP)

#### 8.2.1.1 WEP overview

WEP-40 was defined as a means of protecting (using a 40-bit key) the confidentiality of data exchanged among authorized users of a WLAN from casual eavesdropping. Implementation of WEP is optional. The same algorithms have been widely used with a 104-bit key instead of a 40-bit key in fielded implementations; this is called WEP-104. The WEP cryptographic encapsulation and decapsulation mechanics are the same whether a 40-bit or a 104-bit key is used. Therefore, subsequently, WEP can refer to either WEP-40 or WEP-104.

#### 8.2.1.2 WEP MPDU format

Figure 8-1 depicts the encrypted frame body as constructed by the WEP algorithm.

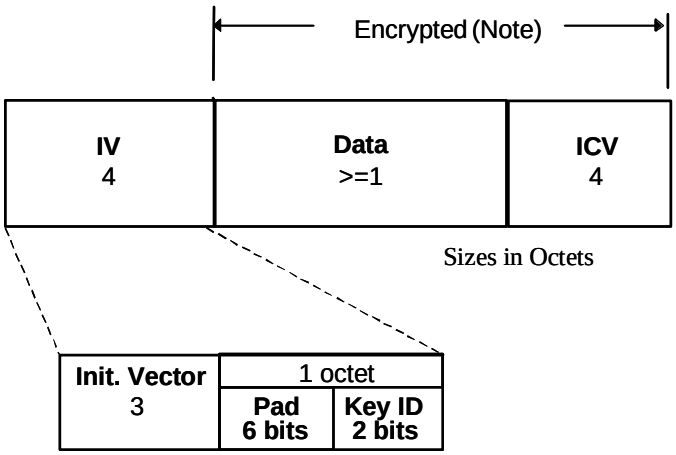



**Figure 8-1—Construction of expanded WEP MPDU**

The WEP ICV field shall be 32 bits in length. The expanded frame body shall start with a 32-bit IV field. This field shall contain three subfields: a 3-octet subfield that contains the IV, a 2-bit Key ID subfield, and a 6-bit Pad subfield. The ordering conventions defined in 7.1.1 apply to the IV field and its subfields and to the ICV field. The Key ID subfield contents select one of four possible secret key values for use in decrypting this frame body. When key-mapping keys are used, the Key ID field value is ignored.

Interpretation of these bits is discussed further in 8.2.1.3. The contents of the Pad subfield shall be 0. The Key ID subfield occupies the 2 MSBs of the last octet of the IV field, while the Pad subfield occupies the 6 LSBs of this octet.

#### 8.2.1.3 WEP state

WEP uses encryption keys only; it performs no data authentication. Therefore, it does not have data integrity keys. WEP uses two types of encryption keys: key-mapping keys and default keys.

A key-mapping key is an unnamed key corresponding to a distinct transmitter address-receiver address <TA,RA> pair. Implementations shall use the key-mapping key if it is configured for a <TA,RA> pair. In other words, the key-mapping key shall be used to WEP-encapsulate or -decapsulate MPDUs transmitted by TA to RA, regardless of the presence of other key types. When a key-mapping key for an address pair is present, the WEP Key ID subfield in the MPDU shall be set to 0 on transmit and ignored on receive.


Copyright © 2007 IEEE. All rights reserved.

A default key is an item in a four-element MIB array called dot11WEPDefaultKeys, named by the value of a related array index called dot11WEPDefaultKeyID. If a key-mapping key is not configured for a WEP MPDU's <TA,RA> pair, WEP shall use a default key to encapsulate or decapsulate the MPDU. On transmit, the key selected is the element of the dot11DefaultKeys array given by the index dot11WEPDefault-KeyID—a value of 0, 1, 2, or 3—corresponding to the first, second, third, or fourth element, respectively, of dot11WEPDefaultKeys. The value the transmitter encodes in the WEP Key ID subfield of the transmitted MPDU shall be the dot11WEPDefaultKeyID value. The receiver shall use the Key ID subfield of the MPDU to index into dot11WEPDefaultKeys to obtain the correct default key. All WEP implementations shall support default keys.

NOTE—Many implementations also support 104-bit WEP keys. These are used exactly like 40-bit WEP keys: a 24-bit WEP IV is prepended to the 104-bit key to construct a 128-bit WEP seed, as explained in 8.2.1.4.3. The resulting 128-bit WEP seed is then consumed by the ARC4 stream cipher.

This construction based on 104-bit keys affords no more assurance than the 40-bit construction, and its implementation and use are in no way condoned by this standard. Rather, the 104-bit construction is noted only to document de facto practice.

The default value for all WEP keys shall be null. WEP implementations shall discard the MSDU and generate an MA-UNITDATA.confirm with transmission status indicating that a frame may not be encapsulated with a null key in response to any request to encapsulate an MPDU with a null key.

#### 8.2.1.4 WEP procedures

##### 8.2.1.4.1 WEP ICV algorithm

The WEP ICV shall be computed using the CRC-32, as defined in 7.1.3.6, calculated over the plaintext MPDU Data (PDU) field.

##### 8.2.1.4.2 WEP encryption algorithm

A WEP implementation shall use the ARC4 stream cipher from RSA Security, Inc., as its encryption and decryption algorithm. ARC4 uses a pseudo-random number generator (PRNG) to generate a key stream that it exclusive-ORs (XORs) with a plaintext data stream to produce cipher text or to recover plaintext from a cipher text.

##### 8.2.1.4.3 WEP seed construction

A WEP implementation shall construct a per-MPDU key, called a *seed*, by concatenating an encryption key to an IV.

For WEP-40, bits 0–39 of the WEP key correspond to bits 24–63 of the seed, and bits 0–23 of the IV correspond to bits 0–23 of the seed, respectively. The bit numbering conventions in 7.1.1 apply to the seed. The seed shall be the input to ARC4, in order to encrypt or decrypt the WEP Data and ICV fields.

NOTE—For WEP-104, bits 0–103 of the WEP key correspond to bits 24–127 of the seed, and bit 0–23 of the IV correspond to bits 0–23 of the seed, respectively.

The WEP implementation encapsulating an MPDU's plaintext data should select a new IV for every MPDU it WEP-protects. The IV selection algorithm is unspecified. The algorithm used to select the encryption key used to construct the seed is also unspecified.

The WEP implementation decapsulating an MPDU shall use the IV from the received MPDU's Init Vector subfield. See 8.2.1.4.5 for the specification of how the decapsulator selects the key to use to construct the per-MPDU key.

Copyright © 2007 IEEE. All rights reserved.

#### 8.2.1.4.4 WEP MPDU cryptographic encapsulation

WEP shall apply three transformations to the plaintext MPDU to effect the WEP cryptographic encapsulation. WEP computes the ICV over the plaintext data and appends this after the MPDU data. WEP encrypts the MPDU plaintext data and ICV using ARC4 with a seed constructed as specified in 8.2.1.4.3. WEP encodes the IV and key identifier into the IV field, prepended to the encrypted Data field.

Figure 8-2 depicts the WEP cryptographic encapsulation process. The ICV shall be computed and appended to the plaintext data prior to encryption, but the IV encoding step may occur in any order convenient for the implementation.

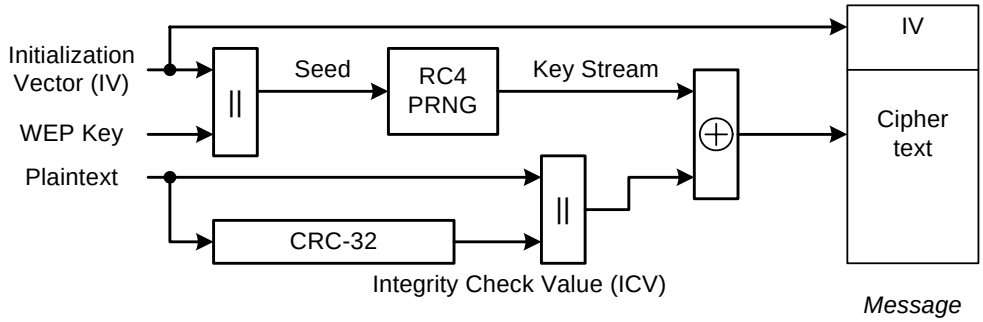



**Figure 8-2—WEP encapsulation block diagram**

#### 8.2.1.4.5 WEP MPDU decapsulation

WEP shall apply three transformations to the WEP MPDU to decapsulate its payload. WEP extracts the IV and key identifier from the received MPDU. If a key-mapping key is present for the <TA,RA> pair, then this shall be used as the WEP key. Otherwise, the key identifier is extracted from the Key ID subfield of the WEP IV field in the received MPDU, identifying the default key to use.

WEP uses the constructed seed to decrypt the Data field of the WEP MPDU; this produces plaintext data and an ICV. Finally WEP recomputes the ICV and bit-wise compares it with the decrypted ICV from the MPDU. If the two are bit-wise identical, then WEP removes the IV and ICV from the MPDU, which is accepted as valid. If they differ in any bit position, WEP generates an error indication to MAC management. MSDUs with erroneous MPDUs (due to inability to decrypt) shall not be passed to LLC.

Figure 8-3 depicts a block diagram for WEP decapsulation. Unlike cryptographic encapsulation, the decapsulation steps shall be in the indicated order.

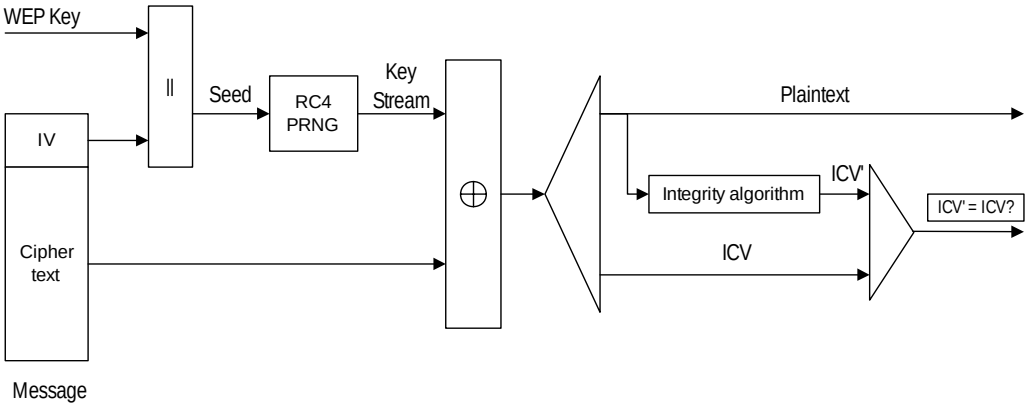



**Figure 8-3—WEP decapsulation block diagram**


Copyright © 2007 IEEE. All rights reserved.

### 8.2.2 Pre-RSNA authentication

#### 8.2.2.1 Overview

In an ESS, a non-AP STA and an AP must both complete an IEEE 802.11 authentication exchange prior to association. Such an exchange is optional in an independent BSS network.

All management frames of subtype Authentication shall be unicast, as IEEE 802.11 authentication is performed between pairs of STAs, i.e., broadcast/multicast authentication is not allowed. Management frames of subtype Deauthentication are advisory and may be sent as group addressed frames.

Shared Key authentication is deprecated and should not be implemented except for backward compatibility with pre-RSNA devices.

#### 8.2.2.2 Open System authentication

Open System authentication is a null authentication algorithm. Any STA requesting Open System authentication may be authenticated if dot11AuthenticationAlgorithm at the recipient STA is set to Open System authentication. A STA may decline to authenticate with another requesting STA. Open System authentication is the default authentication algorithm for pre-RSNA equipment.

Open System authentication utilizes a two-message authentication transaction sequence. The first message asserts identity and requests authentication. The second message returns the authentication result. If the result is "successful," the STAs shall be declared mutually authenticated.

In the description in 8.2.2.2.1 and 8.2.2.2.2, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder*. The specific items in each of the messages described in the following subclauses are defined in 7.2.3.10, Table 7-16, and Table 7-17.

##### 8.2.2.2.1 Open System authentication (first frame)

— Message type: Management
— Message subtype: Authentication
— Information items:
  • Authentication Algorithm Identification = "Open System"
  • STA Identity Assertion (in SA field of header)
  • Authentication transaction sequence number = 1
  • Authentication algorithm dependent information (none)
— Direction of message: From requester to responder

##### 8.2.2.2.2 Open System authentication (final frame)

— Message type: Management
— Message subtype: Authentication
— Information items:
  • Authentication Algorithm Identification = "Open System"
  • Authentication transaction sequence number = 2
  • Authentication algorithm dependent information (none)
  • The result of the requested authentication as defined in 7.3.1.9
— Direction of message: From responder to requester

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 209 of 1232

If dot11AuthenticationAlgorithm does not include the value “Open System,” the result code shall not take the value “successful.”

#### 8.2.2.3 Shared Key authentication

Shared Key authentication seeks to authenticate STAs as either a member of those who know a shared secret key or a member of those who do not.

Shared Key authentication can be used if and only if WEP has been selected.

This mechanism uses a shared key delivered to participating STAs via a secure channel that is independent of IEEE Std 802.11. This shared key is set in a write-only MIB attribute with the intent to keep the key value internal to the STA.

A STA shall not initiate a Shared Key authentication exchange unless its dot11PrivacyOptionImplemented attribute is true.

In the description in 8.2.2.3.1 through 8.2.2.3.5, the STA initiating the authentication exchange is referred to as the *requester*, and the STA to which the initial frame in the exchange is addressed is referred to as the *responder*. The specific items in each of the messages described in the following subclauses are defined in 7.2.3.10, Table 7-16, and Table 7-17.

##### 8.2.2.3.1 Shared Key authentication (first frame)

— Message type: Management

— Message subtype: Authentication

— Information items:

- STA Identity Assertion (in SA field of header)
- Authentication Algorithm Identification = “Shared Key”
- Authentication transaction sequence number = 1
- Authentication algorithm dependent information (none)

— Direction of message: From requester to responder

##### 8.2.2.3.2 Shared Key authentication (second frame)

Before sending the second frame in the Shared Key authentication sequence, the responder shall use WEP to generate a string of octets to be used as the authentication challenge text.

— Message type: Management

— Message subtype: Authentication

— Information items:

- Authentication Algorithm Identification = “Shared Key”
- Authentication transaction sequence number = 2
- Authentication algorithm dependent information = The authentication result
- The status code of the requested authentication as defined in 7.3.1.9.

  If the status code is not “successful,” this shall be the last frame of the transaction sequence; and the content of the challenge text field is unspecified.

  If the status code is “successful,” the following additional information items shall have valid contents:

  Authentication algorithm dependent information = The challenge text


Copyright © 2007 IEEE. All rights reserved.

This authentication result shall be of fixed length of 128 octets. The field shall be filled with octets generated by the WEP PRNG. The actual value of the challenge field is unimportant, but the value shall not be a static value.

— Direction of message: From responder to requester

#### 8.2.2.3.3 Shared Key authentication (third frame)

The requester shall copy the challenge text from the second frame into the third frame. The third frame shall be transmitted after cryptographic encapsulation by WEP, as defined in 8.2.1, using the shared key.

— Message type: Management

— Message subtype: Authentication

— Information items:

- Authentication Algorithm Identification = “Shared Key”
- Authentication transaction sequence number = 3
- Authentication algorithm dependent information = The challenge text from the second frame

— Direction of message: From requester to responder

#### 8.2.2.3.4 Shared Key authentication (final frame)

The responder shall WEP-decapsulate the third frame as described in 8.2.1. If the WEP ICV check is successful, the responder shall compare the decrypted contents of the Challenge Text field with the challenge text sent in second frame. If they are the same, then the responder shall respond with a successful status code in the final frame of the sequence. If the WEP ICV check fails or challenge text comparison fails, the responder shall respond with an unsuccessful status code in final frame.

— Message type: Management

— Message subtype: Authentication

— Information items:

- Authentication Algorithm Identification = “Shared Key”
- Authentication transaction sequence number = 4
- Authentication algorithm dependent information = The authentication result
- The status code of the requested authentication as defined in 7.3.1.9.

  This is a fixed length item with values “successful” and “unsuccessful.”

— Direction of message: From responder to requester

#### 8.2.2.3.5 Shared key MIB attributes

To transmit a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

**if** dot11PrivacyOptionImplemented is “false” **then**

the MMPDU is transmitted with a sequence of 0 octets in the Challenge Text field and a status code value of 13

**else**

the MMPDU is transmitted with a sequence of 128 octets generated using the WEP PRNG and a key whose value is unspecified and beyond the scope of this standard and a randomly chosen IV value (note that this will typically be selected by the same mechanism for choosing IV values for transmitted data MPDUs) in the Challenge Text field and a status code value of 0 (the IV used is immaterial and is not transmitted). Note that there are cryptographic issues involved in the choice of key/IV for this process as the challenge text is sent unencrypted and, therefore, provides a known output sequence from the PRNG.

**endif**

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 211 of 1232

To receive a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 2, the MAC shall operate according to the following decision tree:

**if** the Protected Frame subfield of the Frame Control field is 1 **then**
respond with a status code value of 15
**else**
**if** dot11PrivacyOptionImplemented is "true" **then**
**if** there is a mapping in dot11WEPKeyMappings matching the MSDU's TA **then**
**if** that key is null **then**
respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 15, and no Challenge Text field, without encrypting the contents of the frame
**else**
respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 0, and the identical Challenge Text field, encrypted using that key, and setting the Key ID subfield in the IV field to 0
**endif**
**else**
**if** dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null **then**
respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 15, and no Challenge Text field, without encrypting the contents of the frame
**else**
respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 0, and the identical Challenge Text field, WEP-encapsulating the frame under the key dot11WEPDefaultKeys[dot11WEPDefaultKeyID], and setting the Key ID subfield in the IV field to dot11WEPDefaultKeyID
**endif**
**endif**
**else**
respond with a frame whose Authentication Transaction Sequence Number field is 3 that contains the appropriate authentication algorithm number, a status code value of 13, and no Challenge Text field, without encrypting the contents of the frame
**endif**
**endif**

When receiving a management frame of subtype Authentication, with an Authentication Transaction Sequence Number field value of 3, the MAC shall operate according to the following decision tree:

**if** the Protected Frame subfield of the Frame Control field is 0 **then**
respond with a status code value of 15
**else**
**if** dot11PrivacyOptionImplemented is "true" **then**
**if** there is a mapping in dot11WEPKeyMappings matching the MSDU's TA **then**
**if** that key is null **then**
respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15 without encrypting the contents of the frame
**else**
WEP-decapsulate with that key, incrementing dot11WEPICVErrorCount and responding with a status code value of 15 if the ICV check fails


Copyright © 2007 IEEE. All rights reserved.

**endif**
**else**
**if** dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null **then**
respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15 without encrypting the contents of the frame
**else**
WEP-decapsulate with dot11WEPDefaultKeys[dot11WEPDefault-KeyID], incrementing dot11WEPICVErrorCount and responding with a status code value of 15 if the ICV check fails
**endif**
**endif**
**else**
respond with a frame whose Authentication Transaction Sequence Number field is 4 that contains the appropriate authentication algorithm number and a status code value of 15
**endif**
**endif**

The attribute dot11PrivacyInvoked shall not take the value of true if the attribute dot11PrivacyOption-Implemented is false. Setting dot11WEPKeyMappings to a value that includes more than dot11WEPKey-MappingLength entries is illegal and shall have an implementation-specific effect on the operation of the data confidentiality service. Note that dot11WEPKeyMappings may contain from zero to dot11WEPKey-MappingLength entries, inclusive.

The values of the attributes in the aPrivacygrp should not be changed during the authentication sequence, as unintended operation may result.

## 8.3 RSNA data confidentiality protocols

### 8.3.1 Overview

This standard defines two RSNA data confidentiality and integrity protocols: TKIP and CCMP. Implementation of CCMP shall be mandatory in all IEEE 802.11 devices claiming RSNA compliance. Implementation of TKIP is optional for an RSNA. A design aim for TKIP was that the algorithm should be implementable within the capabilities of most devices supporting only WEP, so that many such devices would be field-upgradeable by the supplier to support TKIP.

NOTE—Use of any of the data confidentiality algorithms depends on local policies. The data confidentiality and integrity mechanisms of TKIP are not as robust as those of CCMP. TKIP is designed to operate within the hardware limitations of a broad class of pre-RSNA devices. TKIP is suitable for firmware-only, hardware-compatible upgrade of fielded equipment. RSNA devices should only use TKIP when communicating with devices that are unable or not configured to communicate using CCMP.

### 8.3.2 Temporal Key Integrity Protocol (TKIP)

#### 8.3.2.1 TKIP overview

The TKIP is a cipher suite enhancing the WEP protocol on pre-RSNA hardware. TKIP modifies WEP as follows:

a) A transmitter calculates a keyed cryptographic message integrity code (MIC) over the MSDU SA and DA, the MSDU priority (see 8.3.2.3), and the MSDU plaintext data. TKIP appends the computed MIC to the MSDU data prior to fragmentation into MPDUs. The receiver verifies the MIC after decryption, ICV checking, and defragmentation of the MPDUs into an MSDU and

Copyright © 2007 IEEE. All rights reserved.

discards any received MSDUs with invalid MICs. TKIP's MIC provides a defense against forgery attacks.

b) Because of the design constraints of the TKIP MIC, it is still possible for an adversary to compromise message integrity; therefore, TKIP also implements countermeasures. The countermeasures bound the probability of a successful forgery and the amount of information an attacker can learn about a key.

c) TKIP uses a per-MPDU TKIP sequence counter (TSC) to sequence the MPDUs it sends. The receiver drops MPDUs received out of order, i.e., not received with increasing sequence numbers. This provides replay protection. TKIP encodes the TSC value from the sender to the receiver as a WEP IV and extended IV.

d) TKIP uses a cryptographic mixing function to combine a temporal key, the TA, and the TSC into the WEP seed. The receiver recovers the TSC from a received MPDU and utilizes the mixing function to compute the same WEP seed needed to correctly decrypt the MPDU. The key mixing function is designed to defeat weak-key attacks against the WEP key.

TKIP defines additional MIB variables; see Annex D.

#### 8.3.2.1.1 TKIP cryptographic encapsulation

TKIP enhances the WEP cryptographic encapsulation with several additional functions, as depicted in Figure 8-4.

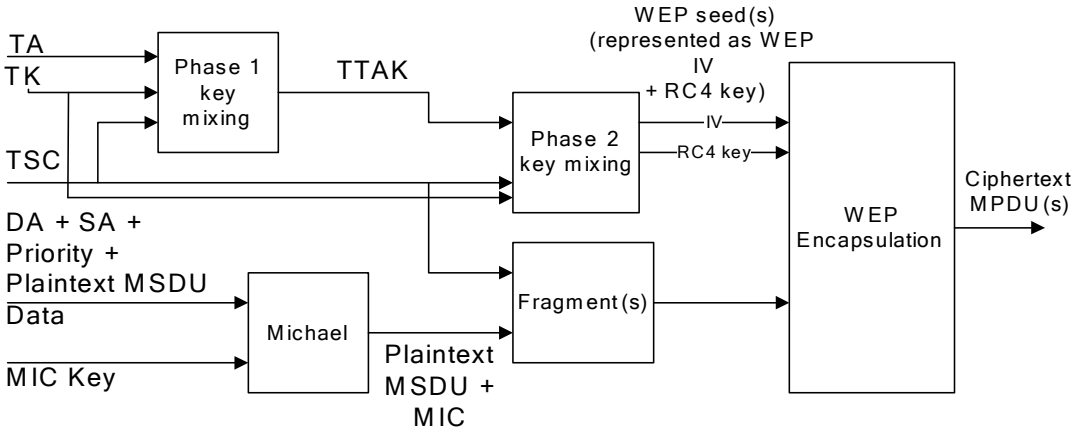



**Figure 8-4—TKIP encapsulation block diagram**

a) TKIP MIC computation protects the MSDU Data field and corresponding SA, DA, and Priority fields. The computation of the MIC is performed on the ordered concatenation of the SA, DA, Priority, and MSDU Data fields. The MIC is appended to the MSDU Data field. TKIP discards any MIC padding prior to appending the MIC.

b) If needed, IEEE Std 802.11 fragments the MSDU with MIC into one or more MPDUs. TKIP assigns a monotonically increasing TSC value to each MPDU, taking care that all the MPDUs generated from the same MSDU have the same value of extended IV (see 8.3.2.2).

c) For each MPDU, TKIP uses the key mixing function to compute the WEP seed.

d) TKIP represents the WEP seed as a WEP IV and ARC4 key and passes these with each MPDU to WEP for generation of the ICV (see 7.1.3.6), and for encryption of the plaintext MPDU, including all or part of the MIC, if present. WEP uses the WEP seed as a WEP default key, identified by a key identifier associated with the temporal key.


Copyright © 2007 IEEE. All rights reserved.

NOTE—When the TSC space is exhausted, the choices available to an implementation are to replace the temporal key with a new one or to end communications. Reuse of any TSC value compromises already sent traffic. Note that retransmitted MPDUs reuse the TSC without any compromise of security. The TSC is large enough, however, that TSC space exhaustion should not be an issue.

In Figure 8-4, the TKIP-mixed transmit address and key (TTAK) denotes the intermediate key produced by Phase 1 of the TKIP mixing function (see 8.3.2.5).

#### 8.3.2.1.2 TKIP decapsulation

TKIP enhances the WEP decapsulation process with the following additional steps:

a) Before WEP decapsulates a received MPDU, TKIP extracts the TSC sequence number and key identifier from the WEP IV and the extended IV. TKIP discards a received MPDU that violates the sequencing rules (see 8.3.2.6) and otherwise uses the mixing function to construct the WEP seed.

b) TKIP represents the WEP seed as a WEP IV and ARC4 key and passes these with the MPDU to WEP for decapsulation.

c) If WEP indicates the ICV check succeeded, the implementation reassembles the MPDU into an MSDU. If the MSDU defragmentation succeeds, the receiver verifies the TKIP MIC. If MSDU defragmentation fails, then the MSDU is discarded.

d) The MIC verification step recomputes the MIC over the MSDU SA, DA, Priority, and MSDU Data fields (but not the TKIP MIC field). The calculated TKIP MIC result is then compared bit-wise against the received MIC.

e) If the received and the locally computed MIC values are identical, the verification succeeds, and TKIP shall deliver the MSDU to the upper layer. If the two differ, then the verification fails; the receiver shall discard the MSDU and shall engage in appropriate countermeasures.

Figure 8-5 depicts this process.

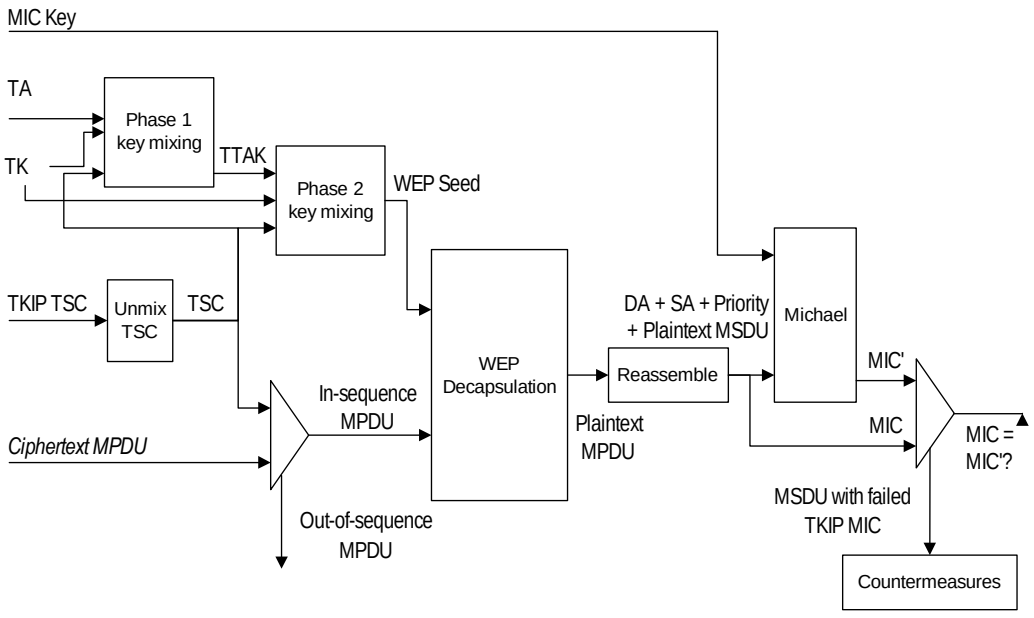



**Figure 8-5—TKIP decapsulation block diagram**

Copyright © 2007 IEEE. All rights reserved.

### 8.3.2.2 TKIP MPDU formats

TKIP reuses the pre-RSNA WEP MPDU format. It extends the MPDU by 4 octets to accommodate an extension to the WEP IV, denoted by the Extended IV field, and extends the MSDU format by 8 octets to accommodate the new MIC field. TKIP inserts the Extended IV field immediately after the WEP IV field and before the encrypted data. TKIP appends the MIC to the MSDU Data field; the MIC becomes part of the encrypted data.

Once the MIC is appended to the MSDU data, the added MIC octets are considered part of the MSDU for subsequent fragmentation.

Figure 8-6 depicts the layout of the encrypted MPDU when using TKIP. Note the figure only depicts the case when the MSDU can be encapsulated in a single MPDU.

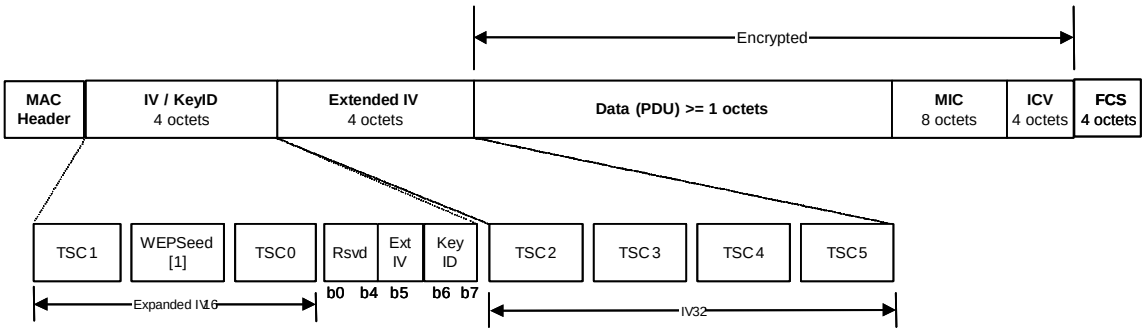



**Figure 8-6—Construction of expanded TKIP MPDU**

The ExtIV bit in the Key ID octet indicates the presence or absence of an extended IV. If the ExtIV bit is 0, only the nonextended IV is transferred. If the ExtIV bit is 1, an extended IV of 4 octets follows the original IV. For TKIP the ExtIV bit shall be set, and the Extended IV field shall be supplied. The ExtIV bit shall be 0 for WEP frames. The Key ID field shall be set to the key index supplied by the MLME-SETKEYS.request primitive for the key used in cryptographic encapsulation of the frame.

TSC5 is the most significant octet of the TSC, and TSC0 is the least significant. Octets TSC0 and TSC1 form the IV sequence number and are used with the TKIP Phase 2 key mixing. Octets TSC2–TSC5 are used in the TKIP Phase 1 key hashing and are in the Extended IV field. When the lower 16-bit sequence number rolls over (0xFFFF → 0x0000), the extended IV value, i.e., the upper 32 bits of the entire 48-bit TSC, shall be incremented by 1.

NOTE—The rationale for this construction is as follows:

— Aligning on word boundaries eases implementation on legacy devices.

— Adding 4 octets of extended IV eliminates TSC exhaustion as a reason to rekey.

— Key ID octet changes. Bit 5 indicates that an extended IV is present. The receiver/transmitter interprets the 4 octets following the Key ID as the extended IV. The receiving/transmitting STA also uses the value of octets TSC0 and TSC1 to detect that the cached TTAK must be updated.

The Extended IV field shall not be encrypted.

WEPSeed[1] is not used to construct the TSC, but is set to (TSC1 | 0x20) & 0x7f.

TKIP shall encrypt all the MPDUs generated from one MSDU under the same temporal key.


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.2.3 TKIP MIC

Flaws in the IEEE 802.11 WEP design cause it to fail to meet its goal of protecting data traffic content from casual eavesdroppers. Among the most significant WEP flaws is the lack of a mechanism to defeat message forgeries and other active attacks. To defend against active attacks, TKIP includes a MIC, named Michael. This MIC offers only weak defenses against message forgeries, but it constitutes the best that can be achieved with the majority of legacy hardware. TKIP uses different MIC keys depending on the direction of the transfer as described in 8.6.1 and 8.6.2.

Annex H contains an implementation of the TKIP MIC. It also provides test vectors for the MIC.

##### 8.3.2.3.1 Motivation for the TKIP MIC

Before defining the details of the MIC, it is useful to review the context in which this mechanism operates. Active attacks enabled by the original WEP design include the following:

— Bit-flipping attacks

— Data (payload) truncation, concatenation, and splicing

— Fragmentation attacks

— Iterative guessing attacks against the key

— Redirection by modifying the MPDU DA or RA field

— Impersonation attacks by modifying the MPDU SA or TA field

The MIC makes it more difficult for any of these attacks to succeed.

All of these attacks remain at the MPDU level with the TKIP MIC. The MIC, however, applies to the MSDU, so it blocks successful MPDU-level attacks. TKIP applies the MIC to the MSDU at the transmitter and verifies it at the MSDU level at the receiver. If a MIC check fails at the MSDU level, the implementation shall discard the MSDU and invoke countermeasures (see 8.3.2.4).

Informative Figure 8-7 depicts different peer layers communicating.

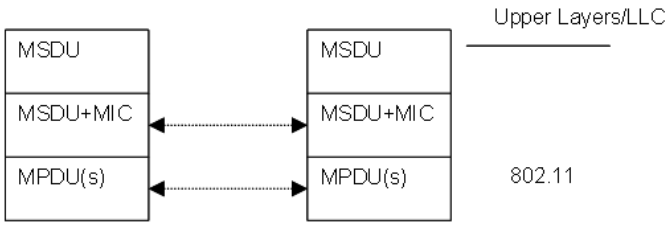



**Figure 8-7—TKIP MIC relation to IEEE 802.11 processing (informative)**

This figure depicts an architecture where the MIC is logically appended to the raw MSDU in response to the MA-UNITDATA.request primitive. The TKIP MIC is computed over

— The MSDU DA

— The MSDU SA

— The MSDU Priority

— The entire unencrypted MSDU data (payload)

Copyright © 2007 IEEE. All rights reserved.

The DA field, SA field, three reserved octets, and a 1-octet Priority field are used only for calculating the MIC. The Priority field refers to the priority parameter of the MA-UNITDATA.request service primitive. The fields in Figure 8-8 are treated as an octet stream using the conventions described in 7.1.1.

| 6 | 6 | 1 | 3 | M | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | octets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DA | SA | Priority | 0 | Data | M 0 | M 1 | M 2 | M 3 | M 4 | M 5 | M 6 | M 7 | |

**Figure 8-8—TKIP MIC processing format**

TKIP appends the MIC at the end of the MSDU payload. The MIC is 8 octets in size for Michael. The IEEE 802.11 MAC then applies its normal processing to transmit this MSDU-with-MIC as a sequence of one or more MPDUs. In other words, the MSDU-with-MIC can be partitioned into one or more MPDUs, the WEP ICV is calculated over each MPDU, and the MIC can even be partitioned to lie in two MPDUs after fragmentation. The TKIP MIC augments, but does not replace, the WEP ICV. Because the TKIP MIC is a weak construction, TKIP protects the MIC with encryption, which makes TKIP MIC forgeries more difficult. The WEP ICV helps to prevent false detection of MIC failures that would cause countermeasures to be invoked.

The receiver reverses this procedure to reassemble the MSDU; and, after the MSDU has been logically reassembled, the IEEE 802.11 MAC verifies the MIC prior to delivery of the MSDU to upper layers. If the MIC validation succeeds, the MAC delivers the MSDU. If the MIC validation fails, the MAC shall discard the MSDU and invoke countermeasures (see 8.3.2.4).

NOTE—TKIP calculates the MIC over the MSDU rather than the MPDU, because doing so increases the implementation flexibility with pre-existing WEP hardware.

It should be noted that a MIC alone cannot provide complete forgery protection, as it cannot defend against replay attacks. Therefore, TKIP provides replay detection by TSC sequencing and ICV validation. Furthermore, if TKIP is utilized with a GTK, an insider STA can masquerade as any other STA belonging to the group.

#### 8.3.2.3.2 Definition of the TKIP MIC

Michael generates a 64-bit MIC. The Michael key consists of 64 bits, represented as an 8-octet sequence, $k_0...k_7$. This is converted to two 32-bit words, $K_0$ and $K_1$. Throughout this subclause, all conversions between octets and 32-bit words shall use the little endian conventions, given in 7.1.1.

Michael operates on each MSDU including the Priority field, 3 reserved octets, SA field, and DA field. An MSDU consists of octets $m_0...m_{n-1}$ where $n$ is the number of MSDU octets, including SA, DA, Priority, and Data fields. The message is padded at the end with a single octet with value 0x5a, followed by between 4 and 7 zero octets. The number of zero octets is chosen so that the overall length of the padded MSDU is a multiple of four. The padding is not transmitted with the MSDU; it is used to simplify the computation over the final block. The MSDU is then converted to a sequence of 32-bit words $M_0 ...M_{N-1}$, where $N = \lceil (n+5)/4 \rceil$, and where $\lceil a \rceil$ means to round $a$ up to the nearest integer. By construction, $M_{N-1} = 0$ and $M_{N-2} \neq 0$.

The MIC value is computed iteratively starting with the key value ($K_0$ and $K_1$) and applying a block function $b$ for every message word, as shown in Figure 8-9. The algorithm loop runs a total of $N$ times ($i$ takes on the values 0 to $N$–1 inclusive), where $N$ is as above, the number of 32-bit words composing the padded MSDU. The algorithm results in two words ($l$ and $r$), which are converted to a sequence of 8 octets using the least-significant-octet-first convention:

— M0 = $l$ & 0xff

— M1 = ($l$/0x100) & 0xff


Copyright © 2007 IEEE. All rights reserved.

— M2 = (*l*/0x10000) & 0xff

— M3 = (*l*/0x1000000) & 0xff

— M4 = *r* & 0xff

— M5 = (*r*/0x100) & 0xff

— M6 = (*r*/0x10000) & 0xff

— M7 = (*r*/0x1000000) & 0xff

This is the MIC value. The MIC value is appended to the MSDU as data to be sent.

```
Input: Key (K0, K1) and padded MSDU (represented as 32-bit words) M0...MN-1
Output: MIC value (V0, V1)
        MICHAEL((K0, K1) , (M0,...,MN))
        (l,r) ← (K0, K1)
        for i = 0 to N–1 do
                l ← l ⊕ M-i
                (l, r) ← b(l, r)
        return (l,r)
```

**Figure 8-9—Michael message processing**

Figure 8-10 defines the Michael block function *b*. It is a Feistel-type construction with alternating additions and XOR operations. It uses <<< to denote the rotate-left operator on 32-bit values, >>> for the rotate-right operator, and XSWAP for a function that swaps the position of the 2 least significant octets. It also uses the position of the two most significant octets in a word.

```
Input: (l,r)
Output: (l,r)
b(L,R)
        r ← r ⊕ (l <<< 17)
        l ← (l + r) mod 232
        r ← r ⊕ XSWAP(l)
        l ← (l + r) mod 232
        r ← r ⊕ (l <<< 3)
        l ← (l + r) mod 232
        r ← r ⊕ (l >>> 2)
        l ← (l + r) mod 232
        return (l, r)
```

**Figure 8-10—Michael block function**

### 8.3.2.4 TKIP countermeasures procedures

The TKIP MIC trades off security in favor of implementability on pre-RSNA devices. Michael provides only weak protection against active attacks. A failure of the MIC in a received MSDU indicates a probable active attack. A successful attack against the MIC would mean an attacker could inject forged data frames and perform further effective attacks against the encryption key itself. If TKIP implementation detects a probable active attack, TKIP shall take countermeasures as specified in this subclause. These countermeasures accomplish the following goals:

— MIC failure events *should* be logged as a security-relevant matter. A MIC failure is an almost certain indication of an active attack and warrants a follow-up by the system administrator.

— The rate of MIC failures *must* be kept below two per minute. This implies that STAs and APs detecting two MIC failure events within 60 s must disable all receptions using TKIP for a period of 60 s. The slowdown makes it difficult for an attacker to make a large number of forgery attempts in a short time.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 219 of 1232

— As an additional security feature, the PTK and, in the case of the Authenticator, the GTK should be changed.

Before verifying the MIC, the receiver shall check the FCS, ICV, and TSC for all related MPDUs. Any MPDU that has an invalid FCS, an incorrect ICV, or a TSC value that is less than or equal to the TSC replay counter shall be discarded before checking the MIC. This avoids unnecessary MIC failure events. Checking the TSC before the MIC makes countermeasure-based denial-of-service attacks harder to perform. While the FCS and ICV mechanisms are sufficient to detect noise, they are insufficient to detect active attacks. The FCS and ICV provide error detection, but not integrity protection.

A single counter or timer shall be used to log MIC failure events. These failure events are defined as follows:

— For an Authenticator:
  — Detection of a MIC failure on a received unicast frame.
  — Receipt of Michael MIC Failure Report frame.
— For a Supplicant:
  — Detection of a MIC failure on a received unicast or broadcast/multicast frame.
  — Attempt to transmit a Michael MIC Failure Report frame.

The number of MIC failures is accrued independent of the particular key context. Any single MIC failure, whether detected by the Supplicant or the Authenticator and whether resulting from a group MIC key failure or a pairwise MIC key failure, shall be treated as cause for a MIC failure event.

The Supplicant uses a single Michael MIC Failure Report frame to report a MIC failure event to the Authenticator. A Michael MIC Failure Report is an EAPOL-Key frame with the following Key Information field bits set to 1: MIC bit, Error bit, Request bit, Secure bit. The Supplicant protects this message with the current PTK; the Supplicant uses the KCK portion of the PTK to compute the IEEE 802.1X EAPOL MIC.

MLME-MICHEALMICFAILURE.indication primitive is used by the IEEE 802.11 MAC to attempt to indicate a MIC failure to the local IEEE 802.1X Supplicant or Authenticator. MLME-EAPOL.request primitive is used by the Supplicant to send the EAPOL-Key frame containing the Michael MIC Failure Report. MLME-EAPOL.confirm primitive indicates to the Supplicant when an IEEE 802.11 MAC ACK has been received for this EAPOL-Key frame.

The first MIC failure shall be logged, and a timer initiated to enable enforcement of the countermeasures. If the MIC failure event is detected by the Supplicant, it shall also report the event to the AP by sending a Michael MIC Failure Report frame.

If a subsequent MIC failure occurs within 60 s of the most recent previous failure, then a STA whose IEEE 802.1X entity has acted as a Supplicant shall deauthenticate (as defined in 11.3.1.3) itself or deauthenticate all the STAs with a security association if its IEEE 802.1X entity acted as an Authenticator. For an IBSS STA, both Supplicant and Authenticator actions shall be taken. Furthermore, the device shall not receive or transmit any TKIP-encrypted data frames, and shall not receive or transmit any unencrypted data frames other than IEEE 802.1X messages, to or from any peer for a period of at least 60 s after it detects the second failure. If the device is an AP, it shall disallow new associations using TKIP during this 60 s period; at the end of the 60 s period, the AP shall resume normal operations and allow STAs to (re)associate. If the device is an IBSS STA, it shall disallow any new security associations using TKIP during this 60 s period. If the device is a Supplicant, it shall first send a Michael MIC Failure Report frame prior to revoking its PTKSA and deauthenticating itself.

The `aMICFailTime` attribute shall contain the `sysUpTime` value at the time the MIC failure was logged.


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.2.4.1 TKIP countermeasures for an Authenticator

The countermeasures used by an Authenticator are depicted in Figure 8-11 and described as follows:

a) For an Authenticator's STA that receives a frame with a MIC error,
   1) Discard the frame.
   2) Increment the MIC failure counter, dot11RSNAStatsTKIPLocalMICFailures.
   3) Generate a MLME-MICHAELMICFAILURE.indication primitive.

b) For an Authenticator that receives a MLME-MICHAELMICFAILURE.indication primitive or a Michael MIC Failure Report frame,
   1) If it is a Michael MIC Failure Report frame, then increment dot11RSNAStatsTKIP-RemoteMICFailures.
   2) If this is the first MIC failure within the past 60 s, initialize the countermeasures timer.
   3) If less than 60 s have passed since the most recent previous MIC failure, the Authenticator shall deauthenticate and delete all PTKSAs for all STAs using TKIP. If the current GTKSA uses TKIP, that GTKSA shall be discarded, and a new GTKSA constructed, but not used for 60 s. The Authenticator shall refuse the construction of new PTKSAs using TKIP as one or more of the ciphers for 60 s. At the end of this period, the MIC failure counter and timer shall be reset, and creation of PTKSAs accepted as usual.
   4) If the Authenticator is using IEEE 802.1X authentication, the Authenticator shall transition the state of the IEEE 802.1X Authenticator state machine to the INITIALIZE state. This will restart the IEEE 802.1X state machine. If the Authenticator is instead using PSKs, this step is omitted.

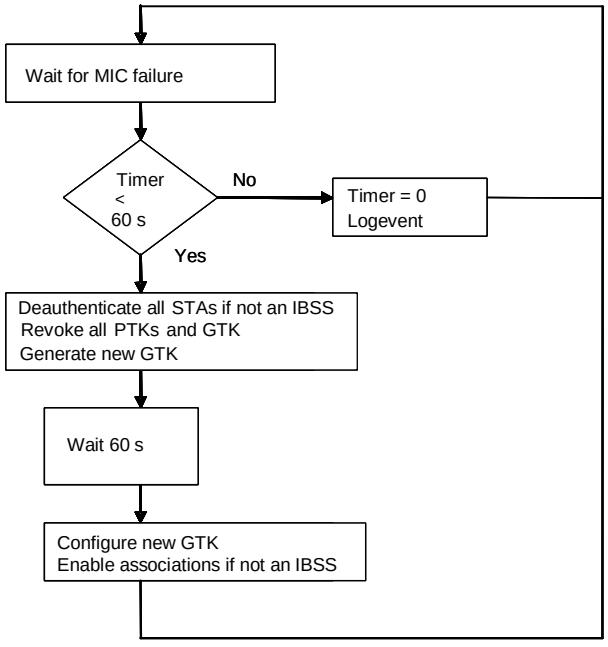



**Figure 8-11—Authenticator MIC countermeasures**

Note that a Supplicant's STA may deauthenticate with a reason code of MIC failure if it is an ESS STA. The Authenticator shall not log the deauthenticate as a MIC failure event to prevent denial-of-service attacks through deauthentications. The Supplicant's STA must report the MIC failure event through the Michael MIC Failure Report frame in order for the AP to log the event.

Copyright © 2007 IEEE. All rights reserved.

The requirement to deauthenticate all STAs using TKIP will include those using CCMP as a pairwise cipher if they are also using TKIP as the group cipher.

#### 8.3.2.4.2 TKIP countermeasures for a Supplicant

The countermeasures used by a Supplicant are depicted in Figure 8-12 and described as follows:

a) For a Supplicant's STA that receives a frame with a MIC error,
   1) Increment the MIC failure counter, dot11RSNAStatsTKIPLocalMICFailures.
   2) Discard the offending frame.
   3) Generate a MLME-MICHAELMICFAILURE.indication primitive.

b) For a Supplicant that receives an MLME-MICHAELMICFAILURE.indication primitive from its STA,
   1) Send a Michael MIC Failure Report frame to the AP.
   2) If this is the first MIC failure within the past 60 s, initialize the countermeasures timer.
   3) If less than 60 s have passed since the most recent previous MIC failure, delete the PTKSA and GTKSA. Deauthenticate from the AP and wait for 60 s before (re)establishing a TKIP association with the same AP. A TKIP association is any IEEE 802.11 association that uses TKIP for its pairwise or group cipher suite.

c) If a non-AP STA receives a deauthenticate frame with the reason code "MIC failure," it cannot be certain that the frame has not been forged, as it does not contain a MIC. The STA may attempt association with this, or another, AP. If the frame was genuine, then it is probable that attempts to associate with the same AP requesting the use of TKIP will fail because the AP will be conducting countermeasures.

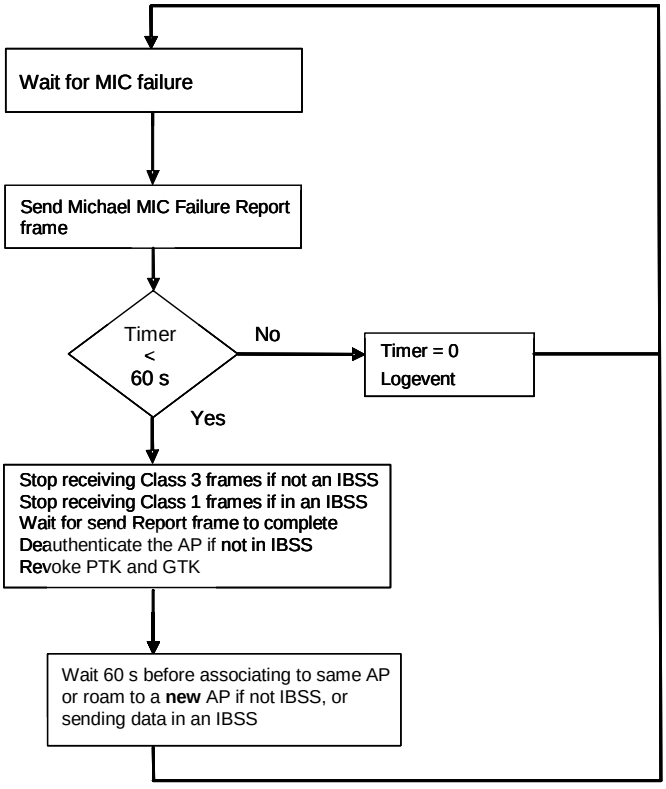



**Figure 8-12—Supplicant MIC countermeasures**


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.2.5 TKIP mixing function

Annex H defines a C-language reference implementation of the TKIP mixing function. It also provides test vectors for the mixing function.

The mixing function has two phases. Phase 1 mixes the appropriate temporal key (pairwise or group) with the TA and TSC. A STA may cache the output of this phase to reuse with subsequent MPDUs associated with the same temporal key and TA. Phase 2 mixes the output of Phase 1 with the TSC and temporal key (TK) to produce the WEP seed, also called the *per-frame key*. The WEP seed may be precomputed before it is used. The two-phase process may be summarized as follows:

```
TTAK := Phase1 (TK, TA, TSC)
WEP seed := Phase2 (TTAK, TK, TSC)
```

##### 8.3.2.5.1 S-Box

Both Phase 1 and Phase 2 rely on an S-box, defined in this subclause. The S-box substitutes one 16-bit value with another 16-bit value. This function may be implemented as a table look up.

NOTE—The S-box is a nonlinear substitution. The table look-up can be organized as either a single table with 65 536 entries and a 16-bit index (128K octets of table) or two tables with 256 entries and an 8-bit index (1024 octets for both tables). When the two smaller tables are used, the high-order octet is used to obtain a 16-bit value from one table, the low-order octet is used to obtain a 16-bit value from the other table, and the S-box output is the XOR (⊕) of the two 16-bit values. The second S-box table is an octet-swapped replica of the first.

```
#define _S_(v16)     (Sbox[0][Lo8(v16)] ^ Sbox[1][Hi8(v16)])

/* 2-byte by 2-byte subset of the full AES S-box table */
const u16b Sbox[2][256]=       /* Sbox for hash (can be in ROM)     */
{ {
   0xC6A5,0xF884,0xEE99,0xF68D,0xFF0D,0xD6BD,0xDEB1,0x9154,
   0x6050,0x0203,0xCEA9,0x567D,0xE719,0xB562,0x4DE6,0xEC9A,
   0x8F45,0x1F9D,0x8940,0xFA87,0xEF15,0xB2EB,0x8EC9,0xFB0B,
   0x41EC,0xB367,0x5FFD,0x45EA,0x23BF,0x53F7,0xE496,0x9B5B,
   0x75C2,0xE11C,0x3DAE,0x4C6A,0x6C5A,0x7E41,0xF502,0x834F,
   0x685C,0x51F4,0xD134,0xF908,0xE293,0xAB73,0x6253,0x2A3F,
   0x080C,0x9552,0x4665,0x9D5E,0x3028,0x37A1,0x0A0F,0x2FB5,
   0x0E09,0x2436,0x1B9B,0xDF3D,0xCD26,0x4E69,0x7FCD,0xEA9F,
   0x121B,0x1D9E,0x5874,0x342E,0x362D,0xDCB2,0xB4EE,0x5BFB,
   0xA4F6,0x764D,0xB761,0x7DCE,0x527B,0xDD3E,0x5E71,0x1397,
   0xA6F5,0xB968,0x0000,0xC12C,0x4060,0xE31F,0x79C8,0xB6ED,
   0xD4BE,0x8D46,0x67D9,0x724B,0x94DE,0x98D4,0xB0E8,0x854A,
   0xBB6B,0xC52A,0x4FE5,0xED16,0x86C5,0x9AD7,0x6655,0x1194,
   0x8ACF,0xE910,0x0406,0xFE81,0xA0F0,0x7844,0x25BA,0x4BE3,
   0xA2F3,0x5DFE,0x80C0,0x058A,0x3FAD,0x21BC,0x7048,0xF104,
   0x63DF,0x77C1,0xAF75,0x4263,0x2030,0xE51A,0xFD0E,0xBF6D,
   0x814C,0x1814,0x2635,0xC32F,0xBEE1,0x35A2,0x88CC,0x2E39,
   0x9357,0x55F2,0xFC82,0x7A47,0xC8AC,0xBAE7,0x322B,0xE695,
   0xC0A0,0x1998,0x9ED1,0xA37F,0x4466,0x547E,0x3BAB,0x0B83,
   0x8CCA,0xC729,0x6BD3,0x283C,0xA779,0xBCE2,0x161D,0xAD76,
   0xDB3B,0x6456,0x744E,0x141E,0x92DB,0x0C0A,0x486C,0xB8E4,
   0x9F5D,0xBD6E,0x43EF,0xC4A6,0x39A8,0x31A4,0xD337,0xF28B,
   0xD532,0x8B43,0x6E59,0xDAB7,0x018C,0xB164,0x9CD2,0x49E0,
   0xD8B4,0xACFA,0xF307,0xCF25,0xCAAF,0xF48E,0x47E9,0x1018,
   0x6FD5,0xF088,0x4A6F,0x5C72,0x3824,0x57F1,0x73C7,0x9751,
   0xCB23,0xA17C,0xE89C,0x3E21,0x96DD,0x61DC,0x0D86,0x0F85,
   0xE090,0x7C42,0x71C4,0xCCAA,0x90D8,0x0605,0xF701,0x1C12,
   0xC2A3,0x6A5F,0xAEF9,0x69D0,0x1791,0x9958,0x3A27,0x27B9,
   0xD938,0xEB13,0x2BB3,0x2233,0xD2BB,0xA970,0x0789,0x33A7,
```

Copyright © 2007 IEEE. All rights reserved.

```
    0x2DB6,0x3C22,0x1592,0xC920,0x8749,0xAAFF,0x5078,0xA57A,
    0x038F,0x59F8,0x0980,0x1A17,0x65DA,0xD731,0x84C6,0xD0B8,
    0x82C3,0x29B0,0x5A77,0x1E11,0x7BCB,0xA8FC,0x6DD6,0x2C3A,
   },
   {  /* second half of table is byte-reversed version of first! */
    0xA5C6,0x84F8,0x99EE,0x8DF6,0x0DFF,0xBDD6,0xB1DE,0x5491,
    0x5060,0x0302,0xA9CE,0x7D56,0x19E7,0x62B5,0xE64D,0x9AEC,
    0x458F,0x9D1F,0x4089,0x87FA,0x15EF,0xEBB2,0xC98E,0x0BFB,
    0xEC41,0x67B3,0xFD5F,0xEA45,0xBF23,0xF753,0x96E4,0x5B9B,
    0xC275,0x1CE1,0xAE3D,0x6A4C,0x5A6C,0x417E,0x02F5,0x4F83,
    0x5C68,0xF451,0x34D1,0x08F9,0x93E2,0x73AB,0x5362,0x3F2A,
    0x0C08,0x5295,0x6546,0x5E9D,0x2830,0xA137,0x0F0A,0xB52F,
    0x090E,0x3624,0x9B1B,0x3DDF,0x26CD,0x694E,0xCD7F,0x9FEA,
    0x1B12,0x9E1D,0x7458,0x2E34,0x2D36,0xB2DC,0xEEB4,0xFB5B,
    0xF6A4,0x4D76,0x61B7,0xCE7D,0x7B52,0x3EDD,0x715E,0x9713,
    0xF5A6,0x68B9,0x0000,0x2CC1,0x6040,0x1FE3,0xC879,0xEDB6,
    0xBED4,0x468D,0xD967,0x4B72,0xDE94,0xD498,0xE8B0,0x4A85,
    0x6BBB,0x2AC5,0xE54F,0x16ED,0xC586,0xD79A,0x5566,0x9411,
    0xCF8A,0x10E9,0x0604,0x81FE,0xF0A0,0x4478,0xBA25,0xE34B,
    0xF3A2,0xFE5D,0xC080,0x8A05,0xAD3F,0xBC21,0x4870,0x04F1,
    0xDF63,0xC177,0x75AF,0x6342,0x3020,0x1AE5,0x0EFD,0x6DBF,
    0x4C81,0x1418,0x3526,0x2FC3,0xE1BE,0xA235,0xCC88,0x392E,
    0x5793,0xF255,0x82FC,0x477A,0xACC8,0xE7BA,0x2B32,0x95E6,
    0xA0C0,0x9819,0xD19E,0x7FA3,0x6644,0x7E54,0xAB3B,0x830B,
    0xCA8C,0x29C7,0xD36B,0x3C28,0x79A7,0xE2BC,0x1D16,0x76AD,
    0x3BDB,0x5664,0x4E74,0x1E14,0xDB92,0x0A0C,0x6C48,0xE4B8,
    0x5D9F,0x6EBD,0xEF43,0xA6C4,0xA839,0xA431,0x37D3,0x8BF2,
    0x32D5,0x438B,0x596E,0xB7DA,0x8C01,0x64B1,0xD29C,0xE049,
    0xB4D8,0xFAAC,0x07F3,0x25CF,0xAFCA,0x8EF4,0xE947,0x1810,
    0xD56F,0x88F0,0x6F4A,0x725C,0x2438,0xF157,0xC773,0x5197,
    0x23CB,0x7CA1,0x9CE8,0x213E,0xDD96,0xDC61,0x860D,0x850F,
    0x90E0,0x427C,0xC471,0xAACC,0xD890,0x0506,0x01F7,0x121C,
    0xA3C2,0x5F6A,0xF9AE,0xD069,0x9117,0x5899,0x273A,0xB927,
    0x38D9,0x13EB,0xB32B,0x3322,0xBBD2,0x70A9,0x8907,0xA733,
    0xB62D,0x223C,0x9215,0x20C9,0x4987,0xFFAA,0x7850,0x7AA5,
    0x8F03,0xF859,0x8009,0x171A,0xDA65,0x31D7,0xC684,0xB8D0,
    0xC382,0xB029,0x775A,0x111E,0xCB7B,0xFCA8,0xD66D,0x3A2C,
   }
};
```

#### 8.3.2.5.2 Phase 1 Definition

The inputs to Phase 1 of the temporal key mixing function shall be a temporal key (*TK*), the TA, and the TSC. The temporal key shall be 128 bits in length. Only the 32 MSBs of the TSC and all of the temporal key are used in Phase 1. The output, *TTAK*, shall be 80 bits in length and is represented by an array of 16-bit values: $TTAK_0$ $TTAK_1$ $TTAK_2$ $TTAK_3$ $TTAK_4$.

The description of the Phase 1 algorithm treats all of the following values as arrays of 8-bit values: $TA_0..TA_5$, $TK_0..TK_{15}$. The *TA* octet order is represented according to the conventions from 7.1.1, and the first 3 octets represent the OUI.

The XOR (⊕) operation, the bit-wise-and (&) operation, and the addition (+) operation are used in the Phase 1 specification. A loop counter, *i*, and an array index temporary variable, *j*, are also employed.

One function, *Mk*16, is used in the definition of Phase 1. The function *Mk*16 constructs a 16-bit value from two 8-bit inputs as $Mk16(X,Y) = (256 \cdot X)+Y$.


Copyright © 2007 IEEE. All rights reserved.

Two steps make up the Phase 1 algorithm. The first step initializes *TTAK* from *TSC* and *TA*. The second step uses an S-box to iteratively mix the keying material into the 80-bit *TTAK*. The second step sets the PHASE1_LOOP_COUNT to 8.

```
Input: transmit address TA0... TA5, Temporal Key TK0..TK15, and TSC0..TSC5
Output: intermediate key TTAK0..TTAK4
PHASE1-KEY-MIXING(TA0... TA5, TK0..TK15, TSC0..TSC5)
    PHASE1_STEP1:
    TTAK0 ← MK16(TSC3, TSC2)
    TTAK1 ← MK16(TSC5, TSC4)
    TTAK2 ← Mk16(TA1,TA0)
    TTAK3 ← Mk16(TA3,TA2)
    TTAK4 ← Mk16(TA5,TA4)
    PHASE1_STEP2:
    for i = 0 to PHASE1_LOOP_COUNT-1
        j ← 2·(i & 1)
        TTAK0 ← TTAK0 + S[TTAK4 ⊕ Mk16(TK1+j,TK0+j)]
        TTAK1 ← TTAK1 + S[TTAK0 ⊕ Mk16(TK5+j,TK4+j)]
        TTAK2 ← TTAK2 + S[TTAK1 ⊕ Mk16(TK9+j,TK8+j)]
        TTAK3 ← TTAK3 + S[TTAK2 ⊕ Mk16(TK13+j,TK12+j)]
        TTAK4 ← TTAK4 + S[TTAK3 ⊕ Mk16(TK1+j,TK0+j)] + i
```

**Figure 8-13—Phase 1 key mixing**

NOTE 1—The TA is mixed into the temporal key in Phase 1 of the hash function. Implementations can achieve a significant performance improvement by caching the output of Phase 1. The Phase 1 output is the same for $2^{16}$ = 65 536 consecutive frames from the same temporal key and TA. Consider the simple case where a STA communicates only with an AP. The STA will perform Phase 1 using its own address, and the *TTAK* will be used to protect traffic sent to the AP. The STA will perform Phase 1 using the AP address, and it will be used to unwrap traffic received from the AP.

NOTE 2—The cached *TTAK* from Phase 1 will need to be updated when the lower 16 bits of the TSC wrap and the upper 32 bits need to be updated.

#### 8.3.2.5.3 Phase 2 definition

The inputs to Phase 2 of the temporal key mixing function shall be the output of Phase 1 (*TTAK*) together with the temporal key and the TSC. The *TTAK* is 80-bits in length. Only the 16 LSBs of the TSC are used in Phase 2. The temporal key is 128 bits. The output is the WEP seed, which is a per-frame key, and is 128 bits in length. The constructed WEP seed has an internal structure conforming to the WEP specification. In other words, the first 24 bits of the WEP seed shall be transmitted in plaintext as the WEP IV. As such, these 24 bits are used to convey lower 16 bits of the TSC from the sender (encryptor) to the receiver (decryptor). The rest of the TSC shall be conveyed in the Extended IV field. The temporal key and *TTAK* values are represented as in Phase 1. The WEP seed is treated as an array of 8-bit values: $WEPSeed_0 \ldots WEPSeed_{15}$. The TSC shall be treated as an array of 8-bit values: $TSC_0\ TSC_1\ TSC_2\ TSC_3\ TSC_4\ TSC_5$.

The pseudo-code specifying the Phase 2 mixing function employs one variable: *PPK*, which is 96 bits long. The *PPK* is represented as an array of 16-bit values: $PPK_0..PPK_5$. The pseudo-code also employs a loop counter, *i*. As detailed in this subclause, the mapping from the 16-bit *PPK* values to the 8-bit *WEPseed* values is explicitly little endian to match the endian architecture of the most common processors used for this application.

The XOR (⊕) operation, the addition (+) operation, the AND (&) operation, the OR (|) operation, and the right bit shift (>>) operation are used in the specification of Phase 2.

Copyright © 2007 IEEE. All rights reserved.

**Input:** intermediate key *TTAK*0…*TTAK*4, *TK*, and TKIP sequence counter *TSC*
**Output:** WEP Seed *WEPSeed*0…*WEPSeed*15
PHASE2-KEY-MIXING(*TTAK*0…*TTAK*4, *TK*0…*TK*15, *TSC*0…*TSC*5)
PHASE2_STEP1:
*PPK*0 ← *TTAK*0
*PPK*1 ← *TTAK*1
*PPK*2 ← *TTAK*2
*PPK*3 ← *TTAK*3
*PPK*4 ← *TTAK*4
*PPK*5 ← *TTAK*4 + Mk16(TSC1, TSC0)
PHASE2_STEP2:
*PPK*0 ← *PPK*0 + S[*PPK*5 ⊕ *Mk*16(*TK*1,*TK*0)]
*PPK*1 ← *PPK*1 + S[*PPK*0 ⊕ *Mk*16(*TK*3,*TK*2)]
*PPK*2 ← *PPK*2 + S[*PPK*1 ⊕ *Mk*16(*TK*5,*TK*4)]
*PPK*3 ← *PPK*3 + S[*PPK*2 ⊕ *Mk*16(*TK*7,*TK*6)]
*PPK*4 ← *PPK*4 + S[*PPK*3 ⊕ *Mk*16(*TK*9,*TK*8)]
*PPK*5 ← *PPK*5 + S[*PPK*4 ⊕ *Mk*16(*TK*11,*TK*10)]
*PPK*0 ← *PPK*0 + *RotR*1(*PPK*5 ⊕ *Mk*16(*TK*13,*TK*12))
*PPK*1 ← *PPK*1 + *RotR*1(*PPK*0 ⊕ *Mk*16(*TK*15,*TK*14))
*PPK*2 ← *PPK*2 + *RotR*1(*PPK*1)
*PPK*3 ← *PPK*3 + *RotR*1(*PPK*2)
*PPK*4 ← *PPK*4 + *RotR*1(*PPK*3)
*PPK*5 ← *PPK*5 + *RotR*1(*PPK*4)
PHASE2_STEP3:
*WEPSeed*0 ← *TSC*1
*WEPSeed*1 ← (TSC1 | 0x20) & 0x7F
*WEPSeed*2 ← TSC0
*WEPSeed*3 ← *Lo*8((*PPK*5 ⊕ *Mk*16(*TK*1,*TK*0)) >> 1)
for *i* = 0 to 5
*WEPSeed*4+(2·*i*) ← *Lo*8(*PPKi*)
*WEPSeed*5+(2·*i*) ← *Hi*8(*PPKi*)
end
**return** *WEPSeed*0…*WEPSeed*15

**Figure 8-14—Phase 2 key mixing**

The algorithm specification relies on four functions:

— The first function, *Lo*8, references the 8 LSBs of the 16-bit input value.

— The second function, *Hi*8, references the 8 MSBs of the 16-bit value.

— The third function, *RotR*1, rotates its 16-bit argument 1 bit to the right.

— The fourth function, *Mk*16, is already used in Phase 1, defined by $Mk16(X,Y) = (256 \cdot X)+Y$, and constructs a 16-bit output from two 8-bit inputs.

NOTE—The rotate and addition operations in STEP2 make Phase 2 particularly sensitive to the endian architecture of the processor, although the performance degradation due to running this algorithm on a big endian processor should be minor.

Phase 2 comprises three steps:

— STEP1 makes a copy of *TTAK* and brings in the TSC.

— STEP2 is a 96-bit bijective mixing, employing an S-box.

— STEP3 brings in the last of the temporal key *TK* bits and assigns the 24-bit WEP IV value.

The WEP IV format carries 3 octets. STEP3 of Phase 2 determines the value of each of these three octets. The construction was selected to preclude the use of known ARC4 weak keys. The recipient can reconstruct the 16 LSBs of the TSC used by the originator by concatenating the third and first octets, ignoring the second octet. The remaining 32 bits of the TSC are obtained from the Extended IV field.


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.2.6 TKIP replay protection procedures

TKIP implementations shall use the TSC field to defend against replay attacks by implementing the following rules:

a) Each MPDU shall have a unique TKIP TSC value.

b) Each transmitter shall maintain a single TSC (48-bit counter) for each PTKSA, GTKSA, and STKSA.

c) The TSC shall be implemented as a 48-bit monotonically incrementing counter, initialized to 1 when the corresponding TKIP temporal key is initialized or refreshed.

d) The WEP IV format carries the 16 LSBs of the 48-bit TSC, as defined by the TKIP mixing function (Phase 2, STEP3). The remainder of the TSC is carried in the Extended IV field.

e) A receiver shall maintain a separate set of TKIP TSC replay counters for each PTKSA, GTKSA, and STKSA.

f) TKIP replay detection takes place after the MIC verification and any reordering required by ACK processing. Thus, a receiver shall delay advancing a TKIP TSC replay counter until an MSDU passes the MIC check, to prevent attackers from injecting MPDUs with valid ICVs and TSCs, but invalid MICs.

   NOTE—This works because if an attacker modifies the TSC, then the encryption key is modified and hence both the ICV and MIC will ordinarily decrypt incorrectly, causing the received MPDU to be dropped.

g) For each PTKSA, GTKSA, and STKSA, the receiver shall maintain a separate replay counter for each frame priority and shall use the TSC recovered from a received frame to detect replayed frames, subject to the limitations on the number of supported replay counters indicated in the RSN Capabilities field, as described in 7.3.2.25. A replayed frame occurs when the TSC extracted from a received frame is less than or equal to the current replay counter value for the frame's priority. A transmitter shall not reorder frames with different priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE 802.11 MSDU priority.

h) A receiver shall discard any MPDU that is received out of order and shall increment the value of dot11RSNAStatsTKIPReplays for this key.

i) For MSDUs sent using the Block Ack feature, reordering of received MSDUs according to the Block Ack receiver operation (described in 9.10.4) is performed prior to replay detection.

### 8.3.3 CTR with CBC-MAC Protocol (CCMP)

This subclause specifies the CCMP, which provides data confidentiality, authentication, integrity, and replay protection. CCMP is mandatory for RSN compliance.

#### 8.3.3.1 CCMP overview

CCMP is based on the CCM of the AES encryption algorithm. CCM combines CTR for data confidentiality and CBC-MAC for authentication and integrity. CCM protects the integrity of both the MPDU Data field and selected portions of the IEEE 802.11 MPDU header.

The AES algorithm is defined in FIPS PUB 197-2001. All AES processing used within CCMP uses AES with a 128-bit key and a 128-bit block size.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 227 of 1232

CCM is defined in IETF RFC 3610. CCM is a generic mode that can be used with any block-oriented encryption algorithm. CCM has two parameters (*M* and *L*), and CCMP uses the following values for the CCM parameters:

— *M* = 8; indicating that the MIC is 8 octets.

— *L* = 2; indicating that the Length field is 2 octets, which is sufficient to hold the length of the largest possible IEEE 802.11 MPDU, expressed in octets.

CCM requires a fresh temporal key for every session. CCM also requires a unique nonce value for each frame protected by a given temporal key, and CCMP uses a 48-bit packet number (PN) for this purpose. Reuse of a PN with the same temporal key voids all security guarantees.

Annex H provides a test vector for CCM.

### 8.3.3.2 CCMP MPDU format

Figure 8-15 depicts the MPDU when using CCMP.

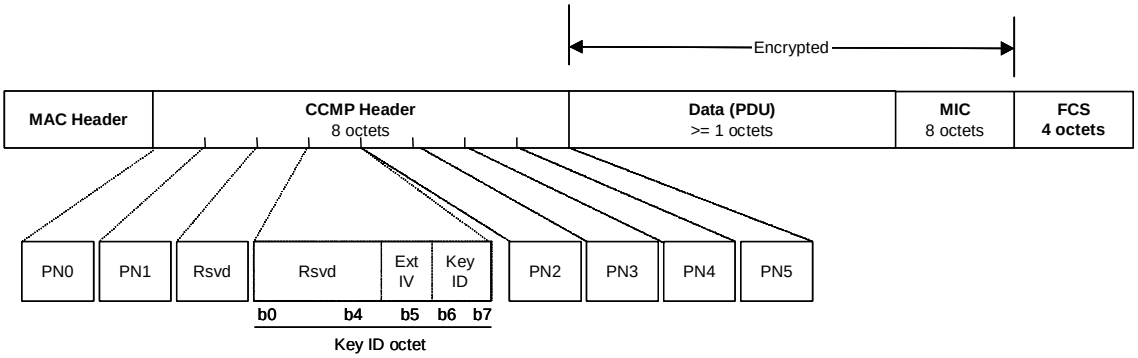



**Figure 8-15—Expanded CCMP MPDU**

CCMP processing expands the original MPDU size by 16 octets, 8 octets for the CCMP Header field and 8 octets for the MIC field. The CCMP Header field is constructed from the PN, ExtIV, and Key ID subfields. PN is a 48-bit PN represented as an array of 6 octets. PN5 is the most significant octet of the PN, and PN0 is the least significant. Note that CCMP does not use the WEP ICV.

The ExtIV subfield (bit 5) of the Key ID octet signals that the CCMP Header field extends the MPDU header by a total of 8 octets, compared to the 4 octets added to the MPDU header when WEP is used. The ExtIV bit (bit 5) is always set to 1 for CCMP.

Bits 6–7 of the Key ID octet are for the Key ID subfield.

The reserved bits shall be set to 0 and shall be ignored on reception.


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.3.3 CCMP cryptographic encapsulation

The CCMP cryptographic encapsulation process is depicted in Figure 8-16.

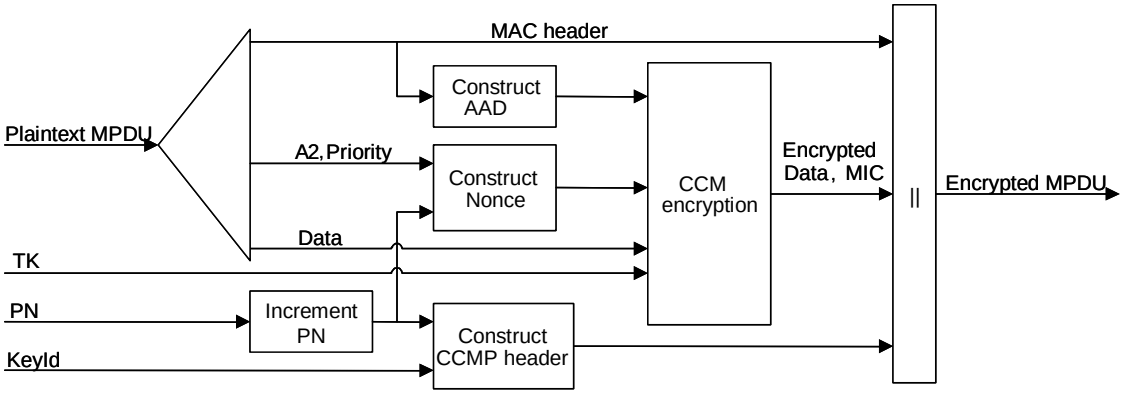



**Figure 8-16—CCMP encapsulation block diagram**

CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following steps:

a) Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission.

b) Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that may change when retransmitted are muted by being masked to 0 when calculating the AAD.

c) Construct the CCM Nonce block from the PN, A2, and the Priority field of the MPDU where A2 is MPDU Address 2.

d) Place the new PN and the key identifier into the 8-octet CCMP header.

e) Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.

f) Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC, as described in 8.3.3.2.

The CCM reference describes the processing of the key, nonce, AAD, and data to produce the encrypted output. See 8.3.3.3.1 through 8.3.3.3.5 for details of the creation of the AAD and nonce from the MPDU and the associated MPDU-specific processing.

##### 8.3.3.3.1 PN processing

The PN is incremented by a positive number for each MPDU. The PN shall never repeat for a series of encrypted MPDUs using the same temporal key.

Copyright © 2007 IEEE. All rights reserved.

#### 8.3.3.3.2 Construct AAD

The format of the AAD is shown in Figure 8-17.

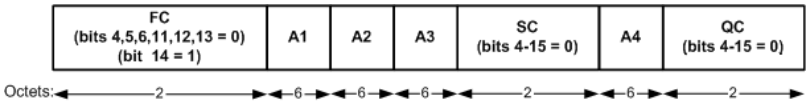



**Figure 8-17—AAD construction**

The length of the AAD varies depending on the presence or absence of the QC and A4 fields and is shown in Table 8-1:

**Table 8-1—AAD length**

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

The AAD is constructed from the MPDU header. The AAD does not include the header Duration field, because the Duration field value can change due to normal IEEE 802.11 operation (e.g., a rate change during retransmission). For similar reasons, several subfields in the Frame Control field are masked to 0. AAD construction is performed as follows:

a) FC – MPDU Frame Control field, with
   1) Subtype bits (bits 4 5 6) masked to 0
   2) Retry bit (bit 11) masked to 0
   3) PwrMgt bit (bit 12) masked to 0
   4) MoreData bit (bit 13) masked to 0
   5) Protected Frame bit (bit 14) always set to 1

b) A1 – MPDU Address 1 field.

c) A2 – MPDU Address 2 field.

d) A3 – MPDU Address 3 field.

e) SC – MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked to 0. The Fragment Number subfield is not modified.

f) A4 – MPDU Address field, if present in the MPDU.

g) QC – QoS Control field, if present, a 2-octet field that includes the MSDU priority. 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).


Copyright © 2007 IEEE. All rights reserved.

#### 8.3.3.3.3 Construct CCM nonce

The Nonce field occupies 13 octets, and its structure is shown in Figure 8-18.

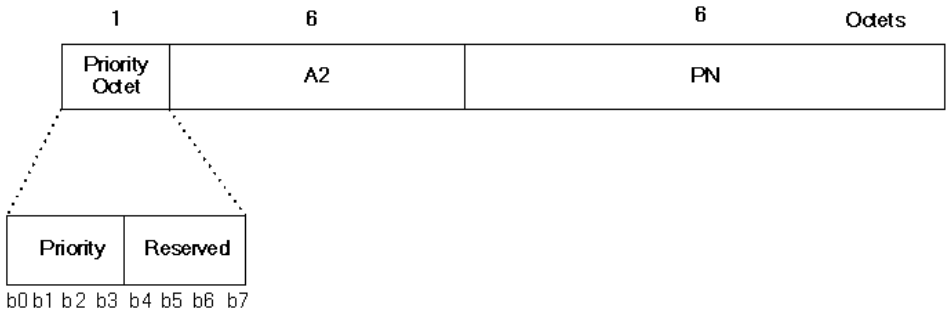



**Figure 8-18—Nonce construction**

The Nonce field has an internal structure of Priority Octet || A2 || PN ("||" is concatenation), where

— 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.

— MPDU address A2 field occupies octets 1–6. This shall be encoded with the octets ordered with A2 octet 0 at octet index 1 and A2 octet 5 at octet index 6.

— The PN field occupies octets 7–12. The octets of PN shall be ordered so that PN0 is at octet index 12 and PN5 is at octet index 7.

#### 8.3.3.3.4 Construct CCMP header

The format of the 8-octet CCMP header is given in 8.3.3.2. The header encodes the PN, Key ID, and ExtIV field values used to encrypt the MPDU.

#### 8.3.3.3.5 CCM originator processing

CCM is a generic authenticate-and-encrypt block cipher mode, and in this standard, CCM is used with the AES block cipher.

There are four inputs to CCM originator processing:

a) *Key:* the temporal key (16 octets).

b) *Nonce:* the nonce (13 octets) constructed as described in 8.3.3.3.3.

c) *Frame body:* the frame body of the MPDU (1–2296 octets; 2296 = 2312 – 8 MIC octets – 8 CCMP header octets).

d) *AAD:* the AAD (22–30 octets) constructed from the MPDU header as described in 8.3.3.3.2.

The CCM originator processing provides authentication and integrity of the frame body and the AAD as well as data confidentiality of the frame body. The output from the CCM originator processing consists of the encrypted data and 8 additional octets of encrypted MIC (see Figure 8-15).

Copyright © 2007 IEEE. All rights reserved.

#### 8.3.3.4 CCMP decapsulation

Figure 8-19 depicts the CCMP decapsulation process.

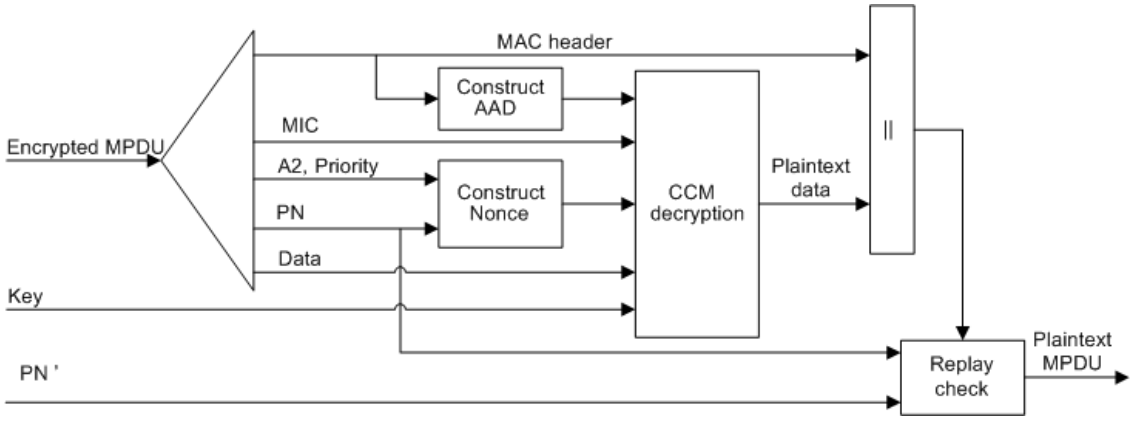



**Figure 8-19—CCMP decapsulation block diagram**

CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU using the following steps:

a) The encrypted MPDU is parsed to construct the AAD and nonce values.

b) The AAD is formed from the MPDU header of the encrypted MPDU.

c) The Nonce value is constructed from the A2, PN, and Priority Octet fields.

d) The MIC is extracted for use in the CCM integrity checking.

e) The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.

f) The received MPDU header and the MPDU plaintext data from the CCM recipient processing may be concatenated to form a plaintext MPDU.

g) The decryption processing prevents replay of MPDUs by validating that the PN in the MPDU is greater than the replay counter maintained for the session.

See 8.3.3.4.1 through 8.3.3.4.3 for details of this processing.

#### 8.3.3.4.1 CCM recipient processing

CCM recipient processing must use the same parameters as CCM originator processing.

There are four inputs to CCM recipient processing:

— *Key:* the temporal key (16 octets).

— *Nonce:* the nonce (13 octets) constructed as described in 8.3.3.3.3.

— *Encrypted frame body:* the encrypted frame body from the received MPDU. The encrypted frame body includes an 8-octet MIC (9–2304 octets).

— *AAD:* the AAD (22–30 octets) that is the canonical MPDU header as described in 8.3.3.3.2.

The CCM recipient processing checks the authentication and integrity of the frame body and the AAD as well as decrypting the frame body. The plaintext is returned only if the MIC check is successful.


Copyright © 2007 IEEE. All rights reserved.

There is one output from error-free CCM recipient processing:

— *Frame body:* the plaintext frame body, which is 8 octets smaller than the encrypted frame body.

#### 8.3.3.4.2 Decrypted CCMP MPDU

The decapsulation process succeeds when the calculated MIC matches the MIC value obtained from decrypting the received encrypted MPDU. The original MPDU header is concatenated with the plaintext data resulting from the successful CCM recipient processing to create the plaintext MPDU.

#### 8.3.3.4.3 PN and replay detection

To effect replay detection, the receiver extracts the PN from the CCMP header. See 8.3.3.2 for a description of how the PN is encoded in the CCMP header. The following processing rules are used to detect replay:

a) The PN values sequentially number each MPDU.

b) Each transmitter shall maintain a single PN (48-bit counter) for each PTKSA, GTKSA, and STKSA.

c) The PN shall be implemented as a 48-bit monotonically incrementing non-negative integer, initialized to 1 when the corresponding temporal key is initialized or refreshed.

d) A receiver shall maintain a separate set of PN replay counters for each PTKSA, GTKSA, and STKSA. The receiver initializes these replay counters to 0 when it resets the temporal key for a peer. The replay counter is set to the PN value of accepted CCMP MPDUs.

e) For each PTKSA, GTKSA, and STKSA, the recipient shall maintain a separate replay counter for each IEEE 802.11 MSDU priority and shall use the PN recovered from a received frame to detect replayed frames, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 7.3.2.25). A replayed frame occurs when the PN extracted from a received frame is less that or equal to the current replay counter value for the frame's MSDU priority. A transmitter shall not use IEEE 802.11 MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE 802.11 MSDU priority.

f) The receiver shall discard MSDUs whose constituent MPDU PN values are not sequential. A receiver shall discard any MPDU that is received with its PN less than or equal to the replay counter and shall increment the value of dot11RSNAStatsCCMPReplays for this key.

g) For MSDUs sent using the Block Ack feature, reordering of received MSDUs according to the Block Ack receiver operation (described in 9.10.4) is performed prior to replay detection.

## 8.4 RSNA security association management

### 8.4.1 Security associations

#### 8.4.1.1 Security association definitions

IEEE Std 802.11 uses the notion of a security association to describe secure operation. Secure communications are possible only within the context of a security association, as this is the context providing the state—cryptographic keys, counters, sequence spaces, etc.—needed for correct operation of the IEEE 802.11 cipher suites.

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, must be consistent among all parties, and must have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. There are four types of security associations supported by an RSN STA:

Copyright © 2007 IEEE. All rights reserved.

— PMKSA: A result of a successful IEEE 802.lX exchange, preshared PMK information, or PMK cached via some other mechanism.
— PTKSA: A result of a successful 4-Way Handshake.
— GTKSA: A result of a successful Group Key Handshake or successful 4-Way Handshake.
— SMKSA: A result of a successful initial SMK Handshake.
— STKSA: A result of a successful 4-Way STK Handshake following the initial SMK Handshake or subsequent rekeying.

#### 8.4.1.1.1 PMKSA

When the PMKSA is the result of a successful IEEE 802.1X authentication, it is derived from the EAP authentication and authorization parameters provided by the AS. This security association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving. The PMKSA is created by the Supplicant's SME when the EAP authentication completes successfully or the PSK is configured. The PMKSA is created by the Authenticator's SME when the PMK is created from the keying information transferred from the AS or the PSK is configured. The PMKSA is used to create the PTKSA. PMKSAs are cached for up to their lifetimes. The PMKSA consists of the following elements:

— PMKID, as defined in 8.5.1.2. The PMKID identifies the security association.
— Authenticator MAC address.
— PMK.
— Lifetime, as defined in 8.5.1.2.
— AKMP.
— All authorization parameters specified by the AS or local configuration. This can include parameters such as the STA's authorized SSID.

#### 8.4.1.1.2 PTKSA

The PTKSA is a result of the 4-Way Handshake. This security association is also bidirectional. The PTKSA is used to create the key hierarchy. PTKSAs are cached for the life of the PMKSA. Because the PTKSA is tied to the PMKSA, it only has the additional information from the 4-Way Handshake. There shall be only one PTKSA with the same Supplicant and Authenticator MAC addresses. There is state created between Message 1 and Message 3 of a 4-Way Handshake. This does not create a PTKSA until Message 3 is validated on the Supplicant and Message 4 is validated by the Authenticator. The PTKSA consists of the following elements:

— PTK
— Pairwise cipher suite selector
— Supplicant MAC address
— Authenticator MAC address

#### 8.4.1.1.3 GTKSA

The GTKSA results from a successful 4-Way Handshake or the Group Key Handshake and is unidirectional. In an ESS, there is one GTKSA, used exclusively for encrypting broadcast/multicast MPDUs that are transmitted by the AP and for decrypting broadcast/multicast transmissions that are received by the STAs. In an IBSS, each STA defines its own GTKSA, which is used to encrypt its broadcast/multicast transmissions, and stores a separate GTKSA for each peer STA so that encrypted broadcast/multicast traffic received from other STAs may be decrypted. A GTKSA is created by the Supplicant's SME when Message 3 of the 4-Way Handshake is received or when Message 1 of the Group Key Handshake is received. The GTKSA is created by the Authenticator's SME when the SME changes the GTK and has sent the GTK to all STAs with which it has a PTKSA. A GTKSA consists of the following elements:


Copyright © 2007 IEEE. All rights reserved.

— Direction vector (whether the GTK is used for transmit or receive).
— Group cipher suite selector.
— GTK.
— Authenticator MAC address.
— All authorization parameters specified by local configuration. This can include parameters such as the STA's authorized SSID.

When the GTK is used to encrypt unicast traffic (the selectable cipher suite is "Use group key"), the GTKSA is bidirectional.

#### 8.4.1.1.4 SMKSA

An SMKSA is the result of a successful SMK Handshake by the initiator STA (described in 8.5.8). It is derived from parameters provided by the STAs and AP. This security association is bidirectional between the initiator and the peer STA. In other words, both parties use the information in the security association for both sending and receiving. The SMKSA is created as a result of a successful SMK Handshake (see 8.5.8). The SMKSA is used to create the STKSA. The SMKSA consists of the following elements:

— SMKID, as defined in 8.5.8. The SMKID identifies the security association.
— BSSID
— Initiator MAC address
— Peer MAC address
— SMK
— Lifetime, as defined in 8.5.8.
— Pairwise cipher suite selector list, as proposed by initiator STA
— Pairwise cipher suite selector, as selected by peer STA

#### 8.4.1.1.5 STKSA

The STKSA is a result of successful completion of the 4-Way STK Handshake. This security association is bidirectional between the initiator and the peer STAs. The STKSA is used to create session keys to protect this STSL. STKSAs are cached for the life of the SMKSA or until the STSL ends, whichever comes first. There shall be only one STKSA with the same initiator STA and peer MAC addresses at any one time. STKSA is created as a result of PeerKey Handshake (see 8.5.8). The STKSA consists of the following elements:

— STK
— Pairwise cipher suite selector
— Initiator MAC address
— Peer MAC address

### 8.4.1.2 Security association life cycle

A STA can operate in either an ESS or in an IBSS, and a security association has a distinct life cycle for each.

#### 8.4.1.2.1 Security association in an ESS

In an ESS there are two cases:

— Initial contact between the STA and the ESS
— Roaming by the STA within the ESS

Copyright © 2007 IEEE. All rights reserved.

A STA and AP establish an initial security association via the following steps:

a) The STA selects an authorized ESS by selecting among APs that advertise an appropriate SSID.

b) The STA then uses IEEE 802.11 Open System authentication followed by association to the chosen AP. Negotiation of security parameters takes place during association.

NOTE 1—It is possible for more than one PMKSA to exist. As an example, a second PMKSA may come into existence through PMKSA caching. A STA might leave the ESS and flush its cache. Before its PMKSA expires in the AP's cache, the STA returns to the ESS and establishes a second PMKSA from the AP's perspective.

NOTE 2—An attack altering the security parameters will be detected by the key derivation procedure.

NOTE 3—IEEE 802.11 Open System authentication provides no security, but is included to maintain backward compatibility with the IEEE 802.11 state machine (see 11.3).

c) The AP's Authenticator or the STA's Supplicant initiates IEEE 802.1X authentication. The EAP method used by IEEE Std 802.1X-2004 will support mutual authentication, as the STA needs assurance that the AP is a legitimate AP.

NOTE 1—Prior to the completion of IEEE 802.1X authentication and the installation of keys, the IEEE 802.1X Controlled Port in the AP will block all data frames. The IEEE 802.1X Controlled Port returns to the unauthorized state and blocks all data frames before invocation of an MLME-DELETEKEYS.request primitive. The IEEE 802.1X Uncontrolled Port allows IEEE 802.1X frames to pass between the Supplicant and Authenticator. Although IEEE Std 802.1X-2004 does not require a Supplicant Controlled Port, this standard assumes that the Supplicant has a Controlled Port in order to provide the needed level of security. Supplicants without a Controlled Port compromise RSN security and should not be used.

NOTE 2—Any secure network cannot support promiscuous association, e.g., an unsecured operation of IEEE Std 802.11. A trust relationship must exist between the STA and the AS of the targeted SSID prior to association and secure operation, in order for the association to be trustworthy. The reason is that an attacker can deploy a rogue AP just as easily as a legitimate network provider can deploy a legitimate AP, so some sort of prior relationship is necessary to establish credentials between the ESS and the STA.

d) The last step is key management. The authentication process creates cryptographic keys shared between the IEEE 802.1X AS and the STA. The AS transfers these keys to the AP, and the AP and STA use one key confirmation handshake, called the 4-Way Handshake, to complete security association establishment. The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic.

A STA roaming within an ESS establishes a new PMKSA by one of three schemes:

— In the case of (re)association followed by IEEE 802.1X or PSK authentication, the STA repeats the same actions as for an initial contact association, but its Supplicant also deletes the PTKSA when it roams from the old AP. The STA's Supplicant also deletes the PTKSA when it disassociates/deauthenticates from all BSSIDs in the ESS.

— A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSN information element of its (Re)Association Request frame. An AP whose Authenticator has retained the PMK for one or more of the PMKIDs can skip the IEEE 802.1X authentication and proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, then the Authenticator shall perform another IEEE 802.1X authentication. Similarly, if the STA fails to send a PMKID, the STA and AP must perform a full IEEE 802.1X authentication.

— A STA already associated with the ESS can request its IEEE 802.1X Supplicant to authenticate with a new AP before associating to that new AP. The normal operation of the DS via the old AP provides the communication between the STA and the new AP. The STA's IEEE 802.11 management entity delays reassociation with the new AP until IEEE 802.1X authentication completes via the DS. If IEEE 802.1X authentication completes successfully, then PMKSAs shared between the new AP and the STA will be cached, thereby enabling the possible usage of reassociation without requiring a subsequent full IEEE 802.1X authentication procedure.


Copyright © 2007 IEEE. All rights reserved.

The MLME-DELETEKEYS.request primitive destroys the temporal keys established for the security association so that they cannot be used to protect subsequent IEEE 802.11 traffic. A STA's SME uses this primitive when it deletes a PTKSA or GTKSA.

#### 8.4.1.2.2 Security association in an IBSS

In an IBSS, when a STA's SME establishes a security association with a peer STA, it creates both an IEEE 802.1X Supplicant and Authenticator for the peer.

A STA can receive IEEE 802.1X messages from a previously unknown MAC address.

Any STA within an IBSS may decline to form a security association with a STA joining the IBSS. An attempt to form a security association may also fail because, for example, the peer uses a different PSK from what the STA expects.

In an IBSS each STA defines its own group key, i.e., GTK, to secure its broadcast/multicast transmissions. Each STA shall use either the 4-Way Handshake or the Group Key Handshake to distribute its transmit GTK to its new peer STA. When the STA generates a new GTK, it also uses the Group Key Handshake to distribute the new GTK to each established peer.

### 8.4.2 RSNA selection

A STA prepared to establish RSNAs shall advertise its capabilities by including the RSN information element in Beacon and Probe Response messages. The included RSN information element shall specify all the authentication and cipher suites enabled by the STA's policy. A STA shall not advertise any authentication or cipher suite that is not enabled.

The STA's IEEE 802.11 management entity shall utilize the MLME-SCAN.request primitive to identify neighboring STAs that assert robust security and advertise an SSID identifying an authorized ESS or IBSS. A STA may decline to communicate with STAs that fail to advertise an RSN information element in their Beacon and Probe Response frames or that do not advertise an authorized SSID. A STA may also decline to communicate with other STAs that do not advertise authorized authentication and cipher suites within their RSN information elements.

A STA shall advertise the same RSN information element in both its Beacon and Probe Response frames.

NOTE 1—Whether a STA with robust security enabled may attempt to communicate with a STA that does not include the RSN information element is a matter of policy.

NOTE 2—As a practical matter, if maximal interoperability is a goal, an AP will support TKIP as well as CCMP.

A STA shall observe the following rules when processing an RSN information element:

— A STA shall advertise the highest version it supports.
— A STA shall request the highest Version field value it supports that is less than or equal to the version advertised by the peer STA.
— Two peer STAs without overlapping supported Version field values shall not use RSNA methods to secure their communication.
— A STA shall ignore suite selectors that it does not recognize.

### 8.4.3 RSNA policy selection in an ESS

RSNA policy selection in an ESS utilizes the normal IEEE 802.11 association procedure. RSNA policy selection is performed by the associating STA. The STA does this by including an RSN information element in its (Re)Association Requests.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 237 of 1232

In an RSN, an AP shall not associate with pre-RSNA STAs, i.e., with STAs that fail to include the RSN information element in the Association or Reassociation Request frame.

The STA's SME initiating an association shall insert an RSN information element into its (Re)Association Request; via the MLME-ASSOCIATE.request primitive, when the targeted AP indicates RSNA support. The initiating STA's RSN information element shall include one authentication and pairwise cipher suite from among those advertised by the targeted AP in its Beacon and Probe Response frames. It shall also specify the group cipher suite specified by the targeted AP. If at least one RSN information element field from the AP's RSN information element fails to overlap with any value the STA supports, the STA shall decline to associate with that AP.

If an RSNA-capable AP receives a (Re)Association Request including an RSN information element and if it chooses to accept the association as a secure association, then it shall use the authentication and pairwise cipher suites in the (Re)Association Request, unless the AP includes an optional second RSN information element in Message 3 of the 4-Way Handshake. If the second RSN information element is supplied in Message 3, then the pairwise cipher suite used by the security association, if established, shall be the pairwise cipher from the second RSN information element.

In order to accommodate local security policy, a STA may choose not to associate with an AP that does not support any pairwise cipher suites. An AP indicates that it does not support any pairwise keys by advertising "Use group key" as the pairwise cipher suite selector.

NOTE—When an ESS uses PSKs, STAs negotiate a pairwise cipher. However, any STA in the ESS can derive the pairwise keys of any other that uses the same PSK by capturing the first two messages of the 4-Way Handshake. This provides malicious insiders with the ability to eavesdrop as well as the ability to establish a man-in-the-middle attack.

#### 8.4.3.1 TSN policy selection in an ESS

In a TSN, an RSN STA shall include the RSN information element in its (Re)Association Requests.

An RSNA-capable AP configured to operate in a TSN shall include the RSN information element and may associate with both RSNA and pre-RSNA STAs. In other words, an RSNA-capable AP shall respond to an associating STA that includes the RSN information element just as in an RSN.

If an AP operating within a TSN receives a (Re)Association Request without an RSN information element, its IEEE 802.1X Controlled Port shall initially be blocked. The SME shall unblock the IEEE 802.1X Controlled Port when WEP has been enabled.

### 8.4.4 RSNA policy selection in an IBSS

In an IBSS, all STAs must use a single group cipher suite, and all STAs must support a common subset of pairwise cipher suites. However, the SMEs of any pair of STAs may negotiate to use any common pairwise cipher suite they both support. Each STA shall include the group cipher suite and its list of pairwise cipher suites in its Beacon and Probe Response messages. Two STAs shall not establish a PMKSA unless they have advertised the same group cipher suite. Similarly, the two STAs shall not establish a PMKSA if the STAs have advertised disjoint sets of pairwise cipher suites.

When an IBSS STA's SME wants to set up a security association with a peer STA, but does not know the peer's policy, it must first obtain the peer's security policy using a Probe Request frame. The SME entities of the two STAs select the pairwise cipher suites using one of the 4-Way Handshakes. The SMEs of each pair of STAs within an IBSS may use the EAPOL-Key 4-Way Handshake to select a pairwise cipher suite. As specified in 8.5.2, Message 2 and Message 3 of the 4-Way Handshake convey an RSN information element. The Message 2 RSN information element includes the selected pairwise cipher suite, and Message 3 includes the RSN information element that the STA would send in a Probe Response frame.


Copyright © 2007 IEEE. All rights reserved.

The pair of STAs shall use the pairwise cipher suite specified in Message 3 of the 4-Way Handshake sent by the Authenticator STA with the higher MAC address (see 8.5.1).

The SME shall check that the group cipher suite and AKMP match those in the Beacon and Probe Response frames for the IBSS.

NOTE 1—The RSN information elements in Message 2 and Message 3 are not the same as in the Beacon frame. The group cipher and AKMP are the same, but the pairwise ciphers may differ because Beacon frames from different STAs may advertise different pairwise ciphers. Thus, STAs in an IBSS use the same AKM suite and group cipher, while different pairwise ciphers can be used between STA pairs.

NOTE 2—When an IBSS network uses PSKs, STAs can negotiate a pairwise cipher. However, any STA in the IBSS can derive the PTKs of any other that uses the same PSK by capturing the first two messages of the 4-Way Handshake. This provides malicious insiders with the ability to eavesdrop as well as the ability to establish a man-in-the-middle attack.

#### 8.4.4.1 TSN policy selection in an IBSS

Pre-RSNA STAs generate Beacon and Probe Response frames without an RSN information element and will ignore the RSN information element because it is unknown to them. This allows an RSNA STA to identify the pre-RSNA STAs from which it has received Beacon and Probe Response frames.

If an RSNA STA's SME instead identifies a possible IBSS member on the basis of a received broadcast/multicast message, via MLME-PROTECTEDFRAMEDROPPED.indication primitive, it cannot identify the peer's security policy directly. The SME can attempt to obtain the peer STA's security policy via a Probe Request frame.

### 8.4.5 RSN management of the IEEE 802.1X Controlled Port

When the policy selection process chooses IEEE 802.1X authentication, this standard assumes that IEEE 802.1X Supplicants and Authenticators exchange protocol information via the IEEE 802.1X Uncontrolled port. The IEEE 802.1X Controlled Port is blocked from passing general data traffic between the STAs until an IEEE 802.1X authentication procedure completes successfully over the IEEE 802.1X Uncontrolled Port. The security of an RSNA depends on this assumption being true.

In an ESS, the STA indicates the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.confirm or MLME-REASSOCIATE.confirm primitive. This signals the Supplicant that the MAC has transitioned from the disabled to enabled state. At this point, the Supplicant's Controlled Port is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the port is not authorized.

In an ESS, the AP indicates that the IEEE 802.11 link is available by invoking the MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive. At this point the Authenticator's Controlled Port corresponding to the STA's association is blocked, and communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

In an IBSS, the STA shall block all IEEE 802.1X ports at initialization. Communication of all non-IEEE-802.1X MSDUs sent or received via the Controlled Port is not authorized.

This standard assumes each Controlled Port remains blocked until the IEEE 802.1X state variables portValid and keyDone both become true. This assumption means that the IEEE 802.1X Controlled Port discards MSDUs sent across the IEEE 802.11 channel prior to the installation of cryptographic keys into the MAC. This protects the STA's host from forged MSDUs written to the channel while it is still being initialized.

The MAC does not distinguish between MSDUs for the Controlled Port, and MSDUs for the Uncontrolled Port. In other words, IEEE 802.1X EAPOL frames will only be encrypted after invocation of the MLME-SETPROTECTION.request primitive.

Copyright © 2007 IEEE. All rights reserved.

This standard assumes that IEEE Std 802.1X-2004 does not block the Controlled Port when authentication is triggered through reauthentication. During IEEE 802.1X reauthentication, an existing RSNA can protect all MSDUs exchanged between the STAs. Blocking MSDUs is not required during reauthentication over an RSNA.

### 8.4.6 RSNA authentication in an ESS

When establishing an RSNA, a STA shall use IEEE 802.11 Open System authentication prior to (re)association.

IEEE 802.1X authentication is initiated by any one of the following mechanisms:

— If a STA negotiates to use IEEE 802.1X authentication during (re)association, the STA's management entity can respond to the MLME-ASSOCIATE.confirm (or indication) primitive by requesting the STA's Supplicant (or AP's Authenticator) to initiate IEEE 802.1X authentication. Thus, in this case, authentication is driven by the STA's decision to associate and the AP's decision to accept the association.

— If a STA's MLME-SCAN.confirm primitive finds another AP within the current ESS, a STA may signal its Supplicant to use IEEE Std 802.1X-2004 to preauthenticate with that AP.

NOTE—A roaming STA's IEEE 802.1X Supplicant may initiate preauthentication by sending an EAPOL-Start message via its old AP, through the DS, to a new AP.

— If a STA receives an IEEE 802.1X message, it delivers this to its Supplicant or Authenticator, which may initiate a new IEEE 802.1X authentication.

#### 8.4.6.1 Preauthentication and RSNA key management

A STA shall not use preauthentication except when pairwise keys are employed. Preauthentication shall not be used unless the new AP advertises the preauthentication capability in the RSN information element.

When preauthentication is used, then

a) Authentication is independent of roaming.

b) The STA's Supplicant may authenticate with multiple APs at a time.

NOTE—Preauthentication can be useful as a performance enhancement, as reassociation will not include the protocol overhead of a full reauthentication when it is used.

Preauthentication uses the IEEE 802.1X protocol and state machines with EtherType 88-C7, rather than the EtherType 88-8E. Only IEEE 802.1X frame types EAP-Packet and EAPOL-Start are valid for preauthentication.

NOTE—Some IEEE 802.1X Authenticators may not bridge IEEE 802.1X frames, as suggested in C.1.1 of IEEE Std 802.1X-2004. Preauthentication uses a distinct EtherType to enable such devices to bridge preauthentication frames.

A STA's Supplicant can initiate preauthentication when it has completed the 4-Way Handshake and configured the required temporal keys. To effect preauthentication, the STA's Supplicant sends an IEEE 802.1X EAPOL-Start message with the DA being the BSSID of a targeted AP and the RA being the BSSID of the AP with which it is associated. The target AP shall use a BSSID equal to the MAC address of its Authenticator. As preauthentication frames do not use the IEEE 802.1X EAPOL EtherType field, the AP with which the STA is currently associated need not apply any special handling. The AP and the MAC in the STA shall handle these frames in the same way as other frames with arbitrary EtherType field values that require distribution via the DS.


Copyright © 2007 IEEE. All rights reserved.

An AP's Authenticator that receives an EAPOL-Start message via the DS may initiate IEEE 802.1X authentication to the STA via the DS. The DS will forward this message to the AP with which the STA is associated.

The result of preauthentication may be a PMKSA, if the IEEE 802.1X authentication completes successfully. If preauthentication produces a PMKSA, then, when the Supplicant's STA associates with the preauthenticated AP, the Supplicant can use the PMKSA with the 4-Way Handshake.

Successful completion of EAP authentication over IEEE 802.1X establishes a PMKSA at the Supplicant. The Authenticator has the PMKSA when the AS completes the authentication, passes the keying information (the master session key (MSK), a portion of which is the PMK) to the Authenticator, and the Authenticator creates a PMKSA using the PMK. The PMKSA is inserted into the PMKSA cache. Therefore, if the Supplicant and Authenticator lose synchronization with respect to the PMKSA, the 4-Way Handshake will fail. In such circumstances, the MIB variable dot11RSNAStats-4WayHandshakeFailures shall be incremented.

A STA's Supplicant may initiate preauthentication with any AP within its present ESS with preauthentication enabled regardless of whether the targeted AP is within radio range.

Even if a STA has preauthenticated, it is still possible that it may have to undergo a full IEEE 802.1X authentication, as the AP's Authenticator may have purged its PMKSA due to, for example, unavailability of resources, delay in the STA associating, etc.

#### 8.4.6.2 Cached PMKSAs and RSNA key management

A STA can retain PMKSAs it establishes as a result of previous authentication. The PMKSA cannot be changed while cached. The PMK in the PMKSA is used with the 4-Way Handshake to establish fresh PTKs.

If a non-AP STA in an ESS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it includes the PMKID for the PMKSA in the RSN information element in the (Re)Association Request. Upon receipt of a (Re)Association Request with one or more PMKIDs, an AP checks whether its Authenticator has retained a PMK for the PMKIDs and whether the PMK is still valid. If so, it asserts possession of that PMK by beginning the 4-Way Handshake after association has completed; otherwise it begins a full IEEE 802.1X authentication after association has completed.

If both sides assert possession of a cached PMKSA, but the 4-Way Handshake fails, both sides may delete the cached PMKSA for the selected PMKID.

If a STA roams to an AP with which it is preauthenticating and the STA does not have a PMKSA for that AP, the STA must initiate a full IEEE 802.1X EAP authentication.

### 8.4.7 RSNA authentication in an IBSS

When authentication is used in an IBSS, it is driven by each STA wishing to establish communications. The management entity of this STA chooses a set of STAs with which it may want to authenticate and then may cause the MAC to send an IEEE 802.11 Open System authentication message to each targeted STA. Candidate STAs can be identified from Beacon frames, Probe Response frames, and data frames from the same BSSID. Before communicating with STAs identified from data frames, the security policy of the STAs may be obtained, e.g., by sending a Probe Request frame to the STA and obtaining a Probe Response frame. Targeted STAs that wish to respond may return an IEEE 802.11 Open System authentication message to the initiating STA.

When IEEE 802.1X authentication is used, the STA management entity will then request its local IEEE 802.1X entity to create a Supplicant PAE for the peer STA. The Supplicant PAE will initiate the

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 241 of 1232

authentication to the peer STA by sending an EAPOL-Start message to the peer. The STA management entity will also request its local IEEE 802.1X entity to create an Authenticator PAE for the peer STA on receipt of the EAPOL-Start message. The Authenticator will initiate authentication to the peer STA by sending an EAP-Request message or, if PSK mode is in effect, Message 1 of the 4-Way Handshake.

Upon initial authentication between any pair of STAs, data frames, other than IEEE 802.1X messages, are not allowed to flow between the pair of STAs until both STAs in each pair of STAs have successfully completed AKM and have provided the supplied encryption keys.

Upon the initiation of an IEEE 802.1X reauthentication by any STA of a pair of STAs, data frames will continue to flow between the STAs while authentication completes. Upon a timeout or failure in the authentication process, the Authenticator of the STA initiating the reauthentication shall cause a Deauthentication message to be sent to the Supplicant of the STA targeted for reauthentication. The Deauthentication message will cause both STAs in the pair of STAs to follow the deauthentication procedure defined in 11.3.1.3 and 11.3.1.4.

The IEEE 802.1X reauthentication timers in each STA are independent. If the reauthentication timer of the STA with the higher MAC address (see 8.5.1 for MAC comparison) triggers the reauthentication via its Authenticator, its Supplicant must send an EAPOL-Start message to the authenticator of the STA with the lower MAC address to trigger reauthentication on the other STA. This process keeps the pair of STAs in a consistent state with respect to derivation of fresh temporal keys upon an IEEE 802.1X reauthentication.

When it receives an MLME-AUTHENTICATE.indicate primitive due to an Open System authentication request, the IEEE 802.11 management entity on a targeted STA shall, if it wants to set up a security association with the peer STA, request its Authenticator to begin IEEE 802.1X authentication, i.e., to send an EAP-Request/Identity message or Message 1 of the 4-Way Handshake to the Supplicant.

The EAPOL-Key frame is used to exchange information between the Supplicant and the Authenticator to negotiate a fresh PTK. The 4-Way Handshake produces a single PTK from the PMK. The 4-Way Handshake and Group Key Handshake use the PTK to protect the GTK as it is transferred to the receiving STA.

PSK authentication may also be used in an IBSS. When a single PSK is shared among the IBSS STAs, the STA wishing to establish communication sends 4-Way Handshake Message 1 to the target STA(s). The targeted STA responds to Message 1 with Message 2 of the 4-Way Handshake and begins its 4-Way Handshake by sending Message 1 to the initiating STA. The two 4-Way Handshakes establish PTKSAs and GTKSAs to be used between the initiating STA and the targeted STA. PSK PMKIDs may also be used, enabling support for pairwise PSKs.

The model for security in an IBSS is not general. In particular, it assumes the following:

a) The sets of use cases for which the authentication procedures described in this subclause are valid are as follows:
    1) PSK-based authentication, typically managed by the pass-phrase hash method as described in H.4
    2) EAP-based authentication, using credentials that have been issued and preinstalled on the STAs within a common administrative domain, such as a single organization
b) All of the STAs are in direct radio communication. In particular, there is no routing, bridging, or forwarding of traffic by a third STA to effect communication. This assumption is made, because the model makes no provision to protect IBSS topology information from tampering by one of the members.


Copyright © 2007 IEEE. All rights reserved.

### 8.4.8 RSNA key management in an ESS

When the IEEE 802.1X authentication completes successfully, this standard assumes that the STA's IEEE 802.1X Supplicant and the IEEE 802.1X AS will share a secret, called a PMK. The AS transfers the PMK, within the MSK, to the AP, using a technique that is outside the scope of this standard; the derivation of the PMK from the MSK is EAP-method-specific. With the PMK in place, the AP initiates a key confirmation handshake with the STA. The key confirmation handshake sets the IEEE 802.1X state variable portValid (as described in IEEE Std 802.1X-2004) to TRUE.

The key confirmation handshake is implemented by the 4-Way Handshake. The purposes of the 4-Way Handshake are as follows:

a) Confirm the existence of the PMK at the peer.
b) Ensure that the security association keys are fresh.
c) Synchronize the installation of temporal keys into the MAC.
d) Transfer the GTK from the Authenticator to the Supplicant.
e) Confirm the selection of cipher suites.

NOTE 1—Message 1 of the 4-Way Handshake can be forged. However, the forgery attempt will be detected in the failure of the 4-Way Handshake.

NOTE 2—Neither the AP nor the STA can use the PMK for any purpose but the one specified herein without compromising the key. If the AP uses it for another purpose, then the STA can masquerade as the AP; similarly if the STA reuses the PMK in another context, then the AP can masquerade as the STA.

The Supplicant and Authenticator signal the completion of key management by utilizing the MLME-SETKEYS.request primitive to configure the agreed-upon temporal pairwise key into the IEEE 802.11 MAC and by calling the MLME-SETPROTECTION.request primitive to enable its use.

A second key exchange, the Group Key Handshake, is also defined. It distributes a subsequent GTK. The AP's Authenticator can use the Group Key Handshake to update the GTK at the STA's Supplicant. The Group Key Handshake uses the EAPOL-Key frames for this exchange. When it completes, the STA's Supplicant can use the MLME-SETKEYS.request primitive to configure the GTK into the IEEE 802.11 MAC.

### 8.4.9 RSNA key management in an IBSS

To establish a security association between two STAs in an IBSS, each STA's SME must have an accompanying IEEE 802.1X Authenticator and Supplicant. Each STA's SME initiates the 4-Way Handshake from the Authenticator to the peer STA's Supplicant (see 8.4.7). Two separate 4-Way Handshakes are conducted.

The 4-Way Handshake is used to negotiate the pairwise cipher suites, as described in 8.4.4. The IEEE 802.11 SME configures the temporal key portion of the PTK into the IEEE 802.11 MAC. Each Authenticator uses the KCK and KEK portions of the PTK negotiated by the exchange it initiates to distribute its own GTK. Each Authenticator generates its own GTK and uses either the 4-Way Handshake or the Group Key Handshake to transfer the GTK to other STAs with whom it has completed a 4-Way Handshake. The pairwise key used between any two STAs shall be the pairwise key from the 4-Way Handshake initiated by the STA with the highest MAC address.

A STA joining an IBSS is required to adopt the security configuration of the IBSS, which includes the group cipher suite, pairwise cipher suite, and AKMP (see 8.4.4). The STA shall not set up a security association with any STA having a different security configuration. The Beacon and Probe Response frames of the various STAs within an IBSS must reflect a consistent security policy, as the beacon initiation rotates among the STAs.

Copyright © 2007 IEEE. All rights reserved.

A STA joining an IBSS shall support and advertise in the Beacon frame the security configuration of the IBSS, which includes the group cipher suite, advertised pairwise cipher suite, and AKMP (see 8.4.4). The STA may use the Probe Request frame to discover the security policy of a STA, including additional unicast cipher suites the STA supports. A STA shall ignore Beacon frames that advertise a different security policy.

### 8.4.10 RSNA security association termination

When a non-AP STA SME receives a successful MLME Association or Reassociation confirm primitive or receives or invokes an MLME Disassociation or Deauthentication primitive, it will delete some security associations. Similarly, when an AP SME receives an MLME Association or Reassociation indication primitive, or receives or invokes an MLME Disassociation or Deauthentication primitive it will delete some security associations. In the case of an ESS, the non-AP STA's SME shall delete the PTKSA, GTKSA, SMKSA, and any STKSA, and the AP's SME shall delete the PTKSA and invoke an STSL application teardown procedure for any of its STKSAs. An example of an STSL application teardown procedure is described in 11.7.3. In the case of an IBSS, the STA's SME shall delete the PTKSA and the receive GTKSA. Once the security associations have been deleted, the SME then invokes MLME-DELETEKEYS.request primitive to delete all temporal keys associated with the deleted security associations. The IEEE 802.1X Controlled Port returns to being blocked. As a result, all data frames are unauthorized before invocation of an MLME-DELETEKEYS.request primitive.

If a STA loses key state synchronization, it can apply the following rules to recover:

a) Any protected frame(s) received shall be discarded, and MLME-PROTECTEDFRAMEDROPPED.indication primitive is invoked.

b) If the STA is RSNA-enabled and has joined an IBSS, the SME shall execute the authentication procedure as described in 11.3.1.1.

c) If the STA is RSNA-enabled and has joined an ESS, the SME shall execute the deauthentication procedures as described in 11.3.1.3. However, if the STA has initiated the RSN security association, but has not yet invoked the MLME-SETPROTECTION.request primitive, then no additional action is required.

NOTE 1—There is a race condition between when MLME-SETPROTECTION.request primitive is invoked on the Supplicant and when it is invoked on the Authenticator. During this time, an encrypted MPDU may be received that cannot be decrypted; and the MPDU will be discarded without a deauthentication occurring.

NOTE 2—Because the IEEE 802.11 null data MPDU does not derive from an MA-UNITDATA.request, it is not protected.

If the selected AKMP fails between a STA and an AP that are associated, then both the STA and the AP shall invoke the MAC deauthentication procedure described in 11.3.1.3.

If the SMK Handshake fails between a pair of associated STAs and AP, then the STAs and the AP shall invoke an STSL application teardown procedure.

## 8.5 Keys and key distribution

### 8.5.1 Key hierarchy

RSNA defines two key hierarchies:

a) Pairwise key hierarchy, to protect unicast traffic

b) GTK, a hierarchy consisting of a single key to protect multicast and broadcast traffic

NOTE—Pairwise key support with TKIP or CCMP allows a receiving STA to detect MAC address spoofing and data forgery. The RSNA architecture binds the transmit and receive addresses to the pairwise key. If an attacker creates an MPDU with the spoofed TA, then the decapsulation procedure at the receiver will generate an error. GTKs do not have this property.


Copyright © 2007 IEEE. All rights reserved.

The description of the key hierarchies uses the following two functions:

— L(*Str*, *F*, *L*)From *Str* starting from the left, extract bits *F* through *F*+*L*–1, using the IEEE 802.11 bit conventions from 7.1.1.

— PRF-*n* Pseudo-random function producing *n* bits of output, defined in 8.5.1.1.

In an ESS, the IEEE 802.1X Authenticator MAC address (AA) and the AP's BSSID are the same, and the Supplicant's MAC address (SPA) and the STA's MAC address are equal. For the purposes of comparison, the MAC address is encoded as 6 octets, taken to represent an unsigned binary number. The first octet of the MAC address shall be used as the most significant octet. The bit numbering conventions in 7.1.1 shall be used within each octet.

An RSNA STA using CCMP shall support at least one pairwise key for any <TA,RA> pair. The <TA,RA> identifies the pairwise key, which does not correspond to any WEP key identifer.

In a a mixed environment, an AP may simultaneously communicate with some STAs using WEP with shared WEP keys and to STAs using CCMP or TKIP with pairwise keys. The STAs running WEP use default keys 0–3 for shared WEP keys; the important point here is that WEP can still use WEP default key 0. The AP can be configured to use the WEP key in WEP default key 0 for WEP; if the AP is configured in this way, STAs that cannot support WEP default key 0 simultaneously with a TKIP pairwise key shall specify the No Pairwise subfield in the RSN Capabilities field. If an AP is configured to use WEP default key 0 as a WEP key and a "No Pairwise" STA associates, the AP shall not set the Install bit in the 4-Way Handshake. In other words, the STA will not install a pairwise temporal key and instead will use WEP default key 0 for all traffic.

NOTE—The behavior of "No Pairwise" STAs is only intended to support the migration of WEP to RSNA.

TKIP STAs in a mixed environment are expected to support a single pairwise key either by using a key mapping key or by mapping to default key 0. The AP will use a pairwise key for unicast traffic between the AP and the STA. If a key mapping key is available, the <RA,TA> pair identifies the key; if there is no key mapping key, then the default key 0 is used because the key index in the message will be 0.

A STA that cannot support TKIP keys and WEP default key 0 simultaneously advertises this deficiency by setting the No Pairwise subfield in the RSN information element it sends in the (Re)Association Request to the AP. In response, the AP will, in Message 3 of the 4-Way Handshake, clear the Install bit to notify the STA not to install the pairwise key. The AP will instead send the WEP shared key to the STA to be plumbed as the WEP default key 0; this key will then be used with WEP to send and receive unicast traffic between the AP and the STA.

The TKIP STA that has this limitation may not know that it will be forced to use WEP for all transmissions until it has associated with the AP and been given the keys to use. (The STA cannot know that the AP has been configured to use WEP default key 0 for WEP communication.) If this does not satisfy the security policy configured at the STA, the STA's only recourse is to disassociate and try a different AP.

CCMP STAs in a TSN shall support pairwise keys and WEP default key 0 simultaneously. It is invalid for the STA to negotiate the No Pairwise subfield when CCMP is one of the configured ciphers.

#### 8.5.1.1 PRF

A PRF is used in a number of places in this standard. Depending on its use, it may need to output 128 bits, 192 bits, 256 bits, 384 bits, or 512 bits. This subclause defines five functions:

— PRF-128, which outputs 128 bits

— PRF-192, which outputs 192 bits

— PRF-256, which outputs 256 bits

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 245 of 1232

— PRF-384, which outputs 384 bits
— PRF-512, which outputs 512 bits

In the following, *A* is a unique label for each different purpose of the PRF; *Y* is a single octet containing 0; *X* is a single octet containing the parameter; and || denotes concatenation:

H-SHA-1(*K*, *A*, *B*, *X*) ← HMAC-SHA-1(*K*, *A* || *Y* || *B* || *X*)

PRF(*K*, *A*, *B*, *Len*)
**for** *i* ← 0 **to** (*Len*+159)/160 **do**
*R* ← *R* || H-SHA-1(*K*, *A*, *B*, *i)*
**return** L(*R*, 0, Len)

PRF-128(*K*, *A*, *B*) = PRF(*K*, *A*, *B*, 128)
PRF-192(*K*, *A*, *B*) = PRF(*K*, *A*, *B*, 192)
PRF-256(*K*, *A*, *B*) = PRF(*K*, *A*, *B*, 256)
PRF-384(*K*, *A*, *B*) = PRF(*K*, *A*, *B*, 384)
PRF-512(*K*, *A*, *B*) = PRF(*K*, *A*, *B*, 512)

### 8.5.1.2 Pairwise key hierarchy

The pairwise key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from a PMK, as depicted in Figure 8-20. The PMK shall be 256 bits. The pairwise key hierarchy takes a PMK and generates a PTK. The PTK is partitioned into KCK, KEK, and temporal keys, which are used by the MAC to protect unicast communication between the Authenticator's and Supplicant's respective STAs. PTKs are used between a single Supplicant and a single Authenticator.

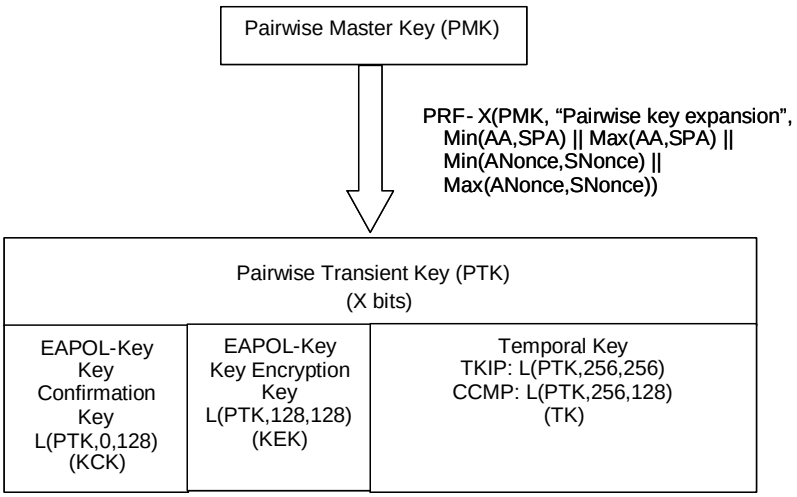



**Figure 8-20—Pairwise key hierarchy**

When not using a PSK, the PMK is derived from the MSK. The PMK shall be computed as the first 256 bits (bits 0–255) of the MSK: PMK ← L(MSK, 0, 256). When this derivation is used, the MSK must consist of at least 256 bits.

The PTK shall not be used longer than the PMK lifetime as determined by the minimum of the PMK lifetime indicated by the AS, e.g., Session-Timeout + dot1xAuthTxPeriod or from the dot11RSNAConfigPMK-Lifetime MIB variable. When RADIUS is used and the Session-Timeout attribute is not in the RADIUS Accept message, and if the key lifetime is not otherwise specified, then the PMK lifetime is infinite.


Copyright © 2007 IEEE. All rights reserved.

NOTE 1—If the protocol between the Authenticator (or AP) and AS is RADIUS, then the MS-MPPE-Recv-Key attribute (vendor-id = 17; see Section 2.4.3 in IETF RFC 2548-1999 [B22]) may be used to transport the PMK to the AP.

NOTE 2—When reauthenticating and changing the pairwise key, a race condition may occur. If a frame is received while MLME-SETKEYS.request primitive is being processed, the received frame may be decrypted with one key and the MIC checked with a different key. Two possible options to avoid this race condition are as follows: the frame can be checked against the old MIC key, and the received frames may be queued while the keys are changed.

NOTE 3—If the AKMP is RSNA-PSK, then a 256-bit PSK may be configured into the STA and AP or a pass-phrase may be configured into the Supplicant or Authenticator. The method used to configure the PSK is outside this standard, but one method is via user interaction. If a pass-phrase is configured, then a 256-bit key is derived and used as the PSK. In any RSNA-PSK method, the PSK is used directly as the PMK. Implementations may support different PSKs for each pair of communicating STAs.

Here, the following assumptions apply:

— SNonce is a random or pseudo-random value contributed by the Supplicant; its value is taken when a PTK is instantiated and is sent to the PTK Authenticator.

— ANonce is a random or pseudo-random value contributed by the Authenticator.

— The PTK shall be derived from the PMK by

PTK ← PRF-X(PMK, “Pairwise key expansion”, Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))

TKIP uses X = 512 and CCMP uses X = 384. The Min and Max operations for IEEE 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The Min and Max operations for nonces are with the nonces treated as positive integers converted as specified in 7.1.1.

NOTE—The Authenticator and Supplicant normally derive a PTK only once per association. A Supplicant or an Authenticator may use the 4-Way Handshake to derive a new PTK. Both the Authenticator and Supplicant create a new nonce value for each 4-Way Handshake instance.

— The KCK shall be computed as the first 128 bits (bits 0–127) of the PTK:

KCK ← L(PTK, 0, 128)

The KCK is used by IEEE Std 802.1X-2004 to provided data origin authenticity in the 4-Way Handshake and Group Key Handshake messages.

— The KEK shall be computed as bits 128–255 of the PTK:

KEK ← L(PTK, 128, 128)

The KEK is used by the EAPOL-Key frames to provide data confidentiality in the 4-Way Handshake and Group Key Handshake messages.

— The temporal key (TK) shall be computed as bits 256–383 (for CCMP) or bits 256–511 (for TKIP) of the PTK:

TK ← L(PTK, 256, 128) or

TK ← L(PTK, 256, 256)

The EAPOL-Key state machines (see 8.5.5 and 8.5.6) use the MLME-SETKEYS.request primitive to configure the temporal key into the STA. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is cipher-suite-specific.

A PMK identifier is defined as

PMKID = HMAC-SHA1-128(PMK, "PMK Name" || AA || SPA)

Here, HMAC-SHA1-128 is the first 128 bits of the HMAC-SHA1 of its argument list.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 247 of 1232

#### 8.5.1.3 Group key hierarchy

The GTK shall be a random number.

Any group master key (GMK) may be reinitialized at a time interval configured into the AP to reduce the exposure of data if the GMK is ever compromised.

The Authenticator can update the GTK for a number of reasons:

a) The Authenticator may change the GTK on disassociation or deauthentication of a STA.

b) An event within the STA's SME can trigger a Group Key Handshake.

Figure 8-21 depicts an example of a relationship among the keys of the group key hierarchy. In this model, the group key hierarchy takes a GMK and generates a GTK. The GTK is partitioned into temporal keys used by the MAC to protect broadcast/multicast communication. GTKs are used between a single Authenticator and all Supplicants authenticated to that Authenticator. The Authenticator derives new GTKs when it wants to update the GTKs.

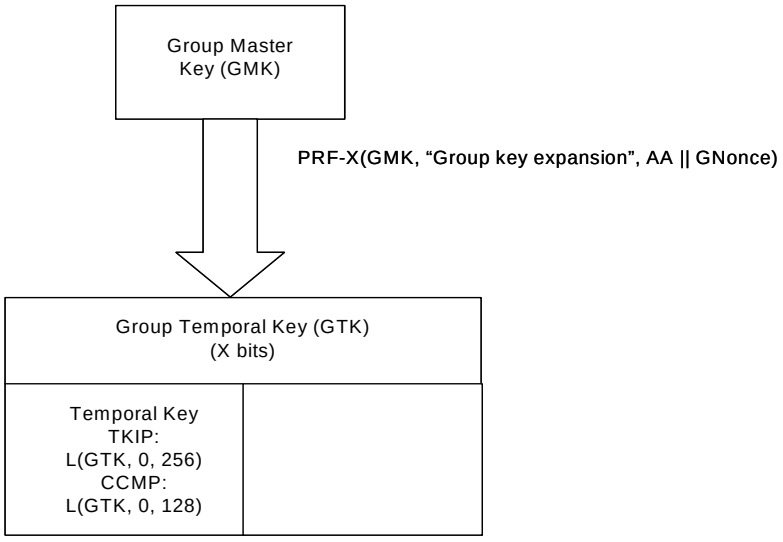



**Figure 8-21—Group key hierarchy (informative)**

In this example, the following assumptions apply:

a) Group nonce (GNonce) is a random or pseudo-random value contributed by the IEEE 802.1X Authenticator.

b) The GTK is derived from the GMK by

c) GTK ← PRF-X(GMK, "Group key expansion" || AA || GNonce)

d) TKIP uses X = 256, CCMP uses X = 128 and WEP use X = 40 or X = 104. AA is represented as an IEEE 802 address and GNonce as a bit string as defined in 7.1.1.

e) The temporal key (TK) is bit 0–39, bits 0–103, bits 0–127, or bits 0–255 of the GTK:
TK ← L(GTK, 0, 40) or
TK ← L(GTK, 0, 104) or


Copyright © 2007 IEEE. All rights reserved.

TK ← L(GTK, 0, 128) or
TK ← L(GTK, 0, 256)

f) The EAPOL-Key state machines (see 8.5.5 and 8.5.6) configure the temporal key into IEEE Std 802.11 via the MLME-SETKEYS.request primitive, and IEEE Std 802.11 uses this key. Its interpretation is cipher-suite-specific.

#### 8.5.1.4 PeerKey key hierarchy

The station-to-station key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from an SMK, as depicted in Figure 8-22. The SMK shall be 256 bits. The pairwise key hierarchy takes an SMK and generates an STK. The STK is partitioned into SKCK, SKEK, and temporal keys, which are used by the MAC to protect unicast communication between the initiator and peer STAs. STKs are used between a single initiator STA and a single peer STA.

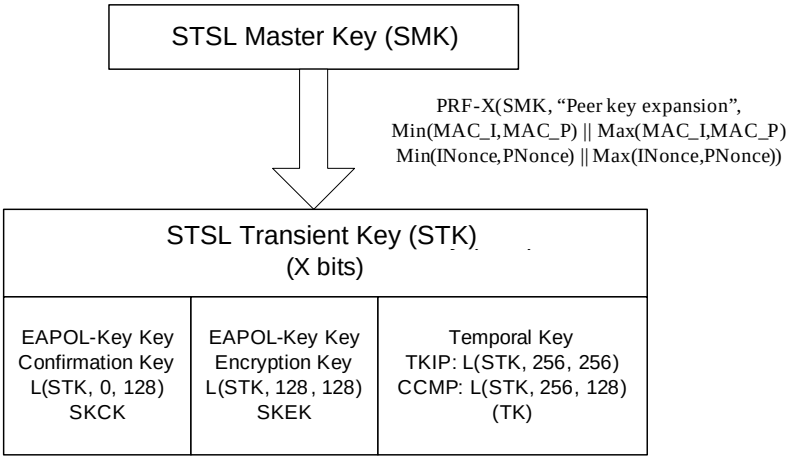



**Figure 8-22—PeerKey hierarchy**

The following apply and are depicted in Figure 8-22:

a) INonce is a random or pseudo-random value contributed by the initiator STA.

b) PNonce is a random or pseudo-random value contributed by the peer STA.

c) The STK shall be derived from the SMK by

STK ← PRF-X(SMK, "Peer key expansion", Min(MAC_I,MAC_P) || Max(MAC_I,MAC_P) || Min(INonce,PNonce) || Max(INonce,PNonce))

TKIP uses X = 512 and CCMP uses X = 384. The Min and Max operations for IEEE 802 addresses are with the address converted to a positive integer treating the first transmitted octet as the most significant octet of the integer. The Min and Max operations for nonces are with the nonces treated as positive integers converted as specified in 7.1.1.

Copyright © 2007 IEEE. All rights reserved.

d) The SKCK shall be computed as the first 128 bits (bits 0–127) of the STK:

SKCK ← L(STK, 0, 128)

The SKCK is used to provide data origin authenticity in the 4-Way STK Handshake.

e) The SKEK shall be computed as bits 128–255 of the STK:

SKEK ← L(STK, 128, 128)

The SKEK is used by the EAPOL-Key frames to provide confidentiality in the 4-Way STK Handshake.

f) The temporal key (TK) shall be computed as bits 256–383 (for CCMP) or bits 256–511 (for TKIP) of the STK:

TK ← L(STK, 256, 128) or TK ← L(STK, 256, 256)

The EAPOL-Key state machines (see 8.5.5 and 8.5.6) use the MLME-SETKEYS.request primitive to configure the temporal key into the STA. The STA uses the temporal key with the pairwise cipher suite; interpretation of this value is cipher-suite-specific.

A SMK identifier is defined as

SMKID = HMAC-SHA1-128(SMK, "SMK Name" || PNonce || MAC_P || INonce || MAC_I)

Here, HMAC-SHA1-128 is the first 128 bits of the HMAC-SHA1 of its argument list.

### 8.5.2 EAPOL-Key frames

IEEE Std 802.11 uses EAPOL-Key frames to exchange information between STAs' Supplicants and Authenticators. These exchanges result in cryptographic keys and synchronization of security association state. EAPOL-Key frames are used to implement three different exchanges:

— 4-Way Handshake, to confirm that the PMK between associated STAs is the same and live and to transfer the GTK to the STA.

— Group Key Handshake, to update the GTK at the STA.

— PeerKey initial SMK Handshake to deliver the SMK and final 4-Way STK Handshake to deliver the STK to the initiating and peer STAs.

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

The RSNA key descriptor used by IEEE Std 802.11 does not use the IEEE 802.1X key descriptor. Instead, it uses the key descriptor described in this subclause.

The bit and octet convention for fields in the EAPOL-Key frame are defined in 7.1 of IEEE Std 802.1X-2004. EAPOL-Key frames containing invalid field values shall be silently discarded. Figure 8-23 depicts the format of an EAPOL-Key frame.


Copyright © 2007 IEEE. All rights reserved.

| Protocol Version – 1 octet | Packet Type – 1 octet | Packet Body Length – 2 octets |
|---|---|---|
| Descriptor Type – 1 octet | | |
| Key Information – 2 octets | | Key Length – 2 octets |
| Key Replay Counter – 8 octets | | |
| Key Nonce – 32 octets | | |
| EAPOL-Key IV – 16 octets | | |
| Key RSC – 8 octets | | |
| Reserved - 8 octets | | |
| Key MIC – 16 octets | | |
| Key Data Length – 2 octets | | Key Data – n octets |

**Figure 8-23—EAPOL-Key frame**

The fields of a EAPOL-Key frame body are as follows:

a) **Descriptor Type.** This field is one octet and has a value defined by IEEE Std 802.1X-2004, identifying the IEEE 802.11 key descriptor.

b) **Key Information**. This field is 2 octets and specifies characteristics of the key. See Figure 8-24.

| B0 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | B12 | B13 | B14 | B15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Key Descriptor Version | Key Type | | Reserved | | Install | Key Ack | Key MIC | Se-cure | Error | Request | Encrypted Key Data | SMK Message | Reserved | |

**Figure 8-24—Key Information bit layout**

The bit convention used is as in 7.1 of IEEE Std 802.1X-2004. The subfields of the Key Information field are as follows:

1) Key Descriptor Version (bits 0–2) specifies the key descriptor version type.
   i) The value 1 shall be used for all EAPOL-Key frames to and from a STA when neither the group nor pairwise ciphers are CCMP for Key Descriptor 1. This value indicates the following:
      — HMAC-MD5 is the EAPOL-Key MIC.
      — ARC4 is the EAPOL-Key encryption algorithm used to protect the Key Data field.
   ii) The value 2 shall be used for all EAPOL-Key frames to and from a STA when either the pairwise or the group cipher is AES-CCMP for Key Descriptor 2. This value indicates the following:
      — HMAC-SHA1-128 is the EAPOL-Key MIC. HMAC is defined in IETF RFC 2104; and SHA1, by FIPS PUB 180-1-1995. The output of the HMAC-SHA1 shall be truncated to its 128 MSBs (octets 0–15 of the digest output by HMAC-SHA1), i.e., the last four octets generated shall be discarded.
      — The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.
2) Key Type (bit 3) specifies whether this EAPOL-Key frame is part of a 4-Way Handshake deriving a PTK.

Copyright © 2007 IEEE. All rights reserved.

i) The value 0 (Group/SMK) indicates the message is not part of a PTK derivation.

ii) The value 1 (Pairwise) indicates the message is part of a PTK derivation.

3) Reserved (bits 4–5). The sender shall set them to 0, and the receiver shall ignore the value of these bits.

4) Install (bit 6).

i) If the value of Key Type (bit 3) is 1, then for the Install bit,

— The value 1 means the IEEE 802.1X component shall configure the temporal key derived from this message into its IEEE 802.11 STA.

— The value 0 means the IEEE 802.1X component shall not configure the temporal key into the IEEE 802.11 STA.

ii) If the value of Key Type (bit 3) is 0, then this bit shall be 0 on transmit and ignored on receive.

5) Key Ack (bit 7) is set in messages from the Authenticator if an EAPOL-Key frame is required in response to this message and is clear otherwise. The Supplicant's response to this message shall use the same replay counter as this message.

6) Key MIC (bit 8) is set if a MIC is in this EAPOL-Key frame and is clear if this message contains no MIC.

7) Secure (bit 9) is set once the initial key exchange is complete.

The Authenticator shall set the Secure bit to 0 in all EAPOL-Key frames sent before the Supplicant has the PTK and the GTK. The Authenticator shall set the Secure bit to 1 in all EAPOL-Key frames it sends to the Supplicant containing the last key needed to complete the Supplicant's initialization.

The Supplicant shall set the Secure bit to 0 in all EAPOL-Key frames it sends before it has the PTK and the GTK and before it has received an EAPOL-Key frame from the Authenticator with the Secure bit set to 1 (this should be before receiving Message 3 of the 4-Way Handshake). The Supplicant shall set the Secure bit to 1 in all EAPOL-Key Frames sent after this until it loses the security association it shares with the Authenticator.

8) Error (bit 10) is set by a Supplicant to report that a MIC failure occurred in a TKIP MSDU or SMK Handshake failure. In case of a MIC failure, a Supplicant shall set this bit only when the Request (bit 11) is set. In the case that the SMK Message bit is set, Error shall be set to indicate the key data field contains an Error KDE.

9) Request (bit 11) is set by a Supplicant to request that the Authenticator initiate either a 4-Way Handshake or Group Key Handshake, is set by a Supplicant in a Michael MIC Failure Report, and is set by the STSL peer STA to request initiator STA rekeying of the STK. The Supplicant shall not set this bit in on-going 4-Way Handshakes, i.e., the Key Ack bit (bit 7) shall not be set in any message with the Request bit set. The Authenticator shall never set this bit.

In a Michael MIC Failure Report, setting the bit is not a request to initiate a new handshake. However the recipient may initiate a new handshake on receiving such a message.

If the EAPOL-Key frame with Request bit set has a key type of Pairwise, the Authenticator shall initiate a 4-Way Handshake. If the EAPOL-Key frame with Request bit set has a key type of Group/SMK, the Authenticator shall change the GTK, initiate a 4-Way Handshake with the Supplicant, and then execute the Group Key Handshake to all Supplicants.

If the EAPOL-Key frame with Request bit set has the SMK Message bit set, the initiator STA shall take appropriate action to create a new STK (based on 8.5.8).

10) Encrypted Key Data (bit 12) is set if the Key Data field is encrypted and is clear if the Key Data field is not encrypted. This subfield shall be set, and the Key Data field shall be encrypted, if any key material (e.g., GTK or SMK) is included in the frame.


Copyright © 2007 IEEE. All rights reserved.

11) SMK Message (bit 13) specifies whether this EAPOL-Key frame is part of an SMK Handshake. If the SMK Handshake is not supported, the STA shall set the SMK message bit to 0 and shall ignore the value of this bit upon receipt.

12) Reserved (bits 14–15). The sender shall set them to 0, and the receiver shall ignore the value of these bits.

c) **Key Length**. This field is 2 octets in length, represented as an unsigned binary number. The value defines the length in octets of the pairwise temporal key to configure into IEEE Std 802.11. See Table 8-2.

**Table 8-2—Cipher suite key lengths**

| **Cipher suite** | CCMP | TKIP | WEP-40 | WEP-104 |
|---|---|---|---|---|
| **Key length (octets)** | 16 | 32 | 5 | 13 |

d) **Key Replay Counter**. This field is 8 octets, represented as an unsigned binary number, and is initialized to 0 when the PMK is established. The Supplicant shall use the key replay counter in the received EAPOL-Key frame when responding to an EAPOL-Key frame. It carries a sequence number that the protocol uses to detect replayed EAPOL-Key frames.

The Supplicant and Authenticator shall track the key replay counter per security association. The key replay counter shall be initialized to 0 on (re)association. The Authenticator shall increment the key replay counter on each successive EAPOL-Key frame.

When replying to a message from the Authenticator, the Supplicant shall use the Key Replay Counter field value from the last valid EAPOL-Key frames received from the Authenticator. The Authenticator should use the key replay counter to identify invalid messages to silently discard. The Supplicant should also use the key replay counter and ignore EAPOL-Key frames with a Key Replay Counter field value smaller than or equal to any received in a valid message. The local Key Replay Counter field should not be updated until the after EAPOL-Key MIC is checked and is valid. In other words, the Supplicant never updates the Key Replay Counter field for Message 1 in the 4-Way Handshake, as it includes no MIC. This implies the Supplicant must allow for retransmission of Message 1 when checking for the key replay counter of Message 3.

The Supplicant shall maintain a separate key replay counter for sending EAPOL-Key request frames to the Authenticator; the Authenticator also shall enforce monotonicity of a separate replay counter to filter received EAPOL-Key Request frames.

NOTE—The key replay counter does not play any role beyond a performance optimization in the 4-Way Handshake. In particular, replay protection is provided by selecting a never-before-used nonce value to incorporate into the PTK. It does, however, play a useful role in the Group Key Handshake.

e) **Key Nonce**. This field is 32 octets. It conveys the ANonce from the Authenticator and the SNonce from the Supplicant. It may contain 0 if a nonce is not required to be sent.

f) **EAPOL-Key IV**. This field is 16 octets. It contains the IV used with the KEK. It shall contain 0 when an IV is not required. It should be initialized by taking the current value of the global key counter (see 8.5.6) and then incrementing the counter. Note that only the lower 16 octets of the counter value will be used.

g) **Key RSC**. This field is 8 octets in length. It contains the receive sequence counter (RSC) for the GTK being installed in IEEE Std 802.11. It is used in Message 3 of the 4-Way Handshake and Message 1 of the Group Key Handshake, where it is used to synchronize the IEEE 802.11 replay state. It may also be used in the Michael MIC Failure Report frame, to report the TSC field value of the frame experiencing a MIC failure. It shall contain 0 in other messages. The Key RSC field gives the current message number for the GTK, to allow a STA to identify replayed MPDUs. If the Key RSC field value is less than 8 octets in length, the remaining octets shall be set to 0. The least significant octet of the TSC or PN should be in the first octet of the Key RSC field.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 253 of 1232

NOTE—The Key RSC field value for TKIP is the TSC in the first 6 octets and for CCMP is the PN in the first 6 octets. See Table 8-3.

**Table 8-3—Key RSC field**

| KeyRSC 0 | KeyRSC 1 | KeyRSC 2 | KeyRSC 3 | KeyRSC 4 | KeyRSC 5 | KeyRSC 6 | KeyRSC 7 |
|---|---|---|---|---|---|---|---|
| TSC0 | TSC1 | TSC2 | TSC3 | TSC4 | TSC5 | 0 | 0 |
| PN0 | PN1 | PN2 | PN3 | PN4 | PN5 | 0 | 0 |

For WEP, the Key RSC value shall be set to 0 on transmit and shall not be used at the receiver.

h) **Key MIC**. This field is 16 octets in length when the Key Descriptor Version subfield is 1 or 2. The EAPOL-Key MIC is a MIC of the EAPOL-Key frames, from and including the EAPOL protocol version field to and including the Key Data field, calculated with the Key MIC field set to 0. If the Encrypted Key Data subfield (of the Key Information field) is set, the Key Data field is encrypted prior to computing the MIC.

1) **Key Descriptor Version 1**: HMAC-MD5; IETF RFC 2104 and IETF RFC 1321 together define this function.

2) **Key Descriptor Version 2**: HMAC-SHA1-128.

i) **Key Data Length**. This field is 2 octets in length, taken to represent an unsigned binary number. This represents the length of the Key Data field in octets. If the Encrypted Key Data subfield (of the Key Information field) is set, the length is the length of the Key Data field after encryption, including any padding.

j) **Key Data**. This field is a variable-length field that is used to include any additional data required for the key exchange that is not included in the fields of the EAPOL-Key frame. The additional data may be zero or more information element(s) (such as the RSN information element) and zero or more key data cryptographic encapsulation(s) (KDEs) (such as GTK(s) or PMKID(s)). Information elements sent in the Key Data field include the Element ID and Length subfields. KDEs shall be encapsulated using the format in Figure 8-25.

| Type (0xdd) | Length | OUI | Data Type | Data |
|---|---|---|---|---|
| 1 octet | 1 octet | 3 octets | 1 octet | (Length – 4) octets |

**Figure 8-25—KDE format**

The Type field shall be set to 0xdd. The Length field specifies the number of octets in the OUI, Data Type, and Data fields. The order of the OUI field shall follow the ordering convention for MAC addresses from 7.1.1.

Table 8-4 lists the KDE selectors defined by this standard.

**Table 8-4—KDE**

| OUI | Data type | Meaning |
|---|---|---|
| 00-0F-AC | 0 | Reserved |
| 00-0F-AC | 1 | GTK KDE |
| 00-0F-AC | 2 | Reserved |
| 00-0F-AC | 3 | MAC address KDE |
| 00-0F-AC | 4 | PMKID KDE |


Copyright © 2007 IEEE. All rights reserved.

**Table 8-4—KDE *(continued)***

| OUI | Data type | Meaning |
|---|---|---|
| 00-0F-AC | 5 | SMK KDE |
| 00-0F-AC | 6 | Nonce KDE |
| 00-0F-AC | 7 | Lifetime KDE |
| 00-0F-AC | 8 | Error KDE |
| 00-0F-AC | 9–255 | Reserved |
| Vendor OUI | Any | Vendor specific |
| Other | Any | Reserved |

STAs shall ignore any information elements and KDEs they do not understand.

If the Encrypted Key Data subfield (of the Key Information field) is set, the entire Key Data field shall be encrypted. If the Key Data field uses the NIST AES key wrap, then the Key Data field shall be padded before encrypting if the key data length is less than 16 octets or if it is not a multiple of 8. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received EAPOL-Key message, the receiver shall ignore this trailing padding. Key Data fields that are encrypted, but do not contain the GroupKey or SMK KDE, shall be accepted.

If the GroupKey or SMK KDE is included in the Key Data field, but the Key Data field is not encrypted, the EAPOL-Key frames shall be ignored.

The format of the GTK KDE is shown in Figure 8-26.

| KeyID (0,1,2, or 3) | Tx | Reserved (0) | Reserved (0) | GTK |
|---|---|---|---|---|
| bits 0–1 | bit 2 | bit 3–7 | 1 octet | (Length – 6) octets |

**Figure 8-26—GTK KDE format**

If the value of the Tx field is 1, then the IEEE 802.1X component shall configure the temporal key derived from this KDE into its IEEE 802.11 STA for both transmission and reception.

If the value of the Tx field is 0, then the IEEE 802.1X component shall configure the temporal key derived from this KDE into its IEEE 802.11 STA for reception only.

The format of the MAC address KDE is shown in Figure 8-27.

| MAC Address |
|---|
| 6 octets |

**Figure 8-27—MAC address KDE format**

Copyright © 2007 IEEE. All rights reserved.

The format of the PMKID KDE is shown in Figure 8-28.

| PMKID |
|---|
| 16 octets |

**Figure 8-28—PMKID KDE format**

The format of the SMK KDE is shown in Figure 8-29.

| SMK | Key Nonce |
|---|---|
| 32 octets | 32 octets |

**Figure 8-29—SMK KDE format**

The format of the Nonce KDE is shown in Figure 8-30.

| Key Nonce |
|---|
| 32 octets |

**Figure 8-30—Nonce KDE format**

The format of the Lifetime KDE is shown in Figure 8-31. The Key Lifetime value is expressed in seconds and uses big endian byte order.

| Key Lifetime (in seconds) |
|---|
| 4 octets |

**Figure 8-31—Lifetime KDE format**

The format of the Error KDE is shown in Figure 8-32. Both MUI and Error Type fields are in big endian byte order. Table 8-5 shows different values of MUI, and Table 8-6 shows different values of SMK error types.

| MUI | Error Type |
|---|---|
| 2 octets | 2 octets |

**Figure 8-32—Error KDE format**

**Table 8-5—MUI values**

| Handshake type | MUI value |
|---|---|
| 4-Way PTK Handshake | 00–01 |
| 4-Way STK Handshake | 00–02 |
| GTK Handshake | 00–03 |
| SMK Handshake | 00–04 |


Copyright © 2007 IEEE. All rights reserved.

**Table 8-6—SMK error types**

| Error name | Error type | Error meaning |
|---|---|---|
| ERR_STA_NR | 1 | STA is not reachable from AP. See 8.5.9.5.1 |
| ERR_STA_NRSN | 2 | STA to AP secure network not present. See 8.5.9.5.2 |
| ERR_CPHR_NS | 3 | Cipher suites not supported. See 8.5.8.4.3. |
| ERR_NO_STSL | 4 | No STSL session present. See 8.5.8.4.4. |

The following EAPOL-Key frames are used to implement the three different exchanges:

— **4-Way Handshake Message 1** is an EAPOL-Key frame with the Key Type subfield set to 1. The Key Data field shall contain an encapsulated PMKID for the PMK that is being used in this key derivation and need not be encrypted.

— **4-Way Handshake Message 2** is an EAPOL-Key frame with the Key Type subfield set to 1. The Key Data field shall contain an RSN information element and need not be encrypted.

An ESS Supplicant's SME shall insert the RSN information element it sent in its (Re)Association Request frame. The RSN information element is included as transmitted in the management frame. On receipt of Message 2, the Authenticator's SME shall validate the selected security configuration against the RSN information element received in the IEEE 802.11 (Re)Association Request.

An IBSS Supplicant's SME shall insert an RSN information element containing the pairwise cipher suite select it wants to negotiate. The Authenticator's SME shall validate that the pairwise cipher suite selected is one of its configured cipher suites and that the group cipher suite and AKM are consistent.

— **4-Way Handshake Message 3** is an EAPOL-Key frame with the Key Type subfield set to 1. The Key Data field shall contain one or two RSN information elements. If a group cipher has been negotiated, this field shall also include an encapsulated GTK. This field shall be encrypted if a GTK is included.

An Authenticator's SME shall insert the RSN information element it sent in its Beacon or Probe Response frame. The Supplicant's SME shall validate the selected security configuration against the RSN information element received in Message 3. If the second optional RSN information element is present, the STA shall either use that cipher suite with its pairwise key or deauthenticate. In either case, if the values do not match, then the receiver shall consider the RSN information element modified and shall use the MLME-DEAUTHENTICATE.request primitive to break the association. A security error should be logged at this time.

It may happen, for example, that a STA's Supplicant selects a pairwise cipher suite which is advertised by an AP, but which policy disallows for this particular STA. An Authenticator may, therefore, insert a second RSN information element to overrule the STA's selection. An Authenticator's SME shall insert the second RSN information element, after the first RSN information element, only for this purpose. The pairwise cipher suite in the second RSN information element included shall be one of the ciphers advertised by the Authenticator. All other fields in the second RSN information element shall be identical to the first RSN information element.

An encapsulated GTK shall be included and the unencrypted length of the GTK is six less than the length of the GTK KDE in octets. The entire Key Data field shall be encrypted as specified by the key descriptor version.

— **4-Way Handshake Message 4** is an EAPOL-Key frame with the Key Type subfield set to 1. The Key Data field can be empty.

— **Group Key Handshake Message 1** is an EAPOL-Key frame with the Key Type subfield set to 0. The Key Data field shall contain a GTK KDE and shall be encrypted.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 257 of 1232

— **Group Key Handshake Message 2** is an EAPOL-Key frame with the Key Type subfield set to 0. The Key Data field can be empty.

PeerKey Handshake Messages use EAPOL-Key frames as defined in 8.5.8.

The key wrap algorithm selected depends on the key descriptor version:

— **Key Descriptor Version 1**: ARC4 is used to encrypt the Key Data field using the KEK field from the derived PTK. No padding shall be used. The encryption key is generated by concatenating the EAPOL-Key IV field and the KEK. The first 256 octets of the ARC4 key stream shall be discarded following ARC4 stream cipher initialization with the KEK, and encryption begins using the 257$^{th}$ key stream octet.

— **Key Descriptor Version 2**: AES key wrap, defined in IETF RFC 3394, shall be used to encrypt the Key Data field using the KEK field from the derived PTK. The key wrap default initial value shall be used.

NOTE—The cipher text output of the AES key wrap algorithm is 8 octets longer than the plaintext input.

#### 8.5.2.1 EAPOL-Key frame notation

The following notation is used throughout the remainder of 8.5 to represent EAPOL-Key frames:

EAPOL-Key(S, M, A, I, K, SM, KeyRSC, ANonce/SNonce, MIC, DataKDs)

where

| | |
|---|---|
| S | means the initial key exchange is complete. This is the Secure bit of the Key Information field. |
| M | means the MIC is available in message. This should be set in all messages except Message 1 of a 4-Way Handshake. This is the Key MIC bit of the Key Information field. |
| A | means a response is required to this message. This is used when the receiver should respond to this message. This is the Key Ack bit of the Key Information field. |
| I | is the Install bit: Install/Not install for the pairwise key. This is the Install bit of the Key Information field. |
| K | is the key type: P (Pairwise), G (Group/SMK). This is the Key Type bit of the Key Information field. |
| SM | is the SMK Message bit: indicates that this message is part of SMK Handshake. |
| KeyRSC | is the key RSC. This is the Key RSC field. |
| ANonce/SNonce | is the Authenticator/Supplicant nonce. This is the Key Nonce field. |
| MIC | is the integrity check, which is generated using the KCK. This is the Key MIC field. |
| DataKDs | is a sequence of zero or more information elements and KDEs, contained in the Key Data field, which may contain the following: |
| RSNIE | is the RSN information element, described in 7.3.2.25. |
| GTK[N] | is the GTK, with the key identifier field set to N. The key identifier specifies which index is used for this GTK. Index 0 shall not be used for GTKs, except in mixed environments, as described in 8.5.1. |
| PMKID | is of type PMKID KDE and is the key identifier used during 4-Way PTK Handshake for PMK key identification and during 4-Way STK Handshake for SMK key identification. |
| Lifetime | is the key lifetime KDE used for sending the expiry timeout value for SMK used during PeerKey Handshake for SMK identification. |
| Initiator MAC | is the Initiator MAC KDE used during PeerKey Handshake |
| Peer MAC | is the Peer MAC KDE used during PeerKey Handshake |
| Initiator Nonce | is the Initiator Nonce KDE used during PeerKey Handshake. This is used when multiple nonces need to be sent. |


Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| Peer Nonce | is the Peer Nonce KDE used during PeerKey Handshake. This is used when multiple nonces need to be sent. |
| SMK KDE | is the encapsulated SMK during SMK Handshake. |
| Error KDE | is an error KDE used when error bit E is set during PeerKey Handshake. |

### 8.5.3 4-Way Handshake

RSNA defines a protocol using IEEE 802.1X EAPOL-Key frames called the 4-Way Handshake. The handshake completes the IEEE 802.1X authentication process. The information flow of the 4-Way Handshake is as follows:

Message 1: Authenticator → Supplicant: EAPOL-Key(0,0,1,0,P,0,0,ANonce,0,DataKD_M1)
where DataKD_M1 = 0 or PMKID for PTK generation, or PMKID KDE (for sending SMKID) for STK generation

Message 2: Supplicant → Authenticator: EAPOL-Key(0,1,0,0,P,0,0,SNonce,MIC,DataKD_M2)
where DataKD_M2 = RSNIE for creating PTK generation or peer RSNIE, Lifetime KDE, SMKID KDE (for sending SMKID) for STK generation

Message 3: Authenticator → Supplicant:
EAPOL-Key(1,1,1,1,P,0,KeyRSC,ANonce,MIC,DataKD_M3)
where DataKD_M3 = RSNIE,GTK[N] for creating PTK generation or initiator RSNIE, Lifetime KDE for STK generation

Message 4: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,P,0,0,0,MIC,DataKD_M4)
where DataKD_M4 = 0.

Here, the following assumptions apply:

— EAPOL-Key(·) denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 8.5.2.1.

— ANonce is a nonce that the Authenticator contributes for PTK generation or that the initiator STA contributes for STK generation. ANonce has the same value in Message 1 and Message 3.

— SNonce is a nonce from the Supplicant for PTK generation or from the peer STA for STK generation.

— P means the pairwise bit is set.

— The MIC is computed over the body of the EAPOL-Key frame (with the Key MIC field first zeroed before the computation) using the KCK defined in 8.5.1.2 for PTK generation or SKCK defined in 8.5.1.4.

— RSNIE represents the appropriate RSN information elements.

— GTK[N] represents the encapsulated GTK with its key identifier.

— SMKID represents the SMKID key identifier used during STK generation.

— Lifetime represents the expiry timeout used for exchanging SMK expiry value.

NOTE—While the MIC calculation is the same in each direction, the Key Ack bit is different in each direction. It is set in EAPOL-Key frames from the Authenticator and clear in EAPOL-Key frames from the Supplicant. 4-Way Handshake requests from the Supplicant have the Request bit set. The Authenticator and Supplicant must check these bits to stop reflection attacks. Message 1 contents must not update state, in particular the keys in use, until the data are validated with Message 3.

#### 8.5.3.1 4-Way Handshake Message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 259 of 1232

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 1 (Pairwise)

SMK Message = 0

Install = 0

Key Ack = 1

Key MIC = 0

Secure = 0

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 8-2

Key Replay Counter = *n* – to allow Authenticator or initiator STA to match the right Message 2 from Supplicant or peer STA

Key Nonce = ANonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = 0

Key Data Length = length of Key Data field in octets

Key Data = PMKID for the PMK being used during PTK generation or SMKID for SMK being used during STK generation

Processing for PTK generation is as follows:

The Authenticator sends Message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new key. On reception of Message 1, the Supplicant determines whether the Key Replay Counter field value has been used before with the current PMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the Supplicant discards the message. Otherwise, the Supplicant

a) Generates a new nonce SNonce.

b) Derives PTK.

c) Constructs Message 2.

Processing for STK generation is as follows:

The initiator STA (STA_I) sends Message 1 to the peer STA (STA_P) at the end of a successful SMK Handshake, when SMKSA is created. On reception of Message 1, the STA_P determines whether the Key Replay Counter field value has been used before with the current SMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the STA_P discards the message. Otherwise, the STA_P

a) Generates 256-bit random number that is sent as a peer Nonce as part Key Nonce field. This Nonce is different from the peer Nonce generated as part SMK Handshake Message 3.

b) Derives STK.

c) Constructs Message 2.


Copyright © 2007 IEEE. All rights reserved.

#### 8.5.3.2 4-Way Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 – same as Message 1

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 0 – same as Message 1

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n* – to let the Authenticator or initiator STA know to which Message 1 this corresponds

Key Nonce = SNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data = included RSNIE – the sending STA's RSNIE for PTK generation or peer RSNIE, Lifetime of SMK and SMKID for STK generation

Processing for PTK generation is as follows:

The Supplicant sends Message 2 to the Authenticator.

On reception of Message 2, the Authenticator checks that the key replay counter corresponds to the outstanding Message 1. If not, it silently discards the message. Otherwise, the Authenticator

a) Derives PTK.
b) Verifies the Message 2 MIC.
    1) If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame, the Authenticator silently discards Message 2.
    2) If the MIC is valid, the Authenticator checks that the RSN information element bit-wise matches that from the (Re)Association Request message.
        i) If these are not exactly the same, the Authenticator uses MLME-DEAUTHENTICATE.request primitive to terminate the association.
        ii) If they do match bit-wise, the Authenticator constructs Message 3.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 261 of 1232

Processing for STK generation is as follows:

The STA_P sends Message 2 to the STA_I. On reception of Message 2, the STA_I checks that the key replay counter corresponds to Message 1. If not, it silently discards the message. Otherwise, the STA_I

a) Derives the STK.

b) Verifies the Message 2 MIC using SKCK key. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 2.

c) If the MIC is valid, the STA_I checks that the RSNIE bit-wise matches that from the SMK Handshake Message 5. If these are not exactly the same, STA_I silently discards the message and restarts the 4-Way Handshake after deleting the existing 4-Way Handshake states.

d) If they do match bit-wise, the STA_I checks SMKID with the value of SMKID in SMKSA. If these are not exactly the same, STA_I silently discards the message and restarts the 4-Way Handshake after deleting the existing 4-Way Handshake states.

e) If they do match, the STA_I constructs Message 3. It also compares the Key Lifetime value from the KDE with value in its SMKSA. If value in its SMKSA is less, it discards the value received in Message 2. Otherwise, it updates the value in SMKSA with value in Message 2.

#### 8.5.3.3 4-Way Handshake Message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 - same as Message 1

Install = 0/1 – For PTK generation, 0 only if the AP does not support key mapping keys, or if the STA has the No Pairwise bit (in the RSN Capabilities field) set and only the group key will be used. For STK generation, this bit is set to 1.

Key Ack = 1

Key MIC = 1

Secure = 1 (keys installed)

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 1

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 8-2

Key Replay Counter = *n+1*

Key Nonce = ANonce – same as Message 1

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = For PTK generation, starting sequence number that the Authenticator's STA will use in MPDUs protected by GTK. For STK generation, this is set to 0.

Key MIC = MIC(KCK, EAPOL) or MIC(SKCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets of included RSN information elements and GTK


Copyright © 2007 IEEE. All rights reserved.

Key Data = For PTK generation, the AP's Beacon/Probe Response frame's RSN information element, and, optionally, a second RSN information element that is the Authenticator's pairwise cipher suite assignment, and, if a group cipher has been negotiated, the encapsulated GTK and the GTK's key identifier (see 8.5.2). For STK generation Initiator RSNIE, Lifetime of SMK is used.

Processing for PTK generation is as follows:

The Authenticator sends Message 3 to the Supplicant.

On reception of Message 3, the Supplicant silently discards the message if the Key Replay Counter field value has already been used or if the ANonce value in Message 3 differs from the ANonce value in Message 1. The Supplicant also

a) Verifies the RSNIE. If it is not identical to that the STA received in the Beacon or Probe Response frame, the STA shall disassociate. If a second RSN information element is provided in the message, the Supplicant uses the pairwise cipher suite specified in the second RSNIE or deauthenticates.
b) Verifies the Message 3 MIC. If the calculated MIC does not match the MIC that the Authenticator included in the EAPOL-Key frame, the Supplicant silently discards Message 3.
c) Updates the last-seen value of the Key Replay Counter field.
d) Constructs Message 4.
e) Sends Message 4 to the Authenticator.
f) Uses the MLME-SETKEYS.request primitive to configure the IEEE 802.11 MAC to send and receive Class 3 unicast MPDUs protected by the PTK. The GTK is also configured by MLME-SETKEYS primitive.

Processing for STK generation is as follows:

The STA_I sends Message 3 to the STA_P. On reception of Message 3, the STA_P silently discards the message if the Key Replay Counter field value has already been used or if the INonce value in Message 3 differs form the INonce value in Message 1. Otherwise,

a) The STA_P verifies the Message 3 MIC using SKCK key in SMKSA. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 3.
b) If the MIC is valid, the STA_P checks that the RSNIE bit-wise matches that from the 4-Way Handshake Message 2. If these are not exactly the same, STA_P silently discards the message and deletes existing 4-Way Handshake states.
c) If they do match, the STA_P constructs Message 4. It also compares the Key Lifetime value from KDE with value in its SMKSA. If value in SMKSA is less, it discards the value received in Message 3. Otherwise, it updates the value in SMKSA with value in Message 3.

#### 8.5.3.4 4-Way Handshake Message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

SMK Message = 0 - same as Message 1

Install = 0

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 263 of 1232

Key Ack = 0 – this is the last message

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n+1*

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL) or MIC(SKCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length of Key Data field in octets

Key Data = none required

Processing for PTK generation is as follows:

The Supplicant sends Message 4 to the Authenticator. Note that when the 4-Way Handshake is first used, Message 4 is sent in the clear.

On reception of Message 4, the Authenticator verifies that the Key Replay Counter field value is one that it used on this 4-Way Handshake; if it is not, it silently discards the message. Otherwise:

a) The Authenticator checks the MIC. If the calculated MIC does not match the MIC that the Supplicant included in the EAPOL-Key frame, the Authenticator silently discards Message 4.

b) If the MIC is valid, the Authenticator uses the MLME-SETKEYS.request primitive to configure the PTK into the IEEE 802.11 MAC.

c) The Authenticator updates the Key Replay Counter field so that it will use a fresh value if a rekey becomes necessary.

Processing for STK generation is as follows:

The STA_P sends Message 4 to the STA_I. On reception of Message 4, the STA_I verifies that the Key Replay Counter field value is one that it used on this 4-Way Handshake; if it is not, it silently discards the message. Otherwise,

a) The STA_I checks the MIC. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the STA_I silently discards Message 4.

b) If the MIC is valid, the STA_I configures the STK into the IEEE 802.11 MAC.

c) The STA_I updates the Key Replay Counter field so that it will use a fresh value if a rekey becomes necessary.

#### 8.5.3.5 4-Way Handshake implementation considerations

When the 4-Way Handshake is used as part of the STK Handshake, the initiator STA acts as Authenticator and peer STA acts as Supplicant.


Copyright © 2007 IEEE. All rights reserved.

If the Authenticator does not receive a reply to its messages, it shall attempt dot11RSNAConfigPairwiseUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval, then 100 ms shall be used for all timeout values. If it still has not received a response after these retries, then for PTK generation the Authenticator should deauthenticate the STA, and for STK generation the STAs should delete the SMKSA and initiate an STSL application teardown procedure.

For PTK generation, if the STA does not receive Message 1 within the expected time interval (prior to IEEE 802.1X timeout), it should disassociate, deauthenticate, and try another AP/STA. For STK generation, if the peer STA does not receive Message 1 or Message 3 within the expected time interval (prior to dot11RSNAConfigSATimeout as specified in 8.5.8), it deletes the SMKSA and invokes an STSL application teardown procedure.

The Authenticator should ignore EAPOL-Key frames it is not expecting in reply to messages it has sent or EAPOL-Key frames with the Ack bit set. This stops an attacker from sending the first message to the Supplicant who responds to the Authenticator.

An implementation should save the KCK and KEK beyond the 4-Way Handshake, as they are needed for Group Key Handshakes, STK Rekeying, and recovery from TKIP MIC failures.

The Supplicant uses the MLME-SETKEYS.request primitive to configure the temporal key from 8.5.1 into its STA after sending Message 4 to the Authenticator.

NOTE 1—If the RSN information element check for Message 2 or Message 3 fails, IEEE Std 802.1X-2004 should log an error and deauthenticate the peer.

NOTE 2—The Supplicant should check that if the RSN information element specified a pairwise cipher suite, then the 4-Way Handshake did specify to configure the temporal key portion of the PTK into the IEEE 802.11 STA.

#### 8.5.3.6 Sample 4-Way Handshake

The following is an informative sample of a 4-Way Handshake for illustration.

After IEEE 802.1X authentication completes by the AP sending an EAP-Success, the AP initiates the 4-Way Handshake. See Figure 8-33.

The 4-Way Handshake consists of the following steps:

a) The Authenticator sends an EAPOL-Key frame containing an ANonce.
b) The Supplicant derives a PTK from ANonce and SNonce.
c) The Supplicant sends an EAPOL-Key frame containing SNonce, the RSN information element from the (Re)Association Request frame, and a MIC.
d) The Authenticator derives PTK from ANonce and SNonce and validates the MIC in the EAPOL-Key frame.
e) The Authenticator sends an EAPOL-Key frame containing ANonce, the RSN information element from its Beacon or Probe Response messages, MIC, whether to install the temporal keys, and the encapsulated GTK.
f) The Supplicant sends an EAPOL-Key frame to confirm that the temporal keys are installed.

Copyright © 2007 IEEE. All rights reserved.

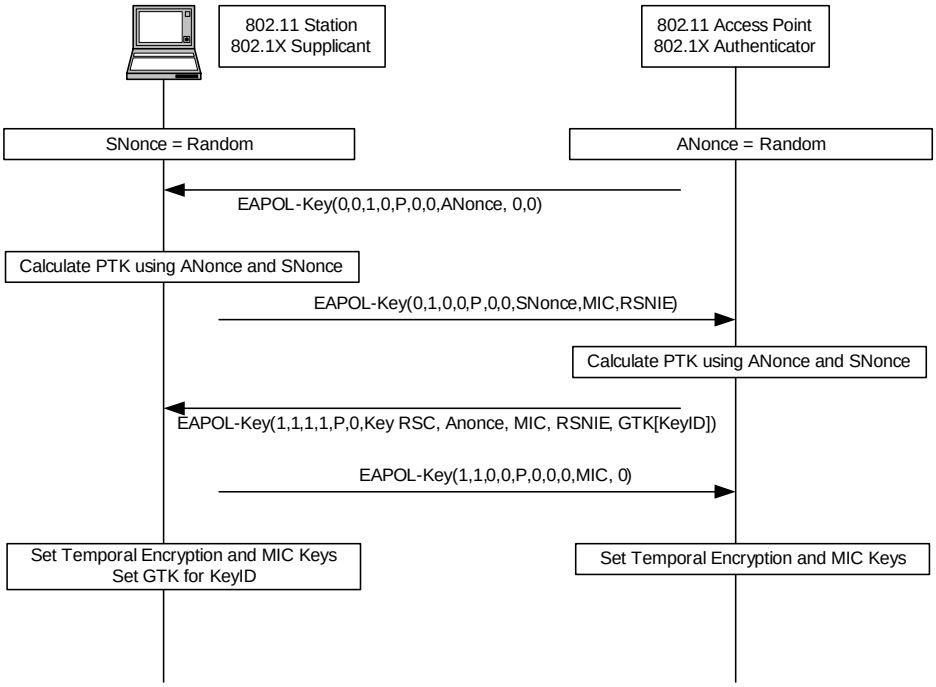



**Figure 8-33—Sample 4-Way Handshake**

### 8.5.3.7 4-Way Handshake analysis

The following is an informative analysis of the the 4-Way Handshake.

This subclause makes the trust assumptions used in this protocol explicit. The protocol assumes the following:

— The PMK is known only by the Supplicant's STA and the Authenticator's STA.

— The Supplicant's STA uses IEEE 802 address SPA.

— The Authenticator's STA uses IEEE 802 address AA.

In many instantiations the RSNA architecture immediately breaks the first assumption because the IEEE 802.1X AS also knows the PMK. Therefore, additional assumptions are required:

— The AS does not expose the PMK to other parties.

— The AS does not masquerade as the Supplicant to the Authenticator.

— The AS does not masquerade as the Authenticator to the Supplicant.

— The AS does not masquerade as the Supplicant's STA.

— The AS does not masquerade as the Authenticator's STA.

The protocol also assumes this particular Supplicant-Authenticator pair is authorized to know this PMK and to use it in the 4-Way Handshake. If any of these assumptions are broken, then the protocol fails to provide any security guarantees.

The protocol also assumes that the AS delivers the correct PMK to the AP with IEEE 802 address AA and that the non-AP STA with IEEE 802 address SPA hosts the Supplicant that negotiated the PMK with the AS. None of the protocols defined by this standard and IEEE Std 802.1X-2004 permit the AS, the Authenticator, the Supplicant, or either STA to verify these assumptions.


Copyright © 2007 IEEE. All rights reserved.

The PTK derivation step

PTK ← PRF-X(PMK, "Pairwise key expansion" || Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))

performs a number of functions:

- Including the AA and SPA in the computation
  - Binds the PTK to the communicating STAs and
  - Prevents undetected man-in-the-middle attacks against 4-Way Handshake messages between the STAs with these two IEEE 802 addresses.
- If ANonce is randomly selected, including ANonce
  - Guarantees the STA at IEEE 802 address AA that PTK is fresh,
  - Guarantees that Message 2 and Message 4 are live, and
  - Uniquely identifies PTK as <AA, ANonce>.
- If SNonce is randomly selected, including SNonce
  - Guarantees the STA at IEEE 802 address SPA that PTK is fresh,
  - Guarantees that Message 3 is live, and
  - Uniquely identifies PTK as <SPA, SNonce>.

Choosing the nonces randomly helps prevent precomputation attacks. With unpredictable nonces, a man-in-the-middle attack that uses the Supplicant to precompute messages to attack the Authenticator cannot progress beyond Message 2, and a similar attack against the Supplicant cannot progress beyond Message 3. The protocol can be executed further before an error if predictable nonces are used.

Message 1 delivers ANonce to the Supplicant and initiates negotiation for a new PTK. It identifies AA as the peer STA to the Supplicant's STA. If an adversary modifies either of the addresses or ANonce, the Authenticator will detect the result when validating the MIC in Message 2. Message 1 does not carry a MIC, as it is impossible for the Supplicant to distinguish this message from a replay without maintaining state of all security associations through all time (PMK might be a static key).

Message 2 delivers SNonce to the Authenticator so it can derive the PTK. If the Authenticator selected ANonce randomly, Message 2 also demonstrates to the Authenticator that the Supplicant is live, that the PTK is fresh, and that there is no man-in-the-middle attack, as the PTK includes the IEEE 802 MAC addresses of both. Inclusion of ANonce in the PTK derivation also protects against replay. The MIC prevents undetected modification of Message 2 contents.

Message 3 confirms to the Supplicant that there is no man-in-the-middle attack. If the Supplicant selected SNonce randomly, it also demonstrates that the PTK is fresh and that the Authenticator is live. The MIC again prevents undetected modification of Message 3.

While Message 4 serves no cryptographic purpose, it serves as an acknowledgment to Message 3. It is required to ensure reliability and to inform the Authenticator that the Supplicant has installed the PTK and GTK and hence can receive encrypted frames.

The PTK and GTK are installed by using MLME-SETKEYS.request primitive after Message 4 is sent. The PTK is installed before the GTK.

Then the 4-Way Handshake uses a correct, but unusual, mechanism to guard against replay. As noted earlier in this subclause, ANonce provides replay protection to the Authenticator, and SNonce to the Supplicant. In most session initiation protocols, replay protection is accomplished explicitly by selecting a nonce randomly and requiring the peer to reflect the received nonce in a response message. The 4-Way Handshake instead mixes ANonce and SNonce into the PTK, and replays are detected implicitly by MIC failures. In particular,

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 267 of 1232

the Key Replay Counter field serves no cryptographic purpose in the 4-Way Handshake. Its presence is not detrimental, however, and it plays a useful role as a minor performance optimization for processing stale instances of Message 2. This replay mechanism is correct, but its implicit nature makes the protocol harder to understand than an explicit approach.

It is critical to the correctness of the 4-Way Handshake that at least one bit differs in each message. Within the 4-Way Handshake, Message 1 can be recognized as the only one with the Key MIC bit clear, meaning Message 1 does not include the MIC, while Message 2 through Message 4 do. Message 3 differs from Message 2 by not asserting the Ack bit and from Message 4 by asserting the Ack Bit. Message 2 differs from Message 4 by including the RSN information element.

Request messages cannot be confused with 4-Way Handshake messages because the former asserts the Request bit and 4-Way Handshake messages do not. Group Key Handshake messages cannot be mistaken for 4-Way Handshake messages because they assert a different key type.

### 8.5.4 Group Key Handshake

The Authenticator uses the Group Key Handshake to send a new GTK to the Supplicant.

The Authenticator may initiate the exchange when a Supplicant is disassociated or deauthenticated.

Message 1: Authenticator → Supplicant: EAPOL-Key(1,1,1,0,G,0,Key RSC,0, MIC,GTK[N])

Message 2: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,G,0,0,0,MIC,0)

Here, the following assumptions apply:

— Key RSC denotes the last frame sequence number sent using the GTK.

— GTK[N] denotes the GTK encapsulated with its key identifier as defined in 8.5.2 using the KEK defined in 8.5.1.2 and associated IV.

— The MIC is computed over the body of the EAPOL-Key frame (with the MIC field zeroed for the computation) using the KCK defined in 8.5.1.2.

The Supplicant may trigger a Group Key Handshake by sending an EAPOL-Key frame with the Request bit set to 1 and the type of the Group Key bit.

An Authenticator shall do a 4-Way Handshake before a Group Key Handshake if both are required to be done.

NOTE—The Authenticator cannot initiate the Group Key Handshake until the 4-Way Handshake completes successfully.

#### 8.5.4.1 Group Key Handshake Message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 0

Install = 0

Key Ack = 1

Key MIC = 1


Copyright © 2007 IEEE. All rights reserved.

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = 0

Key Replay Counter = *n*+2

Key Nonce = 0

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

Key RSC = last transmit sequence number for the GTK

Key MIC = MIC(KCK, EAPOL)

Key Data Length = Cipher-suite-specific; see Table 8-2

Key Data = encrypted, encapsulated GTK and the GTK's key identifier (see 8.5.2)

The Authenticator sends Message 1 to the Supplicant.

On reception of Message 1, the Supplicant

a) Verifies that the Key Replay Counter field value has not yet been seen before, i.e., its value is strictly larger than that in any other EAPOL-Key frame received thus far during this session.

b) Verifies that the MIC is valid, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error.

c) Uses the MLME-SETKEYS.request primitive to configure the temporal GTK into its IEEE 802.11 MAC.

d) Responds by creating and sending Message 2 of the Group Key Handshake to the Authenticator and incrementing the replay counter.

NOTE—The Authenticator must increment and use a new Key Replay Counter field value on every Message 1 instance, even retries, because the Message 2 responding to an earlier Message 1 may have been lost. If the Authenticator did not increment the replay counter, the Supplicant will discard the retry, and no responding Message 2 will ever arrive.

#### 8.5.4.2 Group Key Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (ARC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 0 (Group/SMK) – same as Message 1

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 269 of 1232

Key Replay Counter = *n*+2 – same as Message 1

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL)

Key Data Length = 0

Key Data = none required

On reception of Message 2, the Authenticator

a) Verifies that the Key Replay Counter field value matches one it has used in the Group Key Handshake.

b) Verifies that the MIC is valid, i.e., it uses the KCK that is part of the PTK to verify that there is no data integrity error.

#### 8.5.4.3 Group Key Handshake implementation considerations

If the Authenticator does not receive a reply to its messages, its shall attempt dot11RSNAConfigGroupUpdateCount transmits of the message, plus a final timeout. The retransmit timeout value shall be 100 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval, then 100 ms shall be used for all timeout values. If it still has not received a response after this, then the Authenticator's STA should use the MLME-DEAUTHENTICATE.request primitive to deauthenticate the STA.

#### 8.5.4.4 Sample Group Key Handshake

The following is an informative sample of a Group Key Handshake for illustration.

The state machines in 8.5.5 and 8.5.6 change the GTK in use by the network. See Figure 8-34.

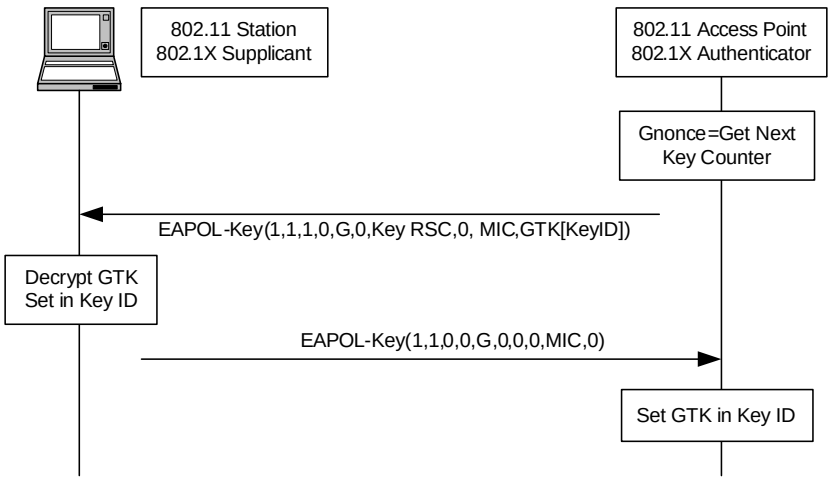



**Figure 8-34—Sample Group Key Handshake**

The following steps occur:

a) The Authenticator generates a new GTK. It encapsulates the GTK and sends an EAPOL-Key frame containing the GTK (Message 1), along with the last sequence number used with the GTK (RSC).


Copyright © 2007 IEEE. All rights reserved.

b) On receiving the EAPOL-Key frame, the Supplicant validates the MIC, decapsulates the GTK, and uses the MLME-SETKEYS.request primitive to configure the GTK and the RSC in its STA.

c) The Supplicant then constructs and sends an EAPOL-Key frame in acknowledgment to the Authenticator.

d) On receiving the EAPOL-Key frame, the Authenticator validates the MIC. If the GTK is not already configured into IEEE 802.11 MAC, after the Authenticator has delivered the GTK to all associated STAs, it uses the MLME-SETKEYS.request primitive to configure the GTK into the IEEE 802.11 STA.

### 8.5.5 RSNA Supplicant key management state machine

The Supplicant shall reinitialize the Supplicant state machine whenever its system initializes. A Supplicant enters the AUTHENTICATION state on an event from the MAC that requests another STA to be authenticated. A Supplicant enters the STAKEYSTART state on receiving an EAPOL-Key frame from the Authenticator. If the MIC or any of the EAPOL-Key frames fails, the Supplicant silently discards the frame. Figure 8-35 depicts the RSNA Supplicant state machine.

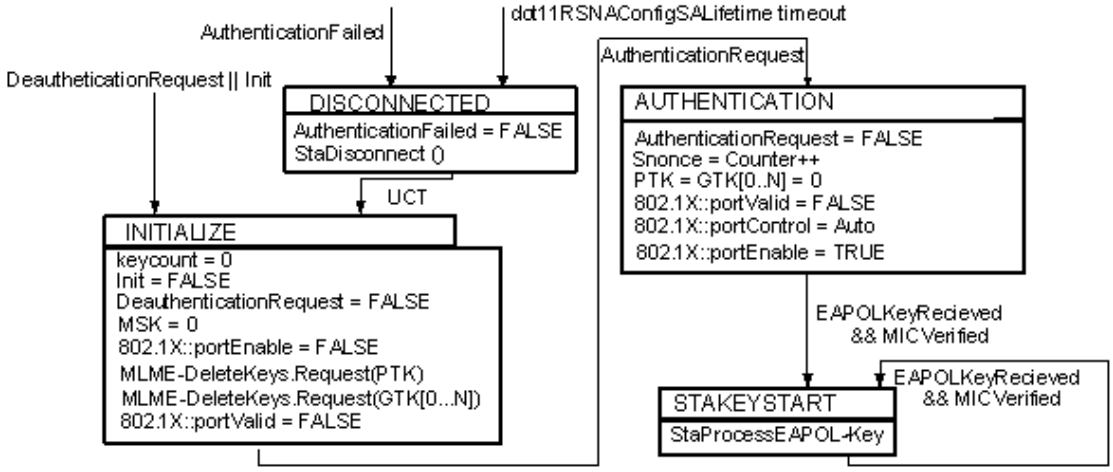



**Figure 8-35—RSNA Supplicant key management state machine**

Unconditional transfer (UCT) means the event triggers an immediate transition.

This state machine does not use timeouts or retries. The IEEE 802.1X state machine has timeouts that recover from authentication failures, etc.

The management entity will send an authentication request event when it wants an Authenticator authenticated. This can be before or after the STA associates to the AP. In an IBSS environment, the event will be generated when a Probe Response frame is received.

#### 8.5.5.1 Supplicant state machine states

The following list summarizes the states of the Supplicant state machine:

— **AUTHENTICATION**: A STA's Supplicant enters this state when it sends an IEEE 802.1X AuthenticationRequest to authenticate to an SSID.

— **DISCONNECTED**: A STA's Supplicant enters this state when IEEE 802.1X authentication fails. The Supplicant executes StaDisconnect and enters the INITIALIZE state.

Copyright © 2007 IEEE. All rights reserved.

— **INITIALIZE**: A STA's Supplicant enters this state from the DISCONNECTED state, when it receives Disassociation or Deauthentication messages or when the STA initializes, causing the STA's Supplicant to initialize the key state variables.

— **STAKEYSTART:** A STA's Supplicant enters this state when it receives an EAPOL-Key frame. All the information to process the EAPOL-Key frame is in the message and is described in the StaProcessEAPOL-Key procedure.

#### 8.5.5.2 Supplicant state machine variables

The following list summarizes the variables used by the Supplicant state machine:

— *DeauthenticationRequest* – The Supplicant sets this variable to TRUE if the Supplicant's STA reports it has received Disassociation or Deauthentication messages.

— *AuthenticationRequest* – The Supplicant sets this variable to TRUE if its STA's IEEE 802.11 management entity reports it wants an SSID authenticated. This can be on association or at other times.

— *AuthenticationFailed* – The Supplicant sets this variable to TRUE if the IEEE 802.1X authentication failed. The Supplicant uses the MLME-DISASSOCIATE.request primitive to cause its STA to disassociate from the Authenticator's STA.

— *EAPOLKeyReceived* – The Supplicant sets this variable to TRUE when it receives an EAPOL-Key frame.

— *IntegrityFailed* **–** The Supplicant sets this variable to TRUE when its STA reports that a fatal data integrity error (e.g., Michael failure) has occurred.

NOTE—A Michael failure is not the same as MICVerified because IntegrityFailed is generated if the Michael integrity check fails; MICVerified is generated from validating the EAPOL-Key integrity check. Note also the STA does not generate this event for CCMP because countermeasures are not required.

— *MICVerified* **–** The Supplicant sets this variable to TRUE if the MIC on the received EAPOL-Key frame verifies as correct. The Supplicant silently discards any EAPOL-Key frame received with an invalid MIC.

— *Counter* **–** The Supplicant uses this variable as a global counter used for generating nonces.

— *SNonce* **–** This variable represents the Supplicant's nonce.

— *PTK* **–** This variable represents the current PTK.

— *TPTK* **–** This variable represents the current PTK until Message 3 of the 4-Way Handshake arrives and is verified.

— *GTK[]* **–** This variable represents the current GTKs for each group key index.

— *PMK* **–** This variable represents the current PMK.

— *keycount* – This variable is used in IBSS mode to decide when all the keys have been delivered and an IBSS link is secure.

— *802.1X::XXX* **–** This variable denotes another IEEE 802.1X state variable *XXX* not specified in this standard.

#### 8.5.5.3 Supplicant state machine procedures

The following list summarizes the procedures used by the Supplicant state machine:

— **STADisconnect** – The Supplicant invokes this procedure to disassociate and deauthenticate its STA from the AP.

— **MIC**(*x*) – The Supplicant invokes this procedure to compute a MIC of the data *x*.

— **CheckMIC**() – The Supplicant invokes this procedure to verify a MIC computed by the MIC() function.


Copyright © 2007 IEEE. All rights reserved.

— **StaProcessEAPOL-Key –** The Supplicant invokes this procedure to process a received EAPOL-Key frame. The pseudo-code for this procedure is as follows:

**StaProcessEAPOL-Key** (*S*, *M*, *A*, *I*, *K*, *RSC*, *ANonce*, RSC, *MIC*, RSNIE, *GTK[N]*)

```
TPTK ← PTK
TSNonce ← 0
PRSC ← 0
UpdatePTK ← 0
State ← UNKNOWN
if M = 1 then
    if Check MIC(PTK, EAPOL-Key frame) fails then
        State ← FAILED
    else
        State ← MICOK
    endif
endif
if K = P then
    if State ≠ FAILED then
        if PSK exists then – PSK is a preshared key
            PMK ← PSK
        else
            PMK ← L(MSK, 0, 256)
        endif
        TSNonce ← SNonce
        if ANonce ≠ PreANonce then
            TPTK ← Calc PTK(PMK, ANonce, TSNonce)
            PreANonce ← ANonce
        endif
        if State = MICOK then
            PTK ← TPTK
            UpdatePTK ← I
            if UpdatePTK = 1 then
                if no GTK then
                    PRSC ← RSC
                endif
                if MLME-SETKEYS.request(0, TRUE, PRSC, PTK) fails then
                    invoke MLME-
                    DEAUTHENTICATE.request
                endif
                MLME-SETPROTECTION.request(TA, Rx)
            endif
            if GTK then
                if (GTK[N] ← Decrypt GTK) succeeds then
                    if MLME-SETKEYS.request(N, 0, RSC, GTK[N]) fails then
                        invoke MLME-DEAUTHENTICATE.request
                    endif
                else
                    State ← FAILED
                endif
            endif
        endif
    endif
else if KeyData = GTK then
    if State = MICOK then
```

Copyright © 2007 IEEE. All rights reserved.

```
                    if (GTK[N] ← Decrypt GTK) succeeds then
                        if MLME-SETKEYS.request(N, T, RSC, GTK[N]) fails then
                            invoke MLME-
                            DEAUTHENTICATE.request
                        endif
                    else
                        State ← FAILED
                    endif
                else
                    State ← FAILED
                endif
            endif
            if A = 1 && State ≠ Failed then
                Send EAPOL-Key(0,1,0,0,K,0,0,TSNonce,MIC(TPTK),RSNIE)
            endif
            if UpdatePTK = 1 then
                MLME-SETPROTECTION.request(TA, Tx_Rx)
            endif
            if State = MICOK && S = 1 then
                MLME-SETPROTECTION.request(TA, Tx_Rx)
                if IBSS then
                    keycount++
                    if keycount = 2 then
                        802.1X::portValid← TRUE
                    endif
                else
                    802.1X::portValid← TRUE
                endif
            endif
```

Here UNKNOWN, MICOK, and FAILED are values of the variable State used in the Supplicant pseudo-code. State is used to decide how to do the key processing. MICOK is set when the MIC of the EAPOL-Key has been checked and is valid. FAILED is used when a failure has occurred in processing the EAPOL-Key frame. UNKNOWN is the initial value of the variable State.

When processing 4-Way Handshake Message 3, the GTK is decrypted from the EAPOL-Key frame and installed. The PTK shall be installed before the GTK.

The Key Replay Counter field used by the Supplicant for EAPOL-Key frames that are sent in response to a received EAPOL-Key frame shall be the received Key Replay Counter field. Invalid EAPOL-Key frames such as invalid MIC, GTK without a MIC, etc., shall be ignored.

NOTE 1—TPTK is used to stop attackers changing the PTK on the Supplicant by sending the first message of the 4-Way Handshake. An attacker can still affect the 4-Way Handshake while the 4-Way Handshake is being carried out.

NOTE 2—The PMK will be supplied by the authentication method used with IEEE Std 802.1X-2004 if preshared mode is not used.

NOTE 3—A PTK is configured into the encryption/integrity engine depending on the Tx/Rx bit, but if configured, is always a transmit key. A GTK is configured into the encryption/integrity engine independent of the state of the Tx/Rx bit, but whether the GTK is used as a transmit key is dependent on the state of the Tx/Rx bit.

— **CalcGTK**(x) – Generates the GTK.

— **DecryptGTK(x)** – Decrypt the GTK from the EAPOL-Key frame.


Copyright © 2007 IEEE. All rights reserved.

#### 8.5.5.4 Supplicant PeerKey state machine states

Figure 8-36 depicts the PeerKey Handshake Supplicant key management state machine. The following list summarizes the states the Supplicant state machine uses to support the PeerKey Handshake:

— **STKINIT:** This state is the idle state and is entered when the IEEE 802.1X Supplicant completes successful Authentication.

— **SMKNEGOTIATING1:** This state is entered when the MLME-STKSTART.Request message is received for the SMK Handshake by the initiator STA.

— **SMKNEGOTIATING2:** This state is entered when the first EAPOL-Key frame of the SMK Handshake is received by the peer STA.

— **SMKNEGOTIATING3:** This state is entered when the fifth EAPOL-Key frame of the SMK Handshake is received by the initiator STA.

— **SMKNEGOTIATING4:** This state is entered when the fourth EAPOL-Key frame of the SMK Handshake is received by the peer STA.

— **STKSTART:** Once the SMKSA is created, the initiator STA enters this state. This is the start of 4-Way STK Handshake.

— **STKCALCNEGOTIATING:** This state is entered when the second EAPOL-Key frame of the 4-Way STK Handshake is received by the initiator STA and the MIC is verified.

— **STKCALCNEGOTIATING1:** This state is entered when the first EAPOL-Key frame of the 4-Way STK Handshake is received by the peer STA and the MIC is verified.

— **STKCALCNEGOTIATING2:** This state is entered unconditionally by the initiator STA.

— **STKCALCNEGOTIATING3:** This state is entered unconditionally by the peer STA.

— **STKCALCNEGOTIATING4:** This state is entered when the third EAPOL-Key frame of the 4-Way STK Handshake is received by the peer STA and the MIC is verified.

— **STKINITDONE:** This state is entered by the initiator STA when the fourth EAPOL-Key frame of the 4-Way STK Handshake is received. This state is entered by the peer STA when the fourth EAPOL-Key frame of the 4-Way STK Handshake is sent.

#### 8.5.5.5 Supplicant PeerKey state machine variables

The following list summarizes the variables used by the Supplicant state machine:

— *PeerKeyInit* – This variable is used to initialize the PeerKey state machine.

— *TimeoutEvt* – This variable is set to TRUE if the EAPOL-Key frame sent fails to obtain a response from the Supplicant. The variable may be set by management action or set by the operation of a timeout while in the different states.

— *TimeoutCtr* – This variable maintains the count of EAPOL-Key receive timeouts. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. Annex D contains details of the timeout values. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.

— *MICVerified* – This variable is set to TRUE if the MIC on the received EAPOL-Key frame is verified and is correct. Any EAPOL-Key frames with an invalid MIC are dropped and ignored.

— *SMKMesgNo* – This variable indicates SMK Handshake EAPOL-Key frame types. Details for each message type (1-5) are provided in 8.5.8.

— *STKMesgNo* – This variable indicates 4-Way STK Handshake EAPOL-Key frame types. Details for each message type (1-4) are provided in 8.5.3.

— *STA_P* – This variable indicates the MAC address of the peer STA participating in the PeerKey Handshake.

— *STA_I* – This variable indicates the MAC address of the initiator STA participating in the PeerKey Handshake.

Copyright © 2007 IEEE. All rights reserved.

— *STKKey* – The STK Key generated as a result of the 4-Way STK Handshake.

— *EAPOLKeyReceived* – The Supplicant sets this variable to TRUE when it receives an EAPOL-Key frame.

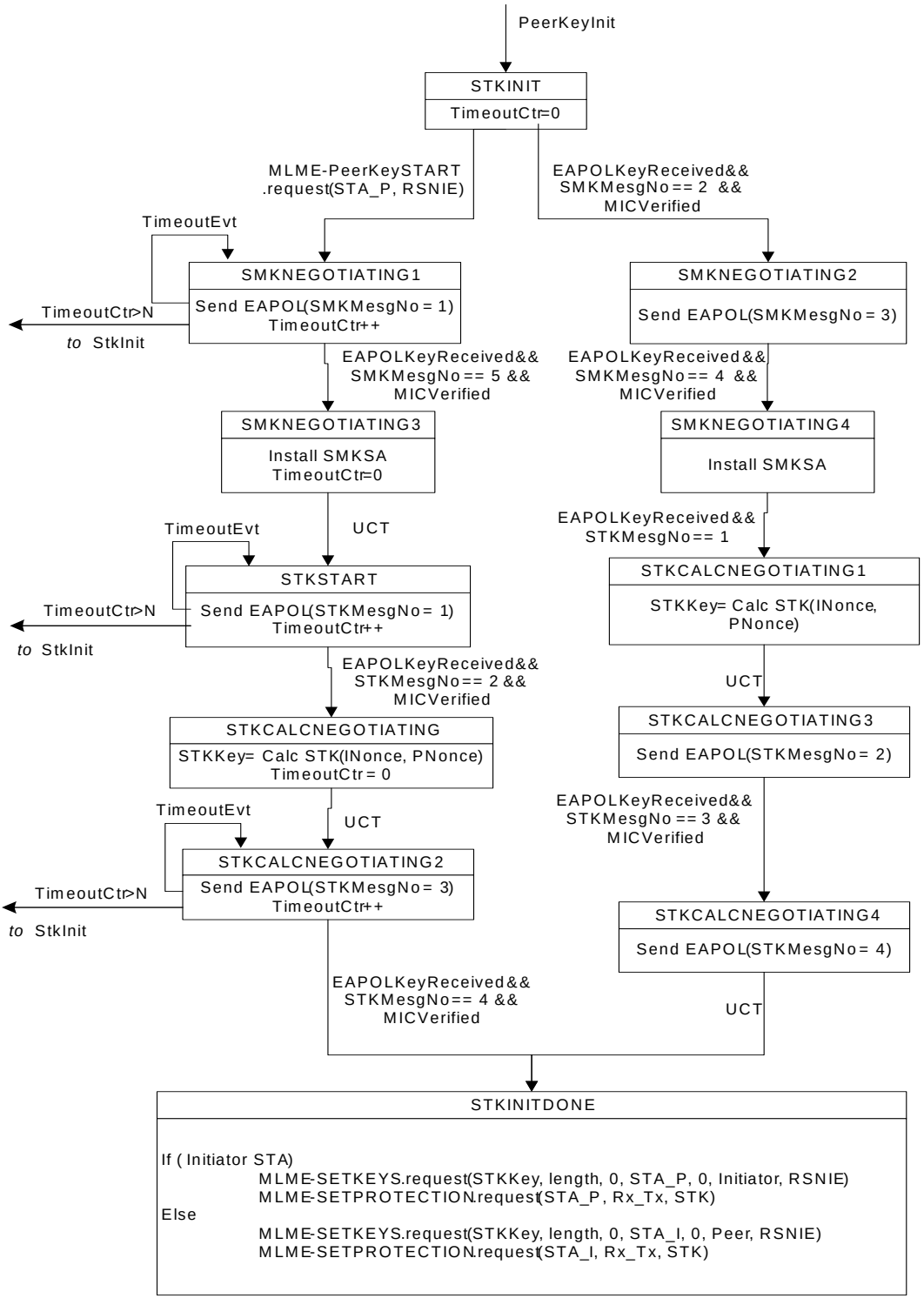



**Figure 8-36—PeerKey Handshake Supplicant key management state machine**


Copyright © 2007 IEEE. All rights reserved.

### 8.5.6 RSNA Authenticator key management state machine

There is one state diagram for the Authenticator. In an ESS, the Authenticator will always be on the AP; and in an IBSS environment, the Authenticator will be on every STA.

The state diagram shown in parts in Figure 8-37 through Figure 8-40 consists of the following states:

a) The AUTHENTICATION, AUTHENTICATION2, INITPMK, INITPSK, PTKSTART, PTKCALCNEGOTIATING, PTKCALCNEGOTIATING2, PTKINITNEGOTIATING, PTKINIT-DONE, DISCONNECT, DISCONNECTED, and INITIALIZE states. These states handle the initialization, 4-Way Handshake, tear-down, and general clean-up. These states are per associated STA.

b) The IDLE, REKEYNEGOTIATING, KEYERROR, and REKEYESTABLISHED states. These states handle the transfer of the GTK to the associated client. These states are per associated STA.

c) The GTK_INIT, SETKEYS, and SETKEYSDONE states. These states change the GTK when required, trigger all the PTK group key state machines, and update the IEEE 802.11 MAC in the Authenticator's AP when all STAs have the updated GTK. These states are global to the Authenticator.

Because there are two GTKs, responsibility for updating these keys is given to the group key state machine (see Figure 8-39). In other words, this state machine determines which GTK is in use at any time.

When a second STA associates, the group key state machine is already initialized, and a GTK is already available and in use.

When the GTK is to be updated the variable GTKReKey is set. The SETKEYS state updates the GTK and triggers all the PTK group key state machines that currently exist—one per associated STA. Each PTK group key state machine sends the GTK to its STA. When all the STAs have received the GTK (or failed to receive the key), the SETKEYSDONE state is executed which updates the APs encryption/integrity engine with the new key.

Both the PTK state machine and the PTK group key state machine use received EAPOL-Key frames as an event to change states. The PTK state machine only uses EAPOL-Key frames with the Key Type field set to

Pairwise, and the PTK group key state machine only uses EAPOL-Key frames with the Key Type field set to Group.

#### 8.5.6.1 Authenticator state machine states

##### 8.5.6.1.1 Authenticator state machine: 4-Way Handshake (per STA)

The following list summarizes the states the Authenticator state machine uses to support the 4-Way Handshake:

— **AUTHENTICATION**: This state is entered when an AuthenticationRequest is sent from the management entity to authenticate a BSSID.

— **AUTHENTICATION2**: This state is entered from the AUTHENTICATION state or from the PTKINITDONE state.

— **DISCONNECT**: This state is entered if an EAPOL-Key frame is received and fails its MIC check. It sends a Deauthentication message to the STA and enters the INITIALIZE state.

— **DISCONNECTED**: This state is entered when Disassociation or Deauthentication messages are received.

— **INITIALIZE**: This state is entered from the DISCONNECTED state, when a deauthentication request event occurs, or when the STA initializes. The state initializes the key state variables.

Copyright © 2007 IEEE. All rights reserved.

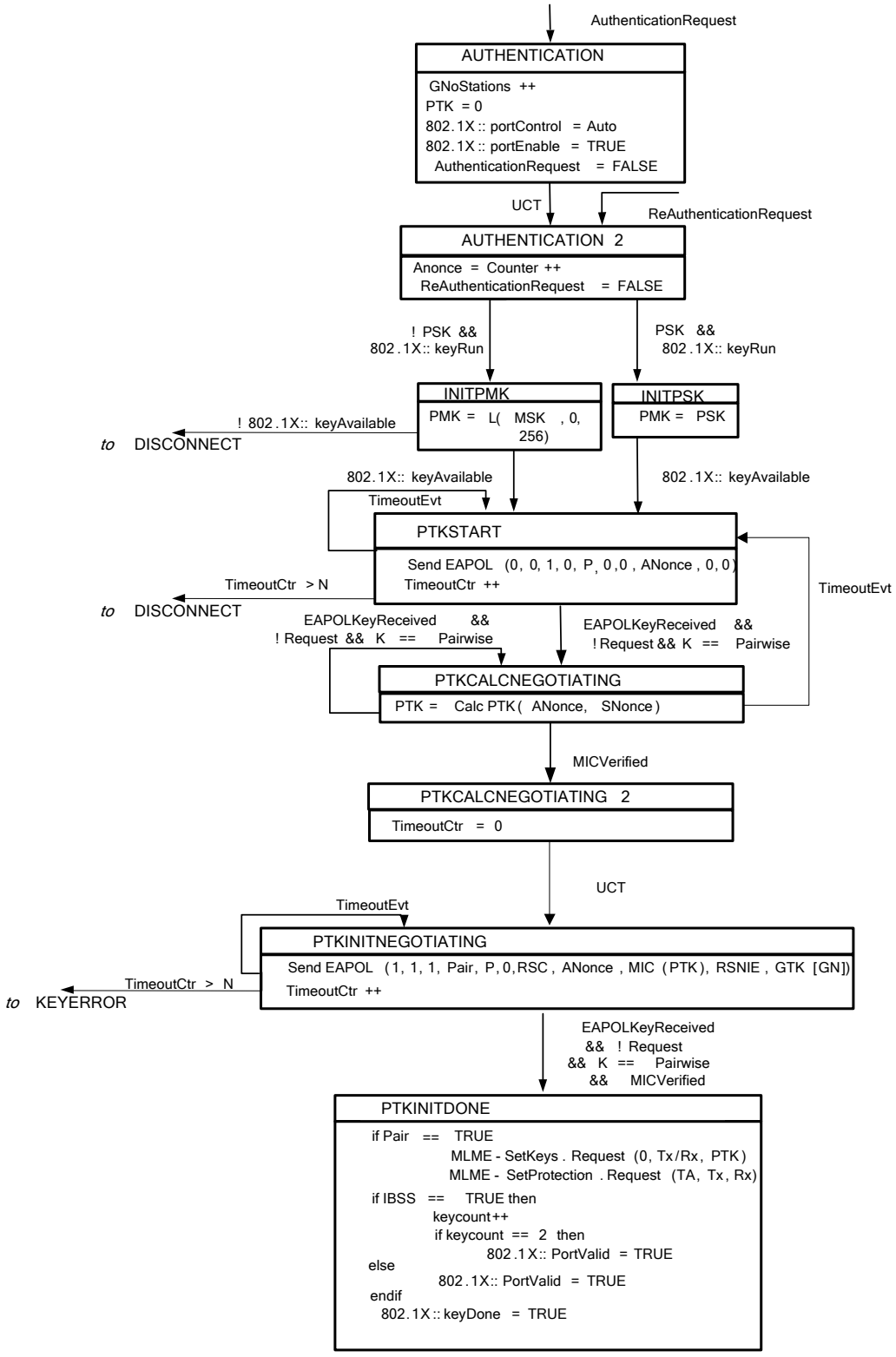



**Figure 8-37—Authenticator state machines, part 1**


Copyright © 2007 IEEE. All rights reserved.

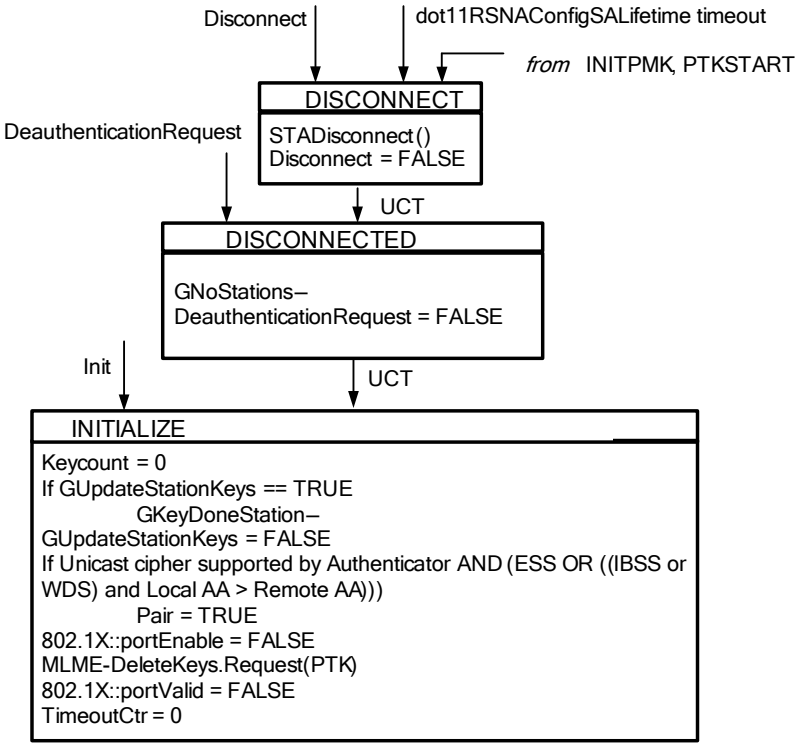



**Figure 8-38—Authenticator state machines, part 2**

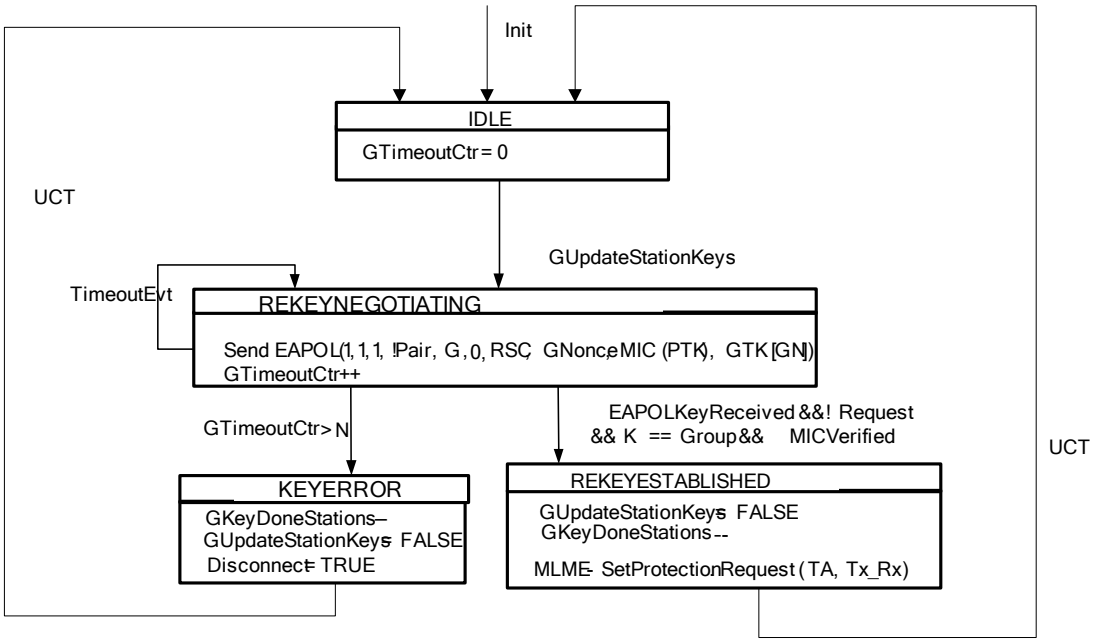



**Figure 8-39—Authenticator state machines, part 3**

Copyright © 2007 IEEE. All rights reserved.

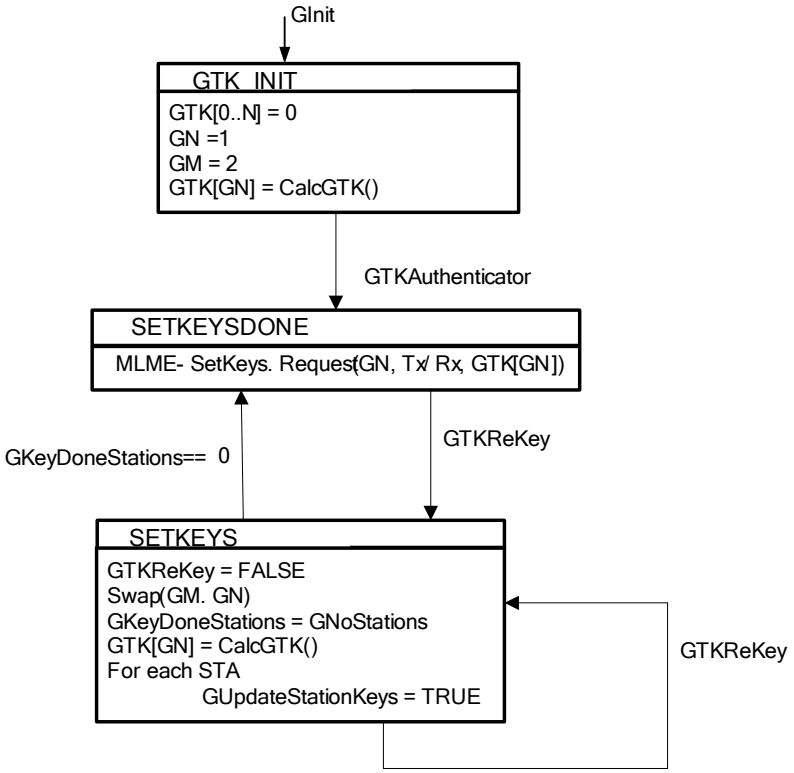



**Figure 8-40—Authenticator state machines, part 4**

— **INITPMK**: This state is entered when the IEEE 802.1X backend AS completes successfully. If a PMK is supplied, it goes to the PTKSTART state; otherwise, it goes to the DISCONNECTED state.

— **INITPSK:** This state is entered when a PSK is configured.

— **PTKCALCNEGOTIATING:** This state is entered when the second EAPOL-Key frame for the 4-Way Handshake is received with the key type of Pairwise.

— **PTKCALNEGOTIATING2:** This state is entered when the MIC for the second EAPOL-Key frame of the 4-Way Handshake is verified.

— **PTKINITNEGOTIATING:** This state is entered when the MIC for the second EAPOL-Key frame for the 4-Way Handshake is verified. When Message 3 of the 4-Way Handshake is sent in state PTKINITNEGOTIATING, the encrypted GTK shall be sent at the end of the data field, and the GTK length is put in the GTK Length field.

— **PTKINITDONE:** This state is entered when the last EAPOL-Key frame for the 4-Way Handshake is received with the key type of Pairwise. This state may call SetPTK; if this call fails, the AP should detect and recover from the situation, for example, by doing a disconnect event for this association.

— **PTKSTART:** This state is entered from INITPMK or INITPSK to start the 4-Way Handshake or if no response to the 4-Way Handshake occurs.


Copyright © 2007 IEEE. All rights reserved.

#### 8.5.6.1.2 Authenticator state machine: Group Key Handshake (per STA)

The following list summarizes the states the Authenticator state machine uses to support the Group Key Handshake:

— **IDLE:** This state is entered when no Group Key Handshake is occurring.

— **KEYERROR:** This state is entered if the EAPOL-Key acknowledgment for the Group Key Handshake is not received.

— **REKEYESTABLISHED:** This state is entered when an EAPOL-Key frame is received from the Supplicant with the Key Type subfield set to Group.

— **REKEYNEGOTIATING:** This state is entered when the GTK is to be sent to the Supplicant.

NOTE—The TxRx flag for sending a GTK is always the opposite of whether the pairwise key is used for data encryption/integrity or not. If a pairwise key is used for encryption/integrity, then the STA never transmits with the GTK; otherwise, the STA uses the GTK for transmit.

#### 8.5.6.1.3 Authenticator state machine: Group Key Handshake (global)

The following list summarizes the states the Authenticator state machine uses to coordinate a group key update of all STAs:

— **GTK_INIT:** This state is entered on system initialization.

— **SETKEYS:** This state is entered if the GTK is to be updated on all Supplicants.

— **SETKEYSDONE:** This state is entered if the GTK has been updated on all Supplicants.

NOTE—SETKEYSDONE calls SetGTK to set the GTK for all associated STA. If this fails, all communication via this key will fail, and the AP needs to detect and recover from this situation.

### 8.5.6.2 Authenticator state machine variables

The following list summarizes the variables used by the Authenticator state machine:

— *AuthenticationRequest* – This variable is set to TRUE if the STA's IEEE 802.11 management entity wants an association to be authenticated. This can be set when the STA associates or at other times.

— *ReAuthenticationRequest* – This variable is set to TRUE if the IEEE 802.1X Authenticator received an eapStart or 802.1X::reAuthenticate is set.

— *DeauthenticationRequest* – This variable is set to TRUE if a Disassociation or Deauthentication message is received.

— *Disconnect* – This variable is set to TRUE when the STA should initiate a deauthentication.

— *EAPOLKeyReceived* – This variable is set to TRUE when an EAPOL-Key frame is received. EAPOL-Key frames that are received in response to an EAPOL-Key frame sent by the Authenticator must contain the same Key Replay Counter field value as the Key Replay Counter field in the transmitted message. EAPOL-Key frames that contain different Key Replay Counter field values should be discarded. An EAPOL-Key frame that is sent by the Supplicant in response to an EAPOL-Key frame from the Authenticator must not have the Ack bit set. EAPOL-Key frames sent by the Supplicant not in response to an EAPOL-Key frame from the Authenticator must have the Request bit set.

NOTE 1—EAPOL-Key frames with a key type of Pairwise and a nonzero key index should be ignored.

NOTE 2—EAPOL-Key frames with a key type of Group and an invalid key index should be ignored.

NOTE 3—When an EAPOL-Key frame with the Ack bit cleared is received, then it is expected as a reply to a message that the Authenticator sent, and the replay counter is checked against the replay counter used in the sent EAPOL-Key frame. When an EAPOL-Key frame with the Request bit set is received, then a replay counter for these messages is used that is a different replay counter than the replay counter used for sending messages to the Supplicant.

Copyright © 2007 IEEE. All rights reserved.

— *GTimeoutCtr* – This variable maintains the count of EAPOL-Key receive timeouts for the Group Key Handshake. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. Annex D details the timeout values. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.

— *GInit* – This variable is used to initialize the group key state machine. This is a group variable.

— *Init* – This variable is used to initialize per-STA state machine

— *TimeoutEvt* – This variable is set to TRUE if the EAPOL-Key frame sent out fails to obtain a response from the Supplicant. The variable may be set by management action or set by the operation of a timeout while in the PTKSTART and REKEYNEGOTIATING states.

— *TimeoutCtr* **–** This variable maintains the count of EAPOL-Key receive timeouts. It is incremented each time a timeout occurs on EAPOL-Key receive event and is initialized to 0. Annex D contains details of the timeout values. The Key Replay Counter field value for the EAPOL-Key frame shall be incremented on each transmission of the EAPOL-Key frame.

— *MICVerified* – This variable is set to TRUE if the MIC on the received EAPOL-Key frame is verified and is correct. Any EAPOL-Key frames with an invalid MIC are dropped and ignored.

— *GTKAuthenticator* – This variable is set to TRUE if the Authenticator is on an AP or it is the designated Authenticator for an IBSS.

— *GKeyDoneStations* – Count of number of STAs left to have their GTK updated. This is a global variable.

— *GTKRekey* – This variable is set to TRUE when a Group Key Handshake is required. This is a global variable.

— *GUpdateStationKeys* – This variable is set to TRUE when a new GTK is available to be sent to Supplicants.

— *GNoStations* – This variable counts the number of Authenticators so it is known how many Supplicants need to be sent the GTK. This is a global variable.

— *Counter* – This variable is the global STA key counter.

— *ANonce* – This variable holds the current nonce to be used if the STA is an Authenticator.

— *GN***,** *GM* – These are the current key indices for GTKs. Swap(GM, GN) means that the global key index in GN is swapped with the global key index in GM, so now GM and GN are reversed.

— *PTK* – This variable is the current PTK.

— *GTK[]* – This variable is the current GTKs for each GTK index.

— *PMK* – This variable is the buffer holding the current PMK.

— *802.1X::XXX* – This variable is the IEEE 802.1X state variable *XXX*.

— *keycount* – This variable is used in IBSS mode to decide when all the keys have been delivered and an IBSS link is secure.

#### 8.5.6.3 Authenticator state machine procedures

The following list summarizes the procedures used by the Authenticator state machine:

— **STADisconnect**() – Execution of this procedure deauthenticates the STA.

— **CalcGTK**(x) – Generates the GTK.

— **MIC**(x) – Computes a MIC over the plaintext data.

### 8.5.7 Nonce generation

The following is an informative description of Nonce generation.


Copyright © 2007 IEEE. All rights reserved.

All STAs contain a global key counter, which is 256 bits in size. It should be initialized at system boot-up time to a fresh cryptographic-quality random number. Refer to H.6 on random number generation. It is recommended that the counter value is initialized to the following:

PRF-256(Random number, "Init Counter", Local MAC Address || Time)

The local MAC address should be AA on the Authenticator and SPA on the Supplicant.

The random number is 256 bits in size. Time should be the current time [from Network Time Protocol (NTP) or another time in NTP format] whenever possible. This initialization is to ensure that different initial key counter values occur across system restarts regardless of whether a real-time clock is available. The key counter must be incremented (all 256 bits) each time a value is used as an IV. The key counter must not be allowed to wrap to the initialization value.

### 8.5.8 PeerKey Handshake

The PeerKey Handshake occurs after any other STSL setup procedures and is used to create an STKSA providing data confidentiality between the two STAs. The AP must establish an RSNA with each STA prior to PeerKey setup. The initiator STA starts the PeerKey Handshake, at the conclusion of which a key is established to secure the connection.

STSL security PeerKey Handshake is used to establish security for data frames passed directly between two STAs associated with the same AP. The AP must establish an RSNA with each STA prior to the PeerKey Handshake. After the STAs establish the STSL, the initiator STA starts the PeerKey Handshake, at the conclusion of which a key is established to secure the connection. The PeerKey Handshake is used to create an STKSA between the two STAs.

The PeerKey EAPOL-Key exchange provides a mechanism for obtaining the keys to be used for direct station-to-station communication. The initiator STA shall start a timer when it sends the first EAPOL-Key message, and the peer STA shall do the same on receipt of the first EAPOL-Key message. On expiration of this timer, the STA shall transition to the STKINIT state.

A STA should use the PeerKey Handshake prior to transferring any direct station-to-station data frames. The STKSA should be deleted when the station-to-station connection is terminated.

Here, the following assumptions apply:

— EAPOL-Key() denotes an EAPOL-Key frame conveying the specified argument list, using the notation introduced in 8.5.2.1.

— STA_I is the initiator STA.

— STA_P is the peer STA.

— AP is the access point with which both the STA_I and the STA_P are associated.

— MAC_I is the MAC address of the STA_I.

— MAC_P is the MAC address of the STA_I.

— INonce is the nonce generated by the STA_I.

— PNonce is the nonce generated by the STA_P.

The PeerKey Handshake has two components:

a) **SMK Handshake:** This handshake is initiated by the initiator STA, and as a result of this handshake, the SMKSA is installed in both the STAs. This message exchange goes through the AP and is protected using the PTK.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 283 of 1232

b) **4-Way STK Handshake:** Using the installed SMKSA, the initiator STA initiates the 4-Way Handshake (per 8.5.3.4), and as a result of this, the STKSA gets installed in both the STAs. The STKSA is used for securing data exchange between the initiator STA and the peer STA.

#### 8.5.8.1 SMK Handshake

The initiator STA initiates the SMK Handshake by sending the first message to the AP to establish an SMKSA between itself and another STA associated with the same AP. Unlike the 4-Way Handshake and the Group Key Handshake, the SMK Handshake is initiated by the initiator STA.

Message 1: Initiator STA → AP: EAPOL-Key(1,1,0,0,0,1,0, INonce, MIC, RSNIE_I, MAC_P KDE)

Message 2: AP → Peer STA: EAPOL-Key(1,1,1,0,0,1,0, INonce, MIC, RSNIE_I, MAC_I KDE)

Message 3: Peer STA → AP: EAPOL-Key(1,1,0,0,0,1,0, PNonce, MIC, RSNIE_P, MAC_I KDE, INonce KDE)

Message 4: AP → Peer STA: EAPOL-Key(1,1,0,1,0,1,0, PNonce, MIC, MAC_I KDE, INonce KDE, SMK KDE, Lifetime KDE)

Message 5: AP → Initiator STA: EAPOL-Key(1,1,0,0,0,1,0, INonce, MIC, RSNIE_P, MAC_P KDE, PNonce KDE, SMK KDE, Lifetime KDE)

##### 8.5.8.1.1 SMK Handshake Message 1

The initiator STA creates the RSN information element (see 7.3.2.25) by including the element ID, length, version, and pairwise cipher suite list fields. Since the group cipher suit field is before the pairwise cipher suite list field (so the STA needs to include it), the STA includes any value in this field, and the receiving STA ignores it. The initiator STA also generates a 256-bit random number that is sent in the Key Nonce field.

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

- Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)
- Key Type = 0 (Group/SMK)
- SMK Message = 1 (SMK)
- Install = 0
- Key Ack = 0
- Key MIC = 1
- Secure = 1
- Error = 0
- Request = 1
- Encrypted Key Data = 0
- Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of initiating STA

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (initiating STA's KCK, EAPOL)


Copyright © 2007 IEEE. All rights reserved.

Key Data Length = Length of Key Data field in octets

Key Data = Initiator RSNIE, peer MAC address KDE

The STA_I sends Message 1 to the AP.

On receipt of Message 1, the AP checks that the key replay counter corresponds to Message 1. If not, it silently discards the message. Otherwise,

a) The AP verifies the Message 1 MIC using the STA_I PTKSA. If the calculated MIC does not match the MIC that the STA_I included in the EAPOL-Key frame, the AP silently discards Message 1.

b) If the MIC is correct, the AP checks if the STA_P is reachable. If it is not reachable, the AP shall send an error EAPOL-Key message to STA_I per 8.5.8.4.1. After sending the message, AP silently discards Message 1.

c) The AP checks if the AP has a secure connection with STA_P. If not, the AP shall send an error EAPOL-Key message to STA_I per 8.5.8.4.2. After sending the message, AP silently discards Message 1.

d) If all checks succeed, the AP creates Message 2 using the STA_P PTKSA. The AP copies the contents of Message 1 to create Message 2.

#### 8.5.8.1.2 SMK Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 1

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of AP and STA_P

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_P, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Initiator RSNIE, initiator MAC address KDE

The AP sends Message 2 to the STA_P. On receipt of Message 2, the STA_P checks that the key replay counter corresponds to Message 2. If not, it silently discards the message. Otherwise,

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 285 of 1232

a) The STA_P verifies the Message 2 MIC using the STA_P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_P silently discards Message 2.

b) If the MIC is correct, the STA_P checks if it supports at least one cipher suites proposed by the STA_I. If it does not, the STA_P shall send an error EAPOL-Key message to STA_I through the AP per 8.5.8.4.3. After sending the errror message, the STA_P silently discards Message 2.

c) STA_O checks if it has already created an STSL with STA_I. If it has not, STA_P shall send an error EAPOL-Key message to STA_I through the AP per 8.5.8.4.4. After sending the error message, the STA_P silently discards Message 2.

d) If all checks succeed, the STA_P creates the state of PeerKey Handshake and stores the initiator Nonce and the RSNIE received in Message 2.

e) STA_P selects a support cipher suite from the cipher suite list proposed by the STA_I and creates the peer RSNIE, which is sent with Message 3.

f) STA_P generates a 256-bit random number that is sent as the peer Nonce KDE with Message 3.

g) Using all the information, STA_P creates Message 3.

#### 8.5.8.1.3 SMK Handshake Message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter of peer STA

Key Nonce = PNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Peer RSNIE, initiator MAC address KDE, initiator Nonce KDE

The STA_P sends Message 3 to the AP. On receipt of Message 3, the AP checks that the key replay counter corresponds to Message 3. If not, it silently discards the message. Otherwise,

a) The AP verifies the Message 1 MIC using the STA_I PTKSA. If the calculated MIC does not match the MIC that the STA_P included in the EAPOL-Key frame, the AP silently discards Message 1.


Copyright © 2007 IEEE. All rights reserved.

b) If MIC is correct, the AP checks if the STA_I is reachable. If it is not reachable, the AP shall send an error EAPOL-Key message to the STA_P per 8.5.8.4.1. After sending the message, the AP silently discards Message 3.

c) The AP checks if the AP has secure connection with STA_I. If it does not, the AP shall send an error EAPOL-Key message to STA_P per 8.5.8.4.2. After sending the message, the AP silently discards Message 3.

d) If all checks succeed, the AP generates a 256-bit random number that is used as the SMK, which is sent with Message 4 and Message 5 as SMK KDEs.

e) Depending on the strength of random number generator, the AP sets the lifetime of the SMK, which is sent with Message 4 and Message 5 as Lifetime KDEs.

f) Using all the information, the AP creates Message 4 and Message 5.

#### 8.5.8.1.4 SMK Handshake Message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 1

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = Cipher-suite-specific; see Table 8-2

Key Replay Counter = request EAPOL replay counter of AP

Key Nonce = PNonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Encrypted initiator MAC address KDE, INonce KDE, SMK KDE (contains SMK and PNonce), Lifetime KDE

The AP sends Message 4 to the STA_P. On receipt of Message 4, the STA_P checks that the key replay counter corresponds to Message 4. If it does not, STA_P silently discards the message. Otherwise,

a) The STA_P verifies the Message 4 MIC using STA_P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_P silently discards Message 4.

b) If the MIC is correct, STA_P identifies the PeerKey session using peer Nonce sent as part of the Key Nonce field of Message 4. If STA_P has an existing PeerKey state for this session, i.e., STA_P has

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 287 of 1232

received Message 2 and this message is a follow-up to that. If STA_P has an existing PeerKey state for this session, STA_P silently discards Message 4.

c) If all checks succeed, STA_P decrypts the Key Data field of Message 4 and extracts the MAC_I, the INonce, the PNonce, the SMK, and the lifetime from Message 4. The STA_P verifies the extracted INonce against the INonce originally received as part of Message 2.

d) The STA_P calculates the SMKID per 8.5.1.4.

e) The STA_P checks the value of the lifetime with the maximum value it can support. If the lifetime suggested by the AP is too long, the STA_P selects a lower value that it can support.

f) Using all the information, the STA_P creates the SMKSA for this PeerKey session.

#### 8.5.8.1.5 SMK Handshake Message 5

Message 5 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 0

Encrypted Key Data = 1

Reserved = 0

Key Length = Cipher-suite-specific; see Table 8-2

Key Replay Counter = request EAPOL replay counter of AP

Key Nonce = INonce

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC (KCK of the STA_I, EAPOL)

Key Data Length = Length of Key Data field in octets

Key Data = Encrypted peer RSNIE, peer MAC address KDE, peer Nonce KDE, SMK KDE (contains SMK and INonce), Lifetime KDE

The AP sends Message 5 to the STA_I. On receipt of Message 5, the STA_I checks that the key replay counter corresponds to Message 5. If it does not, the STA_I silently discards the message. Otherwise,

a) STA_I verifies the Message 4 MIC using STA P PTKSA. If the calculated MIC does not match the MIC that the AP included in the EAPOL-Key frame, the STA_I silently discards Message 5.

b) If the MIC is correct, the STA_I identifies the PeerKey session using the initiator Nonce sent as part of the Key Nonce field of Message 5. If STA_I has an existing PeerKey state for this session, i.e., STA_I has initiated this message exchange using Message 1 and this message is a follow-up to that. If STA_I has an exisiing PeerKey state for this session, STA_I shall silently discard Message 5.

c) If all checks succeed, STA_I decrypts the Key Data field of Message 5 and extracts the peer RSNIE, the MAC_P, the INonce, the PNonce, the SMK, and the lifetime from Message 5.


Copyright © 2007 IEEE. All rights reserved.

d) The STA_I verifies that the peer RSNIE includes a valid cipher (i.e., one that was included in an initiator RSNIE). If not, STA_I discards the message and sends an Error KDE ERR_CPHR_NS.

e) The STA_I calculates SMKID per 8.5.1.4.

f) The STA_I checks the value of the lifetime with the maximum value it can support. If the lifetime suggested by the AP is too long, STA_I selects a lower value that can support.

g) Using all the information, the STA_I creates the SMKSA for this PeerKey session.

#### 8.5.8.2 PeerKey setup and handshake error conditions

If the STA_P does not receive a valid SMK Message 2 or a 4-Way STK Message 1 after sending the EAPOL request message to initiate the PeerKey rekey within a 200 ms timeout, the STA_P shall invoke an STSL application teardown procedure.

If the STA_I does not receive an SMK Message 5 from the AP, the STA_I shall attempt dot11RSNAConfigSMKUpdateCount transmits of the SMK Handshake Message 1 plus a final timeout. If the STA_I still has not received a response after these retries, it shall invoke an STSL application teardown procedure. The retransmit timeout value shall be 200 ms for the first timeout, the listen interval for the second timeout, and twice the listen interval for subsequent timeouts. If there is no listen interval, then 200 ms shall be used for all timeout values.

There is no specific recovery mechanism at the AP if the SMK Message 3 is dropped. This will result in a timeout by the STA_I after nonreceipt of SMK Message 5, as described in the preceding paragraph.

If the SMK Message 4 is not received by the STA_P, a failure will be detected during the 4-Way STK Handshake. In this case, the STA_P will discard the EAPOL-Key messages without the proper key. This failure is covered by behavior described in 8.5.3.5 and will result in teardown of the STSL.

Upon receipt of the SMK Message 5, the STA_I will transmit Message 1 of the 4-Way STK Handshake to the STA_P. If the STA_I does not receive Message 2 of the 4-Way STK Handshake from the STA_P, it shall attempt dot11RSNAConfigSMKUpdateCount transmits of 4-Way STK Handshake Message 1, plus a final timeout. If STA_I still has not received a response after these retries, it shall invoke an STSL application teardown procedure. 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.

There is no specific recovery mechanism at the STA_P if the SMK Message 3 is lost. This will result in a timeout on the STA_I, as described in the preceding paragraph, and a subsequent reinitiation of the SMK Handshake. The STA_P shall allow reinitiation of the SMK Handshake at any point prior to receipt of SMK Message 4.

#### 8.5.8.3 STKSA rekeying

Rekeying is always initiated by the STA_I. When needed, the STA_P sends an EAPOL request message to the STA_I to request rekeying. The STA_P shall wait a minimum of one half the IEEE 802.1X timeout after the STSL setup before initiating a PeerKey rekey procedure. To perform rekeying, there are two cases:

a) If SMK timer has not expired, the STAs will initiate a 4-Way Handshake to create a new STK. The 4-Way Handshake is always initiated by the STA_I. In this case, the STA_P should not delete any existing STKSA prior to verifying Message 3 of the 4-Way Handshake with STA_I for this session.

b) If the SMK has expired, the STA_I shall not use an existing STKSA and shall start the SMK Handshake followed by a 4-Way Handshake to create new keys.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 289 of 1232

The format of the EAPOL-Key request message in case a) from STA_P to STA_I is as follows:

**Request Message:** STA_P → STA_I: EAPOL-Key(1,1,0,0,1,0,0,0, MIC, PMKID KDE)

The request message uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 1 (PTK)

SMK Message = 0

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 0

Request = 1

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request replay counter of peer STA

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC computed over the body of this EAPOL-Key frame

Key Data Length = Length of Key Data field in octets

Key Data = SMKID in SMKID KDE

#### 8.5.8.4 Error Reporting

Error reporting messages are defined in this subclause and used to report errors whenever STAs or an AP detect an error during the SMK Handshake.

The AP, upon receiving the error messages defined in this subclause or upon generating the error messages defined in this subclause, should log the error. The STA, upon receipt of the error messages defined in this subclause, shall tear down the STSL with the other STA and clear all the PeerKey states.

The format of EAPOL-Key request message for reporting an error message is as follows:

**Error Message:** EAPOL-Key(1,1,0,0,0,1,0, 0, MIC, Error KDE, MAC Address KDE).

The request message uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 0 (Group/SMK)

SMK Message = 1 (SMK)


Copyright © 2007 IEEE. All rights reserved.

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 1

Error = 1

Request = 1 when the message is going from the STA to an AP or 0 when the message is going from an AP to the STA

Encrypted Key Data = 0

Reserved = 0

Key Length = 0

Key Replay Counter = request EAPOL replay counter

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC computed over the body of this EAPOL-Key frame

Key Data Length = Length of Key Data field in octets

Key Data = Error KDE (different types defined in Table 8-6), MAC Address KDE

#### 8.5.8.4.1 Error ERR_STA_NR

This error message is sent whenever an AP finds that the STA to which it needs to send a message is not reachable. In response to this error, the AP creates an Error KDE with error type ERR_STA_NR and sends the message back to the other STA involved in the handshake. The MAC address KDE contains the MAC address of the unreachable STA.

#### 8.5.8.4.2 Error ERR_STA_NRSN

This error message is sent whenever the AP finds that the STA to which it needs to send the message does not have a secure RSNA connection. In response of this error, the AP creates an Error KDE with error type ERR_STA_NRSN and sends the message back to the STA from which it received the last message. The MAC address KDE contains the MAC address of the STA with which the AP does not have a secure RSNA connection.

#### 8.5.8.4.3 Error ERR_CPHR_NS

This error message is sent whenever a STA finds that it does not support any of the cipher suites proposed by the other STA. In response to this error, the STA creates an Error KDE with error type ERR_CPHR_NS and sends the message back to the other STA. The MAC address KDE contains the MAC address of the other STA.

#### 8.5.8.4.4 Error ERR_NO_STSL

This error message is sent whenever a STA finds that it does not have an existing STSL with the other STA. In response of this error, the STA creates an Error KDE with error type ERR_NO_STSL and sends the message back to the other STA. The MAC address KDE contains the MAC address of the other STA.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 291 of 1232

## 8.6 Mapping EAPOL keys to IEEE 802.11 keys

### 8.6.1 Mapping PTK to TKIP keys

See 8.5.1.2 for the definition of the EAPOL temporal key derived from PTK.

A STA shall use bits 0–127 of the temporal key as its input to the TKIP Phase 1 and Phase 2 mixing functions.

A STA shall use bits 128–191 of the temporal key as the Michael key for MSDUs from the Authenticator's STA to the Supplicant's STA.

A STA shall use bits 192–255 of the temporal key as the Michael key for MSDUs from the Supplicant's STA to the Authenticator's STA.

### 8.6.2 Mapping GTK to TKIP keys

See 8.5.1.3 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–127 of the temporal key as the input to the TKIP Phase 1 and Phase 2 mixing functions.

A STA shall use bits 128–191 of the temporal key as the Michael key for MSDUs from the Authenticator's STA to the Supplicant's STA.

A STA shall use bits 192–255 of the temporal key as the Michael key for MSDUs from the Supplicant's STA to the Authenticator's STA.

### 8.6.3 Mapping PTK to CCMP keys

See 8.5.1.2 for the definition of the EAPOL temporal key derived from PTK.

A STA shall use the temporal key as the CCMP key for MSDUs between the two communicating STAs.

### 8.6.4 Mapping GTK to CCMP keys

See 8.5.1.3 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use the temporal key as the CCMP key.

### 8.6.5 Mapping GTK to WEP-40 keys

See 8.5.1.3 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–39 of the temporal key as the WEP-40 key.

### 8.6.6 Mapping GTK to WEP-104 keys

See 8.5.1.3 for the definition of the EAPOL temporal key derived from GTK.

A STA shall use bits 0–103 of the temporal key as the WEP-104 key.


Copyright © 2007 IEEE. All rights reserved.

## 8.7 Per-frame pseudo-code

### 8.7.1 WEP frame pseudo-code

An MPDU of type Data with the Protected Frame subfield of the Frame Control field equal to 1 is called a WEP MPDU. Other MPDUs of type Data are called non-WEP MPDUs.

A STA shall not transmit WEP-encapsulated MPDUs when value of the MIB variable dot11PrivacyInvoked is set to FALSE. This MIB variable does not affect MPDU or MMPDU reception.

**if** dot11PrivacyInvoked is “false” **then**
the MPDU is transmitted without WEP cryptographic encapsulation
**else**
**if** (the MPDU has an individual RA and there is an entry in dot11WEPKeyMappings for that RA) **then**
**if** that entry has WEPOn set to “false” **then**
the MPDU is transmitted without WEP cryptographic encapsulation
**else**
**if** that entry contains a key that is null **then**
discard the MPDU’s entire MSDU and generate an MA-UNITDATA.confirm primitive to notify LLC that the MSDU was undeliverable due to a null WEP key
**else**
encrypt the MPDU using that entry’s key, setting the Key ID subfield of the IV field to zero
**endif**
**endif**
**else**
**if** (the MPDU has a group RA and the Privacy subfield of the Capability Information field in this BSS is set to 0) **then**
the MPDU is transmitted without WEP cryptographic encapsulation
**else**
**if** dot11WEPDefaultKeys[dot11WEPDefaultKeyID] is null **then**
discard the MPDU’s entire MSDU and generate an MA-UNITDATA.confirm primitive to notify LLC that the MSDU was undeliverable due to a null WEP key
**else**
WEP-encapsulate the MPDU using the key dot11WEPDefaultKeys-[dot11WEPDefaultKeyID], setting the Key ID subfield of the IV field to dot11WEPDefaultKeyID
**endif**
**endif**
**endif**
**endif**

When the boolean attribute aExcludeUnencrypted is set to TRUE, non-WEP MPDUs shall not be indicated at the MAC service interface, and only MSDUs successfully reassembled from successfully decrypted MPDUs shall be indicated at the MAC service interface. When receiving a frame of type Data, the values of dot11PrivacyOptionImplemented, dot11WEPKeyMappings, dot11WEPDefaultKeys, dot11WEPDefault-KeyID, and aExcludeUnencrypted in effect at the time the PHY-RXSTART.indication primitive is received by the MAC shall be used according to the following decision tree:

**if** the Protected Frame subfield of the Frame Control field is zero **then**
**if** `aExcludeUnencrypted` is “true” **then**

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 293 of 1232

```
            discard the frame body without indication to LLC and increment
              dot11WEPExcludedCount
        else
            receive the frame without WEP decapsulation
        endif
    else
        if dot11PrivacyOptionImplemented is "true" then
            if (the MPDU has individual RA and there is an entry in dot11WEPKeyMappings match-
              ing the MPDU's TA) then
                if that entry has WEPOn set to "false" then
                    discard the frame body and increment dot11WEPUndecryptableCount
                else
                    if that entry contains a key that is null then
                        discard the frame body and increment dot11WEPUndecryptableCount
                    else
                        WEP-decapsulate with that key, incrementing dot11WEPICVErrorCount if
                          the ICV check fails
                    endif
                endif
            else
                if dot11WEPDefaultKeys[Key ID] is null then
                    discard the frame body and increment dot11WEPUndecryptableCount
                else
                    WEP-decapsulate with dot11WEPDefaultKeys[Key ID], incrementing
                      dot11WEPICVErrorCount if the ICV check fails
                endif
            endif
        else
            discard the frame body and increment dot11WEPUndecryptableCount
        endif
    endif
```

### 8.7.2 RSNA frame pseudo-code

STAs transmit protected MSDUs to a RA when temporal keys are configured and an MLME-SETPROTECTION.request primitive has been invoked for transmit to that RA. STAs expect to receive protected MSDUs from a TA when temporal keys are configured and an MLME-SET-PROTECTION.request primitive has been invoked for receive from that TA. MSDUs that do not match these conditions are sent in the clear and are received in the clear.

#### 8.7.2.1 Per-MSDU Tx pseudo-code

```
    if dot11RSNAEnabled = true then
        if MSDU has an individual RA and Protection for RA is off for Tx then
            transmit the MSDU without protections
        else if (MPDU has individual RA and Pairwise key exists for the MPDU's RA) or (MPDU has
          a multicast or broadcast RA and network type is IBSS and IBSS GTK exists for MPDU's
          TA) then
            // If we find a suitable Pairwise or GTK for the mode we are in…
            if key is a null key then
                discard the entire MSDU and generate an MA-UNITDATA.confirm primitive to
                  notify LLC that the MSDU was undeliverable due to a null key
            else
                // Note that it is assumed that no entry will be in the key
```


Copyright © 2007 IEEE. All rights reserved.

```
                    // mapping table of a cipher type that is unsupported.
                    Set the Key ID subfield of the IV field to zero.
                    if cipher type of entry is AES-CCM then
                        Transmit the MSDU, to be protected after fragmentation using AES-CCM
                    else if cipher type of entry is TKIP then
                        Compute MIC using Michael algorithm and entry's Tx MIC key.
                        Append MIC to MSDU
                        Transmit the MSDU, to be protected with TKIP
                    else if cipher type of entry is WEP then
                        Transmit the MSDU, to be protected with WEP
                    endif
                endif
            else // Else we did not find a key but we are protected, so handle the default key case or discard
                if GTK entry for Key ID contains null then
                    discard the MSDU and generate an MA-UNITDATA.confirm primitive to notify
                      LLC that the entire MSDU was undeliverable due to a null GTK
                else if GTK entry for Key ID is not null then
                    Set the Key ID subfield of the IV field to the Key ID.
                    if MPDU has an individual RA and cipher type of entry is not TKIP then
                        discard the entire MSDU and generate an MA-UNITDATA.confirm primitive
                          to notify LLC that the MSDU was undeliverable due to a null key
                    else if cipher type of entry is AES-CCM then
                        Transmit the MSDU, to be protected after fragmentation using AES-CCM
                    else if cipher type of entry is TKIP then
                        Compute MIC using Michael algorithm and entry's Tx MIC key.
                        Append MIC to MSDU
                        Transmit the MSDU, to be protected with TKIP
                    else if cipher type of entry is WEP then
                        Transmit the MSDU, to be protected with WEP
                    endif
                endif
            endif
        endif
```

#### 8.7.2.2 Per-MPDU Tx pseudo-code

```
    if dot11RSNAEnabled = TRUE then
        if MPDU is member of an MSDU that is to be transmitted without protections
            transmit the MPDU without protections
        else if MSDU that MPDU is a member of is to be protected using AES-CCM
            Protect the MPDU using entry's key and AES-CCM
            Transmit the MPDU
        else if MSDU that MPDU is a member of is to be protected using TKIP
            Protect the MPDU using TKIP encryption
            Transmit the MPDU
        else if MSDU that MPDU is a member of is to be protected using WEP
            Encrypt the MPDU using entry's key and WEP
            Transmit the MPDU
        else
            // should not arrive here
        endif
    endif
```

Copyright © 2007 IEEE. All rights reserved.

### 8.7.2.3 Per-MPDU Rx pseudo-code

```
if dot11RSNAEnabled = TRUE then
    if the Protected Frame subfield of the Frame Control field is zero then
        if Protection for TA is off for Rx then
            Receive the unencrypted MPDU without protections
        else
            discard the frame body without indication to LLC and increment
              dot11WEPExcludedCount
        endif
    else if Protection is true for TA then
        if ((MPDU has individual RA and Pairwise key exists for the MPDU's TA) or (MPDU
          has a broadcast/multicast RA and network type is IBSS and IBSS GTK exists for
          MPDU's RA)) then
            if key is null then
                discard the frame body and increment dot11WEPUndecryptableCount
            else if entry has an AES-CCM key then
                decrypt frame using AES-CCM key
                discard the frame if the integrity check fails and increment dot11RSNAStats-
                  CCMPDecryptErrors
            else if entry has a TKIP key then
                prepare a temporal key from the TA, TKIP key and PN
                decrypt the frame using ARC4
                discard the frame if the ICV fails and increment dot11RSNAStatsTKIPLocal-
                  MicFailures
            else if entry has a WEP key then
                decrypt the frame using WEP decryption
                discard the frame if the ICV fails and increment dot11WEPICVErrorCount
            else
                discard the frame body and increment dot11WEPUndecryptableCount
            endif
        else if GTK for the Key ID does not exist then
            discard the frame body and increment dot11WEPUndecryptableCount
        else if GTK for the Key ID is null then
            discard the frame body and increment dot11WEPUndecryptableCount
        else if the GTK for the Key ID is a CCM key then
            decrypt frame using AES-CCM key
            discard the frame if the integrity check fails and increment dot11RSNAStatsCCMP-
              DecryptErrors
        else if the GTK for the Key ID is a TKIP key then
            prepare a temporal key from the TA, TKIP key and PN
            decrypt the frame using ARC4
            discard the frame if the ICV fails and increment dot11RSNAStatsTKIPICVErrors
        else if the GTK for the Key ID is a WEP key then
            decrypt the frame using WEP decryption
            discard the frame if the ICV fails and increment dot11WEPICVErrorCount
        endif
    else
        MLME-PROTECTEDFRAMEDROPPED.indication
        discard the frame body and increment dot11WEPUndecryptableCount
    endif
endif
```


Copyright © 2007 IEEE. All rights reserved.

#### 8.7.2.4 Per-MSDU Rx pseudo-code

**if** *dot11RSNAEnabled* = TRUE **then**
  **if** the frame was not protected **then**
    Receive the MSDU unprotected
    Make MSDU available to higher layers
  **else//** Have a protected MSDU
    **if** Pairwise key is an AES-CCM key **then**
      Accept the MSDU if its MPDUs had sequential PNs (or if it consists of only one MPDU), otherwise discard the MSDU as a replay attack and increment dot11RSNAStatsCCMPReplays
      Make MSDU available to higher layers
    **else if** Pairwise key is a TKIP key **then**
      Compute the MIC using the Michael algorithm
      Compare the received MIC against the computed MIC
      discard the frame if the MIC fails increment dot11RSNAStatsTKIPLocalMIC-Failures and invoke countermeasures if appropriate
      compare TSC against replay counter, if replay check fails increment dot11RSNA-StatsTKIPReplays
      otherwise accept the MSDU
      Make MSDU available to higher layers
    **else if** dot11WEPKeyMappings has a WEP key **then**
      Accept the MSDU since the decryption took place at the MPDU
      Make MSDU available to higher layers
    **endif**
  **endif**
**endif**

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 297 of 1232

Copyright © 2007 IEEE. All rights reserved.
Samsung Ex. 1044, Page 298 of 1232

# 9. MAC sublayer functional description

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), 9.3 (PCF), and 9.9 (HCF). Fragmentation and defragmentation are defined in 9.4 and 9.5. Multirate support is addressed in 9.6. A number of additional restrictions to limit the cases in which MSDUs are reordered or discarded are described in 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

The MAC architecture can be described as shown in Figure 9-1 as providing the PCF and HCF through the services of the DCF. Note that in a non-QoS STA, HCF is not present. In a QoS STA implementation, both DCF and HCF are present. PCF is optional in all STAs.

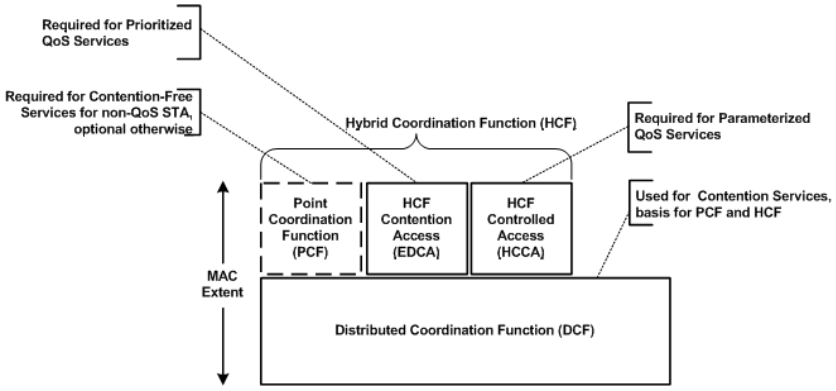



**Figure 9-1—MAC architecture**

### 9.1.1 DCF

The fundamental access method of the IEEE 802.11 MAC is a DCF known as *carrier sense multiple access with collision avoidance* (CSMA/CA). The DCF shall be implemented in all STAs, for use within both IBSS and infrastructure network configurations.

For a STA to transmit, it shall sense the medium to determine if another STA is transmitting. If the medium is not determined to be busy (see 9.2.1), the transmission may proceed. The CSMA/CA distributed algorithm mandates that a gap of a minimum specified duration exist between contiguous frame sequences. A transmitting STA shall ensure that the medium is idle for this required duration before attempting to transmit. If the medium is determined to be busy, the STA shall defer until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the STA shall select a random backoff interval and shall decrement the backoff interval counter while the medium is idle. A transmission is successful either when an ACK frame is received from the STA addressed by the RA field of the transmitted frame or when a frame with a group address in the RA field is transmitted completely. A refinement of the method may be used under various circumstances to further minimize collisions—here the transmitting and receiving STA exchange short control frames (RTS and CTS frames) after determining that the medium is idle and after any deferrals or backoffs, prior to data transmission. The details of CSMA/CA, deferrals, and backoffs are described in 9.2. RTS/CTS exchanges are also presented in 9.2.

Copyright © 2007 IEEE. All rights reserved.

### 9.1.2 PCF

The IEEE 802.11 MAC may also incorporate an optional access method called a PCF, which is only usable on infrastructure network configurations. This access method uses a PC, which shall operate at the AP of the BSS, to determine which STA currently has the right to transmit. The operation is essentially that of polling, with the PC performing the role of the polling master. The operation of the PCF may require additional coordination, not specified in this standard, to permit efficient operation in cases where multiple point-coordinated BSSs are operating on the same channel, in overlapping physical space.

The PCF uses a virtual carrier sense (CS) mechanism aided by an access priority mechanism. The PCF shall distribute information within Beacon management frames to gain control of the medium by setting the NAV in STAs. In addition, all frame transmissions under the PCF may use an interframe space (IFS) that is smaller than the IFS for frames transmitted via the DCF. The use of a smaller IFS implies that point-coordinated traffic shall have priority access to the medium over STAs in overlapping BSSs operating under the DCF access method.

The access priority provided by a PCF may be utilized to create a CF access method. The PC controls the frame transmissions of the STAs so as to eliminate contention for a limited period of time.

### 9.1.3 Hybrid coordination function (HCF)

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

STAs 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 STAs using eight different UPs. The EDCA mechanism defines four access categories (ACs) that provide support for the delivery of traffic with UPs at the STAs. The AC is derived from the UPs as shown in Table 9-1.

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 AP of the BSS with which the STA is associated, where

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

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 an AP and in the MIB table dot11EDCATableAIFSN for a non-AP STA; see 9.9.1) assigned either by a management entity or by an AP.


Copyright © 2007 IEEE. All rights reserved.

**Table 9-1—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 an AP and in the MIB attribute tables dot11EDCATableCWmin and dot11EDCATableCWmax for a non-AP STA) and assigned by a management entity or by an AP.

d) Collisions between contending EDCAFs within a STA are resolved within the STA 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 STA 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 an AP and in dot11EDCATableTXOPLimit MIB table for a non-AP STA. 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 BSS.

The QoS AP 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 STAs 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 STAs are able to receive the updated EDCA parameters. A QoS STA 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. QoS STAs update the MIB attributes and store the EDCA Parameter Set update count value in the QoS Info field. An AP may change the EDCA access parameters by changing the EDCA Parameter Set element in the Beacon and Probe Response frames. However, the AP should change them only rarely. A QoS STA shall use the EDCA Parameter Set Update Count Value subfield in the QoS Capability element of all Beacon frames to determine whether the STA 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 QoS STA shall query the updated EDCA parameter values by sending a Probe Request frame to the AP.

Copyright © 2007 IEEE. All rights reserved.

The AP may use a different set of EDCA parameters than it advertises to the STAs in its BSS.

The management frames shall be sent using the access category AC_VO without being restricted by admission control procedures. A QoS STA shall also send management frames using the access category AC_VO before associating with any BSS, 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 AP of the BSS 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 STAs 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 BSS-wide knowledge of the amounts of pending traffic belonging to different TS and/or TCs and are subject to BSS-specific QoS policies.

An AP may indicate availability of CF-Polls to non-QoS STAs, thereby providing non-QoS contention-free transfers during the CFP. This provisioning of contention-free transfers during the CFP to non-QoS STAs, however, is not recommended. Implementers are cautioned that QoS STAs 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. QoS STAs 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, QoS STAs cannot be treated as CF-Pollable STAs. This requires an AP 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 non-QoS STAs are associated and being polled.

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

A STA 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.

### 9.1.4 Combined use of DCF, PCF, and HCF

The DCF and a centralized coordination function (either PCF or HCF) are defined so they may operate within the same BSS. When a PC is operating in a BSS, the PCF and DCF access methods alternate, with a CFP followed by a CP. This is described in greater detail in 9.3. When an HC is operating in a BSS, 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.


Copyright © 2007 IEEE. All rights reserved.

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.

### 9.1.5 Fragmentation/defragmentation overview

The process of partitioning an 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 (as defined in 9.1.1) of the MSDU or MMPDU in cases where channel characteristics limit reception reliability for longer frames. STAs 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 dot11FragmentationThreshold.

When an individually addressed MSDU is received from the LLC or an individually addressed MMPDU is received from the MLME that would result in a length greater than dot11FragmentationThreshold when the MAC header and FCS are added, the MSDU or MMPDU shall be fragmented. The MSDU or MMPDU is divided into MPDUs. Each fragment is a frame no larger than dot11FragmentationThreshold. It is possible that any fragment may be a frame smaller than dot11FragmentationThreshold. An illustration of fragmentation is shown in Figure 9-2.

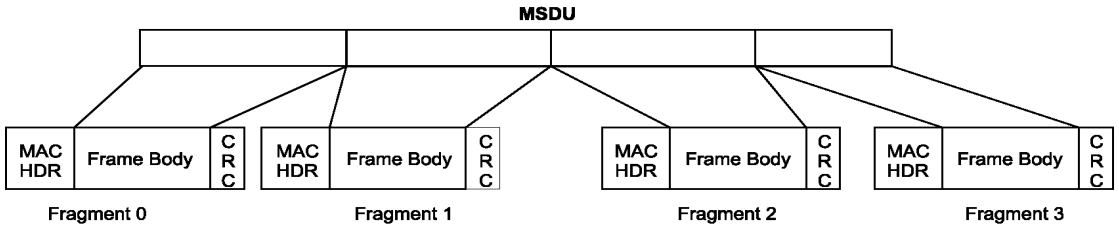



**Figure 9-2—Fragmentation**

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 STA, 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.

### 9.1.6 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.

Copyright © 2007 IEEE. All rights reserved.

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.

In a QoS STA, the TID parameter of the MA-UNITDATA.request results in a TID being specified for the transmitted MSDU. This TID associates the MSDU with the AC or TS queue for the indicated traffic.

## 9.2 DCF

The basic medium access protocol is a DCF that allows for automatic medium sharing between compatible PHYs through the use of CSMA/CA and a random backoff time following a busy medium condition. In addition, all individually addressed traffic uses immediate positive acknowledgment (ACK frame) where retransmission is scheduled by the sender if no ACK is received.

The CSMA/CA protocol is designed to reduce the collision probability between multiple STAs accessing a medium, at the point where collisions would most likely occur. Just after the medium becomes idle following a busy medium (as indicated by the CS function) is when the highest probability of a collision exists. This is because multiple STAs could have been waiting for the medium to become available again. This is the situation that necessitates a random backoff procedure to resolve medium contention conflicts.

CS shall be performed both through physical and virtual mechanisms.

The virtual CS mechanism is achieved by distributing reservation information announcing the impending use of the medium. The exchange of RTS and CTS frames prior to the actual data frame is one means of distribution of this medium reservation information. The RTS and CTS frames contain a Duration field that defines the period of time that the medium is to be reserved to transmit the actual data frame and the returning ACK frame. All STAs within the reception range of either the originating STA (which transmits the RTS) or the destination STA (which transmits the CTS) shall learn of the medium reservation. Thus, a STA can be unable to receive from the originating STA and yet still know about the impending use of the medium to transmit a data frame.

Another means of distributing the medium reservation information is the Duration/ID field in individually addressed frames. This field gives the time that the medium is reserved, either to the end of the immediately following ACK, or in the case of a fragment sequence, to the end of the ACK following the next fragment.

The RTS/CTS exchange also performs both a type of fast collision inference and a transmission path check. If the return CTS is not detected by the STA originating the RTS, the originating STA may repeat the process (after observing the other medium-use rules) more quickly than if the long data frame had been transmitted and a return ACK frame had not been detected.

Another advantage of the RTS/CTS mechanism occurs where multiple BSSs utilizing the same channel overlap. The medium reservation mechanism works across the BSA boundaries. The RTS/CTS mechanism may also improve operation in a typical situation where all STAs can receive from the AP, but may not be able to receive from all other STAs in the BSA.

The RTS/CTS mechanism cannot be used for MPDUs with broadcast and multicast immediate destination because there are multiple recipients for the RTS, and thus potentially multiple concurrent senders of the CTS in response. The RTS/CTS mechanism need not be used for every data frame transmission. Because the additional RTS and CTS frames add overhead inefficiency, the mechanism is not always justified, especially for short data frames.


Copyright © 2007 IEEE. All rights reserved.

The use of the RTS/CTS mechanism is under control of the dot11RTSThreshold attribute. This attribute may be set on a per-STA basis. This mechanism allows STAs to be configured to use RTS/CTS either always, never, or only on frames longer than a specified length.

A STA configured not to initiate the RTS/CTS mechanism shall still update its virtual CS mechanism with the duration information contained in a received RTS or CTS frame, and shall always respond to an RTS addressed to it with a CTS if permitted by medium access rules.

The medium access protocol allows for STAs to support different sets of data rates. All STAs shall be able to receive and transmit at all the data rates in the BSSBasicRateSet parameter of the MLME-START.request or BSSBasicRateSet parameter of the BSSDescription representing the SelectedBSS parameter of the MLME-JOIN.request. To support the proper operation of the RTS/CTS and the virtual CS mechanism, all STAs shall be able to detect the RTS and CTS frames.

Data frames sent under the DCF shall use the frame type Data and subtype Data or Null Function. STAs receiving Data type frames shall not indicate a data frame to LLC when the subtype is Null Function, but shall indicate a data frame to LLC when the subtype is Data, even if the frame body contains zero octets.

### 9.2.1 CS mechanism

Physical and virtual CS functions are used to determine the state of the medium. When either function indicates a busy medium, the medium shall be considered busy; otherwise, it shall be considered idle.

A physical CS mechanism shall be provided by the PHY. See Clause 12 for how this information is conveyed to the MAC. The details of physical CS are provided in the individual PHY specifications.

A virtual CS mechanism shall be provided by the MAC. This mechanism is referred to as the NAV. The NAV maintains a prediction of future traffic on the medium based on duration information that is announced in RTS/CTS frames prior to the actual exchange of data. The duration information is also available in the MAC headers of all frames sent during the CP other than PS-Poll Control frames. The mechanism for setting the NAV using RTS/CTS in the DCF is described in 9.2.5.4, use of the NAV in PCF is described in 9.3.2.2, and use of the NAV in HCF is described in 9.9.2.2.1. Additional details regarding NAV usage and update appear in 9.2.5.6, 9.2.11, and 9.13.

The CS mechanism combines the NAV state and the STA's transmitter status with physical CS to determine the busy/idle state of the medium. The NAV may be thought of as a counter, which counts down to zero at a uniform rate. When the counter is zero, the virtual CS indication is that the medium is idle; when nonzero, the indication is busy. The medium shall be determined to be busy when the STA is transmitting.

### 9.2.2 MAC-Level acknowledgments

The reception of some frames, as described in 9.2.8, 9.3.3.4, and 9.12, requires the receiving STA to respond with an acknowledgment, generally an ACK frame, if the FCS of the received frame is correct. This technique is known as positive acknowledgment.

Lack of reception of an expected ACK frame indicates to the STA initiating the frame exchange that an error has occurred. Note, however, that the destination STA may have received the frame correctly, and that the error may have occurred in the transfer or reception of the ACK frame. To the initiator of the frame exchange, this condition is indistinguishable from an error occurring in the initial frame.

Copyright © 2007 IEEE. All rights reserved.

### 9.2.3 IFS

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. Five different IFSs are defined to provide priority levels for access to the wireless media. Figure 9-3 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 defined as time gaps on the medium, and the IFS timings except AIFS are fixed for each PHY (even in multirate-capable PHYs). The IFS values are determined from attributes specified by the PHY.

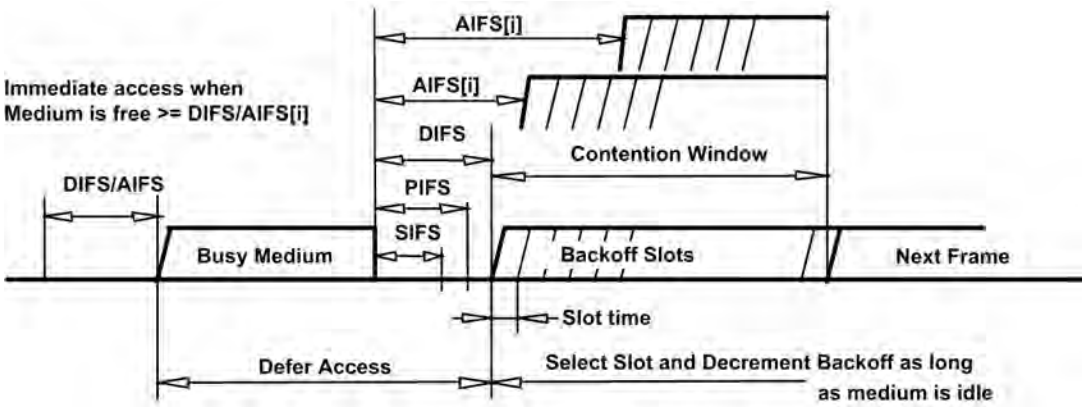



**Figure 9-3—Some IFS relationships**

#### 9.2.3.1 SIFS

The SIFS shall be used prior to transmission of an ACK frame, a CTS frame, the second or subsequent MPDU of a fragment burst, and by a STA responding to any polling by the PCF. The SIFS may also be used by a PC for any types of frames during the CFP (see 9.3). The SIFS is the time from the end of the last symbol of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface. The valid cases where the SIFS may or shall be used are listed in the frame exchange sequences in 9.12.

The SIFS timing shall be achieved when the transmission of the subsequent frame is started at the TxSIFS Slot boundary as specified in 9.2.10. An IEEE 802.11 implementation shall not allow the space between frames that are defined to be separated by a SIFS time, as measured on the medium, to vary from the nominal SIFS value by more than ±10% of aSlotTime for the PHY in use.

SIFS is the shortest of the IFSs. SIFS shall be used when STAs have seized the medium and need to keep it for the duration of the frame exchange sequence to be performed. Using the smallest gap between transmissions within the frame exchange sequence prevents other STAs, which are required to wait for the medium to be idle for a longer gap, from attempting to use the medium, thus giving priority to completion of the frame exchange sequence in progress.


Copyright © 2007 IEEE. All rights reserved.

#### 9.2.3.2 PIFS

The PIFS shall be used only by STAs operating under the PCF to gain priority access to the medium at the start of the CFP or by a STA to transmit a Channel Switch Announcement frame. A STA using the PCF shall be allowed to transmit CF traffic after its CS mechanism (see 9.2.1) determines that the medium is idle at the TxPIFS slot boundary as defined in 9.2.10. A STA may also transmit a Channel Switch Announcement frame after its CS mechanism (see 9.2.1) determines that the medium is idle at the TxPIFS slot boundary. The use of the PIFS by STAs operating under the PCF is described in 9.3. The use of PIFS by STAs transmitting a Channel Switch Announcement frame is described in 11.9.

#### 9.2.3.3 DIFS

The DIFS shall be used by STAs operating under the DCF to transmit data frames (MPDUs) and management frames (MMPDUs). A STA using the DCF shall be allowed to transmit if its CS mechanism (see 9.2.1) determines that the medium is idle at the TxDIFS slot boundary as defined in 9.2.10 after a correctly received frame, and its backoff time has expired. A correctly received frame is one where the PHY-RXEND.indication does not indicate an error and the FCS indicates the frame is error free.

#### 9.2.3.4 AIFS

The AIFS shall be used by QoS STAs 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 STA using the EDCA shall obtain a TXOP for an AC if the STA'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 STA using the EDCA shall not transmit within an EIFS-DIFS+AIFS[AC] plus any backoff time after that STA 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 STA to transmit using the AIFS[AC] following that subsequent frame, provided that the backoff time for that AC has expired.

A non-AP QoS STA computes the time periods for each AIFS[AC] from the dot11EDCATableAIFSN attributes in the MIB. QoS STAs update their dot11EDCATableAIFSN values using information in the most recent EDCA Parameter Set element of Beacon frames received from the AP of the BSS (see 7.3.2.28). A QoS AP computes the time periods for each AIFS[AC] from the dot11QAPEDCATableAIFSN attributes in its MIB.

#### 9.2.3.5 EIFS

A STA's DCF shall use EIFS before transmission, when it determines that the medium is idle following reception of a frame for which the PHY-RXEND.indication primitive contained an error or a frame for which the MAC FCS value was not correct. Similarly, a STA's EDCA mechanism under HCF shall use the EIFS-DIFS+AIFS[AC] interval. The duration of an EIFS is defined in 9.2.10. The EIFS or EIFS-DIFS+AIFS[AC] 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 STA shall not begin a transmission until the expiration of the later of the NAV and EIFS or EIFS-DIFS+AIFS[AC]. The EIFS and EIFS-DIFS+AIFS[AC] are 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 or EIFS-DIFS+AIFS[AC] resynchronizes the STA to the actual busy/idle state of the medium, so the EIFS or EIFS-DIFS+AIFS[AC] is terminated and normal medium access (using DIFS or AIFS as appropriate and, if necessary, backoff) continues following reception of that frame. At the expiration or termination of the EIFS or EIFS-DIFS+AIFS[AC], the STA reverts to the NAV and physical CS to control access to the medium.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 307 of 1232

### 9.2.4 Random backoff time

A STA desiring to initiate transfer of data MPDUs and/or MMPDUs shall invoke the CS mechanism (see 9.2.1) to determine the busy/idle state of the medium. If the medium is busy, the STA shall defer until the medium is determined to be idle without interruption for a period of time equal to DIFS when the last frame detected on the medium was received correctly, or after the medium is determined to be idle without interruption for a period of time equal to EIFS when the last frame detected on the medium was not received correctly. After this DIFS or EIFS medium idle time, the STA shall then generate a random backoff period for an additional deferral time before transmitting, unless the backoff timer already contains a nonzero value, in which case the selection of a random number is not needed and not performed. This process minimizes collisions during contention between multiple STAs that have been deferring to the same event.

Backoff Time = Random() × aSlotTime

where

Random() = Pseudo-random integer drawn from a uniform distribution over the interval [0,CW], where CW is an integer within the range of values of the PHY characteristics aCWmin and aCWmax, aCWmin ≤ CW ≤ aCWmax. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

aSlotTime = The value of the correspondingly named PHY characteristic.

The contention window (CW) parameter shall take an initial value of aCWmin. Every STA shall maintain a STA short retry count (SSRC) as well as a STA long retry count (SLRC), both of which shall take an initial value of zero. The SSRC shall be incremented when any short retry count (SRC) associated with any MPDU of type Data is incremented. The SLRC shall be incremented when any long retry count (LRC) associated with any MPDU of type Data is incremented. The CW shall take the next value in the series every time an unsuccessful attempt to transmit an MPDU causes either STA retry counter to increment, until the CW reaches the value of aCWmax. A retry is defined as the entire sequence of frames sent, separated by SIFS intervals, in an attempt to deliver an MPDU, as described in 9.12. Once it reaches aCWmax, the CW shall remain at the value of aCWmax until the CW is reset. This improves the stability of the access protocol under high-load conditions. See Figure 9-4.

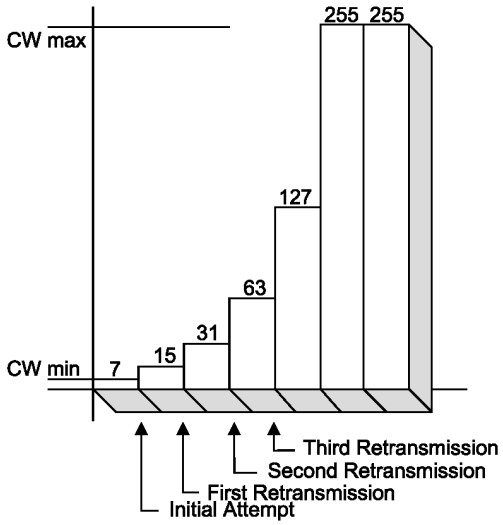



**Figure 9-4—Example of exponential increase of CW**


Copyright © 2007 IEEE. All rights reserved.

The CW shall be reset to aCWmin after every successful attempt to transmit an MPDU or MMPDU, when SLRC reaches dot11LongRetryLimit, or when SSRC reaches dot11ShortRetryLimit. The SSRC shall be reset to 0 when a CTS frame is received in response to an RTS frame, when an ACK frame is received in response to an MPDU or MMPDU transmission, or when a frame with a group address in the Address1 field is transmitted. The SLRC shall be reset to 0 when an ACK frame is received in response to transmission of an MPDU or MMPDU of length greater than dot11RTSThreshold, or when a frame with a group address in the Address1 field is transmitted.

The set of CW values shall be sequentially ascending integer powers of 2, minus 1, beginning with a PHY-specific aCWmin value, and continuing up to and including a PHY-specific aCWmax value.

### 9.2.5 DCF access procedure

The CSMA/CA access method is the foundation of the DCF. The operational rules vary slightly between the DCF and the PCF.

#### 9.2.5.1 Basic access

Basic access refers to the core mechanism a STA uses to determine whether it may transmit.

In general, a STA may transmit a pending MPDU when it is operating under the DCF access method, either in the absence of a PC, or in the CP of the PCF access method, when the STA determines that the medium is idle for greater than or equal to a DIFS period, or an EIFS period if the immediately preceding medium-busy event was caused by detection of a frame that was not received at this STA with a correct MAC FCS value. If, under these conditions, the medium is determined by the CS mechanism to be busy when a STA desires to initiate the initial frame of one of the frame exchanges described in 9.12, exclusive of the CF period, the random backoff procedure described in 9.2.5.2 shall be followed. There are conditions, specified in 9.2.5.2 and 9.2.5.5, where the random backoff procedure shall be followed even for the first attempt to initiate a frame exchange sequence.

In a STA having an FH PHY, control of the channel is lost at the dwell time boundary and the STA shall have to contend for the channel after that dwell boundary. It is required that STAs having an FH PHY complete transmission of the entire MPDU and associated acknowledgment (if required) before the dwell time boundary. If, when transmitting or retransmitting an MPDU, there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment (if required), the STA shall defer the transmission by selecting a random backoff time, using the present CW (without advancing to the next value in the series). The short retry counter and long retry counter for the MSDU are not affected.

The basic access mechanism is illustrated in Figure 9-5.

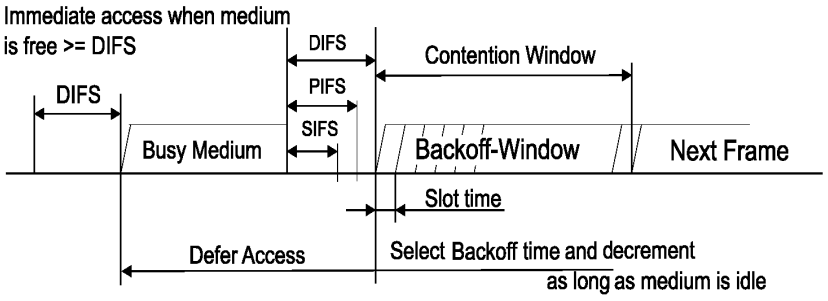



**Figure 9-5—Basic access method**

Copyright © 2007 IEEE. All rights reserved.

### 9.2.5.2 Backoff procedure for DCF

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.

The backoff procedure shall be invoked for a STA to transfer a frame when finding the medium busy as indicated by either the physical or virtual CS mechanism (see Figure 9-6). The backoff procedure shall also be invoked when a transmitting STA infers a failed transmission as defined in 9.2.5.7 or 9.2.8.

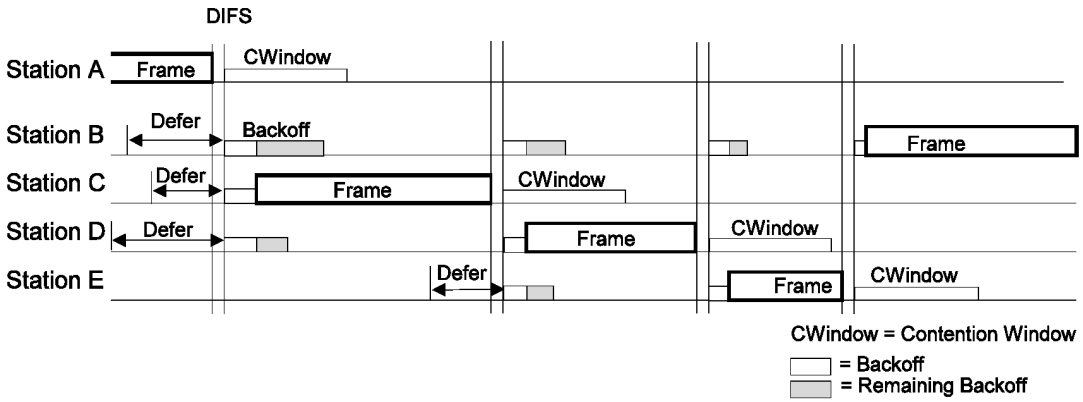



**Figure 9-6—Backoff procedure**

To begin the backoff procedure, the STA shall set its Backoff Timer to a random backoff time using the equation in 9.2.4. All backoff slots occur following a DIFS period during which the medium is determined to be idle for the duration of the DIFS period, or following an EIFS period during which the medium is determined to be idle for the duration of the EIFS period, as appropriate (see 9.2.3).

A STA performing the backoff procedure shall use the CS mechanism (see 9.2.1) to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff time by aSlotTime.

If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended; that is, the backoff timer shall not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS period or EIFS, as appropriate (see 9.2.3), before the backoff procedure is allowed to resume. Transmission shall commence when the Backoff Timer reaches zero.

A backoff procedure shall be performed immediately after the end of every transmission with the More Fragments bit set to 0 of an MPDU of type Data, Management, or Control with subtype PS-Poll, even if no additional transmissions are currently queued. In the case of successful acknowledged transmissions, this backoff procedure shall begin at the end of the received ACK frame. In the case of unsuccessful transmissions requiring acknowledgment, this backoff procedure shall begin at the end of the ACKTimeout interval (as defined in 9.2.8). An unsuccessful transmission is one where an ACK frame is not received from the STA addressed by the RA field of the transmitted frame and the value of the RA field is an individual address. If the transmission is successful, the CW value reverts to aCWmin before the random backoff interval is chosen, and the SSRC and/or SLRC are updated as described in 9.2.4. This assures that transmitted frames from a STA are always separated by at least one backoff interval.

The effect of this procedure is that when multiple STAs are deferring and go into random backoff, then the STA selecting the smallest backoff time using the random function will win the contention (assuming all of the contending STAs detect the same instances of WM activity at their respective receivers).


Copyright © 2007 IEEE. All rights reserved.

In an IBSS, the backoff time for a pending nonbeacon or non-ATIM transmission shall not decrement in the period from the TBTT until the expiration of the ATIM window, and the backoff time for a pending ATIM management frame shall decrement only within the ATIM window. (See Clause 11.) Within an IBSS, a separate backoff interval shall be generated to precede the transmission of a Beacon frame, as described in 11.1.2.2.

#### 9.2.5.3 Recovery procedures and retransmit limits

Error recovery is always the responsibility of the STA that initiates a frame exchange sequence, as defined in 9.12. Many circumstances may cause an error to occur that requires recovery. For example, the CTS frame may not be returned after an RTS frame is transmitted. This may happen due to a collision with another transmission, due to interference in the channel during the RTS or CTS frame, or because the STA receiving the RTS frame has an active virtual CS condition (indicating a busy medium time period).

Error recovery shall be attempted by retrying transmissions for frame exchange sequences that the initiating STA infers have failed. Retries shall continue, for each failing frame exchange sequence, until the transmission is successful, or until the relevant retry limit is reached, whichever occurs first. STAs shall maintain a SRC and a LRC for each MSDU or MMPDU awaiting transmission. These counts are incremented and reset independently of each other.

After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 9.2.5.7. If the RTS transmission fails, the SRC for the MSDU or MMPDU and the SSRC are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11ShortRetryLimit.

After transmitting a frame that requires acknowledgment, the STA shall perform the ACK procedure, as defined in 9.2.8. The SRC for an MPDU of type Data or MMPDU and the SSRC shall be incremented every time transmission of a MAC frame of length less than or equal to dot11RTSThreshold fails for that MPDU of type Data or MMPDU. This SRC and the SSRC shall be reset when a MAC frame of length less than or equal to dot11RTSThreshold succeeds for that MPDU of type Data or MMPDU. The LRC for an MPDU of type Data or MMPDU and the SLRC shall be incremented every time transmission of a MAC frame of length greater than dot11RTSThreshold fails for that MPDU of type Data or MMPDU. This LRC and the SLRC shall be reset when a MAC frame of length greater than dot11RTSThreshold succeeds for that MPDU of type Data or MMPDU. All retransmission attempts for an MPDU of type Data or MMPDU that has failed the ACK procedure one or more times shall be made with the Retry field set to 1 in the Data or Management type frame.

Retries for failed transmission attempts shall continue until the SRC for the MPDU of type Data or MMPDU is equal to dot11ShortRetryLimit or until the LRC for the MPDU of type Data or MMPDU is equal to dot11LongRetryLimit. When either of these limits is reached, retry attempts shall cease, and the MPDU of type Data (and any MSDU of which it is a part) or MMPDU shall be discarded.

A STA in PS mode, in an ESS, initiates a frame exchange sequence by transmitting a PS-Poll frame to request data from an AP. In the event that neither an ACK frame nor a data frame is received from the AP in response to a PS-Poll frame, then the STA shall retry the sequence, by transmitting another PS-Poll frame, at its convenience. If the AP sends a data frame in response to a PS-Poll frame, but fails to receive the ACK frame acknowledging this data frame, the next PS-Poll frame from the same STA may cause a retransmission of the last MSDU. This duplicate MSDU shall be filtered at the receiving STA using the normal duplicate frame filtering mechanism. If the AP responds to a PS-Poll by transmitting an ACK frame, then responsibility for the data frame delivery error recovery shifts to the AP because the data are transferred in a subsequent frame exchange sequence, which is initiated by the AP. The AP shall attempt to deliver one MSDU to the STA that transmitted the PS-Poll, using any frame exchange sequence valid for an individually addressed MSDU. If the PS STA that transmitted the PS-Poll returns to Doze state after transmitting the ACK frame in response to successful receipt of this MSDU, but the AP fails to receive this ACK frame, the AP will retry transmission of this MSDU until the relevant retry limit is reached. See Clause 11 for details on filtering of extra PS-Poll frames.

Copyright © 2007 IEEE. All rights reserved.

### 9.2.5.4 Setting and resetting the NAV

STAs receiving a valid frame shall update their NAV with the information received in the Duration field for all frames where the new NAV value is greater than the current NAV value, except the NAV shall not be updated where the RA is equal to the receiving STA's MAC address. Upon receipt of a PS-Poll frame, a STA shall update its NAV settings as appropriate under the data rate selection rules using a duration value equal to the time, in microseconds, required to transmit one ACK frame plus one SIFS interval, but only when the new NAV value is greater than the current NAV value. If the calculated duration includes a fractional microsecond, that value is rounded up the next higher integer. Various additional conditions may set or reset the NAV, as described in 9.3.2.2. When the NAV is reset, a PHY-CCARESET.request shall be issued.

Figure 9-7 indicates the NAV for STAs that may receive the RTS frame, while other STAs may only receive the CTS frame, resulting in the lower NAV bar as shown (with the exception of the STA to which the RTS was addressed).

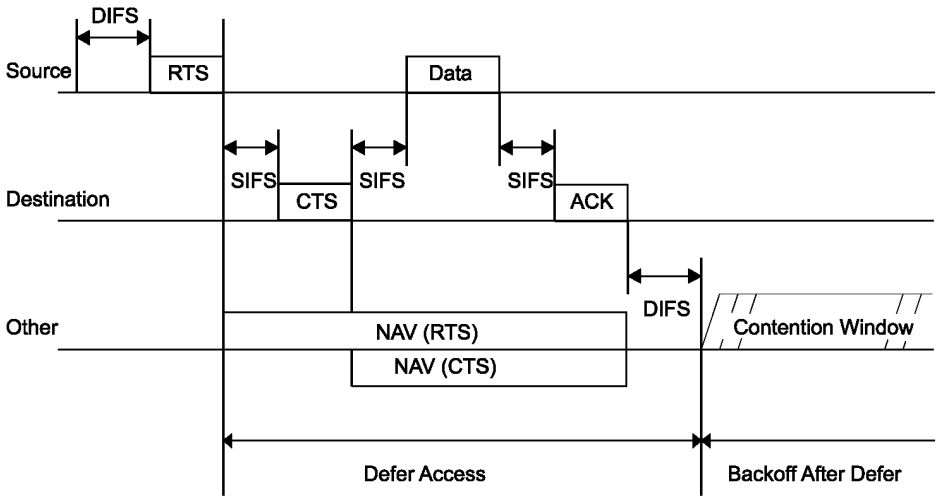



**Figure 9-7—RTS/CTS/data/ACK and NAV setting**

A STA that used information from an RTS frame as the most recent basis to update its NAV setting is permitted to reset its NAV if no PHY-RXSTART.indication is detected from the PHY during a period with a duration of (2 × aSIFSTime) + (CTS_Time) + aPHY-RX-START-Delay + (2 × aSlotTime) starting at the PHY-RXEND.indication corresponding to the detection of the RTS frame. The "CTS_Time" shall be calculated using the length of the CTS frame and the data rate at which the RTS frame used for the most recent NAV update was received.

### 9.2.5.5 Control of the channel

The SIFS is used to provide an efficient MSDU delivery mechanism. Once the STA has contended for the channel, that STA shall continue to send fragments until either all fragments of a single MSDU or MMPDU have been sent, an acknowledgment is not received, or the STA is restricted from sending any additional fragments due to a dwell time boundary. Should the sending of the fragments be interrupted due to one of these reasons, when the next opportunity for transmission occurs the STA shall resume transmission. The algorithm by which the STA decides which of the outstanding MSDUs shall next be attempted after an unsuccessful transmission (as defined in 9.2.5.2) attempt is beyond the scope of this standard, but any such algorithm shall comply with the restrictions listed in 9.7.


Copyright © 2007 IEEE. All rights reserved.

Figure 9-8 illustrates the transmission of a multiple-fragment MSDU using the SIFS.

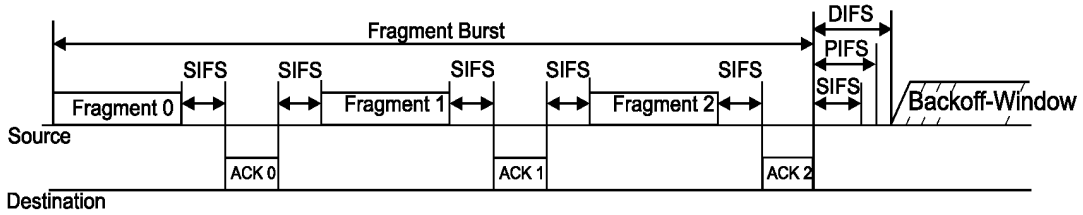



**Figure 9-8—Transmission of a multiple-fragment MSDU using SIFS**

When the source STA transmits a fragment, it shall release the channel, then immediately monitor the channel for an acknowledgment as described in 9.2.8.

When the destination STA has finished sending the acknowledgment, the SIFS following the acknowledgment shall be reserved for the source STA to continue (if necessary) with another fragment. The STA sending the acknowledgment shall not transmit on the channel immediately following the acknowledgment.

The process of sending multiple fragments after contending for the channel is defined as a fragment burst.

If the source STA receives an acknowledgment but there is not enough time to transmit the next fragment and receive an acknowledgment due to an impending dwell boundary, the source STA shall contend for the channel at the beginning of the next dwell time.

If the source STA does not receive an acknowledgment frame, it shall attempt to retransmit the failed MPDU or another eligible MPDU, as defined in 9.7, after performing the backoff procedure and the contention process.

After a STA contends for the channel to retransmit a fragment of an MSDU, it shall start with the last fragment that was not acknowledged. The destination STA shall receive the fragments in order (because the source sends them in order and they are individually acknowledged). It is possible, however, that the destination STA may receive duplicate fragments. It shall be the responsibility of the receiving STA to detect and discard duplicate fragments.

A STA shall transmit after the SIFS only under the following conditions during a fragment burst:

— The STA has just received a fragment that requires acknowledgment.

— The source STA has received an acknowledgment for a previous fragment, has more fragment(s) for the same MSDU to transmit, and there is enough time before the next dwell boundary to send the next fragment and receive its acknowledgment.

The following rules shall also apply:

— When a STA has transmitted a frame other than an initial or intermediate fragment, that STA shall not transmit on the channel following the acknowledgment for that frame, without performing the backoff procedure.

— When an MSDU has been successfully delivered or all retransmission attempts have been exhausted, and the STA has a subsequent MSDU to transmit, then that STA shall perform a backoff procedure.

Copyright © 2007 IEEE. All rights reserved.

### 9.2.5.6 RTS/CTS usage with fragmentation

The following is a description of using RTS/CTS for a fragmented MSDU or MMPDU. The RTS/CTS frames define the duration of the following frame and acknowledgment. The Duration/ID field in the data and ACK frames specifies the total duration of the next fragment and acknowledgment. This is illustrated in Figure 9-9.

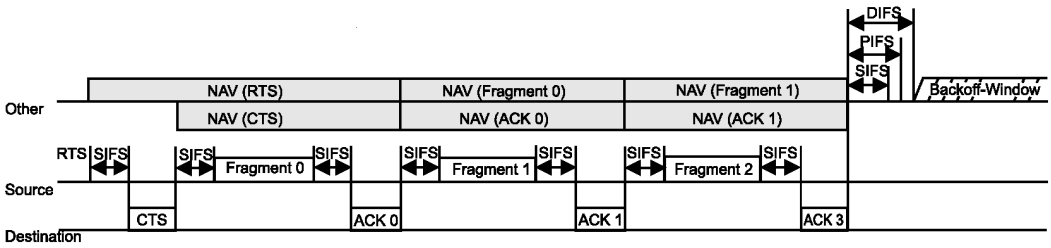



**Figure 9-9—RTS/CTS with fragmented MSDU**

Each frame contains information that defines the duration of the next transmission. The duration information from RTS frames shall be used to update the NAV to indicate busy until the end of ACK 0. The duration information from the CTS frame shall also be used to update the NAV to indicate busy until the end of ACK 0. Both Fragment 0 and ACK 0 shall contain duration information to update the NAV to indicate busy until the end of ACK 1. This shall be done by using the Duration/ID field in the Data and ACK frames. This shall continue until the last fragment, which shall have a duration of one ACK time plus one SIFS time, and its ACK, which shall have its Duration/ID field set to 0. Each fragment and ACK acts as a virtual RTS and CTS; therefore no further RTS/CTS frames need to be generated after the RTS/CTS that began the frame exchange sequence even though subsequent fragments may be larger than dot11RTSThreshold. At STAs using an FH PHY, when there is insufficient time before the next dwell boundary to transmit the subsequent fragment, the STA initiating the frame exchange sequence may set the Duration/ID field in the last data or management frame to be transmitted before the dwell boundary to the duration of one ACK time plus one SIFS time.

In the case where an acknowledgment is sent but not received by the source STA, STAs that heard the fragment, or ACK, will mark the channel busy for the next frame exchange due to the NAV having been updated from these frames. This is the worst-case situation, and it is shown in Figure 9-10. If an acknowledgment is not sent by the destination STA, STAs that can only hear the destination STA will not update their NAV and may attempt to access the channel when their NAV updated from the previously received frame reaches zero. All STAs that hear the source will be free to access the channel after their NAV updated from the transmitted fragment has expired.


Copyright © 2007 IEEE. All rights reserved.

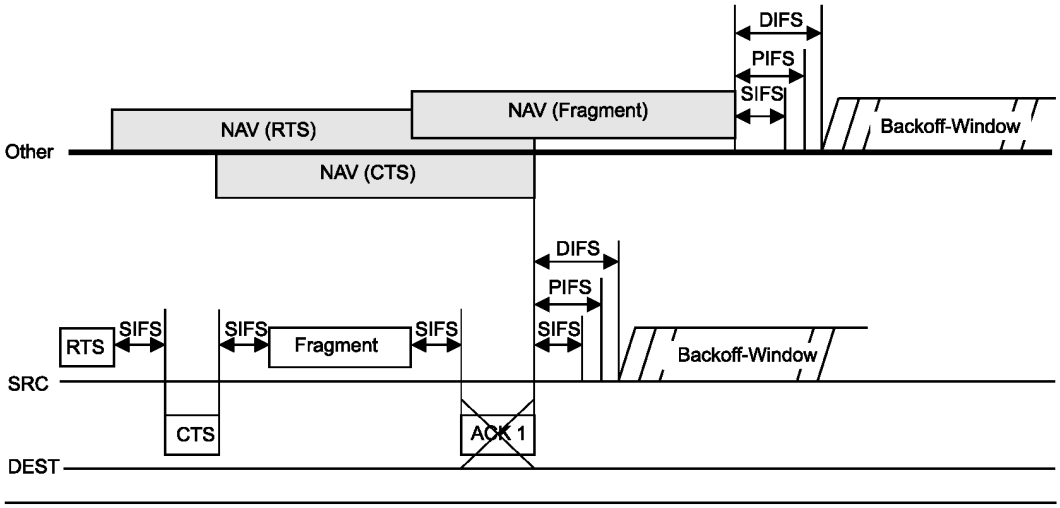



**Figure 9-10—RTS/CTS with transmitter priority and missed acknowledgment**

### 9.2.5.7 CTS procedure

A STA that is addressed by an RTS frame shall transmit a CTS frame after a SIFS period if the NAV at the STA receiving the RTS frame indicates that the medium is idle. If the NAV at the STA receiving the RTS indicates the medium is not idle, that STA shall not respond to the RTS frame. The RA field of the CTS frame shall be the value obtained from the TA field of the RTS frame to which this CTS frame is a response. The Duration field in the CTS frame shall be the duration field from the received RTS frame, adjusted by subtraction of aSIFSTime and the number of microseconds required to transmit the CTS frame at a data rate determined by the rules in 9.6.

After transmitting an RTS frame, the STA shall wait for a CTSTimeout interval, with a value of aSIFSTime + aSlotTime + aPHY-RX-START-Delay, starting at the PHY-TXEND.confirm. If a PHY-RXSTART.indication does not occur during the CTSTimeout interval, the STA shall conclude that the transmission of the RTS has failed, and this STA shall invoke its backoff procedure upon expiration of the CTSTimeout interval. If a PHY-RXSTART.indication does occur during the CTSTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication to determine whether the RTS transmission was successful. The recognition of a valid CTS frame sent by the recipient of the RTS frame, corresponding to this PHY-RXEND.indication, shall be interpreted as successful response, permitting the frame sequence to continue (see 9.12). The recognition of anything else, including any other valid frame, shall be interpreted as failure of the RTS transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication and may process the received frame.

### 9.2.6 Individually addressed MPDU transfer procedure

A STA shall use an RTS/CTS exchange for individually addressed frames only when the length of the MPDU is greater than the length threshold indicated by the dot11RTSThreshold attribute.

The dot11RTSThreshold attribute shall be a managed object within the MAC MIB, and its value may be set and retrieved by the MLME. The value 0 shall be used to indicate that all MPDUs shall be delivered with the use of RTS/CTS. Values of dot11RTSThreshold larger than the maximum MSDU length shall indicate that all MPDUs shall be delivered without RTS/CTS exchanges.

Copyright © 2007 IEEE. All rights reserved.

When an RTS/CTS exchange is used, the asynchronous data frame shall be transmitted starting one SIFS period after the end of the CTS frame. No regard shall be given to the busy or idle status of the medium when transmitting this data frame.

When an RTS/CTS exchange is not used, the asynchronous data frame shall be transmitted following the success of the basic access procedure. With or without the use of the RTS/CTS exchange procedure, the STA that is the destination of an asynchronous data frame shall follow the ACK procedure.

### 9.2.7 Broadcast and multicast MPDU transfer procedure

In the absence of a PCF, when broadcast or multicast MPDUs are transferred from a STA with the To DS field clear, only the basic access procedure shall be used. Regardless of the length of the frame, no RTS/CTS exchange shall be used. In addition, no ACK shall be transmitted by any of the recipients of the frame. Any broadcast or multicast MPDUs transferred from a STA with a To DS field set shall, in addition to conforming to the basic access procedure of CSMA/CA, obey the rules for RTS/CTS exchange and the ACK procedure because the MPDU is directed to the AP. The broadcast/multicast message shall be distributed into the BSS. The STA originating the message shall receive the message as a broadcast/multicast message. Therefore, all STAs shall filter out broadcast/multicast messages that contain their address as the source address. Broadcast and multicast MSDUs shall be propagated throughout the ESS.

There is no MAC-level recovery on broadcast or multicast frames, except for those frames sent with the To DS field set. As a result, the reliability of this traffic is reduced, relative to the reliability of individually addressed traffic, due to the increased probability of lost frames from interference, collisions, or time-varying channel properties.

### 9.2.8 ACK procedure

An ACK frame shall be generated as shown in the frame exchange sequences listed in 9.12.

Upon successful reception of a frame of a type that requires acknowledgment with the To DS field set, an AP shall generate an ACK frame. An ACK frame shall be transmitted by the destination STA that is not an AP, when it successfully receives a unicast frame of a type that requires acknowledgment, but not if it receives a broadcast or multicast frame of such type. After a successful reception of a frame requiring acknowledgment, transmission of the ACK frame shall commence after a SIFS period, without regard to the busy/idle state of the medium. (See Figure 9-11.)

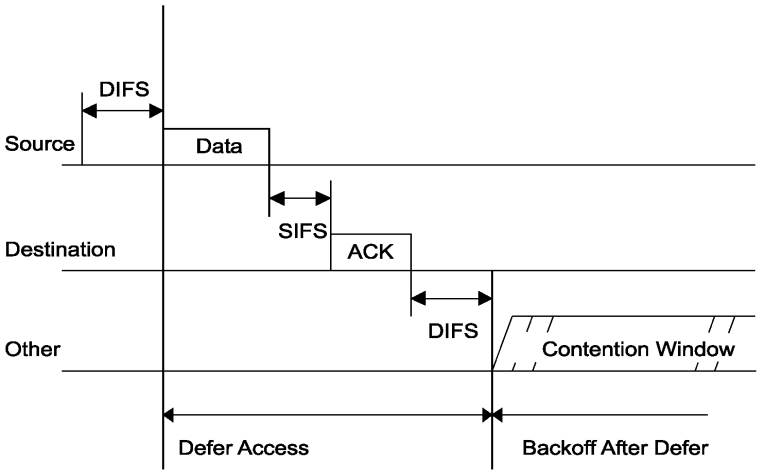



**Figure 9-11—Individually addressed data/ACK MPDU**


Copyright © 2007 IEEE. All rights reserved.

After transmitting an MPDU that requires an ACK frame as a response (see 9.12), the STA shall wait for an ACKTimeout interval, with a value of aSIFSTime + aSlotTime + aPHY-RX-START-Delay, starting at the PHY-TXEND.confirm. If a PHY-RXSTART.indication does not occur during the ACKTimeout interval, the STA concludes that the transmission of the MPDU has failed, and this STA shall invoke its backoff procedure upon expiration of the ACKTimeout interval. If a PHY-RXSTART.indication does occur during the ACKTimeout interval, the STA shall wait for the corresponding PHY-RXEND.indication to determine whether the MPDU transmission was successful. The recognition of a valid ACK frame sent by the recipient of the MPDU requiring acknowledgment, corresponding to this PHY-RXEND.indication, shall be interpreted as successful acknowledgment, permitting the frame sequence to continue, or to end without retries, as appropriate for the particular frame sequence in progress. The recognition of anything else, including any other valid frame, shall be interpreted as failure of the MPDU transmission. In this instance, the STA shall invoke its backoff procedure at the PHY-RXEND.indication and may process the received frame. The sole exception is that recognition of a valid data frame sent by the recipient of a PS-Poll frame shall also be accepted as successful acknowledgment of the PS-Poll frame.

### 9.2.9 Duplicate detection and recovery

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 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 of the Subtype field set to 0, is generated by the transmitting STA as an incrementing sequence of integers. In a QoS STA, 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 QoS STA shall also keep a cache of recently received <Address 2, TID, sequence-number, fragment-number> tuples for all STAs 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 QoS STA 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 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, fragment-number> tuple of an entry in the cache. A receiver QoS STA 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).

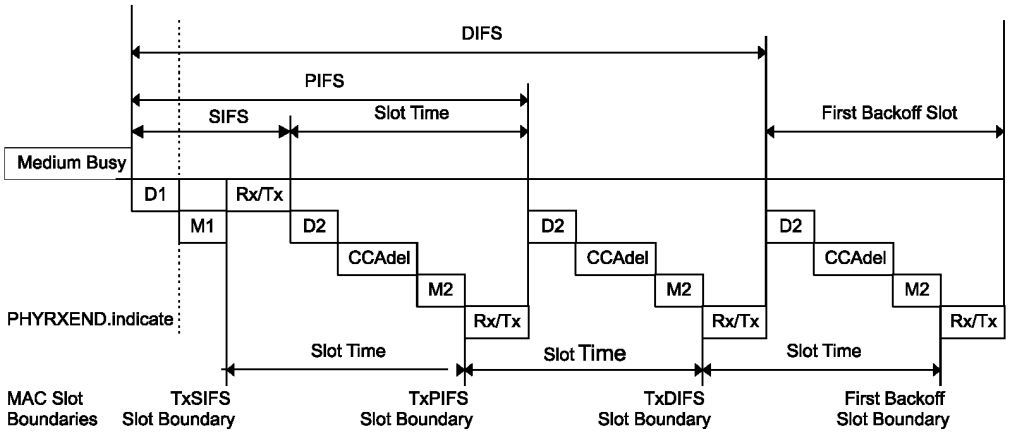
The receiver STA shall perform the ACK procedure on all successfully received frames requiring acknowledgment, even if the frame is discarded due to duplicate filtering.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 317 of 1232

### 9.2.10 DCF timing relations

The relationships between the IFS specifications are defined as time gaps on the medium. The associated attributes are provided by the specific PHY. (See Figure 9-12.)




D1 = aRxRFDelay + aRxPLCPDelay (referenced from the end of the last symbol of a frame on the medium)
D2 = D1 + Air Propagation Time
Rx/Tx = aRXTXTurnaroundTime (begins with a PHYTXSTART.request)
M1 = M2 = aMACProcessingDelay
CCAdel = aCCA Time - D1

**Figure 9-12—DCF timing relationships**

All timings that are referenced from the end of the transmission are referenced from the end of the last symbol of a frame on the medium. The beginning of transmission refers to the first symbol of the preamble of the next frame on the medium.

aSIFSTime and aSlotTime are fixed per PHY.

aSIFSTime is: aRxRFDelay + aRxPLCPDelay + aMACProcessingDelay + aRxTxTurnaroundTime.

aSlotTime is: aCCATime + aRxTxTurnaroundTime + aAirPropagationTime + aMACProcessingDelay.

The PIFS and DIFS are derived by the following equations, as illustrated in Figure 9-12.

PIFS = aSIFSTime + aSlotTime

DIFS = aSIFSTime + 2 × aSlotTime

The EIFS is derived from the SIFS and the DIFS and the length of time it takes to transmit an ACK Control frame at the lowest PHY mandatory rate by the following equation:

EIFS = aSIFSTime + DIFS + ACKTxTime

where

ACKTxTime is the time expressed in microseconds required to transmit an ACK frame, including preamble, PLCP header and any additional PHY dependent information, at the lowest PHY mandatory rate.


Copyright © 2007 IEEE. All rights reserved.

Figure 9-12 illustrates the relation between the SIFS, PIFS, and DIFS as they are measured on the medium and the different MAC slot boundaries TxSIFS, TxPIFS, and TxDIFS. These slot boundaries define when the transmitter shall be turned on by the MAC to meet the different IFS timings on the medium, after subsequent detection of the CCA result of the previous slot time.

The following equations define the MAC Slot Boundaries, using attributes provided by the PHY, which are such that they compensate for implementation timing variations. The starting reference of these slot boundaries is again the end of the last symbol of the previous frame on the medium.

TxSIFS = SIFS – aRxTxTurnaroundTime

TxPIFS = TxSIFS + aSlotTime

TxDIFS = TxSIFS + 2 × aSlotTime.

The tolerances are specified in the physical layer management entity (PLME) SAP interface specification (see 10.4) and shall only apply to the SIFS specification so that tolerances shall not accumulate.

### 9.2.11 NAV distribution

When a node needs to distribute NAV information, for instance, to reserve the medium for a transmission of a nonbasic rate frame (that may not be heard by other nodes in the BSS), the node may first transmit a CTS frame with the RA field equal to its own MAC address (CTS-to-self) and with a duration value that protects the pending transmission, plus possibly an ACK frame.

The CTS-to-self NAV distribution mechanism is lower in network overhead cost than is the RTS/CTS NAV distribution mechanism, but CTS-to-self is less robust against hidden nodes and collisions than RTS/CTS. STAs employing a NAV distribution mechanism should choose a mechanism such as CTS-to-self or RTS/CTS that is appropriate for the given network conditions. If errors occur when employing the CTS-to-self mechanism, STAs should switch to a more robust mechanism.

### 9.2.12 Determination of PLME aCWmin characteristics

In the case of the Clause 19 ERP, the aCWmin value is dependent on the requestor's characteristic rate set. The characteristic rate set is equal to the IBSS's supported rate set when the STA is operating as a member of an IBSS. It is equal to the AP's supported rate set when the STA is associated with an AP. At all other times, it is equal to the STA's mandatory rate set. The MAC variable aCWmin is set to aCWmin(0) if the characteristic rate set includes only rates in the set 1, 2, 5.5, 11; otherwise, aCWmin is set to aCWmin(1). If the returned value for aCWmin is a scalar, then the MAC always sets the variable aCWmin to the returned scalar value of aCWmin.

## 9.3 PCF

The PCF provides CF frame transfer. The PC shall reside in the AP. It is an option for an AP to be able to become the PC. All STAs inherently obey the medium access rules of the PCF, because these rules are based on the DCF, and all STAs set their NAV at the beginning of each CFP. The operating characteristics of the PCF are such that all STAs are able to operate properly in the presence of a BSS in which a PC is operating, and, if associated with a point-coordinated BSS, are able to receive all frames sent under PCF control. It is also an option for a STA to be able to respond to a CF-Poll received from a PC. A STA that is able to respond to CF-Polls is referred to as being CF-Pollable, and may request to be polled by an active PC. CF-Pollable STAs and the PC do not use RTS/CTS in the CFP. When polled by the PC, a CF-Pollable STA may transmit only one MPDU, which can be sent to the PC but may have any destination, and may "piggyback" the acknowledgment of a frame received from the PC using particular data frame subtypes for this transmission. If the data frame is

Copyright © 2007 IEEE. All rights reserved.

not in turn acknowledged, the CF-Pollable STA shall not retransmit the frame unless it is polled again by the PC, or it decides to retransmit during the CP. If the addressed recipient of a CF transmission is not CF-Pollable, that STA acknowledges the transmission using the DCF acknowledgment rules, and the PC retains control of the medium. A PC may use CF frame transfer solely for delivery of frames to STAs, and never to poll CF-Pollable STAs.

A PC may perform a backoff on retransmission of an unacknowledged frame during the CFP. A PC that is maintaining a polling list may retry the unacknowledged frame the next time the particular AID is at the top of the polling list.

A PC may retransmit an unacknowledged frame during the CFP after a PIFS time.

When more than one point-coordinated BSS is operating on the same PHY channel in overlapping space, the potential exists for collisions between PCF transfer activities by the independent PCs. The rules under which multiple, overlapping point-coordinated BSSs may coexist are presented in 9.3.3.2. As shown in Figure 9-1 (in 9.1), the PCF is built on top of the CSMA/CA-based DCF, by utilizing the access priority provisions provided by this scheme. An active PC shall be located at an AP, which restricts PCF operation to infrastructure networks. PCF is activated at a PC-capable AP by setting the CFPMaxDuration parameter in the CF Parameter Set of the MLMEStart.request to a nonzero value.

Data frames sent by, or in response to polling by, the PC during the CFP shall use the appropriate data subtypes based upon the following usage rules:

— Data+CF-Poll, Data+CF-Ack+CF-Poll, CF-Poll, and CF-Ack+CF-Poll shall only be sent by a PC.

— Data, Data+CF-Ack, Null Function, and CF-Ack may be sent by a PC or by any CF-Pollable STA.

STAs receiving Data type frames shall only consider the frame body as the basis of a possible indication to LLC, if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll. CF-Pollable STAs shall interpret all subtype bits of received Data type frames that contain the BSSID of the current BSS for CF purposes, but shall inspect the frame body only if the frame is of subtype Data, Data+CF-Ack, Data+CF-Poll, or Data+CF-Ack+CF-Poll.

### 9.3.1 CFP structure and timing

The PCF controls frame transfers during a CFP. The CFP shall alternate with a CP, when the DCF controls frame transfers, as shown in Figure 9-13. Each CFP shall begin with a Beacon frame that contains a DTIM element (hereafter referred to as a DTIM). The CFPs shall occur at a defined repetition rate, which shall be synchronized with the beacon interval as specified in the following paragraphs.

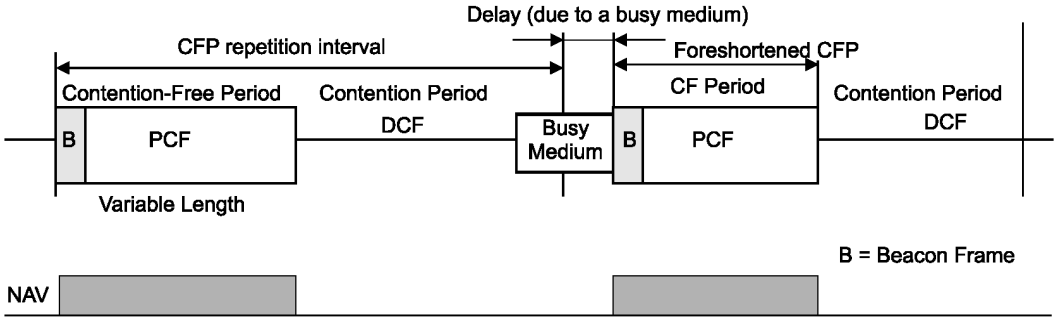



**Figure 9-13—CFP/CP alternation**


Copyright © 2007 IEEE. All rights reserved.

The PC generates CFPs at the CFP repetition interval (CFPPeriod), which is defined as a number of DTIM intervals. The PC shall determine the CFPPeriod (depicted as a repetition interval in the illustrations in Figure 9-13 and Figure 9-14) to use from the CFPPeriod parameter in the CF Parameter Set. This value, in units of DTIM intervals, shall be communicated to other STAs in the BSS in the CFPPeriod field of the CF Parameter Set element of Beacon frames. The CF Parameter Set element shall only be present in Beacon and Probe Response frames transmitted by STAs containing an active PC.

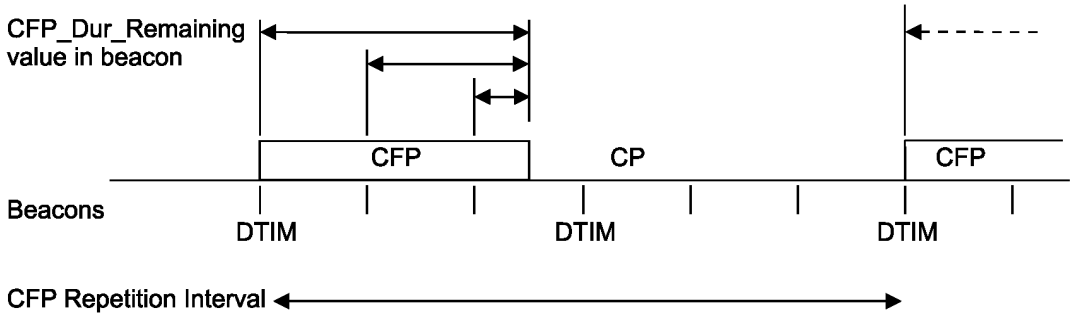



**Figure 9-14—Beacon frames and CFPs**

The length of the CFP is controlled by the PC, with maximum duration specified by the value of the CFPMaxDuration Parameter in the CF Parameter Set at the PC. Neither the maximum duration nor the actual duration (signaled by transmission of a Control frame of subtype CF-End or CF-End+ACK by the PC) is constrained to be a multiple of the beacon interval. If the CFP duration is greater than the beacon interval, the PC shall transmit Beacon frames at the appropriate times during the CFP (subject to delay due to traffic at the nominal times, as with all Beacon frames). The CF Parameter Set element in all Beacon frames at the start of, or within, a CFP shall contain a nonzero value in the CFPDurRemaining field. This value, in units of TU, shall specify the maximum time from the most recent TBTT to the end of this CFP. The value of the CFPDurRemaining field shall be zero in Beacon frames sent during the CP. An example of these relationships is illustrated in Figure 9-14, which shows a case where the CFPPeriod is two DTIM intervals, the DTIM interval is three beacon intervals, and the aCFPMaxDuration value is approximately 2.5 beacon intervals.

The PC may terminate any CFP at or before the aCFPMaxDuration, based on available traffic and size of the polling list. Because the transmission of any Beacon frame may be delayed due to a medium busy condition at the TBTT, a CFP may be foreshortened by the amount of the delay. In the case of a busy medium due to DCF traffic, the Beacon frame shall be delayed for the time required to complete the current DCF frame exchange. In cases where the Beacon frame transmission is delayed, the CFPDurRemaining value in the Beacon frame at the beginning of the CFP shall specify a time that causes the CFP to end no later than TBTT plus the value of aCFPMaxDuration. This is illustrated in Figure 9-15.

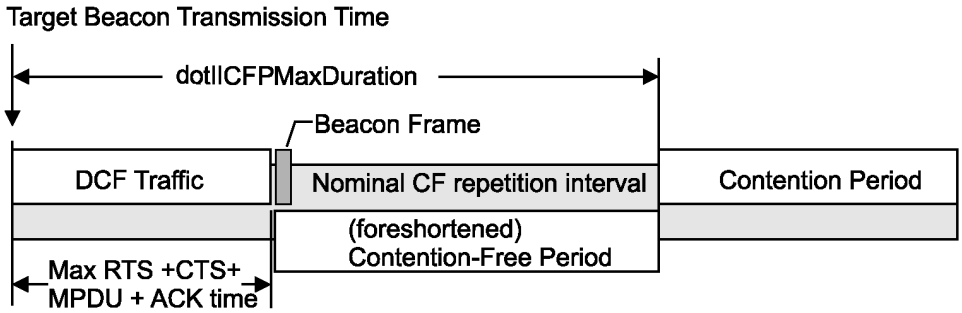



**Figure 9-15—Example of delayed beacon and foreshortened CFP**

Copyright © 2007 IEEE. All rights reserved.

### 9.3.2 PCF access procedure

The CF transfer protocol is based on a polling scheme controlled by a PC operating at the AP of the BSS. The PC gains control of the medium at the beginning of the CFP and attempts to maintain control for the entire CFP by waiting a shorter time between transmissions than the STAs using the DCF access procedure. All STAs that receive Beacon frames containing a CF Parameter Set information element, including STAs not associated with the BSS, set their NAVs to the CFPMaxDuration value at the nominal start time of each CFP. This prevents most contention by preventing nonpolled transmissions by STAs regardless of whether they are CF-Pollable. Acknowledgment of frames sent during the CFP may be accomplished using Data+CF-ACK, CF-ACK, Data+CF-ACK+CF-Poll (only on frames transmitted by the PC), or CF-ACK+CF-Poll (only on frames transmitted by the PC) frames in cases where a Data (or Null) frame immediately follows the frame being acknowledged, thereby avoiding the overhead of separate ACK Control frames. Non-CF-Pollable or unpolled CF-Pollable STAs acknowledge frames during the CFP using the DCF ACK procedure.

#### 9.3.2.1 Fundamental access

At the nominal beginning of each CFP, the PC shall sense the medium. When the medium is determined to be idle for one PIFS period, the PC shall transmit a Beacon frame containing the CF Parameter Set element and a DTIM element.

After the initial Beacon frame, the PC shall wait for one SIFS period, and then transmit one of the following: a data frame, a CF-Poll frame, a Data+CF-Poll frame, a management frame, or a CF-End frame. If the CFP is null, i.e., no traffic is buffered and no polls exist to send at the PC, a CF-End frame shall be transmitted immediately after the initial Beacon frame. If there are buffered multicast or broadcast frames, the PC shall transmit these prior to any unicast frames.

STAs receiving individually addressed, error-free frames from the PC are expected to respond after a SIFS period, in accordance with the transfer procedures defined in 9.3.3. If the recipient STA is not CF-Pollable, the response to receipt of an error-free data frame shall always be an ACK frame.

#### 9.3.2.2 NAV operation during the CFP

The mechanism for handling the NAV during the CFP is designed to facilitate the operation of overlapping CFP coordinated infrastructure BSSs. The mechanism by which infrastructure BSSs coordinate their CFPs is beyond the scope of this standard.

Each STA, except the STA with the PC, shall preset its NAV to the CFPMaxDuration value (obtained from the CF Parameter Set element in Beacon frames from this PC) at each TBTT (see Clause 11) at which a CFP is scheduled to start (based on the CFPCount field in the CF Parameter Set element of the Beacon frames from this PC). Each non-PC STA shall update its NAV using the CFPDurRemaining value in the CF Parameter Set element of any error-free Beacon frame that the STA receives. This includes CFPDurRemaining values in CF Parameter Set elements from Beacon frames received from other (overlapping) BSSs.

These actions prevent STAs from taking control of the medium during the CFP, which is especially important in cases where the CFP spans multiple medium-occupancy intervals, such as dwell periods of an FH PHY. This setting of the NAV also reduces the risk of hidden STAs determining the medium to be idle for a DIFS period during the CFP and possibly corrupting a transmission in progress.

A STA joining a BSS operating with a PC shall use the information in the CFPDurRemaining element of the CF parameter set of any received Beacon or Probe Response frames to update its NAV prior to initiating any transmissions.

The PC shall transmit a CF-End or CF-End+ACK frame at the end of each CFP. A STA that receives either of these frames, from any BSS, shall reset its NAV.


Copyright © 2007 IEEE. All rights reserved.

### 9.3.3 PCF transfer procedure

Frame transfers under the PCF may consist of frames alternately sent from the AP/PC and sent to the AP/PC. During the CFP, the ordering of these transmissions, and the STA allowed to transmit frames to the PC at any given point in time, shall be controlled by the PC. Figure 9-16 depicts frame transfer during a typical CFP. The rules under which this frame transfer takes place are detailed in 9.3.3.1 through 9.3.3.4.

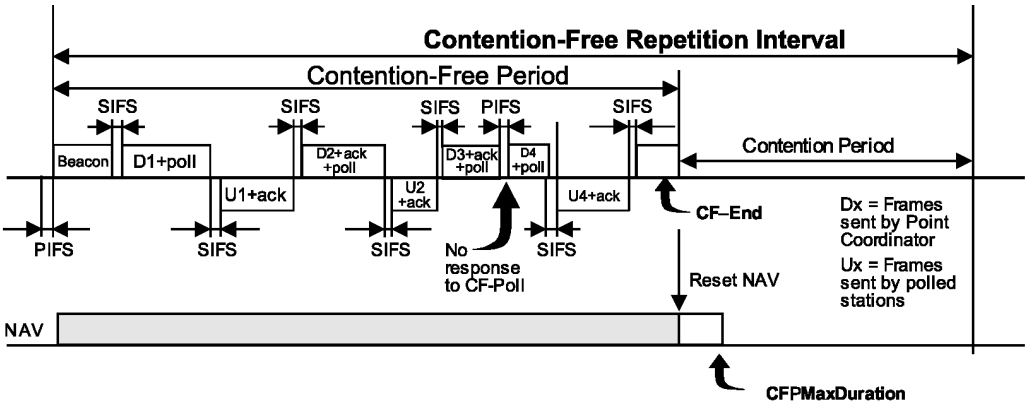



**Figure 9-16—Example of PCF frame transfer**

In a STA having an FH PHY, control of the channel is lost at a dwell time boundary. It is required that the current MPDU transmission and the accompanying acknowledgment of the MPDU be transmitted before the dwell time boundary. After having been polled by the PC, if there is not enough time remaining in the dwell to allow transmission of the MPDU plus the acknowledgment, the STA shall defer the transmission of the MPDU and shall transmit a Null frame or CF-ACK frame. The short retry counter and long retry counter for the MSDU shall not be affected.

MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by security mechanisms, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. In a STA having an FH PHY, the PC shall not transmit a frame with any data subtype that includes CF-Poll to a STA if there is insufficient time remaining before the dwell boundary for the STA to respond with a Null frame or CF-ACK frame.

#### 9.3.3.1 PCF transfers when the PC STA is transmitter or recipient

The PC shall transmit frames between the Beacon that starts the CFP and the CF-End using the SIFS except in cases where a transmission by another STA is expected by the PC and a SIFS period elapses without the receipt of the expected transmission. In such cases the PC may send its next pending transmission as soon as one PIFS after the end of its last transmission. This permits the PC to retain control of the medium in the presence of an overlapping BSS. The PC may transmit any of the following frame types to CF-Pollable STAs:

— Data, used to send data from the PC when the addressed recipient is not being polled and there is no previous frame to acknowledge;

— Data+CF-ACK, used to send data from the PC when the addressed recipient is not being polled or is not CF-Pollable or the DA is a group address and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;

— Data+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;

Copyright © 2007 IEEE. All rights reserved.

— Data+CF-ACK+CF-Poll, used to send data from the PC when the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame received from a CF-Pollable STA a SIFS period before starting this transmission;

— CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and there is no previous frame to acknowledge;

— CF-ACK+CF-Poll, used when the PC is not sending data to the addressed recipient but the addressed recipient is the next STA to be permitted to transmit during this CFP and the PC needs to acknowledge the receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission;

— CF-ACK, used when the PC is not sending data to, or polling, the addressed recipient, but the PC needs to acknowledge receipt of a frame from a CF-Pollable STA a SIFS period before starting this transmission (useful when the next transmission by the PC is a management frame, such as a Beacon frame); or

— Any management frame that is appropriate for the AP to send under the rules for that frame type.

The PC may transmit data or management frames to non-CF-Pollable, non-PS STAs during the CFP. These STAs shall acknowledge receipt with ACK frames after a SIFS, as with the DCF. The PC may also transmit broadcast or multicast frames during the CFP. Because the Beacon frame that initiates the CFP contains a DTIM element, if there are associated STAs using PS mode, the broadcasts and multicasts buffered shall be sent immediately after any Beacon frame containing a TIM element with a DTIM count field with a value of 0.

A CF-Pollable STA that receives an individually addressed data frame of any subtype that includes CF-Poll may transmit one data frame a SIFS period after receiving the CF-Poll. CF-Pollable STAs shall ignore, but not reset, their NAV when performing transmissions in response to a CF-Poll.

Non-CF-Pollable STAs that receive an individually addressed frame during the CFP shall transmit an ACK, but shall not reset their NAV.

For frames that require MAC-level acknowledgment, CF-Pollable STAs that received a CF-Poll (of any type) may perform this acknowledgment using the Data+CF-ACK subtype in the response to the CF-Poll. For example, the U1 frame in Figure 9-16 contains the acknowledgment to the preceding D1 frame. The D2 frame contains the acknowledgment to the preceding U1 frame. The PC may use the CF-ACK subtypes to acknowledge a received frame even if the data frame sent with the CF-ACK subtype is addressed to a different STA than the one being acknowledged. CF-Pollable STAs that are expecting an acknowledgment shall interpret the subtype of the frame (if any) sent by the PC a SIFS period after that STA's transmission to the PC. If a frame that requires MAC-level acknowledgment is received by a non-CF-Pollable STA, that STA shall not interpret the CF-Poll indication (if any), and shall acknowledge the frame by sending an ACK Control frame after a SIFS period.

The lengths of the frames may be variable, only bounded by the frame and/or fragment length limitations that apply for the BSS. If a CF-Pollable STA does not respond to a CF-Poll (of any type) within the SIFS period following a transmission from the PC, or a non-CF-Pollable STA does not return the ACK frame within a SIFS period following a transmission from the PC that requires acknowledgment, then the PC shall resume control and may transmit its next frame after a PIFS period from the end of the PC's last transmission.

A CF-Pollable STA shall always respond to a frame with any data subtype that includes CF-Poll directed to its MAC address and received without error. If the STA has no frame to send when polled, the response shall be a Null frame. If the STA has no frame to send when polled, but an acknowledgment is required for the frame that conveyed the CF-Poll, the response shall be a CF-ACK (no data) frame. The null response is required to permit a "no-traffic" situation to be distinguished from a collision between overlapping PCs.


Copyright © 2007 IEEE. All rights reserved.

The CFP shall end when the CFPDurRemaining time has elapsed since the Beacon frame originating the CFP or when the PC has no further frames to transmit nor STAs to poll. In either case, the end of the CFP shall be signaled by the transmission of a CF-End by the PC. If there is a received frame that requires acknowledgment at the time the CF-End is to be transmitted, the PC shall transmit a CF-End+ACK frame instead. All STAs of the BSS receiving a CF-End or CF-End+ACK shall reset their NAVs so they may attempt to transmit during the CP.

#### 9.3.3.2 Operation with overlapping point-coordinated BSSs

Because the PCF operates without the CSMA/CA CW randomization and backoff of the DCF, there is a risk of repeated collisions if multiple, overlapping, point-coordinated BSSs are operating on the same PHY channel, and their CFP Rates and beacon intervals are approximately equal. To minimize the risk of significant frame loss due to CF collisions, the PC shall use a DIFS plus a random backoff delay (with CW in the range of 1 to aCWmin) to start a CFP when the initial Beacon frame is delayed because of deferral due to a busy medium. The PC may optionally use this backoff during the CFP prior to retransmitting an unacknowledged, individually addressed data or management frame.

To further reduce the susceptibility to inter-PC collisions, the PC shall require that the medium be determined as being idle for a DIFS period plus a random (over a range of 1 to aCWmin) number of slot times once every aMediumOccupancyLimit TU during the CFP. This results in loss of control of the medium to overlapping BSS or hidden STA traffic, because the STAs in this BSS are prevented from transmitting by their NAV setting to CFPMaxDuration or CFPDurRemaining. For operation of the PCF in conjunction with an FH PHY, aMediumOccupancyLimit shall be set equal to the dwell time. For operation in conjunction with other PHY types, aMediumOccupancyLimit may be set equal to CFPMaxDuration, unless extra protection against PCF collisions is desired. The aMediumOccupancyLimit is also useful for compliance in regulatory domains that impose limits on continuous transmission time by a single STA as part of a spectrum etiquette.

#### 9.3.3.3 CFPMaxDuration limit

The value of CFPMaxDuration shall be limited to allow coexistence between contention and CF traffic.

The minimum value for CFPMaxDuration is two times MaxMPDUTime plus the time required to send the initial Beacon frame and the CF-End frame of the CFP. This may allow sufficient time for the AP to send one data frame to a STA, while polling that STA, and for the polled STA to respond with one data frame.

The maximum value for CFPMaxDuration is the duration of (BeaconPeriod × DTIMPeriod × CFPPeriod) minus [MaxMPDUTime plus (2 × aSIFSTime) plus (2 × aSlotTime) plus (8 × ACKSize)], expressed in microseconds. MaxMPDUTime is the time to transmit the maximum-sized MAC frame, expanded by security mechanisms, plus the time to transmit the PHY preamble, header, trailer, and expansion bits, if any. This allows sufficient time to send at least one data frame during the CP.

#### 9.3.3.4 CF usage rules

A PC may send broadcast or multicast frames, and individually addressed data or management frames to any active STA, as well as to CF-Pollable PS STAs. During the CFP, CF-Pollable STAs shall acknowledge after a SIFS period, the receipt of each Data+CF-Poll frame or Data+CF-ACK+CF-Poll frame using Data+CF-Ack or CF-Ack (no data) frames, the receipt of each CF_Poll (no data) using Data or Null (no data), and the receipt of all other data and management frames using ACK Control frames. Non-CF-Pollable STAs shall acknowledge receipt of data and management frames using ACK Control frames sent after a SIFS period. This non-CF-Pollable operation is the same as that already employed by such STAs for DCF operation.

When polled by the PCF (Data+CF-Poll, Data+CF-ACK+CF-Poll, CF-Poll, or CF-ACK+CF-Poll) a CF-Pollable STA may send one data frame to any destination. Such a frame directed to or through the PC STA shall be acknowledged by the PC, using the CF-ACK indication (Data+CF-ACK, Data+CF-ACK+CF-Poll,

Copyright © 2007 IEEE. All rights reserved.

CF-ACK, CF-ACK+CF-Poll, or CF-End+ACK) sent after a SIFS. Such a frame directed to a non-CF-Pollable STA shall be acknowledged using an ACK Control frame sent after a SIFS period. A polled CF-Pollable STA with neither a data frame nor an acknowledgment to send shall respond by transmitting a Null frame after a SIFS period. A polled CF-Pollable STA with insufficient time before the end of the CFP or current medium occupancy limit, to send its queued MPDU and receive an acknowledgment, shall respond by transmitting a Null frame, or a CF-ACK frame if polled using Data+CF-Poll or Data+CF-ACK+CF-Poll, after a SIFS period. The CF-Pollable STA may set the More Data bit in its response to permit the PC to distinguish between an empty STA queue and a response due to insufficient time to transfer an MPDU.

The PC shall not issue frames with a subtype that includes CF-Polls if insufficient time remains in the current CFP to permit the polled STA to transmit a data frame containing a minimum length MPDU.

### 9.3.4 CF polling list

If the PC supports use of the CFP for inbound frame transfer as well as for frame delivery, the PC shall maintain a "polling list" for use in selecting STAs that are eligible to receive CF-Polls during CFPs. The polling list functional characteristics are defined below. If the PC supports the use of the CFP solely for frame delivery, the PC does not require a polling list, and shall never generate data frames with a subtype that includes CF-Poll. The form of CF support provided by the PC is identified in the Capability Information field of Beacon, Association Response, Reassociation Response, and Probe Response management frames, which are sent from APs. Any such frames sent by STAs, as in noninfrastructure networks, shall always have these bits set to 0.

The polling list is used to force the polling of CF-Pollable STAs, whether or not the PC has pending traffic to transmit to those STAs. The polling list may be used to control the use of Data+CF-Poll and Data+CF-ACK+CF-Poll types for transmission of data frames being sent to CF-Pollable STAs by the PC. The polling list is a *logical* construct, which is not exposed outside of the PC. A minimum set of polling list maintenance techniques are required to ensure interoperability of arbitrary CF-Pollable STAs in BSSs controlled by arbitrary APs with active PCs. APs may also implement additional polling list maintenance techniques that are outside the scope of this standard.

#### 9.3.4.1 Polling list processing

The PC shall send a CF-Poll to at least one STA during each CFP when there are entries in the polling list. During each CFP, the PC shall issue polls to a subset of the STAs on the polling list in order by ascending AID value.

While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC may generate one or more CF-Polls to *any* STAs on the polling list. While time remains in the CFP, all CF frames have been delivered, and all STAs on the polling list have been polled, the PC *may* send data or management frames to *any* STAs.

In order to gain maximum efficiency from the CFP, and the ability to piggyback acknowledgments on successor data frames in the opposite direction, the PC should generally use Data+CF-Poll and Data+CF-ACK+CF-Poll types for each data frame transmitted while sufficient time for the potential response to the CF-Poll remains in the CFP.

#### 9.3.4.2 Polling list update procedure

A STA indicates its CF-Pollability using the CF-Pollable subfield of the Capability Information field of Association Request and Reassociation Request frames. If a STA desires to change the PC's record of CF-Pollability, that STA shall perform a reassociation. During association, a CF-Pollable STA may request to be placed on the polling list, or to never be polled, by appropriate use of bits in the Capability Information field of the Associate Request or Reassociate Request frame, as shown in Table 7-21 (see 7.3.1.4).


Copyright © 2007 IEEE. All rights reserved.

CF-Pollable STAs that are not on the polling list, but did not request never to be polled during their most recent association, may be dynamically placed on the polling list by the PC to handle bursts of frame transfer activity by that STA.

## 9.4 Fragmentation

The MAC may fragment and reassemble individually addressed MSDUs or MMPDUs. The fragmentation and defragmentation mechanisms allow for fragment retransmission.

The length of each fragment shall be an equal number of octets for all fragments except the last, which may be smaller. The length of each fragment shall always be an even number of octets, except for the last fragment of an MSDU or MMPDU, which may be either an even or an odd number of octets. The length of a fragment shall never be larger than dot11FragmentationThreshold unless WEP is invoked for the MPDU. If WEP is active for the MPDU, then the MPDU shall be expanded by IV and ICV (see 8.2.1); this may result in a fragment larger than dot11FragmentationThreshold.

A fragment is an MPDU, the payload of which carries all or a portion of an MSDU or MMPDU. When data are to be transmitted, the number of octets in the fragment (before processing by the security mechanism) shall be determined by dot11FragmentationThreshold and the number of octets in the MPDU that have yet to be assigned to a fragment at the instant the fragment is constructed for the first time. Once a fragment is transmitted for the first time, its frame body content and length shall be fixed until it is successfully delivered to the immediate receiving STA. A STA shall be capable of receiving fragments of arbitrary length.

If a fragment requires retransmission, its frame body content and length shall remain fixed for the lifetime of the MSDU or MMPDU at that STA. After a fragment is transmitted once, contents and length of that fragment are not allowed to fluctuate to accommodate the dwell time boundaries. Each fragment shall contain a Sequence Control field, which is comprised of a sequence number and fragment number. When a STA is transmitting an MSDU or MMPDU, the sequence number shall remain the same for all fragments of that MSDU or MMPDU. The fragments shall be sent in order of lowest fragment number to highest fragment number, where the fragment number value starts at zero, and increases by one for each successive fragment. The Frame Control field also contains a bit, the More Fragments bit, that is equal to zero to indicate the last (or only) fragment of the MSDU or MMPDU.

The source STA shall maintain a transmit MSDU timer for each MSDU being transmitted. The attribute dot11MaxTransmitMSDULifetime specifies the maximum amount of time allowed to transmit an MSDU. The timer starts on the initial attempt to transmit the first fragment of the MSDU. If the timer exceeds dot11MaxTransmitMSDULifetime, then all remaining fragments are discarded by the source STA and no attempt is made to complete transmission of the MSDU.

## 9.5 Defragmentation

Each fragment contains information to allow the complete MSDU or MMPDU to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the destination STA to reassemble the MSDU or MMPDU:

- Frame type
- Address of the sender, obtained from the Address2 field
- Destination address
- *Sequence Control field:* This field allows the destination STA to check that all incoming fragments belong to the same MSDU or MMPDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of an MSDU or MMPDU, while the fragment number within the Sequence Control field increments for each fragment.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 327 of 1232

— *More Fragments indicator:* Indicates to the destination STA that this is not the last fragment of the MSDU or MMPDU. Only the last or sole fragment of the MSDU or MMPDU shall have this bit set to 0. All other fragments of the MSDU or MMPDU shall have this bit set to 1.

The destination STA shall reconstruct the MSDU or MMPDU by combining the fragments in order of fragment number subfield of the Sequence Control field. If WEP has been applied to the fragment, it shall be decrypted before the fragment is used for defragmentation of the MSDU or MMPDU. If the fragment with the More Fragments bit set to 0 has not yet been received, then the destination STA knows that the MSDU or MMPDU is not yet complete. As soon as the STA receives the fragment with the More Fragments bit set to 0, the STA knows that no more fragments may be received for the MSDU or MMPDU.

All STAs shall support the concurrent reception of fragments of at least three MSDUs or MMPDUs. Note that a STA receiving more than three fragmented MSDUs or MMPDUs concurrently may experience a significant increase in the number of frames discarded.

The destination STA shall maintain a Receive Timer for each MSDU or MMPDU being received, for a minimum of three MSDUs or MMPDUs. The STA may implement additional timers to be able to receive additional concurrent MSDUs or MMPDUs. The receiving STA shall discard all fragments that are part of an MSDU or MMPDU for which a timer is not maintained. There is also an attribute, aMaxReceiveLifetime, that specifies the maximum amount of time allowed to receive an MSDU. The receive MSDU or MMPDU timer starts on the reception of the first fragment of the MSDU or MMPDU. If the receive MSDU timer exceeds aMaxReceiveLifetime, then all received fragments of this MSDU or MMPDU are discarded by the destination STA. If additional fragments of an individually addressed MSDU or MMPDU are received after its aMaxReceiveLifetime is exceeded, those fragments shall be acknowledged and discarded.

To properly reassemble MPDUs into an MSDU or MMPDU, a destination STA shall discard any duplicated fragments received. A STA shall discard duplicate fragments as described in 9.2.9. However, an acknowledgment shall be sent in response to a duplicate fragment of an individually addressed MSDU.

## 9.6 Multirate support

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 to be followed by all STAs.

Control frames that initiate a frame exchange shall be transmitted at one of the rates in the BSS-BasicRateSet parameter 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

In 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 9.13. 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 BSSBasicRateSet parameter, 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 BSSBasicRateSet parameter 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 data, BlockAckReq, and BlockAck frames


Copyright © 2007 IEEE. All rights reserved.

and/or management MPDUs with unicast in the Address 1 field shall be sent using any data rate subject to the following constraints. No STA shall transmit a unicast frame at a rate that is not supported by the receiver STA, as reported in any Supported Rates and Extended Supported Rates element in the management frames transmitted by that STA. 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 and modulation class 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. In the case where the supported rate set of the receiving STA is not known, the transmitting STA shall transmit at a rate contained in the BSSBasicRateSet parameter or a rate at which the transmitting STA has received a frame from the receiving STA.

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), 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 9.12) and that is of the same modulation class (see 9.6.1) as the received frame. If no rate contained in the BSSBasicRateSet parameter 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 class 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 origi-nally 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 QoS STAs, the Duration/ID field may cover multiple frames and may involve using the PLME-TXTIME.request primitive several times.

### 9.6.1 Modulation classes

In order to determine the rules for response frames given in 9.6, the following modulation classes are defined in Table 9-2. Each row defines a modulation class. Modulations described within the same row have the same modulation class, while modulations described in different rows have different modulation classes.

**Table 9-2—Modulation classes**

| Modulation class | Description of modulation |
|---|---|
| 1 | Infrared (IR) PHY (Clause 16) |
| 2 | Frequency-hopping spread spectrum (FHSS) PHY (Clause 14) |
| 3 | DSSS PHY (Clause 15) and HR/DSSS PHY (Clause 18) |
| 4 | ERP-PBCC PHY (19.6) |

Copyright © 2007 IEEE. All rights reserved.

**Table 9-2—Modulation classes *(continued)***

| Modulation class | Description of modulation |
|---|---|
| 5 | DSSS-OFDM PHY (19.7) |
| 6 | ERP-OFDM PHY (19.5) |
| 7 | OFDM PHY (Clause 17) |

## 9.7 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.

A non-QoS STA 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 QoS STA 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 individual or group-addressed). This is because a group-addressed MSDU is implicitly addressed to a collection of peer STAs that could include any individual RA.

It is recommended that the STA select a value of dot11MaxTransmitMSDULifetime 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 QoS STA, 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 QoS STA, 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.

## 9.8 Operation across regulatory domains

The PHY of a WLAN is subject to regulations that can vary significantly from one regulatory domain to another. This clause provides the framework for operation across regulatory domains and describes the mechanism that supports cross-domain mobility and operation in multiple regulatory domains. When this mechanisms is active, the dot11MultiDomainCapabilityEnabled attribute shall be true.

NOTE—This clause does not eliminate the need to obtain type acceptance, regulatory approval, equipment authorization, or equipment certification in each of the regulatory domains in which the equipment will operate. The mechanisms described in this clause provide the information to the STA to identify the regulatory domain in which it is


Copyright © 2007 IEEE. All rights reserved.

located and to cease operation while in those domains for which it does not have type approval. It is incumbent upon the implementer to provide proof of compliance to the requirements of individual regulatory agencies.

The method for configuring individual STAs is outside the scope of this standard. A STA must be properly configured for operation in a particular regulatory domain prior to beginning normal operation. Particular care must be taken when operating in an IBSS configuration.

### 9.8.1 Operation upon entering a regulatory domain

A STA that is enabled for operation across regulatory domains shall default to passive scanning when it has lost connectivity with its ESS. Passive scanning is performed using only the receive capabilities of the STA and is, thus, compatible with regulatory requirements. The timeout for determining the loss of connectivity is system dependent and beyond the scope of this standard.

When a STA enters a regulatory domain, it shall passively scan to learn at least one valid channel, i.e., a channel upon which it detects IEEE 802.11 frames. The Beacon frame contains information on the country code, the maximum allowable transmit power, and the channels to be used for the regulatory domain. Optionally, the Beacon frame may also include, on a periodic basis, the regulatory information that would be returned in a Probe Response frame. Once the STA has acquired the information so that it is able to meet the transmit requirements of the regulatory domain, it shall transmit a Probe Request to an AP to gain the additional regulatory domain information contained in the Probe Response frame, unless the information was previously received in a Beacon frame. The STA then has sufficient information available to configure its PHY for operation in the regulatory domain.

### 9.8.2 Support for FH PHYs

#### 9.8.2.1 Determination of hopping patterns

When operating in a regulatory domain that does not have a method for determining a hopping pattern described in 14.6.8 or a hopping table in Annex B, hopping patterns shall be determined by an AP using HCCs or EHCCs. The HCC hopping sequences are derived from a simple formula that uses field operations on a group. For full details of the HCC placement operator function, please see the references. The placement operator function shall be as shown in Equation (9-4).

$$y_{HCC}(k;a) = \frac{a}{k} \bmod N \qquad \text{for } k,\ a \in J'_N \tag{9-4}$$

where

$N$ is the prime radix
$a$ is the family index
$k$ is in the group $J'_N$
$J'_N$ is the group remaining when the element containing the value zero is removed from the field $J_N$

Therefore, $k$ does not take the value 0. The value $1/k$ is the multiplicative inverse of $k$ on the field $J_N$. The multiplicative inverse of $k$ on the group $J_N$ is the integer value $w$, such that $(k \times w) \bmod N = 1$. The values computed for $y_{HCC}$ are the channel numbers, $a$ corresponds to the hopping pattern number, and $k$ corresponds to the index into the hopping pattern. A code family is the set of $N$–1 hopping patterns generated for the prime radix $N$. There is no value equivalent to the hopping set of Clause 14. Each hopping pattern comprises $N$–1 channels.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 331 of 1232

As an example, consider a code family that supports 10 channels. The prime radix for such a family is 11. The code family generated by the HCC algorithm is shown below in Table 9-3.

**Table 9-3—HCC family – *N* = 11; Family indices (SEQ) 1 through 10**

| INDEX (k) ----> | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| SEQ | | | | | | | | | | |
| 1 | 1 | 6 | 4 | 3 | 9 | 2 | 8 | 7 | 5 | 10 |
| 2 | 2 | 1 | 8 | 6 | 7 | 4 | 5 | 3 | 10 | 9 |
| 3 | 3 | 7 | 1 | 9 | 5 | 6 | 2 | 10 | 4 | 8 |
| 4 | 4 | 2 | 5 | 1 | 3 | 8 | 10 | 6 | 9 | 7 |
| 5 | 5 | 8 | 9 | 4 | 1 | 10 | 7 | 2 | 3 | 6 |
| 6 | 6 | 3 | 2 | 7 | 10 | 1 | 4 | 9 | 8 | 5 |
| 7 | 7 | 9 | 6 | 10 | 8 | 3 | 1 | 5 | 2 | 4 |
| 8 | 8 | 4 | 10 | 2 | 6 | 5 | 9 | 1 | 7 | 3 |
| 9 | 9 | 10 | 3 | 5 | 4 | 7 | 6 | 8 | 1 | 2 |
| 10 | 10 | 5 | 7 | 8 | 2 | 9 | 3 | 4 | 6 | 1 |

The HCC method to calculate hopping sequences only generates sequences of a length that is one less than the prime radix. The EHCC algorithm extends the original HCC algorithm to support a larger number of possible hopping sequence lengths. The EHCC algorithm works through a process known as “deletion of the diagonals.” This creates hopping patterns with *N*–2 and *N*–3 channels.

Using the same example in Table 9-3, a family of codes cannot be generated for code lengths of 9 or 8 by the HCC algorithm because neither 9 or 8 is equal to a prime number minus one. However, the diagonals of the array in Table 9-3 represent the end points of the group. Thus, code families for code lengths 9 and 8 can be easily generated from the table simply by removing the diagonals. Table 9-4 shows such a code family with a code length of 9 (constructed by removing the diagonal of 10s).

Table 9-4 now contains a near diagonal (upper left to lower right) consisting entirely of ones. This is a necessary mathematical property of the result of removing the initial diagonal of (p–1) in the previous operation, and will be exhibited for any prime p.

Extending the process, Table 9-5 shows a code family with a code length of 8 (constructed from Table 9-4 by removing the entry from each row that has a value of 1, subtracting 1 from each value remaining in the row, and discarding the last row).


Copyright © 2007 IEEE. All rights reserved.

**Table 9-4—EHCC family – Code length = 9, *N* = 11; Family Indices (SEQ) 1 through 9**

| INDEX (k) ----> | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| SEQ | | | | | | | | | |
| 1 | 1 | 6 | 4 | 3 | 9 | 2 | 8 | 7 | 5 |
| 2 | 2 | 1 | 8 | 6 | 7 | 4 | 5 | 3 | 9 |
| 3 | 3 | 7 | 1 | 9 | 5 | 6 | 2 | 4 | 8 |
| 4 | 4 | 2 | 5 | 1 | 3 | 8 | 6 | 9 | 7 |
| 5 | 5 | 8 | 9 | 4 | 1 | 7 | 2 | 3 | 6 |
| 6 | 6 | 3 | 2 | 7 | 1 | 4 | 9 | 8 | 5 |
| 7 | 7 | 9 | 6 | 8 | 3 | 1 | 5 | 2 | 4 |
| 8 | 8 | 4 | 2 | 6 | 5 | 9 | 1 | 7 | 3 |
| 9 | 9 | 3 | 5 | 4 | 7 | 6 | 8 | 1 | 2 |

**Table 9-5—EHCC family – Code length = 8, *N* = 11; Family indices (SEQ) 1 through 8**

| INDEX (k) ----> | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| SEQ | | | | | | | | |
| 1 | 5 | 3 | 2 | 8 | 1 | 7 | 6 | 4 |
| 2 | 1 | 7 | 5 | 6 | 3 | 4 | 2 | 8 |
| 3 | 2 | 6 | 8 | 4 | 5 | 1 | 3 | 7 |
| 4 | 3 | 1 | 4 | 2 | 7 | 5 | 8 | 6 |
| 5 | 4 | 7 | 8 | 3 | 6 | 1 | 2 | 5 |
| 6 | 5 | 2 | 1 | 6 | 3 | 8 | 7 | 4 |
| 7 | 6 | 8 | 5 | 7 | 2 | 4 | 1 | 3 |
| 8 | 7 | 3 | 1 | 5 | 4 | 8 | 6 | 2 |

Copyright © 2007 IEEE. All rights reserved.

To obtain a code family of length *N*–2, the code family of length *N*–1 calculated by the HCC algorithm shall be modified by deleting the diagonal of the code family with the value of *N*–1 and removing the row of the code family with the family coefficient of *N*–1. This results in a code family represented in a square (*N*–2) by (*N*–2) array.

To obtain a code family of length *N*–3, the code family of length *N*–1 calculated by the HCC algorithm shall be modified by deleting the diagonals of the code family with the values of 1 and *N*–1, removing the rows of the code family with the family coefficients of *N*–1 and *N*–2, and subtracting 1 from all remaining values in the code family array. This results in a code family represented in a square (*N*–3) by (*N*–3) array with array values of 1 through *N*–3.

When using hopping patterns calculated using the HCC or EHCC algorithms, the values in the FH parameter set element shall be set as follows:

a) The Hop Set field shall be 0; the value of 0 for the Hop Set field indicates that the HCC/EHCC algorithm is in use and that the Hop Pattern field contains the HCC/EHCC family index and the Hop Index field contains the HCC/EHCC index.

b) The family index of the code being used shall be placed in the Hop Pattern field.

c) The index shall be placed in the Hop Index field.

## 9.9 HCF

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 STA winning an instance of EDCA contention (see 9.9.1) during the CP or by a non-AP STA 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 9-17 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 9-1. The mapping of frame types to ACs is described in 9.1.3.1.


Copyright © 2007 IEEE. All rights reserved.

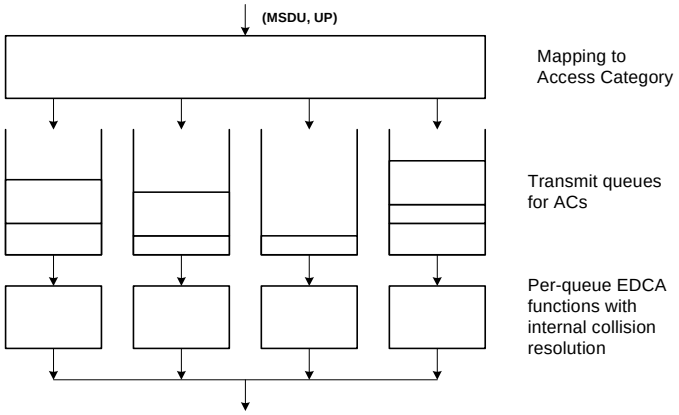



**Figure 9-17—Reference implementation model**

### 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 AP in the EDCA Parameter Set information element in Beacon and Probe Response frames transmitted by the AP. 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 STAs 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 STA 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 QoS AP may send broadcast/multicast frames without using any protection mechanism. In a QoS IBSS, 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.

Copyright © 2007 IEEE. All rights reserved.

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 STAs and is advertised by the AP in the EDCA Parameter Set information element in Beacon and Probe Response frames transmitted by the AP. The value of AIFSN[AC] shall be greater than or equal to 1 for APs. 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 STA 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.


Copyright © 2007 IEEE. All rights reserved.

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 9-18. 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 9-18). 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.

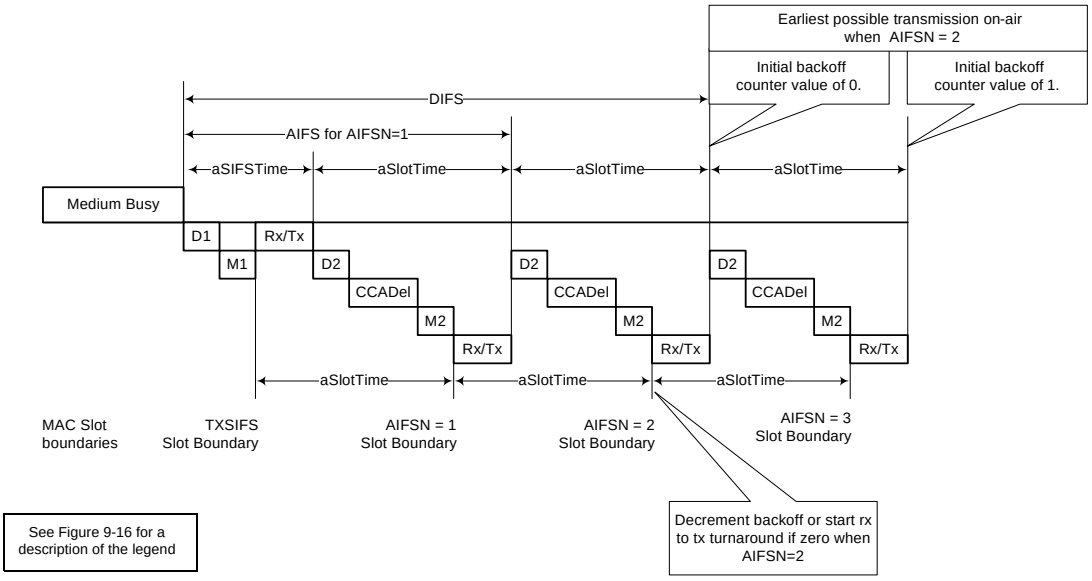



**Figure 9-18—EDCA mechanism timing relationships**

### 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 is 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 STA 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

Copyright © 2007 IEEE. All rights reserved.

the STA 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 STA 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 STA shall transmit before the expiry of the NAV set by the frame that resulted in a transmission failure. All other ACs at the STA 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 STA.

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 STA or an AP.

Note that, as for an EDCA TXOP, a multiple frame transmission is granted to an EDCAF, not to a non-AP STA or AP, 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 STA 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:


Copyright © 2007 IEEE. All rights reserved.

a) If the QSRC[AC] or the QLRC[AC] for the QoS STA 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, as appropriate (see 9.2.3).

#### 9.9.1.6 Retransmit procedures

QoS STAs 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. QoS STAs 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 STA 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 QoS STA 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. STAs 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 QoS STA 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).

QoS STAs 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

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 339 of 1232

remaining, undelivered fragments of that MSDU, shall be discarded by the source STA 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 STAs and to allocate TXOPs to non-AP STAs.

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 STAs associated in a BSS during both the CP and the CFP. Another significant difference is that the HC grants a non-AP STA a polled TXOP with duration specified in a QoS (+)CF-Poll frame. Non-AP STAs 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 STA contains a duration value chosen to cause STAs in the BSS to set their NAVs to protect the expected subsequent frames.

All non-AP QoS STAs 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 Beacon frames. 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.[22]

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 standard.

#### 9.9.2.1 HCCA procedure

The HC gains control of the WM as needed to send QoS traffic to non-AP STAs and to issue QoS (+)CF-Poll frames to non-AP STAs 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.

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


Copyright © 2007 IEEE. All rights reserved.

#### 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 9-13, 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 STAs 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 an AP 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 non-QoS STA. The use of a non-QoS CF-Poll frame by an AP to a non-AP QoS STA 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 9-19.

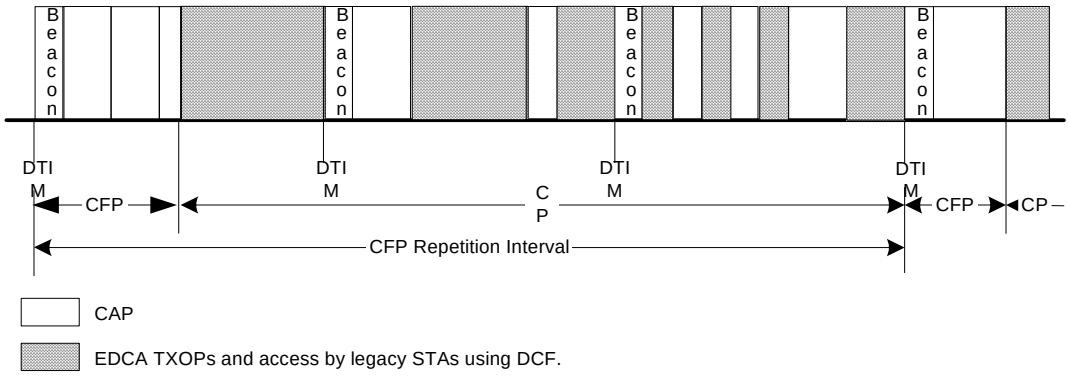



**Figure 9-19—CAP/CFP/CP periods**

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.

#### 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 BSS 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 AP, the AP may recover using the rules described in this subclause even if the recovery is from the absence of an expected reception.

Copyright © 2007 IEEE. All rights reserved.

If the beginning of reception of an expected response, as detected by the occurrence of PHY-CCA.indication(busy) primitive at the STA that is expecting the response, does not occur during the first slot time following SIFS, then[23]

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 QoS STA, 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 QoS STA 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 STAs 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 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 STA. 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 STA with a duration/ID covering only the response frame, the HC shall assume that the STA 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 QoS STA 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 QoS STA 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 QoS STA 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

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


Copyright © 2007 IEEE. All rights reserved.

either the QoS Null frame or the frame with the Duration/ID field covering only the response frame, then the non-AP STA shall cease control of the channel.[24] 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 QoS STA that is expecting the response during the first slot time following SIFS, the non-AP QoS STA 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 STA 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 STA 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 9-20. Within a polled TXOP, a STA may initiate the transmission of one or more frame exchange sequences, with all such sequences nominally separated by a SIFS interval. The STA 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 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 STA 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 STAs 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 STA that holds the TXOP.[25, 26]

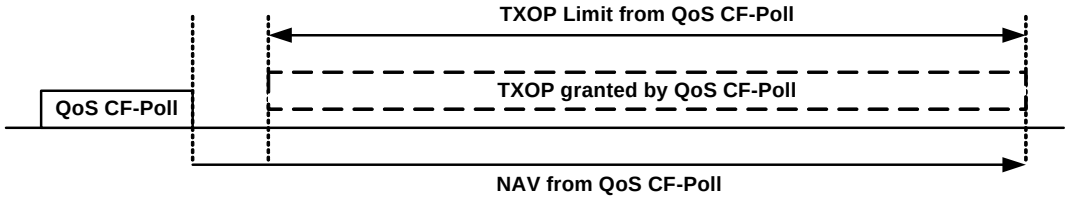



**Figure 9-20—Polled TXOP**

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

[25]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 STA needs to sense the channel and is given by floor(TXOP limit/limit imposed in the regulatory domain).

[26]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 STA may send in the granted TXOP.

Copyright © 2007 IEEE. All rights reserved.

#### 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 STA through an HCCA poll. However, the HC may reclaim the TXOP if a STA 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 STAs to poll and it has no more data, management, BlockAckReq, or BlockAck frames to send, it may reset the NAVs of all STAs in the BSS 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 AP contains a PC, during the CFP, it may reset the NAVs of all STAs within the BSS (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 STA that receives a CF-End frame containing the BSSID of the BSS, in which the non-AP STA is associated, shall reset the NAV value to 0. A non-AP STA 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 STA receives a QoS (+)CF-Poll frame containing the BSSID of the BSS in which the STA is associated, that STA shall update the NAV if necessary and shall save the TXOP holder address for that BSS, 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 STA and the MAC address in the TA field in the RTS frame matches the saved TXOP holder address, then the STA 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 STA 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 STA 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 STA may respond with any frame. Upon receiving a QoS (+)CF-Poll frame, a non-AP STA 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 STAs are not allowed to send QoS (+)CF-Poll frames. Non-AP STAs shall not send QoS (+)Data frames in response to any data frame other than the QoS (+)CF-Poll frames.

If a STA has set up at least one TS for which the Aggregation subfield in the associated TSPEC is set to 0, the AP shall use only QoS CF-Poll or QoS CF-Ack+CF-Poll frames to poll the STA and shall never use QoS (+)Data+CF-Poll to poll the STA. 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 BSSBasicRateSet parameter in order to set the NAV of all STAs 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.

QoS STAs shall use QoS data frames for all MSDU transfers to another QoS STA. 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 STA uses for traffic belonging to this TC or TS, or the TXOP Duration Requested subfield shall indicate the duration that the


Copyright © 2007 IEEE. All rights reserved.

STA 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 STA 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).

QoS STAs 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 STA 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 STA a SIFS after receiving the final transmission of the previous TXOP, if the non-AP STA 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, STAs 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 STA for which the +CF-Ack is intended, unless the STA to which the +CF-Ack is intended, sets the Q-Ack subfield in the QoS Capability information element in the (Re)Association Request frame. STAs are not required to be able to transmit QoS data frames with subtypes that include +CF-Ack.

An AP that has dot11QAckOptionImplemented true may allow associations with STAs advertising support for the Q-Ack option. Such associations do not require the AP to employ piggyback acknowledgments directed toward that associated STA in frames that are not directed to that associated STA. QoS STAs shall be able to process received QoS data frames with subtypes that include +CF-Ack when the STA to which the acknowledgment is directed is the same as the STA addressed by the Address 1 field of that data frame. A STA 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 STAs. 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 STA 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 STA address. QoS (Data+)CF-Poll+CF-Ack frames are useful if the HC wants to grant another TXOP to the same non-AP STA a SIFS after receiving the final transmission of that non-AP STA'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 STA a SIFS after receiving the final transmission of a non-AP STA's previous TXOP, if the non-AP STA 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 STA utilized the full TXOP provided by the HC, and it has to send more MPDUs. When this STA 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 AP grants a TSPEC with the Access Policy subfield set to HEMM and if the corresponding AC needs admission control, the AP 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 STAs 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. APs indicate whether they process TXOP request or queue size in the QoS Info field in the Beacon, Probe Response, and (Re)Association Response frames. APs shall process requests in at least one format. The AP may reallocate TXOPs if the request belongs to TS or update the EDCA parameter set if the above request belongs to TC. Non-AP STAs shall use only the request format that the AP indicates it can process.

Copyright © 2007 IEEE. All rights reserved.

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

STAs 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.[27]

If a QoS STA 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 AP in response to a QoS data frame with a subtype that includes CF-Poll, the polled non-AP STA is allowed to send a CTS frame (as a CTS frame is shorter 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.[28]

### 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 STA desires guarantee on the amount of time that it can access the channel. The HC, which is in the AP, 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 STA 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 9-1, that does not require admission control. APs shall support admission control procedures, at least to the minimal extent of advertising that admission is not mandatory on its ACs.

The AP 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 AP, 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 STA 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 AP shall associate the received UP of the ADDTS Request frame with the appropriate AC per the UP-to-AC mappings described in 9.1.3.1. The non-AP STA may

[27]The sending of an RTS frame during the CFP is usually unnecessary, but may be used to ensure that the addressed recipient QoS STA 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 QoS STA, because a non-QoS STA in the same BSS has its NAV set to protect the CFP, which renders those non-QoS STAs 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.

[28]This is unnecessary because the NAVs in the vicinity of the QoS AP were set by the QoS (+)CF-Poll frame.


Copyright © 2007 IEEE. All rights reserved.

transmit unadmitted traffic for the ACs for which the AP does not require admission control. If a STA desires to send data without admission control using an AC that mandates admission control, the STA 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 AP

Regardless of the AC's ACM setting, the AP 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 STA, the AP shall make a determination about whether to

a) Accept the request, or

b) Deny the request.

The algorithm used by the AP to make this determination is a local matter. If the AP decides to accept the request, the AP shall also derive the medium time from the information conveyed in the TSPEC element in the ADDTS Request frame. The AP 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 AP shall transmit a TSPEC element to the requesting non-AP STA contained in an ADDTS Response frame. If the AP is accepting the request, the Medium Time field shall be specified.

#### 9.9.3.1.2 Procedure at non-AP STAs

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 STA 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 STA 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 STA shall recompute the admitted_time for the specified EDCAF as follows:

admitted_time = admitted_time + dot11EDCAAveragingPeriod * (medium time of TSPEC).

The non-AP STA may choose to tear down the explicit request at any time. For the teardown of an explicit admission, the non-AP STA shall transmit a DELTS frame containing the TSID and direction that specify the TSPEC to the AP.

If the non-AP STA 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 STA, two parameters are defined. The parameter used_time signifies the amount of time used, in units of 32 μs, by the non-AP STA in dot11EDCAAveragingPeriod. The parameter admitted_time is the medium time allowed by the AP, in units of 32 μs, in dot11EDCA-AveragingPeriod. The non-AP STA shall update the value of used_time:

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 347 of 1232

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 STA 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 STA 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 STA may make a request for more admitted_time to the AP 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 STAs, 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 STA 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 STA. An SP starts at fixed intervals of time specified in Service Interval field. The first SP starts when the lower order 4 octets of the TSF timer equals the value specified in Service Start Time field.[29] 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.

[29]The lower order 4 octets 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 octets match the Start Time field in the Schedule element.


Copyright © 2007 IEEE. All rights reserved.

When the HC aggregates the admitted TS, it shall set the Aggregation field in the granted TSPEC to 1. An AP shall schedule the transmissions in HCCA TXOPs and communicate the service schedule to the non-AP STA. The HC shall provide an aggregate service schedule if the non-AP STA sets the Aggregation field in its TSPEC request. If the AP establishes an aggregate service schedule for a non-AP STA, it shall aggregate all HCCA streams for the STA. The service schedule is communicated to the non-AP STA 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 an AP 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 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 STA 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 STA from the non-AP STA'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 STAs 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 STA 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 standard. The mandatory set of parameters can be set by any higher layer entity or may be generated autonomously by the MAC.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 349 of 1232

If a STA 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.

## 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.[30] In this subclause, the STA 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, QoS STAs 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 9-21 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.

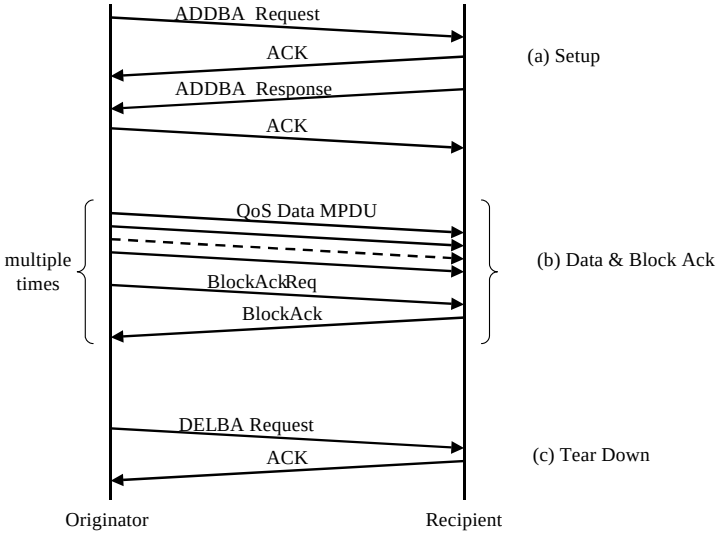



**Figure 9-21—Message sequence chart for Block Ack mechanism: (a) setup, (b) data and acknowledgment transfer and (c) tear down**

[30] 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.


Copyright © 2007 IEEE. All rights reserved.

### 9.10.2 Setup and modification of the Block Ack parameters

A STA 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 STA 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 STA 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 STA shall respond by an ADDBA Response frame. The receiving STA, which is the intended peer, has the option of accepting or rejecting the request. When the STA accepts, then a Block Ack agreement exists between the originator and recipient. When the STA 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 STA 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.

### 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

— Split transmission of data MPDUs sent under Block Ack policy across multiple TXOPs

— 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 BSS.

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.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 351 of 1232

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 STA 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 STA 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 9-22.

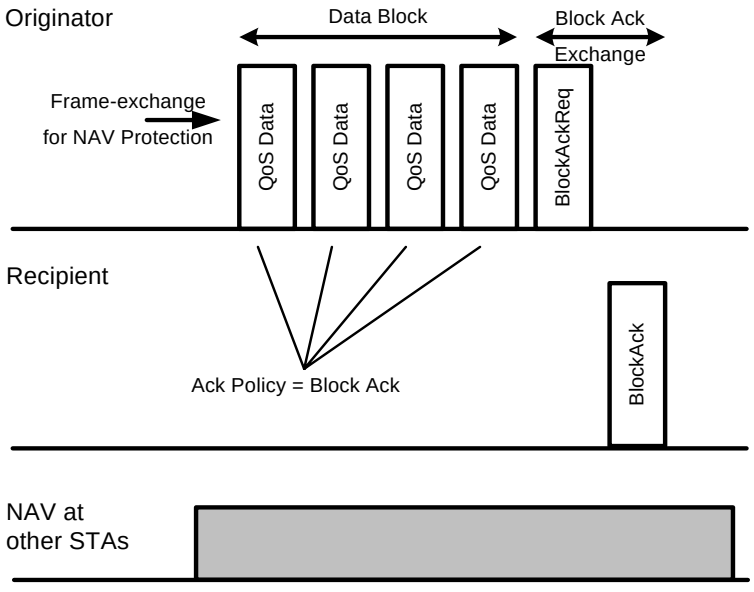



**Figure 9-22—A typical Block Ack sequence when immediate policy is used**


Copyright © 2007 IEEE. All rights reserved.

A typical Block Ack sequence using the delayed Block Ack is shown in Figure 9-23.

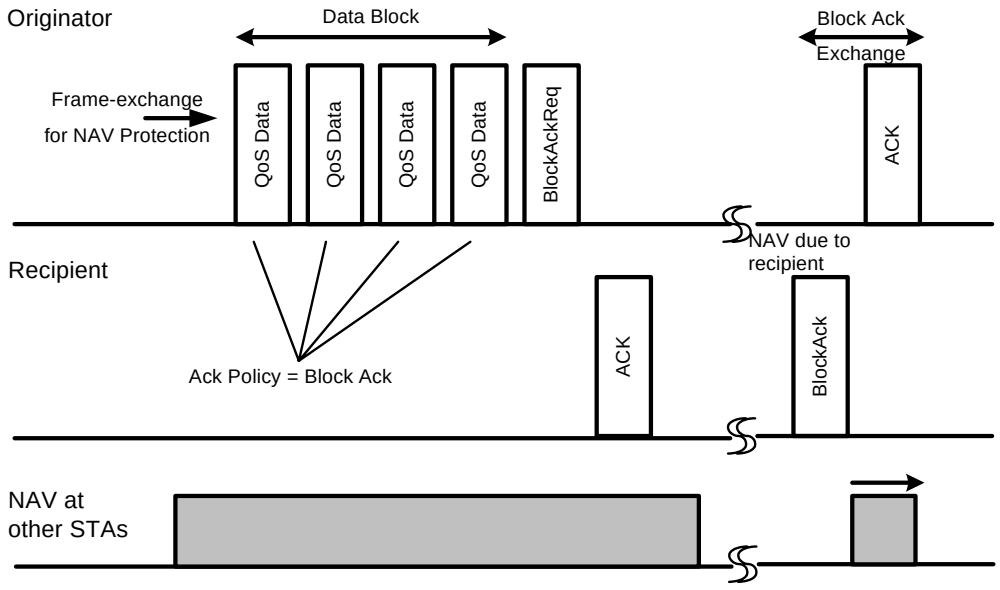



**Figure 9-23—A typical BlockAck sequence when delayed policy is used**

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.

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.

Copyright © 2007 IEEE. All rights reserved.

### 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.[31] 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).

## 9.11 No Acknowledgment (No Ack)

The usage of No Ack is determined by the policy at the QoS STA. 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.

## 9.12 Frame exchange sequences

The allowable frame exchange sequences are defined using an extension of the EBNF format as defined in ISO/IEC 14977 : 1996(E). The elements of this syntax that are used here are as follows:

— [a] = a is optional.

— {a} = a is repeated zero or more times.

— n{a} = a is repeated n or more times. For example, 3{a} requires 3 or more "a".

[31]Normal Ack rules apply.


Copyright © 2007 IEEE. All rights reserved.

— a|b = a or b.

— ( ) = grouping, so "a (b|c)" is equivalent to "a b | a c".

— (* a *) = “a” is a comment. Comments are placed before the text they relate to.

— < > = order of frames not relevant. For example, <a b> is either "a b" or "b a."

— A rule is termined by a semicolon ";"

— Whitespace is not significant, but it is used to highlight the nesting of grouped terms.

Two types of terminals are defined:

— **Frames.** A frame is shown in Bold and identified by its type/subtype (e.g., Beacon, Data). Frames are shown in an initial capital letter.

— *Attributes.* Attributes are shown in italic. An attribute is introduced by the the "+" character. The attribute specifies a condition that applies to the frame that preceeds it. Where there are multiple attributes applied, they are generally ordered in the same order of the fields in the frame to which they refer.

Nonterminals of this syntax are shown in a normal font, i.e., a sequence of words joined by hyphens (e.g., cf-frame-exchange-sequence).

The attributes are defined in Table 9-6.

**Table 9-6—Attributes applicable to frame exchange sequence definition**

| Attribute | Description |
|---|---|
| *block-ack* | QoS Data frame has ack policy set to Block Ack |
| *broadcast* | Frame RA is the broadcast address |
| *CF* | Beacon contains a CFP element |
| *CF-Ack* | Data type CF-Ack subtype bit set or CF-End+CF-Ack frame |
| *CF-Poll* | Data type CF-Poll subtype bit set |
| *delayed* | BlockAck or BlockAckReq under a delayed policy |
| *DTIM* | Beacon is a DTIM |
| *frag* | Frame has its More Fragments field set to 1 |
| *group* | Frame RA has i/g bit set to 1 |
| *individual* | Frame RA has i/g bit set to 0 |
| *last* | Frame has its More Fragments field set to 0 |
| *no-ack* | QoS Data frame has ack policy set to No Ack |
| *normal-ack* | QoS Data frame has ack policy set to Normal Ack |
| *null* | Data type Null Data subtype bit set |
| *pifs* | Frame is transmitted using a PIFS |
| *QAP* | Frame is transmitted by a QoS AP |
| *QoS* | Data type QoS subtype bit set |
| *self* | Frame RA = TA |

Copyright © 2007 IEEE. All rights reserved.

The allowable frame exchange sequence is defined by the rule frame sequence. Except where modified by the *pifs* attribute, frames are separated by a SIFS.

(* This rule defines all the allowable frame exchange sequences *)
frame-sequence =

( [**CTS**] (**Management** +*broadcast* | **Data** +*group*) ) |

( [**CTS** | **RTS CTS** | **PS-Poll**] {frag-frame **Ack**} last-frame **Ack** ) |

(**PS-Poll Ack**) |

( [**Beacon** +*DTIM* ] {cf-sequence} [**CF-End** [+*CF-Ack*] ] )|

hcf-sequence;

(* A frag-frame is a nonfinal part of an individually addressed MSDU or MMPDU *)
frag-frame = (**Data** | **Management**) +*individual* +*frag*;

(* This is the last (or only) part of a an individually addressed MSDU or MMPDU *)
last-frame = (**Data** | **Management**) +*individual* +*last*;

(* A cf-sequence expresses all the sequences that may be generated within a contention free period. The first frame in this sequence is sent by the AP. *)
cf-sequence =

(*Broadcast *)

**Beacon** | **Management** +*broadcast* | **Data** +*group* [+*QoS*] |

(* CF poll with data *)

(**Data**+*individual* +*CF-Poll* [+*CF-Ack*]

(**Data** +*individual* +*CF-Ack* [**Data** +*null* +*CF-Ack*] |
**Data** +*null* +*CF-Ack*) ) |

(* CF poll without data *)

**Data** +*individual* +*null* +*CF-Poll* [+*CF-Ack*]
(**Data** +*null* |
(**Data** +*individual* (**Data** +*null* +*CF-Ack* | **Ack** ) ) )|

(* individual management *)

(**Management** +*individual* Ack) |

(* All the sequences initiated by an HC *)

hcf-sequence;

(* An hcf-sequence represents all the sequences that may be generated under HCCA. The sequence may be initiated by an HC within a CFP, or it may be initiated by a STA using EDCA channel access. *)
hcf-sequence =

( [**CTS**] 1{(**Data** +*group* [+*QoS*] ) | **Management** +*broadcast*) +pifs} |

( [**CTS**] 1{txop-sequence} ) |

(* HC only, polled TXOP delivery *)

( [**RTS CTS**] non-cf-ack-piggybacked-qos-poll-sequence )


Copyright © 2007 IEEE. All rights reserved.

(* HC only, polled TXOP delivery *)

cf-ack-piggybacked-qos-poll-sequence |

(* HC only, self TXOP delivery or termination *)

**Data** +*self* +*null* +*CF-Poll* +*QoS*;

(* A poll-sequence is the start of a polled TXOP, in which the HC delivers a polled TXOP to a STA. The poll may or may not piggyback a CF-Ack according to whether the previous frame received by the HC was a Data frame. *)

poll-sequence =

non-cf-ack-piggybacked-qos-poll-sequence |

cf-ack-piggybacked-qos-poll-sequence;

(* A cf-ack-piggybacked-qos-poll-sequence is the start of a polled TXOP that also delivers a CF-Ack. There are two main variants, polls that deliver data and, therefore, need acknowledgment and polls that do not. *)

cf-ack-piggybacked-qos-poll-sequence=

(qos-poll-requiring-no-ack +*CF-Ack* (
[**CTS** +*self*] polled-txop-content |
polled-txop-termination) ) |

(qos-poll-requiring-ack +*CF-Ack* (
**Ack** (
polled-txop-content |
polled-txop-termination) ) |

cf-ack-piggybacked-qos-data-sequence);

(* A non-cf-ack-piggybacked-qos-poll-sequence is the start of a polled TXOP that does not deliver a CF-Ack. Except for this, it is identical to the CF-Ack version. *)

non-cf-ack-piggybacked-qos-poll-sequence=

(qos-poll-requiring-no-ack (
[**CTS** +*self*] polled-txop-content |
polled-txop-termination) ) |

(qos-poll-requiring-ack (
**Ack** (
polled-txop-content |
polled-txop-termination) ) |

cf-ack-piggybacked-qos-data-sequence);

(* This sequence is the delivery of a single frame that is the TXOP poll frame that does not require acknowledgment either because the frame carries no data or because the frame carries data that do not require immediate acknowledgment. *)

qos-poll-requiring-no-ack =

**Data** +*null* +*CF-Poll* +*QoS* |

**Data** +*individual* +*CF-Poll* +*QoS* +(*no-ack*|*block-ack*);

(* A qos-poll-requiring-ack is the delivery of a single frame that is a TXOP poll frame, but also carries data that require immediate acknowledgment. *)

qos-poll-requiring-ack =

**Data** +*individual* +*CF-Poll* [+*CF-Ack*] +*QoS* +*normal-ack*;

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 357 of 1232

(* Polled-txop-content is what may occur after the delivery of a polled TXOP. A QoS STA transmits the first frame in this sequence *)
polled-txop-content =

1{txop-sequence} [polled-txop-termination];

(* A polled-txop-termination may be used by a QoS STA to terminate the polled TXOP. The data frame is addressed to the HC, which regains control of the medium and may reuse any unused polled TXOP duration. *)
polled-txop-termination =

**Data** +*individual* +*null* +*QoS* +*normal-ack* **Ack**;

(* A TXOP (either polled or EDCA) may be filled with txop-sequences, which are initiated by the TXOP holder. *)
txop-sequence =

( ( (**RTS CTS**) | **CTS** +*self*) **Data** +*individual* +*QoS* +(*block-ack* | *no-ack*) ) |

[**RTS CTS**] (txop-part-requring-ack txop-part-providing-ack )|

[**RTS CTS**] **(Management** | (**Data** +*QAP*)) +*individual* **Ack** |

[**RTS CTS**] **(BlockAckReq BlockAck**);

(* These frames require acknowledgment *)
txop-part-requring-ack =

**Data** +*individual* [+*null*] +*QoS* +*normal-ack* |

**BlockAckReq** +delayed |

**BlockAck** +delayed;

(* These frames provide acknowledgment to the txop-part-requiring-ack *)
txop-part-providing-ack=

**Ack** |

cf-ack-piggybacked-poll-sequence | (* An HC responds with a new polled TXOP on expiry of current TXOP *)

cf-ack-piggybacked-data-sequence | (* An HC responds with CF-Ack and its own data on expiry of TXOP *)

**Data** +*CF-Ack*;

(* An HC has received a frame requiring Ack with a duration value indicating the end of the TXOP. The HC continues the CAP by transmitting its own data. *)
cf-ack-piggybacked-qos-data-sequence =

( **Data** +*individual* +*CF-Ack* +*QoS* +(*no-ack*|*block-ack*) polled-txop-content ) |

( **Data** +*individual* +*CF-Ack* +*QoS*+*normal-ack* (
**Ack** polled-txop-content |
**Data** +*CF-Ack* |
cf-ack-piggybacked-qos-poll-sequence ) ) ;

## 9.13 Protection mechanism for non-ERP receivers

The intent of a protection mechanism is to ensure that a STA does not transmit an MPDU of type Data or an MMPDU with an ERP-OFDM preamble and header unless it has attempted to update the NAV of receiving NonERP STAs. The updated NAV period shall be longer than or equal to the total time required to send the data and any required response frames. ERP STAs shall use protection mechanisms (such as RTS/CTS or CTS-to-self) for ERP-OFDM MPDUs of type Data or an MMPDU when the Use_Protection field of the ERP


Copyright © 2007 IEEE. All rights reserved.

Information element is set to 1 (see the requirements of 9.6). Protection mechanisms frames shall be sent using one of the mandatory Clause 15 or Clause 18 rates and using one of the mandatory Clause 15 or Clause 18 waveforms, so all STAs in the BSA will know the duration of the exchange even if they cannot detect the ERP-OFDM signals using their CCA function.

Note that when using the Clause 19 options, ERP-PBCC or DSSS-OFDM, there is no need to use protection mechanisms, as these frames start with a DSSS header.

In the case of a BSS composed of only ERP STAs, but with knowledge of a neighboring co-channel BSS having NonERP traffic, the AP may require protection mechanisms to protect the BSS's traffic from interference. This will provide propagation of NAV to all attached STAs and all STAs in a neighboring co-channel BSS within range by messages sent using rates contained in the BSSBasicRateSet parameter. The frames that propagate the NAV throughout the BSS include RTS/CTS/ACK frames, all data frames with the "more fragments" field set to 1, all data frames sent in response to PS-Poll that are not proceeded in the frame sequence by a data frame with the "more fragments" field set to 1, Beacon frames with nonzero CFDurRemaining, CF-End frames, and CF-End+ACK frames.

When RTS/CTS is used as the protection mechanism, cases exist such as NAV resetting (discretionary, as indicated in 9.2.5.4), where a hidden STA may reset its NAV and this may cause a collision. The likelihood of occurrence is low, and it is not considered to represent a significant impairment to overall system operation. A mechanism to address this possible situation would be to use alternative protection mechanisms or to revert to alternative modulation methods.

If a protection mechanism is being used, a fragment sequence shall use ERP-OFDM modulation for the final fragment and control response.

The rules for calculating RTS/CTS NAV fields are unchanged when using RTS/CTS as a protection mechanism.

Additionally, if any of the rates in the BSSBasicRateSet parameter of the protection mechanism frame transmitting STA's BSS are Clause 15 or Clause 18 rates, then the protection mechanism frames shall be sent at one of those Clause 15 or Clause 18 basic rates.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 359 of 1232

Copyright © 2007 IEEE. All rights reserved.
Samsung Ex. 1044, Page 360 of 1232

# 10. Layer management

## 10.1 Overview of management model

Both the MAC sublayer and PHY conceptually include management entities, called MLME and PLME, respectively. These entities provide the layer management service interfaces through which layer management functions can be invoked.

In order to provide correct MAC operation, an SME is present within each STA. The SME is a layer-independent entity that can be viewed as residing in a separate management plane or as residing "off to the side." The exact functions of the SME are not specified in this standard, but in general this entity can be viewed as being responsible for such functions as the gathering of layer-dependent status from the various layer management entities (LMEs), and similarly setting the value of layer-specific parameters. SME would typically perform such functions on behalf of general system management entities and would implement standard management protocols. Figure 5-10 (in 5.7) depicts the relationship among management entities.

The various entities within this model interact in various ways. Certain of these interactions are defined explicitly within this standard, via a SAP across which defined primitives are exchanged. Other interactions are not defined explicitly within this standard, such as the interfaces between MAC and MLME and between PLCP and PLME, represented as double arrows within Figure 10-1. The specific manner in which these MAC and PHY LMEs are integrated into the overall MAC sublayer and PHY is not specified within this standard.

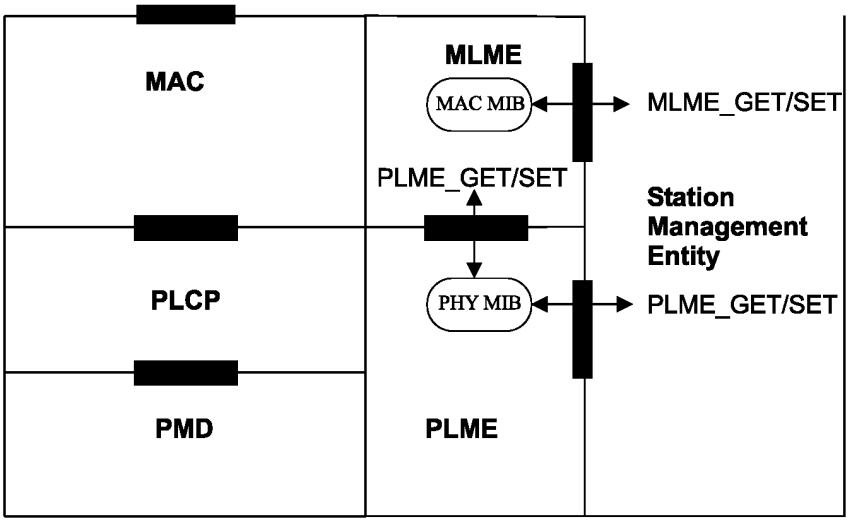



**Figure 10-1—GET and SET operations**

The management SAPs within this model are the following:

— SME-MLME SAP

— SME-PLME SAP

— MLME-PLME SAP

The latter two SAPs support identical primitives, and in fact can be viewed as a single SAP (called the PLME SAP) that can be used either directly by MLME or by SME. In this fashion, the model reflects what is anticipated to be a common implementation approach in which PLME functions are controlled by the

Copyright © 2007 IEEE. All rights reserved.

MLME (on behalf of SME). In particular, PHY implementations are not required to have separate interfaces defined other than their interfaces with the MAC and MLME.

## 10.2 Generic management primitives

The management information specific to each layer is represented as a MIB for that layer. The MLME and PLME are viewed as "containing" the MIB for that layer. The generic model of MIB-related management primitives exchanged across the management SAPs is to allow the SAP user-entity to either GET the value of a MIB attribute, or to SET the value of a MIB attribute. The invocation of a SET.request primitive can require that the layer entity perform certain defined actions.

Figure 10-1 depicts these generic primitives.

The GET and SET primitives are represented as REQUESTs with associated CONFIRM primitives. These primitives are prefixed by MLME or PLME depending upon whether the MAC sublayer or PHY management SAP is involved. In the following, XX denotes MLME or PLME:

XX-GET.request(MIBattribute)
: Requests the value of the given MIBattribute.

XX-GET.confirm(status, MIBattribute, MIBattributevalue)
: Returns the appropriate MIB attribute value if status = "success," otherwise returns an error indication in the Status field. Possible error status values include "invalid MIB attribute" and "attempt to get write-only MIB attribute."

XX-SET.request(MIBattribute, MIBattributevalue)
: Requests that the indicated MIB attribute be set to the given value. If this MIBattribute implies a specific action, then this requests that the action be performed.

XX-SET.confirm(status, MIBattribute)
: If status = "success," this confirms that the indicated MIB attribute was set to the requested value, otherwise it returns an error condition in status field. If this MIBattribute implies a specific action, then this confirms that the action was performed. Possible error status values include "invalid MIB attribute" and "attempt to set read-only MIB attribute."

Additionally, there are certain requests (with associated confirms) that can be invoked across a given SAP that do not involve the setting or getting of a specific MIB attribute. One of these is supported by each SAP, as follows:

— XX-RESET.request: where XX is MLME or PLME as appropriate
— XX-RESET.confirm

This service is used to initialize the management entities, the MIBs, and the datapath entities. It can include a list of attributes for items to be initialized to nondefault values. The corresponding .confirm indicates success or failure of the request.

Other SAP-specific primitives are identified in 10.3.

## 10.3 MLME SAP interface

The services provided by the MLME to the SME are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface. MLME SAP primitives are of the general form ACTION.request followed by ACTION.confirm. The SME uses the services provided by the MLME through the MLME SAP.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.1 Power management

This mechanism supports the process of establishment and maintenance of the power management mode of a STA.

#### 10.3.1.1 MLME-POWERMGT.request

##### 10.3.1.1.1 Function

This primitive requests a change in the power management mode.

##### 10.3.1.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-POWERMGT.request(
                        PowerManagementMode,
                        WakeUp,
                        ReceiveDTIMs
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PowerManagementMode | Enumeration | ACTIVE, POWER_SAVE | An enumerated type that describes the desired power management mode of the STA. |
| WakeUp | Boolean | True, false | When true, the MAC is forced immediately into the Awake state. This parameter has no effect if the current power management mode is ACTIVE. |
| ReceiveDTIMs | Boolean | True, false | When true, this parameter causes the STA to awaken to receive all DTIM frames. When false, the STA is not required to awaken for every DTIM frame. |

##### 10.3.1.1.3 When generated

This primitive is generated by the SME to implement the power-saving strategy of an implementation.

##### 10.3.1.1.4 Effect of receipt

This request sets the STA's power management parameters. The MLME subsequently issues a MLME-POWERMGT.confirm that reflects the results of the power management change request.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.1.2 MLME-POWERMGT.confirm

##### 10.3.1.2.1 Function

This primitive confirms the change in power management mode.

##### 10.3.1.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-POWERMGT.confirm(
ResultCode
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, NOT_SUPPORTED | Indicates the result of the MLME-POWERMGT.request. |

##### 10.3.1.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-POWERMGT.request to establish a new power management mode. It is not generated until the change has completed as defined in 11.2.1.

##### 10.3.1.2.4 Effect of receipt

The SME is notified of the change of power management mode.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.2 Scan

This mechanism supports the process of determining the characteristics of the available BSSs.

#### 10.3.2.1 MLME-SCAN.request

##### 10.3.2.1.1 Function

This primitive requests a survey of potential BSSs that the STA can later elect to try to join.

##### 10.3.2.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SCAN.request(
BSSType,
BSSID,
SSID,
ScanType,
ProbeDelay,
ChannelList,
MinChannelTime,
MaxChannelTime,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| BSSType | Enumeration | INFRASTRUCTURE, INDEPENDENT, ANY_BSS | Determines whether infrastructure BSS, IBSS, or both, are included in the scan. |
| BSSID | MACAddress | Any valid individual or broadcast MAC address | Identifies a specific or wildcard BSSID. |
| SSID | Octet string | 0–32 octets | Specifies the desired SSID or the wildcard SSID. |
| ScanType | Enumeration | ACTIVE, PASSIVE | Indicates either active or passive scanning. |
| ProbeDelay | Integer | N/A | Delay (in microseconds) to be used prior to transmitting a Probe frame during active scanning. |
| ChannelList | Ordered set of integers | Each channel will be selected from the valid channel range for the appropriate PHY and carrier set. | Specifies a list of channels that are examined when scanning for a BSS. |
| MinChannelTime | Integer | ≥ ProbeDelay | The minimum time (in TU) to spend on each channel when scanning. |
| MaxChannelTime | Integer | ≥ MinChannelTime | The maximum time (in TU) to spend on each channel when scanning. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.2.1.3 When generated

This primitive is generated by the SME for a STA to determine if there are other BSSs that it can join.

##### 10.3.2.1.4 Effect of receipt

This request initiates the scan process when the current frame exchange sequence is completed.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.2.2 MLME-SCAN.confirm

##### 10.3.2.2.1 Function

This primitive returns the descriptions of the set of BSSs detected by the scan process.

##### 10.3.2.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SCAN.confirm(
BSSDescriptionSet,
ResultCode,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| BSSDescriptionSet | Set of BSSDescriptions | N/A | The BSSDescriptionSet is returned to indicate the results of the scan request. It is a set containing zero or more instances of a BSSDescription. |
| ResultCode | Enumeration | SUCCESS, INVALID_ PARAMETERS, NOT_SUPPORTED | Indicates the result of the MLME-SCAN.confirm. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Each BSSDescription consists of the following elements:

| Name | Type | Valid range | Description |
|---|---|---|---|
| BSSID | MACAddress | N/A | The BSSID of the found BSS. |
| SSID | Octet string | 1–32 octets | The SSID of the found BSS. |
| BSSType | Enumeration | INFRASTRUC-TURE, INDEPENDENT | The type of the found BSS. |
| Beacon Period | Integer | N/A | The Beacon period of the found BSS (in TU). |
| DTIM Period | Integer | As defined in frame format | The DTIM period of the BSS (in beacon periods). |
| Timestamp | Integer | N/A | The timestamp of the received frame (probe response/beacon) from the found BSS. |
| Local Time | Integer | N/A | The value of the STA's TSF timer at the start of reception of the first octet of the timestamp field of the received frame (probe response or beacon) from the found BSS. |
| PHY Parameter Set | As defined in frame format or according to the relevant PHY clause. | As defined in frame format or according to the relevant PHY clause. | The parameter sets relevant to the PHY from the received Beacon or Probe Response frame. If no PHY Parameter Set information element is present in the received frame, this parameter contains the channel number on which the frame was received. Valid channel numbers are defined in the relevant PHY clause. |


Copyright © 2007 IEEE. All rights reserved.

| Name | Type | Valid range | Description |
|---|---|---|---|
| CF Parameter Set | As defined in frame format | As defined in frame format | The parameter set for the CF periods, if found BSS supports CF mode. |
| IBSS Parameter Set | As defined in frame format | As defined in frame format | The parameter set for the IBSS, if found BSS is an IBSS. |
| CapabilityInformation | As defined in frame format | As defined in frame format | The advertised capabilities of the BSS. |
| BSSBasicRateSet | Set of integers | 1–127 inclusive (for each integer in the set) | The set of data rates that must be supported by all STAs that desire to join this BSS. The STAs must be able to receive and transmit at each of the data rates listed in the set. |
| OperationalRateSet | Set of integers | 1–127 inclusive (for each integer in the set) | The set of data rates that the STA desires to use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. This set is a superset of the rates contained in the BSSBasicRateSet parameter. |
| Country | As defined in the Country element | As defined in the Country element | The information required to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain. Present only when TPC functionality is required, as specified in 11.8, or when dot11MultiDomainCapabilityEnabled is true. |
| IBSS DFS Recovery Interval | Integer | 1–255 | Only present if BSSType = INDEPENDENT.<br>The time interval that is used for DFS recovery.<br>Present only when DFS functionality is required, as specified in 11.9. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites and AKM suites supported in the BSS. |
| Load | As defined in frame format | As defined in frame format | The values from the BSS Load information element if such an element was present in the probe response or Beacon frame, 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 frame, 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 frame, else null. |

#### 10.3.2.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-SCAN.request to ascertain the operating environment of the STA.

#### 10.3.2.2.4 Effect of receipt

The SME is notified of the results of the scan procedure.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 367 of 1232

### 10.3.3 Synchronization

This mechanism supports the process of selection of a peer in the authentication process.

#### 10.3.3.1 MLME-JOIN.request

##### 10.3.3.1.1 Function

This primitive requests synchronization with a BSS.

##### 10.3.3.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-JOIN.request(
SelectedBSS,
JoinFailureTimeout,
ProbeDelay,
OperationalRateSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| SelectedBSS | BSSDescription | N/A | The BSSDescription of the BSS to join. The SelectedBSS is a member of the set of descriptions that was returned as a result of a MLME-SCAN.request. |
| JoinFailureTimeout | Integer | ≥ 1 | The time limit, in units of beacon intervals, after which the join procedure will be terminated. |
| ProbeDelay | Integer | N/A | Delay (in microseconds) to be used prior to transmitting a Probe frame during active scanning. |
| OperationalRateSet | Set of integers | 1–127 inclusive (for each integer in the set) | The set of data rates that the STA desires to use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. This set is a superset of the rates contained in the BSSBasicRateSet parameter. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.3.1.3 When generated

This primitive is generated by the SME for a STA to establish synchronization with a BSS.

##### 10.3.3.1.4 Effect of receipt

This primitive initiates a synchronization procedure once the current frame exchange sequence is complete. The MLME synchronizes its timing with the specified BSS based on the elements provided in the SelectedBSS parameter. The MLME subsequently issues a MLME-JOIN.confirm that reflects the results.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.3.2 MLME-JOIN.confirm

##### 10.3.3.2.1 Function

This primitive confirms synchronization with a BSS.

##### 10.3.3.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-JOIN.confirm(
                    ResultCode,
                    VendorSpecificInfo
                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_ PARAMETERS, TIMEOUT | Indicates the result of the MLME-JOIN.request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.3.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-JOIN.request to establish synchronization with a BSS.

##### 10.3.3.2.4 Effect of receipt

The SME is notified of the results of the synchronization procedure.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.4 Authenticate

This mechanism supports the process of establishing an authentication relationship with a peer MAC entity.

#### 10.3.4.1 MLME-AUTHENTICATE.request

##### 10.3.4.1.1 Function

This primitive requests authentication with a specified peer MAC entity.

##### 10.3.4.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-AUTHENTICATE.request(
PeerSTAAddress,
AuthenticationType,
AuthenticateFailureTimeout,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the authentication process. |
| AuthenticationType | Enumeration | OPEN_SYSTEM, SHARED_KEY | Specifies the type of authentication algorithm to use during the authentication process. |
| AuthenticationFail-ureTimeout | Integer | ≥ 1 | Specifies a time limit (in TU) after which the authentication procedure will be terminated. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.4.1.3 When generated

This primitive is generated by the SME for a STA to establish authentication with a specified peer MAC entity in order to permit Class 2 frames to be exchanged between the two STAs. During the authentication procedure, the SME can generate additional MLME-AUTHENTICATE.request primitives.

##### 10.3.4.1.4 Effect of receipt

This primitive initiates an authentication procedure. The MLME subsequently issues a MLME-AUTHENTICATE.confirm that reflects the results.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.4.2 MLME-AUTHENTICATE.confirm

##### 10.3.4.2.1 Function

This primitive reports the results of an authentication attempt with a specified peer MAC entity.

##### 10.3.4.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-AUTHENTICATE.confirm(
                                    PeerSTAAddress,
                                    AuthenticationType,
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the authentication process was attempted. This value must match the peerSTAAddress parameter specified in the corresponding MLME-AUTHENTICATE.request. |
| AuthenticationType | Enumeration | OPEN_SYSTEM, SHARED_KEY | Specifies the type of authentication algorithm that was used during the authentication process. This value must match the authenticationType parameter specified in the corresponding MLME-AUTHENTICATE.request. |
| ResultCode | Enumeration | SUCCESS, INVALID_ PARAMETERS, TIMEOUT, TOO_MANY_ SIMULTANEOUS_ REQUESTS, REFUSED | Indicates the result of the MLME-AUTHENTICATE.request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.4.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-AUTHENTICATE.request to authenticate with a specified peer MAC entity.

##### 10.3.4.2.4 Effect of receipt

The SME is notified of the results of the authentication procedure.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.4.3 MLME-AUTHENTICATE.indication

##### 10.3.4.3.1 Function

This primitive indicates receipt of a request from a specific peer MAC entity to establish an authentication relationship with the STA processing this primitive.

##### 10.3.4.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-AUTHENTICATE.indication(
                                PeerSTAAddress,
                                AuthenticationType,
                                VendorSpecificInfo
                                )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the authentication relationship was established. |
| AuthenticationType | Enumeration | OPEN_SYSTEM, SHARED_KEY | Specifies the type of authentication algorithm that was used during the authentication process. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.4.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of an authentication request from a specific peer MAC entity.

##### 10.3.4.3.4 Effect of receipt

The SME is notified of the receipt of the authentication request.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.4.4 MLME-AUTHENTICATE.response

##### 10.3.4.4.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested authentication with the STA that issued this primitive.

##### 10.3.4.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-AUTHENTICATE.response(
                                    PeerSTAAddress,
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the authentication request was received. |
| ResultCode | Enumeration | SUCCESS, REFUSED | Indicates the result response to the authentication request from the peer MAC entity. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.4.4.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-AUTHENTICATE.indication primitive.

##### 10.3.4.4.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that requested authentication.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.5 Deauthenticate

This mechanism supports the process of invalidating an authentication relationship with a peer MAC entity.

#### 10.3.5.1 MLME-DEAUTHENTICATE.request

##### 10.3.5.1.1 Function

This primitive requests that the authentication relationship with a specified peer MAC entity be invalidated.

##### 10.3.5.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.request(
                                    PeerSTAAddress,
                                    ReasonCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the deauthentication process. |
| ReasonCode | As defined in frame format | As defined in frame format | Specifies the reason for initiating the deauthentication procedure. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.5.1.3 When generated

This primitive is generated by the SME for a STA to invalidate authentication with a specified peer MAC entity in order to prevent the exchange of Class 2 frames between the two STAs. During the deauthentication procedure, the SME can generate additional MLME-DEAUTHENTICATE.request primitives.

##### 10.3.5.1.4 Effect of receipt

This primitive initiates a deauthentication procedure. The MLME subsequently issues a MLME-DEAUTHENTICATE.confirm that reflects the results.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.5.2 MLME-DEAUTHENTICATE.confirm

#### 10.3.5.2.1 Function

This primitive reports the results of a deauthentication attempt with a specified peer MAC entity.

#### 10.3.5.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.confirm(
                                    PeerSTAAddress,
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the deauthentication process was attempted. |
| ResultCode | Enumeration | SUCCESS, INVALID_ PARAMETERS, TOO_MANY_ SIMULTANEOUS_ REQUESTS | Indicates the result of the MLME-DEAUTHENTICATE.request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.5.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DEAUTHENTICATE.request to invalidate the authentication relationship with a specified peer MAC entity.

#### 10.3.5.2.4 Effect of receipt

The SME is notified of the results of the deauthentication procedure.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.5.3 MLME-DEAUTHENTICATE.indication

##### 10.3.5.3.1 Function

This primitive reports the invalidation of an authentication relationship with a specific peer MAC entity.

##### 10.3.5.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DEAUTHENTICATE.indication(
                                    PeerSTAAddress,
                                    ReasonCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the authentication relationship was invalidated. |
| ReasonCode | As defined in frame format. | As defined in frame format | Specifies the reason the deauthentication procedure was initiated. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.5.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an authentication relationship with a specific peer MAC entity.

##### 10.3.5.3.4 Effect of receipt

The SME is notified of the invalidation of the specific authentication relationship.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.6 Associate

The following primitives describe how a STA becomes associated with an AP.

#### 10.3.6.1 MLME-ASSOCIATE.request

##### 10.3.6.1.1 Function

This primitive requests association with a specified peer MAC entity that is within an AP.

##### 10.3.6.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ASSOCIATE.request(
PeerSTAAddress,
AssociateFailureTimeout,
CapabilityInformation,
ListenInterval,
Supported Channels,
RSN,
QoSCapability,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the association process. |
| AssociateFailureTime-out | Integer | ≥ 1 | Specifies a time limit (in TU) after which the associate procedure will be terminated. |
| CapabilityInformation | As defined in frame format | As defined in frame format | Specifies the requested operational capabilities to the AP. |
| ListenInterval | Integer | ≥ 0 | Specifies the number of beacon intervals that can pass before the STA awakens and listens for the next Beacon frame. |
| Supported Channels | As defined in the Supported Channels element | As defined in the Supported Channels element | The list of channels in which the STA is capable of operating.<br>Present only when DFS functionality is required, as specified in 11.9. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites and AKM suites supported in the BSS. |
| 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Additional parameters needed to perform the association procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

##### 10.3.6.1.3 When generated

This primitive is generated by the SME when a STA wishes to establish association with an AP.

##### 10.3.6.1.4 Effect of receipt

This primitive initiates an association procedure. The MLME subsequently issues an MLME-ASSOCIATE.confirm that reflects the results.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.6.2 MLME-ASSOCIATE.confirm

#### 10.3.6.2.1 Function

This primitive reports the results of an association attempt with a specified peer MAC entity that is within an AP.

#### 10.3.6.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ASSOCIATE.confirm(
ResultCode,
CapabilityInformation,
AssociationID,
SupportedRates,
EDCAParameterSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED_REASON_UNSPECIFIED, REFUSED_NOT_AUTHENTICATED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH | Indicates the result of the MLME-ASSOCIATE.request. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the operational capabilities advertised by the AP. |
| AssociationID | Integer | 1–2007 inclusive | If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned by the AP. |
| SupportedRates | Set of integers | 2–127 inclusive (for each integer in the set), bit 7 is set to 1 to indicate that a rate is a member of the BBSBasicRateSet. | The set of data rates (in units of 500 kb/s) that are supported by AP, including indication of which rates are part of the BSSBasicRateSet (according to 7.3.2.2). |
| EDCA-ParameterSet | As defined in frame format | As defined in frame format | Specifies the EDCA parameter set that the STA should use. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true. |
| Vendor-SpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.6.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-ASSOCIATE.request or receipt of an association response frame from the peer MAC entity to associate with a specified peer MAC entity that is within an AP.

#### 10.3.6.2.4 Effect of receipt

The SME is notified of the results of the association procedure.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.6.3 MLME-ASSOCIATE.indication

#### 10.3.6.3.1 Function

This primitive indicates that a specific peer MAC entity is requesting association with the local MAC entity, which is within an AP.

#### 10.3.6.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-ASSOCIATE.indication(
                                    PeerSTAAddress,
                                    CapabilityInformation,
                                    ListenInterval,
                                    SSID,
                                    SupportedRates,
                                    RSN,
                                    QoSCapability,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the association was received. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the operational capability definitions provided by the peer MAC entity as part of the association request. |
| ListenInterval | Integer | ≥ 0 | Specifies the listen interval value provided by the peer MAC as part of the association request. |
| SSID | Octet string | 0–32 octets | Specifies the SSID provided by the peer MAC entity as part of the association request. |
| SupportedRates | Set of integers | 2–127 inclusive (for each integer in the set) | The set of data rates (in units of 500 kb/s) that are supported by the STA that is requesting association. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites and AKM suites supported in the BSS. Only one pairwise cipher suite and only one authenticated key suite are allowed in the RSN information element. |
| QoSCapability | As defined in frame format | As defined in frame format | Specifies the parameters within the QoSCapability that are supported by the peer MAC entity. The parameter may be present only if the MIB attribute dot11QosOption-Implemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.6.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of an association request from a specific peer MAC entity.

#### 10.3.6.3.4 Effect of receipt

The SME is notified of the receipt of the association request.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.6.4 MLME-ASSOCIATE.response

##### 10.3.6.4.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested an association with the STA that issued this primitive, which is within an AP.

##### 10.3.6.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ASSOCIATE.response(
PeerSTAAddress,
ResultCode,
CapabilityInformation,
AssociationID,
EDCAParameterSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the association request was received. |
| ResultCode | Enumeration | SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH | Indicates the result response to the association request from the peer MAC entity. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the operational capabilities advertised by the AP. |
| AssociationID | Integer | 1–2007 inclusive | If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned to the peer MAC entity by the AP. |
| EDCAParameter-Set | As defined in frame format | As defined in frame format | Specifies the EDCA parameter set that the STA should use. The parameter shall be present only if the MIB attribute dot11Qos-OptionImplemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Additional parameters needed to perform the association response procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

##### 10.3.6.4.3 When generated

This primitive is generated by the SME of a STA that is within an AP as a response to an MLME-ASSOCIATE.indication primitive.

##### 10.3.6.4.4 Effect of receipt

This primitive initiates transmission of an AssociationResponse to the specific peer MAC entity that requested association.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.7 Reassociate

The following primitives describe how a STA becomes associated with another AP.

#### 10.3.7.1 MLME-REASSOCIATE.request

##### 10.3.7.1.1 Function

This primitive requests a change in association to a specified new peer MAC entity that is within an AP.

##### 10.3.7.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REASSOCIATE.request(
NewAPAddress,
ReassociateFailureTimeout,
CapabilityInformation,
ListenInterval,
Supported Channels
RSN,
QoSCapability,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| NewAPAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the reassociation process. |
| ReassociateFailure-Timeout | Integer | ≥ 1 | Specifies a time limit (in TU) after which the reassociate procedure will be terminated. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the requested operational capabilities to the AP. |
| ListenInterval | Integer | ≥ 0 | Specifies the number of beacon intervals that can pass before the STA awakens and listens for the next Beacon frame. |
| Supported Channels | As defined in the Supported Channels element | As defined in the Supported Channels element | The list of channels in which the STA is capable of operating.<br>Present only when DFS functionality is required, as specified in 11.9. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites and AKM suites supported in the BSS. |
| 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Additional parameters needed to perform the reassociation procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

##### 10.3.7.1.3 When generated

This primitive is generated by the SME for a STA to change association to a specified new peer MAC entity that is within an AP.

##### 10.3.7.1.4 Effect of receipt

This primitive initiates a reassociation procedure. The MLME subsequently issues a MLME-REASSOCIATE.confirm that reflects the results.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.7.2 MLME-REASSOCIATE.confirm

#### 10.3.7.2.1 Function

This primitive reports the results of a reassociation attempt with a specified peer MAC entity that is within an AP.

#### 10.3.7.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REASSOCIATE.confirm(
ResultCode,
CapabilityInformation,
AssociationID,
SupportedRates,
EDCAParameterSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED_REASON_UNSPECIFIED, REFUSED_NOT_AUTHENTICATED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH | Indicates the result of the MLME-REASSOCIATE.request. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the operational capabilities advertised by the AP. |
| AssociationID | Integer | 1–2007 inclusive | If the association request result was SUCCESS, then AssociationID specifies the association ID value assigned by the AP. |
| Supported-Rates | Set of integers | 2–127 inclusive (for each integer in the set), bit 7 is set to 1 to indicate that a rate is a member of the BBSBasicRateSet. | The set of data rates (in units of 500 kb/s) that are supported by AP, including indication of which rates are part of the BSSBasicRateSet (according to 7.3.2.2). |
| EDCAParameterSet | As defined in frame format | As defined in frame format | Specifies the EDCA parameter set that the STA should use. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.7.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-REASSOCIATE.request to reassociate with a specified peer MAC entity that is within an AP.

#### 10.3.7.2.4 Effect of receipt

The SME is notified of the results of the reassociation procedure.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.7.3 MLME-REASSOCIATE.indication

#### 10.3.7.3.1 Function

This primitive indicates that a specific peer MAC entity is requesting reassociation with the local MAC entity, which is within an AP.

#### 10.3.7.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-REASSOCIATE.indication(
                                    PeerSTAAddress,
                                    CurrentAPAddress,
                                    CapabilityInformation,
                                    ListenInterval,
                                    SSID,
                                    SupportedRates,
                                    RSN,
                                    QoSCapability,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the reassociation request was received. |
| CurrentAPAddress | MACAddress | Any valid individual MAC address | Specifies the address of the AP with which the peer STA is currently associated. |
| Capability-Information | As defined in frame format | As defined in frame format | Specifies the operational capability definitions provided by the peer MAC entity as part of the association request. |
| ListenInterval | Integer | ≥ 0 | Specifies the listen interval value provided by the peer MAC as part of the association request. |
| SSID | Octet string | 0–32 octets | Specifies the desired SSID provided by the peer MAC as part of the association request. |
| SupportedRates | Set of integers | 2–127 inclusive (for each integer in the set) | The set of data rates (in units of 500 kb/s) that are supported by the STA that is requesting reassociation. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites and AKM suites supported in the BSS. |
| 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 dot11QosOption-Implemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.7.3.3 When generated

This primitive is generated by the MLME as a result of the establishment of a reassociation with a specific peer MAC entity that resulted from a reassociation procedure that was initiated by that specific peer MAC entity.

#### 10.3.7.3.4 Effect of receipt

The SME is notified of the establishment of the reassociation.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.7.4 MLME-REASSOCIATE.response

##### 10.3.7.4.1 Function

This primitive is used to send a response to a specific peer MAC entity that requested a reassociation with the STA that issued this primitive, which is within an AP.

##### 10.3.7.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-REASSOCIATE.response(
PeerSTAAddress,
ResultCode,
CapabilityInformation,
AssociationID,
EDCAParameterSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity from which the reassociation request was received. |
| ResultCode | Enumeration | SUCCESS, REFUSED_REASON_UNSPECIFIED, REFUSED_CAPABILITIES_MISMATCH, REFUSED_EXTERNAL_REASON, REFUSED_AP_OUT_OF_MEMORY, REFUSED_BASIC_RATES_MISMATCH | Indicates the result response to the reassociation request from the peer MAC entity. |
| Capability-Information | As defined in Frame Format | As defined in Frame Format | Specifies the operational capabilities advertised by the AP. |
| AssociationID | Integer | 1–2007 inclusive | If the reassociation request result was SUCCESS, then AssociationID specifies the association ID value assigned to the peer MAC entity by the AP. |
| EDCAParameterSet | As defined in frame format | As defined in frame format | Specifies the EDCA parameter set that the STA should use. The parameter shall be present only if the MIB attribute dot11Qos-OptionImplemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Additional parameters needed to perform the association response procedure are not included in the primitive parameter list since the MLME already has that data (maintained as internal state).

##### 10.3.7.4.3 When generated

This primitive is generated by the SME of a STA that is within an AP as a response to an MLME-REASSOCIATE.indication primitive.

##### 10.3.7.4.4 Effect of receipt

This primitive initiates transmission of a response to the specific peer MAC entity that requested reassociation.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.8 Disassociate

#### 10.3.8.1 MLME-DISASSOCIATE.request

##### 10.3.8.1.1 Function

This primitive requests disassociation with a specified peer MAC entity that is within an AP.

##### 10.3.8.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DISASSOCIATE.request(
PeerSTAAddress,
ReasonCode,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the disassociation process. |
| ReasonCode | As defined in frame format | As defined in frame format | Specifies the reason for initiating the disassociation procedure. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.8.1.3 When generated

This primitive is generated by the SME for a STA to establish disassociation with an AP.

##### 10.3.8.1.4 Effect of receipt

This primitive initiates a disassociation procedure. The MLME subsequently issues an MLME-DISASSOCIATE.confirm that reflects the results.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.8.2 MLME-DISASSOCIATE.confirm

#### 10.3.8.2.1 Function

This primitive reports the results of a disassociation procedure with a specific peer MAC entity that is within an AP.

#### 10.3.8.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.confirm(
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, TIMEOUT, REFUSED | Indicates the result of the MLME-DISASSOCIATE.request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.8.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-DISASSOCIATE.request to disassociate with a specified peer MAC entity that is within an AP.

#### 10.3.8.2.4 Effect of receipt

The SME is notified of the results of the disassociation procedure.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.8.3 MLME-DISASSOCIATE.indication

#### 10.3.8.3.1 Function

This primitive reports disassociation with a specific peer MAC entity.

#### 10.3.8.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DISASSOCIATE.indication(
                                    PeerSTAAddress,
                                    ReasonCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which the association relationship was invalidated. |
| ReasonCode | As defined in frame format | As defined in frame format | Specifies the reason the disassociation procedure was initiated. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.8.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an association relationship with a specific peer MAC entity.

#### 10.3.8.3.4 Effect of receipt

The SME is notified of the invalidation of the specific association relationship.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 387 of 1232

### 10.3.9 Reset

This mechanism supports the process of resetting the MAC.

#### 10.3.9.1 MLME-RESET.request

##### 10.3.9.1.1 Function

This primitive requests that the MAC entity be reset.

##### 10.3.9.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.request(
                        STAAddress,
                        SetDefaultMIB
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| STAAddress | MACAddress | Any valid individual MAC address | Specifies the MAC address that is to be used by the MAC entity that is being reset. This value can be used to provide a locally administered STA address. |
| SetDefaultMIB | Boolean | True, false | If true, all MIB attributes are set to their default values. The default values are implementation dependent.<br>If false, the MAC is reset, but all MIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive. |

##### 10.3.9.1.3 When generated

This primitive is generated by the SME to reset the MAC to initial conditions. The MLME-RESET.request primitive must be used prior to use of the MLME-START.request primitive.

##### 10.3.9.1.4 Effect of receipt

This primitive sets the MAC to initial conditions, clearing all internal variables to the default values. MIB attributes can be reset to their implementation-dependent default values by setting the SetDefaultMIB flag to true. The MLME subsequently issues a MLME-RESET.confirm that reflects the results.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.9.2 MLME-RESET.confirm

##### 10.3.9.2.1 Function

This primitive reports the results of a reset procedure.

##### 10.3.9.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-RESET.confirm(
                        ResultCode
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS | Indicates the result of the MLME-RESET.request. |

##### 10.3.9.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-RESET.request to reset the MAC entity.

##### 10.3.9.2.4 Effect of receipt

The SME is notified of the results of the reset procedure.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.10 Start

This mechanism supports the process of creating a new BSS.

#### 10.3.10.1 MLME-START.request

##### 10.3.10.1.1 Function

This primitive requests that the MAC entity start a new BSS.

##### 10.3.10.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-START.request(
SSID,
BSSType,
BeaconPeriod,
DTIMPeriod,
CF parameter set,
PHY parameter set,
IBSS parameter set,
ProbeDelay.
CapabilityInformation,
BSSBasicRateSet,
OperationalRateSet,
Country,
IBSS DFS Recovery Interval,
EDCAParameterSet,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| SSID | Octet string | 1–32 octets | The SSID of the BSS. |
| BSSType | Enumeration | INFRASTRUCTURE, INDEPENDENT | The type of the BSS. |
| Beacon Period | Integer | ≥ 1 | The Beacon period of the BSS (in TU). |
| DTIM Period | Integer | As defined in 7.3.2.6 | The DTIM Period of the BSS (in beacon periods). |
| CF parameter set | As defined in frame format | As defined in 7.3.2.5 | The parameter set for CF periods, if the BSS supports CF mode. aCFPPeriod is modified as a side effect of the issuance of an MLME-START.request primitive. |
| PHY parameter sets | As defined in frame format | As defined in 7.3.2.3 or 7.3.2.4 | The parameter sets relevant to the PHY. |
| IBSS parameter set | As defined in frame format | As defined in 7.3.2.7 | The parameter set for the IBSS, if BSS is an IBSS. |
| ProbeDelay | Integer | N/A | Delay (in microseconds) to be used prior to transmitting a Probe frame during active scanning. |
| CapabilityInformation | As defined in frame format | As defined in 7.3.1.4 | The capabilities to be advertised for the BSS. |


Copyright © 2007 IEEE. All rights reserved.

| Name | Type | Valid range | Description |
|---|---|---|---|
| BSSBasicRateSet | Set of integers | 1–127 inclusive (for each integer in the set) | The set of data rates that must be supported by all STAs to join this BSS. The STA that is creating the BSS must be able to receive and transmit at each of the data rates listed in the set. |
| OperationalRateSet | Set of integers | 1–127 inclusive (for each integer in the set) | The set of data rates that the STA desires to use for communication within the BSS. The STA must be able to receive at each of the data rates listed in the set. This set is a superset of the rates contained in the BSSBasicRateSet parameter. |
| Country | As defined in the Country element | As defined in the Country element | The information required to identify the regulatory domain in which the STA is located and to configure its PHY for operation in that regulatory domain.<br>Present only when TPC functionality is required, as specified in 11.8 or when dot11MultiDomainCapabilityEnabled is true. |
| IBSS DFS Recovery Interval | Integer | 1–255 | Present only if BSSType = INDEPENDENT. The time interval that is used for DFS recovery.<br>Present only when DFS functionality is required, as specified in 11.9. |
| EDCAParameterSet (QoS only) | As defined in frame format | As defined in frame format | The initial EDCA parameter set values to be used in the BSS. The parameter shall be present only if the MIB attribute dot11QosOptionImplemented is true. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.10.1.3 When generated

This primitive is generated by the SME to start either an infrastructure BSS (with the MAC entity within an AP) or an IBSS (with the MAC entity acting as the first STA in the IBSS).

The MLME-START.request primitive must be generated after an MLME-RESET.request primitive has been used to reset the MAC entity and before an MLME-JOIN.request primitive has been used to successfully join an existing infrastructure BSS or IBSS.

The MLME-START.request primitive must not be used after successful use of the MLME-START.request primitive or successful use of the MLME-JOIN.request without generating an intervening MLME-RESET.request primitive.

#### 10.3.10.1.4 Effect of receipt

This primitive initiates the BSS initialization procedure once the current frame exchange sequence is complete. The MLME subsequently issues an MLME-START.confirm that reflects the results of the creation procedure.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.10.2 MLME-START.confirm

##### 10.3.10.2.1 Function

This primitive reports the results of a BSS creation procedure.

##### 10.3.10.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-START.confirm(
ResultCode,
VendorSpecificInfo
)

| Name | Type | Valid range | resetDescription |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, BSS_ALREADY_STARTED_OR_JOINED, RESET_REQUIRED_BEFORE_ START, NOT_SUPPORTED | Indicates the result of the MLME-START.request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.10.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-START.request to create a new BSS.

##### 10.3.10.2.4 Effect of receipt

The SME is notified of the results of the BSS creation procedure.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.11 Spectrum management protocol layer model

The layer management extensions for measurement and channel switching assume a certain partition of spectrum management functionality between the MLME and SME. This partitioning assumes that policy decisions (e.g., regarding measurement and channel switching) reside in the SME, while the protocol for measurement, switch timing, and the associated frame exchanges resides within the MLME (see Figure 10-2).

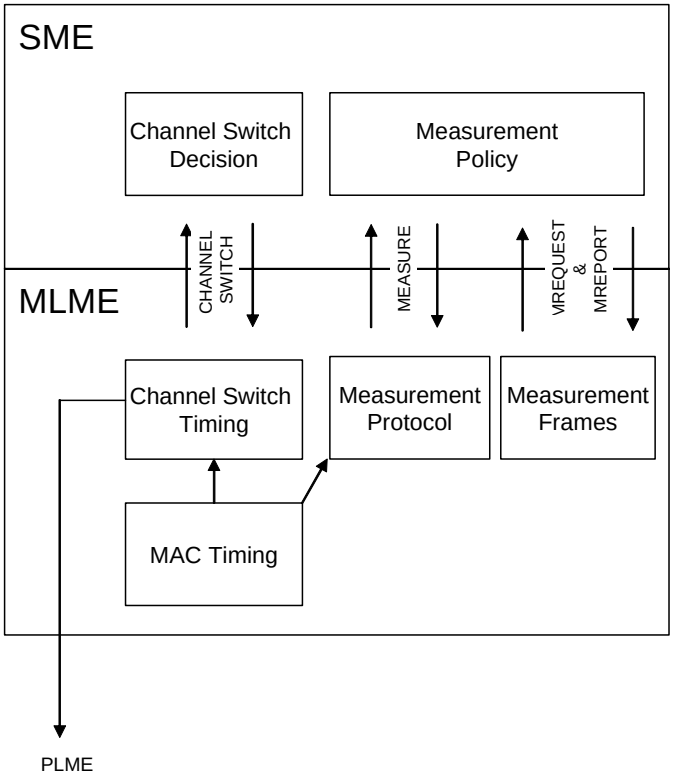



**Figure 10-2—Layer management model**

The informative diagrams within this subclause further illustrate the spectrum management protocol model adopted. Figure 10-3 and Figure 10-4 depict the measurement process for a peer STA to accept and reject a measurement request, respectively. Figure 10-5 illustrates the TPC adaptation process. Lastly, Figure 10-6 depicts the management process for a channel switch using a Channel Switch Announcement frame.

Copyright © 2007 IEEE. All rights reserved.

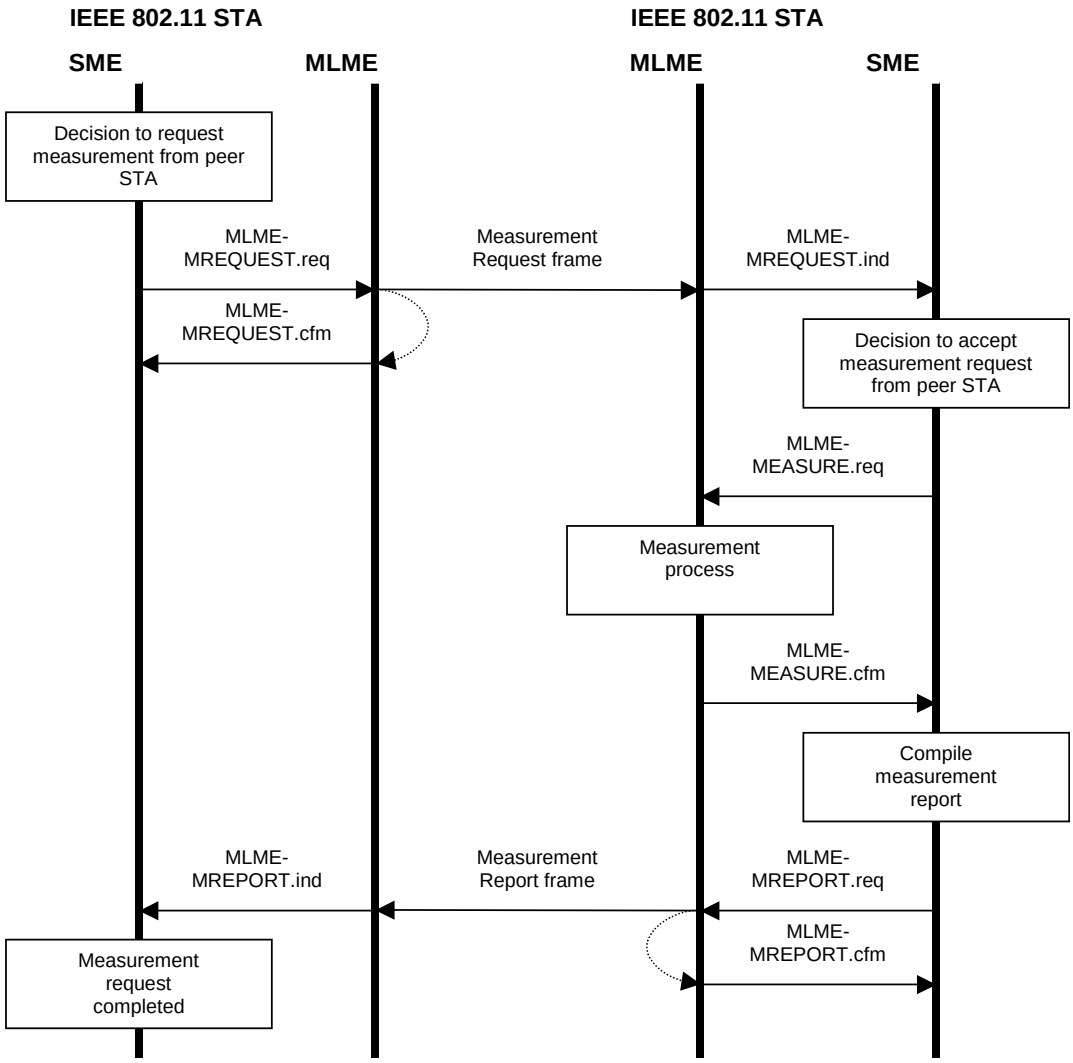



**Figure 10-3—Measurement request—accepted**


Copyright © 2007 IEEE. All rights reserved.

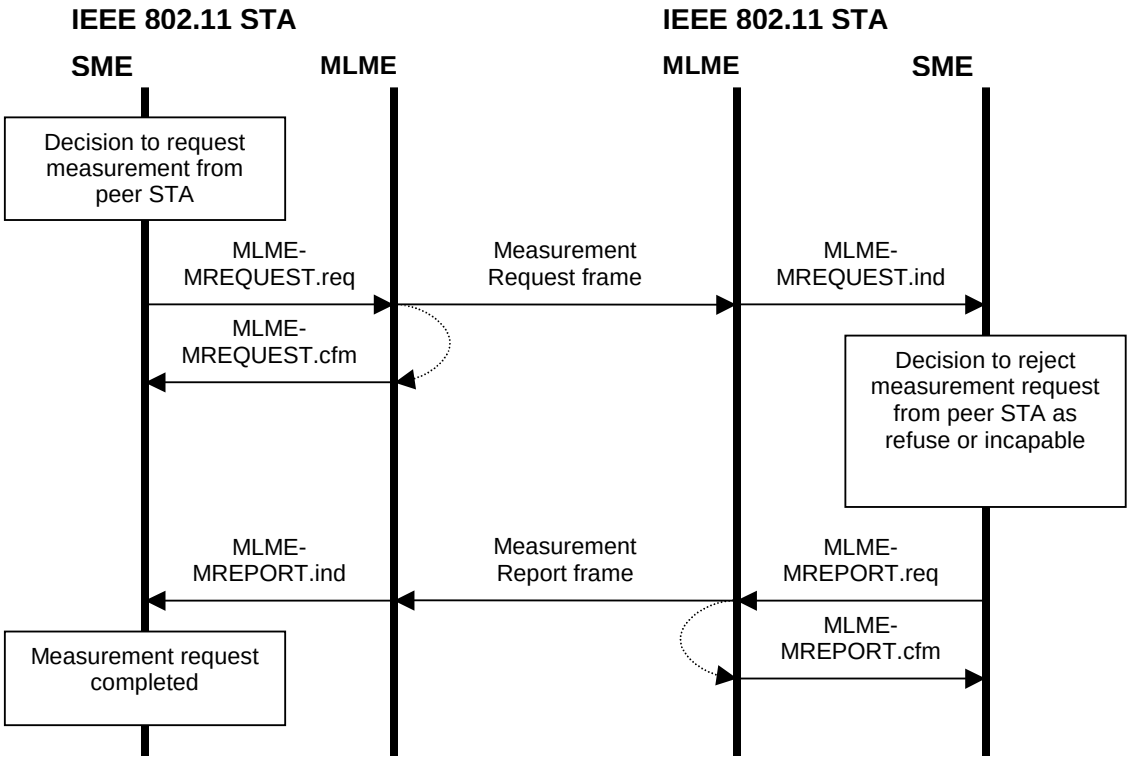



**Figure 10-4—Measurement request—rejected**

Copyright © 2007 IEEE. All rights reserved.

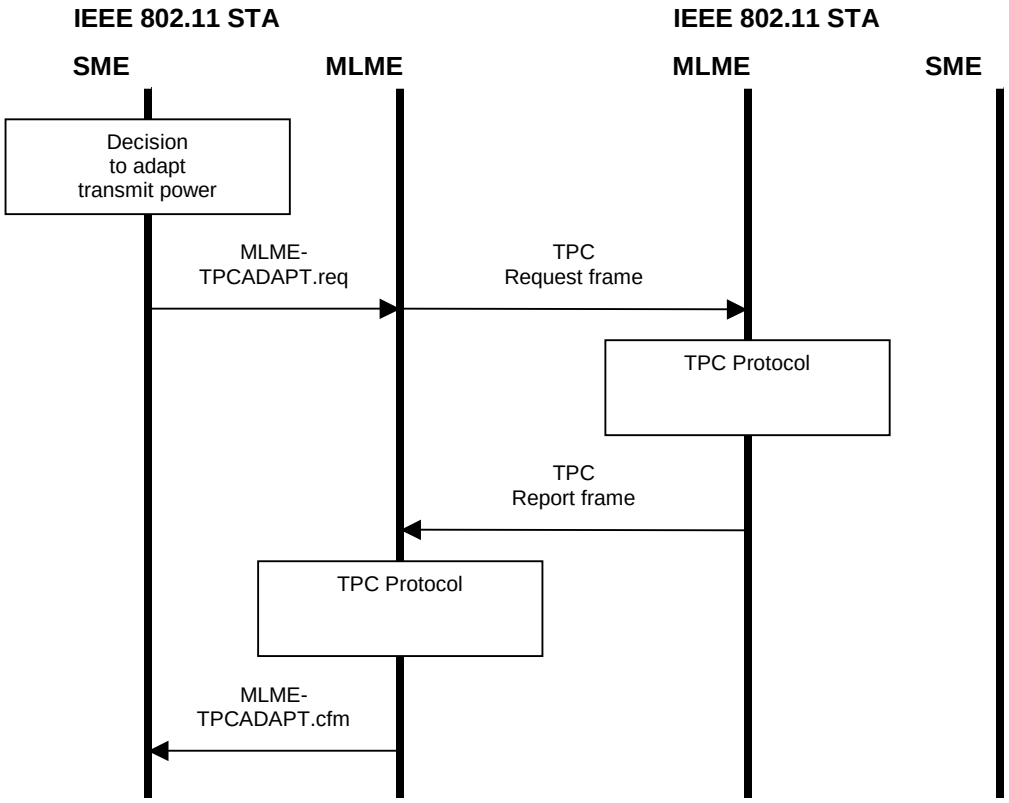



**Figure 10-5—TPC adaptation**


Copyright © 2007 IEEE. All rights reserved.

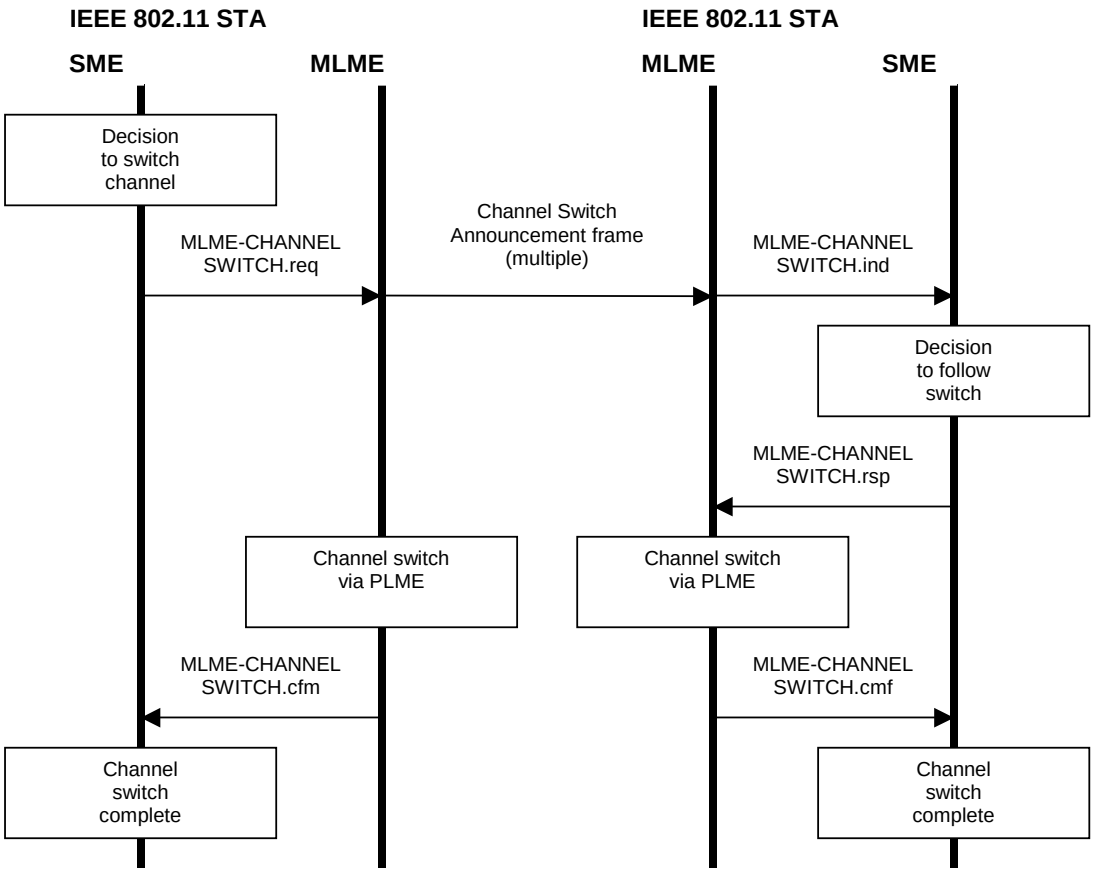



**Figure 10-6—Channel switch**

Copyright © 2007 IEEE. All rights reserved.

### 10.3.12 Measurement request

This set of primitives supports the signaling of measurement requests between peer SMEs.

#### 10.3.12.1 MLME-MREQUEST.request

##### 10.3.12.1.1 Function

This primitive requests the transmission of a measurement request to a peer entity.

##### 10.3.12.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREQUEST.request(
                        Peer MAC Address,
                        Dialog Token,
                        Measurement Request Set,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual or group MAC address | The address of the peer MAC entity to which the measurement request is set. |
| Dialog Token | Integer | 1–255 | The dialog token to identify the measurement transaction. |
| Measurement Request Set | Set of measurement requests, each as defined in the Measurement Request element | Set of measurement requests, each as defined in the Measurement Request element | A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.12.1.3 When generated

This primitive is generated by the SME to request that a Measurement Request frame be sent to a peer entity to initiate one or more measurements.

##### 10.3.12.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Measurement Request frame containing the set of Measurement Request elements specified. This frame is then scheduled for transmission.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.12.2 MLME-MREQUEST.confirm

#### 10.3.12.2.1 Function

This primitive reports the result of a request to send a Measurement Request frame.

#### 10.3.12.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREQUEST.confirm(
                        ResultCode,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, or UNSPECIFIED_FAILURE | Reports the outcome of a request to send a Measurement Request frame. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.12.2.3 When generated

This primitive is generated by the MLME when the request to transmit a Measurement Request frame completes.

#### 10.3.12.2.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result code.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.12.3 MLME-MREQUEST.indication

#### 10.3.12.3.1 Function

This primitive indicates that a Measurement Request frame has been received requesting the measurement of one or more channels.

#### 10.3.12.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MREQUEST.indication(
Peer MAC Address,
Dialog Token,
Measurement Request Set,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual MAC address | The address of the peer MAC entity from which the measurement request was received. |
| Dialog Token | Integer | 1–255 | The dialog token to identify the measurement transaction. |
| Measurement Request Set | Set of measurement requests, each as defined in the Measurement Request element | Set of measurement requests, each as defined in the Measurement Request element | A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.12.3.3 When generated

This primitive is generated by the MLME when a valid Measurement Request frame is received.

#### 10.3.12.3.4 Effect of receipt

On receipt of this primitive, the SME either rejects the request or commences the requested measurements.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.13 Channel measurement

This set of primitives supports the requesting and reporting of measurement data.

#### 10.3.13.1 MLME-MEASURE.request

##### 10.3.13.1.1 Function

This primitive is generated by the SME to request that the MLME initiate specified measurements.

##### 10.3.13.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MEASURE.request(
                        Dialog Token,
                        Measurement Request Set,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Dialog Token | Integer | 0–255 | The Dialog Token to identify the measurement transaction. |
| Measurement Request Set | Set of measurement requests, each as defined in the Measurement Request element | Set of measurement requests, each as defined in the Measurement Request element | A set of measurement requests, each containing a Measurement Token, Measurement Request Mode, Measurement Type, and Measurement Request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.13.1.3 When generated

This primitive is generated by the SME to request that the MLME initiate the specified measurements.

##### 10.3.13.1.4 Effect of receipt

On receipt of this primitive, the MLME commences the measurement process.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.13.2 MLME-MEASURE.confirm

##### 10.3.13.2.1 Function

This primitive reports the result of a measurement.

##### 10.3.13.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MEASURE.confirm(
                        ResultCode,
                        Dialog Token,
                        Measurement Report Set,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, or UNSPECIFIED_FAILURE | The outcome of the measurement request. |
| Dialog Token | Integer | 0–255 | The dialog token to identify the measurement transaction. |
| Measurement Report Set | Set of measurement reports, each as defined in the Measurement Report element | Set of measurement reports, each as defined in the Measurement Report element | A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.13.2.3 When generated

This primitive is generated by the MLME to report the results when a measurement set completes.

##### 10.3.13.2.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result code and, if appropriate, stores the channel measurements pending communication to the requesting entity or for local use.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.14 Measurement report

This set of primitives supports the signaling of measurement reports.

#### 10.3.14.1 MLME-MREPORT.request

##### 10.3.14.1.1 Function

This primitive supports the signaling of measurement reports between peer SMEs.

##### 10.3.14.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREPORT.request(
                        Peer MAC Address,
                        Dialog Token,
                        Measurement Report Set,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual MAC address | The address of the peer MAC entity to which the measurement report is set. |
| Dialog Token | Integer | 0–255 | The dialog token to identify the measurement transaction. Set to 0 for an autonomous report. |
| Measurement Report Set | Set of measurement reports, each as defined in the Measurement Report element | Set of measurement reports, each as defined in the Measurement Report element | A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.14.1.3 When generated

This primitive is generated by the SME to request that a frame be sent to a peer entity to report the results of measuring one or more channels.

##### 10.3.14.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Measurement Report frame containing the set of measurement reports. This frame isthen scheduled for transmission.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.14.2 MLME-MREPORT.confirm

##### 10.3.14.2.1 Function

This primitive reports the result of a request to send a Measurement Report frame.

##### 10.3.14.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-MREPORT.confirm(
                        ResultCode,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS,<br>INVALID PARAMETERS,<br>or<br>UNSPECIFIED FAILURE | Reports the outcome of a request to send a Measurement Request frame. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.14.2.3 When generated

This primitive is generated by the MLME when the request to transmit a Measurement Report frame completes.

##### 10.3.14.2.4 Effect of receipt

On receipt of this primitive, the SME evaluates the result code.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.14.3 MLME-MREPORT.indication

#### 10.3.14.3.1 Function

This primitive indicates that a Measurement Report frame has been received from a peer entity. This management report can be in response to an earlier measurement request (e.g., MLME-MREQUEST.request) or can be an autonomous report.

#### 10.3.14.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MREPORT.indication(
Peer MAC Address,
Dialog Token,
Measurement Report Set,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual MAC address | The address of the peer MAC entity from which the Measurement Report frame was received. |
| Dialog Token | Integer | 0–255 | The dialog token to identify the measurement transaction. Set to 0 for an autonomous report. |
| Measurement Report Set | Set of measurement reports, each as defined in the Measurement Report element | Set of measurement reports, each as defined in the Measurement Report element | A set of measurement reports, each containing a Measurement Token, Measurement Report Mode, Measurement Type, and Measurement Report. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.14.3.3 When generated

This primitive is generated by the MLME when a valid Measurement Report frame is received.

#### 10.3.14.3.4 Effect of receipt

On receipt of this primitive, measurement data can be available for SME processes, such as channel selection.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.15 Channel switch

#### 10.3.15.1 MLME-CHANNELSWITCH.request

##### 10.3.15.1.1 Function

This primitive requests a switch to a new operating channel.

##### 10.3.15.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-CHANNELSWITCH.request(
Mode,
Channel Number,
Channel Switch Count,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Mode | Integer | 0, 1 | Channel switch mode, as defined for the Channel Switch Announcement element. |
| Channel Number | Integer | As defined in 17.3.8.3.3 | Specifies the new channel number. |
| Channel Switch Count | Integer | 0–255 | Specifies the number of TBTTs until the channel switch event, as described for the Channel Switch Announcement element. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.15.1.3 When generated

This primitive is generated by the SME to schedule a channel switch and announce this switch to peer entities in the BSS.

##### 10.3.15.1.4 Effect of receipt

On receipt of this primitive, the MLME schedules the channel switch event and announces this switch to other STAs in the BSS using the Channel Switch Announcement frame or element. The MLME ensures the timing of frame transmission takes into account the activation delay. The actual channel switch can be achieved at the appropriate time through the MLME-PLME interface using the PLME-SET primitive of the dot11CurrentFrequency MIB attribute.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.15.2 MLME-CHANNELSWITCH.confirm

#### 10.3.15.2.1 Function

This primitive reports the result of a request to switch channel.

#### 10.3.15.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.confirm(
                                    ResultCode,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS,<br>INVALID PARAMETERS,<br>or<br>UNSPECIFIED FAILURE | Reports the result of a channel switch request. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.15.2.3 When generated

This primitive is generated by the MLME when a channel switch request completes. Possible unspecified failure causes include an inability to schedule a channel switch announcement.

#### 10.3.15.2.4 Effect of receipt

The SME is notified of the results of the channel switch procedure.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.15.3 MLME-CHANNELSWITCH.indication

##### 10.3.15.3.1 Function

This primitive indicates that a channel switch announcement has been received from a peer entity.

##### 10.3.15.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-CHANNELSWITCH.indication(
                                    Peer MAC Address,
                                    Mode,
                                    Channel Number,
                                    Channel Switch Count,
                                    VendorSpecificInfo
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual MAC address | The address of the peer MAC entity from which the Measurement Report frame was received. |
| Mode | Integer | 0, 1 | Channel switch mode, as defined for the Channel Switch Announcement element. |
| Channel Number | Integer | As defined in 17.3.8.3.3 | Specifies the new channel number. |
| Channel Switch Count | Integer | 0–255 | Specifies the number of TBTTs until the channel switch event, as described for the Channel Switch Announcement element. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.15.3.3 When generated

This primitive is generated by the MLME when a valid Channel Switch Announcement frame is received.

##### 10.3.15.3.4 Effect of receipt

On receipt of this primitive, the SME decides whether to accept the switch.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.15.4 MLME-CHANNELSWITCH.response

##### 10.3.15.4.1 Function

This primitive is used to schedule an accepted channel switch.

##### 10.3.15.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-CHANNELSWITCH.response(
Mode,
Channel Number,
Channel Switch Count,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Mode | Integer | 0, 1 | Channel switch mode, as defined for the Channel Switch Announcement element. |
| Channel Number | Integer | As defined in 17.3.8.3.3 | Specifies the new channel number. |
| Channel Switch Count | Integer | 0–255 | Specifies the number of TBTTs until the channel switch event. as described for the Channel Switch Announcement element. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.15.4.3 When generated

This primitive is generated by the SME to schedule an accepted channel switch request.

##### 10.3.15.4.4 Effect of receipt

On receipt of this primitive, the MLME schedules the channel switch. The actual channel switch can be achieved at the appropriate time through the MLME-PLME interface using the PLME-SET primitive of the dot11CurrentFrequency MIB attribute.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.16 TPC request

This set of primitives supports the adaptation of transmit power between peer entities as described in 11.8.4.

#### 10.3.16.1 MLME-TPCADAPT.request

##### 10.3.16.1.1 Function

This primitive supports the adaptation of transmit power between peer entities as specified in 11.8.4.

##### 10.3.16.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-TPCADAPT.request(
Peer MAC Address,
Dialog Token,
VendorSpecificInfo
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Peer MAC Address | MACAddress | Any valid individual or group MAC address | The address of the peer MAC entity to which the TPC request is set. |
| Dialog Token | Integer | 1–255 | The dialog token to identify the TPC transaction. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.16.1.3 When generated

This primitive is generated by the SME to request that a TPC Request frame be sent to a peer entity to request that entity to report transmit power and link margin information.

##### 10.3.16.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a TPC Request frame. This frame is then scheduled for transmission.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.16.2 MLME-TPCADAPT.confirm

#### 10.3.16.2.1 Function

This primitive reports the result of the TPC adaptation procedure.

#### 10.3.16.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-TPCADAPT.confirm(
                        ResultCode
                        Transmit Power,
                        Link Margin,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS,<br>INVALID PARAMETERS,<br>or<br>UNSPECIFIED FAILURE | Reports the outcome of a request to send a TPC Request frame. |
| Transmit Power | Integer | –127 to 127 | Value of the Transmit Power field of the TPC Report element of the TPC Report frame. |
| Link Margin | Integer | –127 to 127 | Value of the Link Margin field of the TPC Report element of the TPC Report frame. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.16.2.3 When generated

This primitive is generated by the MLME when the TPC adaptation procedure completes.

#### 10.3.16.2.4 Effect of receipt

The SME is notified of the results of the TPC adaptation procedure.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.17 SetKeys

#### 10.3.17.1 MLME-SETKEYS.request

##### 10.3.17.1.1 Function

This primitive causes the keys identified in the parameters of the primitive to be set in the MAC and enabled for use.

##### 10.3.17.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SETKEYS.request(
Keylist
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Keylist | A set of SetKeyDescriptors | N/A | The list of keys to be used by the MAC. |

Each SetKeyDescriptor consists of the following elements:

| Name | Type | Valid range | Description |
|---|---|---|---|
| Key | Bit string | N/A | The temporal key value |
| Length | Integer | N/A | The number of bits in the Key to be used. |
| Key ID | Integer | 0–3 | Key identifier |
| Key Type | Integer | Group, Pairwise, PeerKey | Defines whether this key is a group key, pairwise key, or PeerKey. |
| Address | MACAddress | Any valid individual MAC address | This parameter is valid only when the Key Type value is Pairwise, when the Key Type value is Group and the STA is in IBSS, or when the Key Type value is PeerKey. |
| Receive Sequence Count | 8 octets | N/A | Value to which the RSC(s) is initialized |
| Authenticator/Supplicant or Initiator/Peer | Boolean | True, false | Whether the key is configured by the Authenticator or Supplicant; true indicates Authenticator or Initiator. |
| Cipher Suite Selector | 4 octets | As defined in the RSN information element format | The cipher suite required for this association. |

##### 10.3.17.1.3 When generated

This primitive is generated by the SME at any time when one or more keys are to be set in the MAC.

##### 10.3.17.1.4 Effect of receipt

Receipt of this primitive causes the MAC to set the appropriate keys and to begin using them for future MA-UNITDATA.request and MA-UNITDATA.indication primitives provided the MLME-SETPROTECTION.request primitive has been issued.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.17.2 MLME-SETKEYS.confirm

##### 10.3.17.2.1 Function

This primitive confirms that the action of the associated MLME-SETKEYS.request primitive has been completed.

##### 10.3.17.2.2 Semantics of the service primitive

This primitive has no parameters.

##### 10.3.17.2.3 When generated

This primitive is generated by the MAC in response to receipt of a MLME-SETKEYS.request primitive. This primitive is issued when the action requested has been completed.

##### 10.3.17.2.4 Effect of receipt

The SME is notified that the requested action of the MLME-SETKEYS.request primitive is completed.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.18 DeleteKeys

#### 10.3.18.1 MLME-DELETEKEYS.request

##### 10.3.18.1.1 Function

This primitive causes the keys identified in the parameters of the primitive to be deleted from the MAC and thus disabled for use.

##### 10.3.18.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-DELETEKEYS.request(
Keylist
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Keylist | A set of DeleteKeyDescriptors | N/A | The list of keys to be deleted from the MAC. |

Each DeleteKeyDescriptor consists of the following elements:

| Name | Type | Valid range | Description |
|---|---|---|---|
| Key ID | Integer | N/A | Key identifier. |
| Key Type | Integer | Group, Pairwise, PeerKey | Defines whether this key is a group key, pairwise key, or PeerKey. |
| Address | MACAddress | Any valid individual MAC address | This parameter is valid only when the Key Type value is Pairwise, or when the Key Type value is Group and is from an IBSS STA, or when the Key Type value is PeerKey. |

##### 10.3.18.1.3 When generated

This primitive is generated by the SME at any time when keys for a security association are to be deleted in the MAC.

##### 10.3.18.1.4 Effect of receipt

Receipt of this primitive causes the MAC to delete the temporal keys identified by the Keylist Address, including Group, Pairwise and PeerKey, and to cease using them.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.18.2 MLME-DELETEKEYS.confirm

##### 10.3.18.2.1 Function

This primitive confirms that the action of the associated MLME-DELETEKEYS.request primitive has been completed.

##### 10.3.18.2.2 Semantics of the service primitive

This primitive has no parameters.

##### 10.3.18.2.3 When generated

This primitive is generated by the MAC in response to receipt of a MLME-DELETEKEYS.request primitive. This primitive is issued when the action requested has been completed.

##### 10.3.18.2.4 Effect of receipt

The SME is notified that the requested action of the MLME-DELETEKEYS.request primitive is completed.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.19 MIC (Michael) failure event

#### 10.3.19.1 MLME-MICHAELMICFAILURE.indication

##### 10.3.19.1.1 Function

This primitive reports that a MIC failure event was detected.

##### 10.3.19.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MICHAELMICFAILURE.indication (
Count,
Address,
Key Type,
Key ID,
TSC
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Count | Integer | 1 or 2 | The current number of MIC failure events. |
| Address | MACAddress | Any valid individual MAC address | The source MAC address of the frame. |
| Key Type | Integer | Group, Pairwise, PeerKey | The key type that the receive frame used. |
| Key ID | Integer | 0–3 | Key identifier. |
| TSC | 6 octets | N/A | The TSC value of the frame that generated the MIC failure. |

##### 10.3.19.1.3 When generated

This primitive is generated by the MAC when it has detected a MIC failure.

##### 10.3.19.1.4 Effect of receipt

The SME is notified that the MAC has detected a MIC failure.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.20 EAPOL

#### 10.3.20.1 MLME-EAPOL.request

##### 10.3.20.1.1 Function

This primitive is generated by the SME when the SME has an IEEE 802.1X EAPOL-Key frame to send.

##### 10.3.20.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EAPOL.request (
                    Source Address,
                    Destination Address,
                    Data
                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Source Address | MACAddress | N/A | The MAC sublayer address from which the EAPOL-Key frame is being sent. |
| Destination Address | MACAddress | N/A | The MAC sublayer entity address to which the EAPOL-Key frame is being sent. |
| Data | IEEE 802.1X EAPOL-Key frame | N/A | The EAPOL-Key frame to be transmitted. |

##### 10.3.20.1.3 When generated

This primitive is generated by the SME when the SME has a 802.1X EAPOL-Key frame to send.

##### 10.3.20.1.4 Effect of receipt

The MAC sends this EAPOL-Key frame.

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.20.2 MLME-EAPOL.confirm

##### 10.3.20.2.1 Function

This primitive indicates that this EAPOL-Key frame has been acknowledged by the IEEE 802.11 MAC.

##### 10.3.20.2.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-EAPOL.confirm (
                    ResultCode
                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, TIMEOUT | Indicates whether the EAPOL-Key frame has been acknowledged by the target STA. |

##### 10.3.20.2.3 When generated

This primitive is generated by the MAC as a result of an MLME-EAPOL.request being generated to send an EAPOL-Key frame.

##### 10.3.20.2.4 Effect of receipt

The SME is always notified whether this EAPOL-Key frame has been acknowledged by the IEEE 802.11 MAC.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.21 MLME-PeerKeySTART

#### 10.3.21.1 MLME- PeerKeySTART.request

##### 10.3.21.1.1 Function

This primitive is generated by the SME when the SME wants to start a PeerKey Handshake with a peer.

##### 10.3.21.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-PeerKeySTART.request (
                                    PeerSTAAddress,
                                    RSN
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the PeerKey Handshake process. |
| RSN | RSN information element | As defined in frame format | A description of the cipher suites supported by initiator STA. |

##### 10.3.21.1.3 When generated

This primitive is generated by the SME for a STA to initiate a PeerKey Handshake with a specified peer MAC entity in order to create a secure link between the two STAs.

##### 10.3.21.1.4 Effect of receipt

This primitive initiates the SMK Handshake as part of PeerKey Handshake by sending an EAPOL-Key message.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.22 SetProtection

#### 10.3.22.1 MLME-SETPROTECTION.request

##### 10.3.22.1.1 Function

This primitive indicates whether protection is required for frames sent to and received from the indicated MAC address.

##### 10.3.22.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SETPROTECTION.request( Protectlist )

| Name | Type | Valid range | Description |
|---|---|---|---|
| Protectlist | A set of protection elements | N/A | The list of how each key is being used currently. |

Each Protectlist consists of the following elements:

| Name | Type | Valid range | Description |
|---|---|---|---|
| Address | MACAddress | Any valid individual MAC address | This parameter is valid only when the Key Type value is Pairwise or when the Key Type value is Group and is from an IBSS STA or PeerKey. |
| ProtectType | Enumeration | None, Rx, Tx, Rx_Tx | The protection value for this MAC. |
| Key Type | Integer | Group, Pairwise, or PeerKey | Defines whether this key is a group key, pairwise key, or PeerKey. |

##### 10.3.22.1.3 When generated

This primitive is generated by the SME when protection is required for frames sent to and received from the indicated MAC address.

##### 10.3.22.1.4 Effect of receipt

Receipt of this primitive causes the MAC to set the protection and to protect data frames as indicated in the ProtectType element of the Protectlist parameter:

— None: Specifies that data frames neither from the MAC address nor to the MAC address is protected.

— Rx: Specifies that data frames from MAC address is protected.

— Tx: Specifies that data frames to MAC address is protected.

— Rx_Tx: Specifies that data frames to and from MAC address is protected.

Once it is specified that a data frame is protected to or from a MAC address, this is reset by the MLME-SETPROTECTION.request primitive. The MLME-SETPROTECTION.request primitive deletes the state by specifying None.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.22.2 MLME-SETPROTECTION.confirm

##### 10.3.22.2.1 Function

This primitive indicates that the frame protection request is completed.

##### 10.3.22.2.2 Semantics of the service primitive

There are no parameters for this primitive.

##### 10.3.22.2.3 When generated

This primitive is generated by the MAC when the protection request is complete.

##### 10.3.22.2.4 Effect of receipt

The SME is notified that the protection request is complete.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.23 MLME-PROTECTEDFRAMEDROPPED

#### 10.3.23.1 MLME- PROTECTEDFRAMEDROPPED.indication

##### 10.3.23.1.1 Function

This primitive notifies the SME that a frame has been dropped because a temporal key was unavailable.

##### 10.3.23.1.2 Semantics of the service primitive

This primitive has two parameters, the MAC addresses of the two STAs.

The primitive parameters are as follows:

MLME-PROTECTEDFRAMEDROPPED.indication (
Address1,
Address2
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Address1 | MACAddress | Any valid individual MAC address | MAC address of SA. |
| Address2 | MACAddress | Any valid individual MAC address | MAC address of RA. |

##### 10.3.23.1.3 When generated

This primitive is generated by the MAC when a frame is dropped because no temporal key is available for the frame.

##### 10.3.23.1.4 Effect of receipt

The SME is notified that a frame was dropped. The SME can use this information in an IBSS to initiate a security association to the peer STA.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.24 TS management interface

This mechanism supports the process of adding, modifying, or deleting a TS in a BSS 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 10-1 defines which primitives are supported by which type of STA.

**Table 10-1—Supported TS Management primitives**

| Primitive | Request | Confirm | Indication | Response |
|---|---|---|---|---|
| ADDTS | non-AP QoS STA | non-AP QoS STA | HC | HC |
| DELTS | non-AP QoS STA and HC | non-AP QoS STA and HC | non-AP QoS STA 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 STA 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,
                    VendorSpecificInfo
                    )
```

Copyright © 2007 IEEE. All rights reserved.

| 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 10.3.24.1.3 When generated

This primitive is generated by the SME at a non-AP STA 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 STA to initiate such an addition (or modification).

#### 10.3.24.1.4 Effect of receipt

The non-AP STA operates according to the procedures defined in 11.4.


Copyright © 2007 IEEE. All rights reserved.

### 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,
VendorSpecificInfo
)

| 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 STA 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

Copyright © 2007 IEEE. All rights reserved.

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 STA, it is the responsibility of the non-AP STA 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 STA 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.


Copyright © 2007 IEEE. All rights reserved.

### 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-APSTAAddress
                        TSPEC,
                        TCLAS,
                        TCLASProcessing,
                        VendorSpecificInfo
                        )
```

| 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-APSTAAddress | MACAddress | | Contains the MAC address of the non-AP STA 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

The TCLAS is optional at the discretion of the non-AP STA 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 STA in the form of an Add TS request in the corresponding QoS Action frame.

Copyright © 2007 IEEE. All rights reserved.

#### 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 STA.

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.


Copyright © 2007 IEEE. All rights reserved.

#### 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 STA MAC entity.

##### 10.3.24.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-ADDTS.response (
ResultCode,
DialogToken,
Non-APSTAAddress,
TSDelay,
TSPEC,
Schedule,
TCLAS,
TCLASProcessing,
VendorSpecificInfo
)

| 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-APSTAAddress | MACAddress | | Contains the non-AP STA address of the matching MLME-ADDTS.indication primitive. |
| TSDelay | Integer | ≥ 0 | When the result code is REJECTED_FOR_DELAY_ PERIOD, provides the amount of time a STA 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. |

Copyright © 2007 IEEE. All rights reserved.

| Name | Type | Valid range | Description |
|---|---|---|---|
| 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

The DialogToken and non-APSTAAddress 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 STA. 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 STA 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 STA containing the specified parameters.


Copyright © 2007 IEEE. All rights reserved.

#### 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 STA or the HC.

##### 10.3.24.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-DELTS.request(
                        Non-APSTAAddress,
                        TSInfo,
                        ReasonCode,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Non-APSTAAddress (HC only) | MACAddress | | (At the HC only) Specifies the MAC address of the non-AP STA that initiated this TS. |
| TSInfo | As defined in frame format | As defined in frame format | Specifies the TS to be deleted. |
| ReasonCode | Enumeration | STA_LEAVING, END_TS, UNKNOWN_TS | Indicates the reason why the TS is being deleted. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 10.3.24.5.3 When generated

This primitive is generated by the SME at a STA to initiate deletion of a TS when a higher layer protocol or mechanism signals the STA 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 STA MAC address. If this primitive was generated at the non-AP STA, 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.

Copyright © 2007 IEEE. All rights reserved.

#### 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-APSTAAddress,
                        TSInfo,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, FAILURE | Indicates the results of the corresponding MLME-DELTS.request primitive. |
| Non-APSTAAddress (HC only) | MACAddress | | (HC only) Contains the non-AP STA 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. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

##### 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.


Copyright © 2007 IEEE. All rights reserved.

### 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-APSTAAddress,
                        TSInfo,
                        ReasonCode,
                        VendorSpecificInfo
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Non-APSTAAddress (HC only) | MACAddress | | (HC only) Specifies the MAC address of the non-AP STA 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 | STA_LEAVING, END_TS, UNKNOWN_TS, TIMEOUT | Indicates the reason why the TS is being deleted. |
| VendorSpecificInfo | A set of information elements | As defined in 7.3.2.26 | Zero or more information elements. |

#### 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 STA 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.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.25 Management of direct links

This subclause 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA 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. |
| DLSResponseTimeout | 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 STA to set up a direct link with another non-AP STA.

##### 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.


Copyright © 2007 IEEE. All rights reserved.

### 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA that is the intended immediate recipient of the data flow. |
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, NOT_ALLOWED, NOT_PRESENT, NOT_QOS_STA, 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.

Copyright © 2007 IEEE. All rights reserved.

### 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA 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.


Copyright © 2007 IEEE. All rights reserved.

#### 10.3.25.4 MLME-DLSTeardown.request

##### 10.3.25.4.1 Function

When initiated by a non-AP STA, this primitive requests the teardown of the direct link with a specified peer MAC entity. When initiated by an AP, 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA that is the intended immediate recipient of the teardown. |
| PeerMACAddress2 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA with which the DLS is being torn down. This parameter is applicable only at the AP. |
| ReasonCode | Enumeration | STA_LEAVING, END_DLS, PEERKEY_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 STA for tearing down a direct link with another non-AP STA.

##### 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.

Copyright © 2007 IEEE. All rights reserved.

#### 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA that is the intended immediate recipient of the teardown. |
| PeerMACAddress2 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA with which the direct link is being torn down. This parameter is applicable only at the AP. |
| 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.


Copyright © 2007 IEEE. All rights reserved.

#### 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 | MACAddress | Any valid individual MAC address | Specifies the MAC address of the non-AP STA that is the sender of the data flow. |
| ReasonCode | Enumeration | STA_LEAVING, END_DLS, PEERKEY_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.

Copyright © 2007 IEEE. All rights reserved.

### 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 standard. 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 | MACAddress | 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.


Copyright © 2007 IEEE. All rights reserved.

### 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.

Copyright © 2007 IEEE. All rights reserved.

### 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 | MACAddress | 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.


Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
DialogToken,
TID,
BlockAckPolicy,
BufferSize,
BlockAckTimeout,
ADDBAFailureTimeout,
BlockAckStartingSequenceControl
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | 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. |

Copyright © 2007 IEEE. All rights reserved.

#### 10.3.27.1.3 When generated

This primitive is generated by the SME at a STA to request initiation (or modification) of Block Ack with the specified peer MAC entity.

#### 10.3.27.1.4 Effect of receipt

The STA sends the ADDBA Request frame to the specified peer MAC entity.


Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
DialogToken,
TID,
ResultCode,
BlockAckPolicy,
BufferSize,
BlockAckTimeout
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | 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 PeerSTAAddress 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).

Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
DialogToken,
TID,
BlockAckPolicy,
BufferSize,
BlockAckTimeout
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | 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 peerSTAAddress 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.


Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
DialogToken,
TID,
ResultCode,
BlockAckPolicy,
BufferSize,
BlockAckTimeout
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | N/A | Specifies the address of the peer MAC entity that attempted the Block Ack initiation (or modification). This value must match the Peer-STAAddress 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.

Copyright © 2007 IEEE. All rights reserved.

### 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(
                    PeerSTAAddress,
                    Direction,
                    TID,
                    ReasonCode
                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | 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 | STA_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 STA to request deletion of Block Ack with the specified peer MAC entity.

#### 10.3.27.5.4 Effect of receipt

The STA sends the DELBA frame to the specified peer MAC entity.


Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
Direction,
TID,
ResultCode,
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | N/A | Specifies the address of the peer MAC entity with which the Block Ack deletion was attempted. This value must match the PeerSTAAddress 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.

Copyright © 2007 IEEE. All rights reserved.

### 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(
PeerSTAAddress,
Direction,
TID,
ReasonCode
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerSTAAddress | MACAddress | 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 | STA_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.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.28 Schedule element management

This subclause 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-APSTAAddress,
                        Schedule
                        )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| Non-APSTAAddress | MACAddress | Any valid individual address | MAC address of the non-AP STA 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 STA, 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 STA when the schedule information for the non-AP STA 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 STA specified in the primitive containing the specified Schedule parameters.

Copyright © 2007 IEEE. All rights reserved.

### 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 STA in the Schedule frame.


Copyright © 2007 IEEE. All rights reserved.

### 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 STA in the form of a Schedule frame. It is valid at the non-AP STA.

#### 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 STA, 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.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.29 Vendor-specific action

This set of primitives supports the signaling of Vendor Specific Action frames between peer SMEs.

#### 10.3.29.1 MLME-VSPECIFIC.request

##### 10.3.29.1.1 Function

This primitive requests transmission of a Vendor Specific Action frame to a peer entity.

##### 10.3.29.1.2 Semantics of the Service Primitive

The primitive parameters are as follows:

MLME-VSPECIFIC.request(
PeerMACAddress,
OUI,
VendorSpecificContent
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerMACAddress | MACAddress | Any valid individual MAC address | The address of the peer MAC entity to which the Vendor Specific Action frame is sent. |
| OUI | 3 octets | 00-00-00 to FF-FF-FF | A public value assigned by the IEEE to identify the entity that has defined the content of the particular vendor-specific action. |
| VendorSpecificContent | Set of information elements and vendor-specific fields | Set of information elements and vendor-specific fields | A set of information elements as defined in 7.3.2. A set of vendor-specific fields specifying the required fields for the Vendor Specific Action frame. |

##### 10.3.29.1.3 When Generated

This primitive is generated by the SME to request that a Vendor Specific Action frame be sent to a peer entity.

##### 10.3.29.1.4 Effect of Receipt

On receipt of this primitive, the MLME constructs a Vendor Specific Action frame containing the set of information elements and vendor-specific fields. The STA then attempts to transmit the frame to the peer entity.


Copyright © 2007 IEEE. All rights reserved.

### 10.3.29.2 MLME-VSPECIFIC.confirm

#### 10.3.29.2.1 Function

This primitive reports the result of a request to send a Vendor Specific Action frame to the peer entity.

#### 10.3.29.2.2 Semantics of the Service Primitive

The primitive parameters are as follows:

```
MLME-VSPECIFIC.confirm(
                                    ResultCode
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| ResultCode | Enumeration | SUCCESS, INVALID_PARAMETERS, TIMEOUT, TRANSMISSION_FAILURE, UNSPECIFIED_FAILURE | Indicates the result of the corresponding MLME-VSPECIFIC.request. |

#### 10.3.29.2.3 When Generated

This primitive is generated by the MLME when the request to transmit a Vendor Specific Action frame completes and indicates the results of the request.

#### 10.3.29.2.4 Effect of Receipt

On receipt of this primitive, the SME evaluates the result code.

Copyright © 2007 IEEE. All rights reserved.

### 10.3.29.3 MLME-VSPECIFIC.indication

#### 10.3.29.3.1 Function

This primitive indicates that a Vendor Specific Action frame has been received from a peer entity.

#### 10.3.29.3.2 Semantics of the Service Primitive

The primitive parameters are as follows:

```
MLME-VSPECIFIC.indication(
                                    PeerMACAddress,
                                    OUI,
                                    VendorSpecificContent
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| PeerMACAddress | MACAddress | Any valid individual MAC address | The address of the peer MAC entity from which the Vendor Specific Action frame was received. |
| OUI | 3 octets | 00-00-00 to FF-FF-FF | A public value assigned by the IEEE to identify the entity that has defined the content of the particular vendor-specific action. |
| VendorSpecificContent | Set of information elements and vendor-specific fields | Set of information elements and vendor-specific fields | A set of information elements as defined in 7.3.2. A set of vendor-specific fields specifying the required fields for the Vendor Specific Action frame. |

#### 10.3.29.3.3 When Generated

This primitive is generated by the MLME when a valid Vendor Specific Action frame is received.

#### 10.3.29.3.4 Effect of Receipt

On receipt of this primitive, the Vendor Specific Content can be made available for SME processes.


Copyright © 2007 IEEE. All rights reserved.

## 10.4 PLME SAP interface

The PHY management service interface consists of the generic PLMEGET and PLMESET primitives on PHY MIB attributes, as described previously, together with the PLME-RESET and PLME-CHARACTERISTICS primitives and the following specific primitives.

### 10.4.1 PLME-RESET.request

#### 10.4.1.1 Function

This primitive is a request by the LME to reset the PHY. The PHY is always reset to the receive state to avoid accidental data transmission.

#### 10.4.1.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PLME-RESET.request( )

This primitive has no parameters.

#### 10.4.1.3 When generated

This primitive is generated at any time to reset the PHY.

#### 10.4.1.4 Effect of receipt

Receipt of this primitive by the PHY causes the PHY entity to reset both the transmit and the receive state machines and places the PHY into the receive state.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 457 of 1232

### 10.4.2 PLME-CHARACTERISTICS.request

#### 10.4.2.1 Function

This primitive is a request by the LME to provide the PHY operational characteristics.

#### 10.4.2.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PLME-CHARACTERISTICS.request( )

This primitive has no parameters.

#### 10.4.2.3 When generated

This primitive is generated by the LME, at initialization time, to request the PHY entity to provide its operational characteristics.

#### 10.4.2.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity will be to generate a PLME-CHARACTERISTICS.confirm primitive that conveys its operational characteristics.


Copyright © 2007 IEEE. All rights reserved.

### 10.4.3 PLME-CHARACTERISTICS.confirm

#### 10.4.3.1 Function

This primitive provides the PHY operational parameters.

#### 10.4.3.2 Semantics of the service primitive

The primitive provides the following parameters:

PLME-CHARACTERISTICS.confirm
(
aSlotTime,
aSIFSTime,
aCCATime,
aPHY-RX-START-Delay,
aRxTxTurnaroundTime,
aTxPLCPDelay,
aRxPLCPDelay,
aRxTxSwitchTime,
aTxRampOnTime,
aTxRampOffTime,
aTxRFDelay,
aRxRFDelay,
aAirPropagationTime,
aMACProcessingDelay,
aPreambleLength,
aPLCPHeaderLength,
aMPDUDurationFactor,
aMPDUMaxLength,
aCWmin,
aCWmax
)

The values assigned to the parameters is as specified in the PLME SAP interface specification contained within each PHY subclass of this standard. The parameter aMPDUDurationFactor is not used by all PHYs defined within this standard.

| Name | Type | Description |
|---|---|---|
| aSlotTime | integer | The Slot Time (in microseconds) that the MAC will use for defining the PIFS and DIFS periods. See 9.2.10. |
| aSIFSTime | integer | The nominal time (in microseconds) that the MAC and PHY will require to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol on the air interface of the earliest possible response frame. See 9.2.10. |
| aCCATime | integer | The minimum time (in microseconds) the CCA mechanism has available to assess the medium within every time slot to determine whether the medium is busy or idle. |
| aPHY-RX-START-Delay | integer | The delay, in microseconds, from a point in time specified by the PHY to the issuance of the PHY-RXSTART.indication primitive. |
| aRxTxTurnaroundTime | integer | The maximum time (in microseconds) that the PHY requires to change from receiving to transmitting the start of the first symbol. The following equation is used to derive the RxTxTurnaroundTime:<br>aTxPLCPDelay + aRxTxSwitchTime + aTxRampOnTime + aTxRFDelay. |
| aTxPLCPDelay | integer | The nominal time (in microseconds) that the PLCP uses to deliver a symbol from the MAC interface to the transmit data path of the physical medium dependent (PMD). |
| aRxPLCPDelay | integer | The nominal time (in microseconds) that the PLCP uses to deliver the last bit of a received frame from the PMD receive path to the MAC. |

Copyright © 2007 IEEE. All rights reserved.

| Name | Type | Description |
|---|---|---|
| aRxTxSwitchTime | integer | The nominal time (in microseconds) that the PMD takes to switch from Receive to Transmit. |
| aTxRampOnTime | integer | The maximum time (in microseconds) that the PMD takes to turn the Transmitter on. |
| aTxRampOffTime | integer | The nominal time (in microseconds) that the PMD takes to turn the Transmit Power Amplifier off. |
| aTxRFDelay | integer | The nominal time (in microseconds) between the issuance of a PMD_DATA.request to the PMD and the start of the corresponding symbol at the air interface. The start of a symbol is defined to be 1/2 symbol period prior to the center of the symbol for FH, or 1/2 chip period prior to the center of the first chip of the symbol for DS, or 1/2 slot time prior to the center of the corresponding slot for infrared (IR). |
| aRxRFDelay | integer | The nominal time (in microseconds) between the end of a symbol at the air interface to the issuance of a PMD_DATA.indicate to the PLCP. The end of a symbol is defined to be 1/2 symbol period after the center of the symbol for FH, or 1/2 chip period after the center of the last chip of the symbol for DS, or 1/2 slot time after the center of the corresponding slot for IR. |
| aAirPropagationTime | integer | Twice the propagation time (in microseconds) for a signal to cross the maximum distance between the most distant allowable STAs that are slot synchronized. |
| aMACProcessingDelay | integer | The maximum time (in microseconds) available for the MAC to issue a PHY-TXSTART.request primitive pursuant to a PHY-RXEND.indication primitive (for response after SIFS) or PHY-CCA.indication(IDLE) primitive (for response at any slot boundary following SIFS). This constraint on MAC performance is defined as a PHY-specific parameter because of its use, along with other PHY-specific time delays, in calculating the two PHY characteristics of primary concern to the MAC: aSlotTime and aSIFSTime. The relationship between aMACProcessingTime and the IFS and slot timing is described in 9.2.10 and illustrated in Figure 9-12. |
| aPreambleLength | integer | The current PHY's preamble length (in microseconds). If the actual value of the length of the modulated preamble is not an integral number of microseconds, the value is rounded up to the next higher value. |
| aPLCPHeaderLength | integer | The current PHY's PLCP header length (in microseconds). If the actual value of the length of the modulated header is not an integral number of microseconds, the value is rounded up to the next higher value. |
| aMPDUDurationFactor | integer | The overhead added by the PHY to the MPDU as it is transmitted through the WM expressed as a scaling factor applied to the number of bits in the MPDU.<br>The value of aMPDUDurationFactor is generated by the following equation:<br>Truncate[((PPDUbits/PSDUbits)–1) × $10^9$)].<br>The total time to transmit a PPDU over the air is generated by the following equation rounded up to the next integer μs:<br>aPreambleLength + aPLCPHeaderLength + ( ( (aMPDUDurationFactor × 8 × PSDUoctets) / $10^9$) + (8 × PSDUoctets) ) / data rate<br>where data rate is in Mb/s.<br>The total time (in μs) to the beginning of any octet in a PPDU from the first symbol of the preamble can be calculated using the duration factor in the following equation:<br>Truncate[aPreambleLength + aPLCPHeaderLength +<br>( ( (aMPDUDurationFactor × 8 × N) / $10^9$) + (8 × N) ) / data rate] + 1,<br>where data rate is in Mb/s and where *N* counts the number of octets in the PPDU prior to the desired octet, but does not count the number of octets in the preamble PLCP header. |
| aMPDUMaxLength | integer | The maximum number of octets in an MPDU that can be conveyed by a PLCP protocol data unit (PPDU). |
| aCWmin | integer | The minimum size of the CW, in units of aSlotTime. |
| aCWmax | integer | The maximum size of the CW, in units of aSlotTime. |

#### 10.4.3.3 When generated

This primitive will be issued by the PHY entity in response to a PLME-CHARACTERISTICS.request.

#### 10.4.3.4 Effect of receipt

The receipt of this primitive provides the operational characteristics of the PHY entity.


Copyright © 2007 IEEE. All rights reserved.

### 10.4.4 PLME-DSSSTESTMODE.request

#### 10.4.4.1 Function

This primitive requests that the DSSS PHY entity enter a test mode operation. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

#### 10.4.4.2 Semantics of the service primitive

The primitive parameters are as follows:

PLME-DSSSTESTMODE.request(
TEST_ENABLE,
TEST_MODE,
SCRAMBLE_STATE,
SPREADING_STATE,
DATA_TYPE,
DATA_RATE;
PREAMBLE_TYPE;
MODULATION_CODE_TYPE;
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| TEST_ENABLE | Boolean | True, false | If true, enables the PHY test mode according to the remaining parameters. |
| TEST_MODE | integer | 1, 2, 3 | TEST_MODE selects one of three operational states:<br>01 = transparent receive<br>02 = continuous transmit<br>03 = 50% duty cycle |
| SCRAMBLE_STATE | Boolean | True, false | If true, sets the operational state of the scrambler to ON. |
| SPREADING_STATE | Boolean | True, false | If true, selects the operational state of the chipping. |
| DATA_TYPE | integer | 1, 2, 3 | Selects one of three data patterns to be used for the transmit portions of the tests, e.g., all ones, all zeros, and random data patterns. |
| DATA_RATE | integer | 2, 3, 4, 5, 6, 9, 11, 12, 18, 22, 24, 27, 36, 44, 48, 54, 66, 72, 96, 108 | Selects among rates:<br>02 = 1 Mb/s<br>03 = 1.5 Mb/s<br>04 = 2 Mb/s<br>05 = 2.5 Mb/s<br>06 = 3 Mb/s<br>09 = 4.5 Mb/s<br>11 = 5.5 Mb/s<br>12 = 6 Mb/s<br>18 = 9 Mb/s<br>22 = 11 Mb/s<br>24 = 12 Mb/s<br>27 = 13.5 Mb/s<br>36 = 18 Mb/s<br>44 = 22 Mb/s<br>48 = 24 Mb/s<br>54 = 27 Mb/s<br>66 = 33 Mb/s<br>72 = 36 Mb/s<br>96 = 48 Mb/s<br>108 = 54 Mb/s |

Copyright © 2007 IEEE. All rights reserved.

| Name | Type | Valid range | Description |
|---|---|---|---|
| PREAMBLE_TYPE | Boolean | null, 0, 1 | Selects the preamble length:<br>0 = long<br>1 = short<br>Can be null. |
| MODULATION_CODE_TYPE | Integer | null, 0, 1, 2 | Selects among modulation options:<br>0 = no optional modulation modes<br>1 = optional ERP-PBCC modes<br>2 = optional DSSS-OFDM modes<br>Can be null. |

The rate for DATA_Rate=05 is rounded up to the next higher 0.5 Mb/s value.

#### 10.4.4.3 When generated

This primitive is generated at any time to enter the DSSS PHY test mode.

#### 10.4.4.4 Effect of receipt

Receipt of this primitive by the PHY causes the DSSS PHY entity to enter the test mode of operation.


Copyright © 2007 IEEE. All rights reserved.

### 10.4.5 PLME-DSSSTESTOUTPUT.request

#### 10.4.5.1 Function

This optional primitive is a request by the LME to enable selected test signals from the PHY. The parameters associated with this primitive are considered as recommendations and are optional in any particular implementation.

#### 10.4.5.2 Semantics of the service primitive

The primitive parameters are as follows:

```
PLME-DSSSTESTOUTPUT.request(
                                    TEST_OUTPUT,
                                    )
```

| Name | Type | Valid range | Description |
|---|---|---|---|
| TEST_OUTPUT | Boolean | True, false | If true, enables the selected test signals for testing DS PHY. |

TEST_OUTPUT enables and disables selected signals for debugging and testing the PHY. Some signals that can be available for output are PHY-TXSTART.request, PHY-RXSTART.indicate(RXVECTOR), PHY-CCA.indicate, the chipping clock, the data clock, the symbol clock, transmit (TX) data, and receive (RX) data.

#### 10.4.5.3 When generated

This primitive is generated at any time to enable the test outputs when in the DSSS PHY test mode.

#### 10.4.5.4 Effect of receipt

Receipt of this primitive by the DSSS PHY causes the DSSS PHY entity to enable the test outputs using the modes set by the most recent PLME-DSSSTESTMODE.request primitive.

Copyright © 2007 IEEE. All rights reserved.

### 10.4.6 PLME-TXTIME.request

#### 10.4.6.1 Function

This primitive is a request for the PHY to calculate the time that will be required to transmit onto the WM a PPDU containing a specified length MPDU, and using a specified format, data rate, and signalling.

#### 10.4.6.2 Semantics of the service primitive

This primitive provides the following parameters:

PLME-TXTIME.request(TXVECTOR)

The TXVECTOR represents a list of parameters that the MAC sublayer provides to the local PHY entity in order to transmit an MPDU, as further described in 12.3.4.4 and 17.4 (which defines the local PHY entity).

#### 10.4.6.3 When generated

This primitive is issued by the MAC sublayer to the PHY entity when the MAC sublayer needs to determine the time required to transmit a particular MPDU.

#### 10.4.6.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to generate a PHY-TXTIME.confirm primitive that conveys the required transmission time.


Copyright © 2007 IEEE. All rights reserved.

### 10.4.7 PLME-TXTIME.confirm

#### 10.4.7.1 Function

This primitive provides the time that will be required to transmit the PPDU described in the corresponding PLME-TXTIME.request.

#### 10.4.7.2 Semantics of the service primitive

This primitive provides the following parameters:

PLME-TXTIME.confirm(TXTIME)

The TXTIME represents the time, in microseconds, required to transmit the PPDU described in the corresponding PLME-TXTIME.request. If the calculated time includes a fractional microsecond, the TXTIME value is rounded up to the next higher integer.

#### 10.4.7.3 When generated

This primitive is issued by the local PHY entity in response to a PLME-TXTIME.request.

#### 10.4.7.4 Effect of receipt

The receipt of this primitive provides the MAC sublayer with the PPDU transmission time.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# 11. MLME

## 11.1 Synchronization

All STAs within a single BSS shall be synchronized to a common clock using the mechanisms defined herein.

### 11.1.1 Basic approach

A Timing Synchronization Function (TSF) keeps the timers for all STAs in the same BSS synchronized. All STAs shall maintain a local TSF timer.

#### 11.1.1.1 TSF for infrastructure networks

In an infrastructure BSS, the AP shall be the timing master for the TSF. The AP shall initialize its TSF timer independently of any simultaneously started APs in an effort to minimize the synchronization of the TSF timers of multiple APs. The AP shall periodically transmit special frames called *Beacon frames* that contain a copy of its TSF timer to synchronize the TSF timers of other STAs in a BSS. A receiving STA shall always accept the timing information in Beacon frames sent from the AP servicing its BSS. If a STA's TSF timer is different from the timestamp in the received Beacon frame, the receiving STA shall set its local TSF timer to the received timestamp value.

Beacon frames shall be generated for transmission by the AP once every dot11BeaconPeriod TUs.

#### 11.1.1.2 TSF for an IBSS

The TSF in an IBSS shall be implemented via a distributed algorithm that shall be performed by all of the members of the BSS. Each STA in the IBSS shall transmit Beacon frames according to the algorithm described in this clause. Each STA in an IBSS shall adopt the TSF value received from any Beacon frame or probe response from the IBSS of which it is a member and which has a TSF value later than its own TSF timer.

### 11.1.2 Maintaining synchronization

Each STA shall maintain a TSF timer with modulus $2^{64}$ counting in increments of microseconds. STAs expect to receive Beacon frames at a nominal rate. The interval between Beacon frames is defined by the dot11BeaconPeriod parameter of the STA. A STA sending a Beacon frame shall set the value of the Beacon frame's timestamp so that it equals the value of the STA's TSF timer at the time that the data symbol containing the first bit of the timestamp is transmitted to the PHY plus the transmitting STA's delays through its local PHY from the MAC-PHY interface to its interface with the WM [e.g., antenna, light-emitting diode (LED) emission surface].

#### 11.1.2.1 Beacon generation in infrastructure networks

The AP shall define the timing for the entire BSS by transmitting Beacon frames according to the dot11BeaconPeriod attribute within the AP. This defines a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time zero is defined to be a TBTT with the Beacon frame being a DTIM and transmitted at the beginning of a CFP. At each TBTT, the AP shall schedule a Beacon frame as the next frame for transmission according to the medium access rules specified in Clause 9. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the BSS.

NOTE—Though the transmission of a Beacon frame may be delayed because of CSMA deferrals, subsequent Beacon frames shall be scheduled at the undelayed nominal beacon interval. This is shown in Figure 11-1.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 467 of 1232

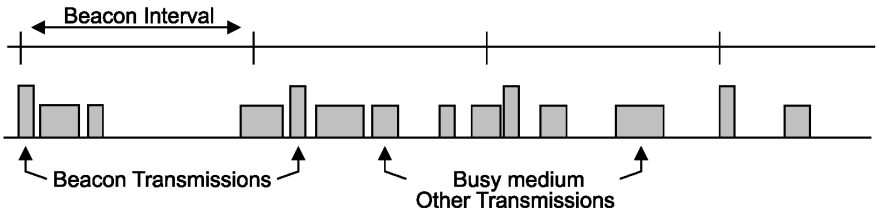



**Figure 11-1—Beacon transmission on a busy network**

### 11.1.2.2 Beacon generation in an IBSS

Beacon generation in an IBSS is distributed. The beacon period is included in Beacon and Probe Response frames, and STAs shall adopt that beacon period when joining the IBSS. All members of the IBSS participate in beacon generation. Each STA shall maintain its own TSF timer that is used for dot11BeaconPeriod timing. The beacon interval within an IBSS is established by the STA at which the MLME-START.request is performed to create the IBSS. This defines a series of TBTTs exactly dot11BeaconPeriod TUs apart. Time zero is defined to be a TBTT. At each TBTT the STA shall

a) Suspend the decrementing of the backoff timer for any pending nonbeacon or non-ATIM transmission,

b) Calculate a random delay uniformly distributed in the range between zero and twice aCWmin × aSlotTime,

c) Wait for the period of the random delay, decrementing the random delay timer using the same algorithm as for backoff,

d) Cancel the remaining random delay and the pending beacon transmission, if a Beacon frame arrives from the IBSS of which the STA is a member before the random delay timer has expired, at which time the ATIM backoff timer shall resume decrementing.

e) Send a Beacon frame if the random delay has expired and no Beacon frame has arrived from the IBSS of which the STA is a member during the delay period.

(See Figure 11-2.)

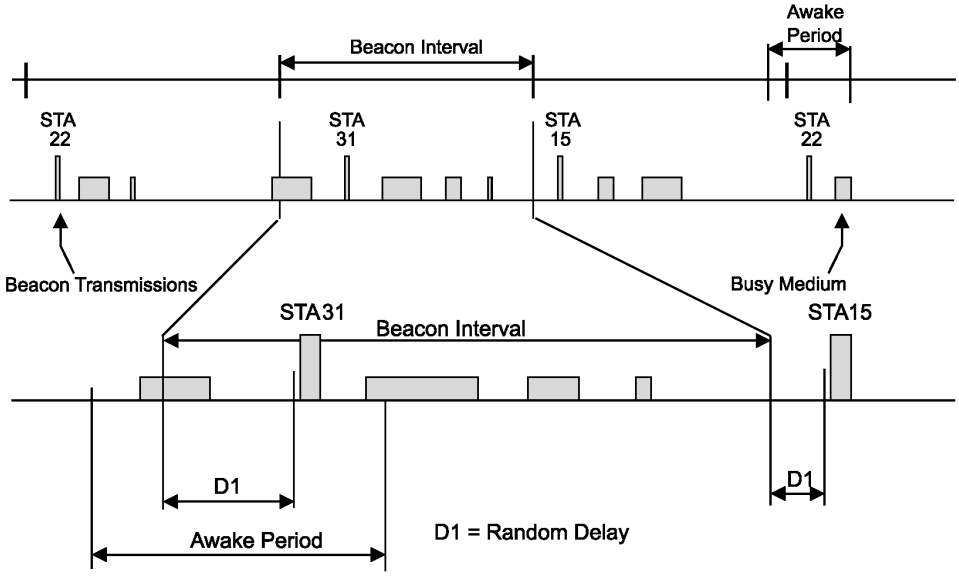



**Figure 11-2—Beacon transmission in an IBSS**


Copyright © 2007 IEEE. All rights reserved.

The beacon transmission shall always occur during the Awake Period of STAs that are operating in a low-power mode. This is described in more detail in 11.2.

#### 11.1.2.3 Beacon reception

STAs shall use information from the CF Parameter Set element of all received Beacon frames, without regard for the BSSID, to update their NAV as specified in 9.3.2.2.

STAs in an infrastructure network shall only use other information in received Beacon frames, if the BSSID field is equal to the MAC address currently in use by the STA contained in the AP of the BSS.

STAs in an IBSS shall use other information in any received Beacon frame for which the IBSS subfield of the Capability field is set to 1, the content of the SSID element is equal to the SSID of the IBSS, and the TSF value is later than the receiving STA's TSF timer. Use of this information is specified in 11.1.4.

#### 11.1.2.4 TSF timer accuracy

Upon receiving a Beacon frame with a valid FCS and BSSID or SSID, as described in 11.1.2.3, a STA shall update its TSF timer according to the following algorithm: The received timestamp value shall be adjusted by adding an amount equal to the receiving STA's delay through its local PHY components plus the time since the first bit of the timestamp was received at the MAC/PHY interface. In the case of an infrastructure BSS, the STA's TSF timer shall then be set to the adjusted value of the timestamp. In the case of an IBSS, the STA's TSF timer shall be set to the adjusted value of the received timestamp, if the adjusted value of the timestamp is later than the value of the STA's TSF timer. The accuracy of the TSF timer shall be no worse than ±0.01%.

### 11.1.3 Acquiring synchronization, scanning

A STA shall operate in either a Passive Scanning mode or an Active Scanning mode depending on the current value of the ScanMode parameter of the MLME-SCAN.request primitive.

Active scanning is prohibited in some frequency bands and regulatory domains. The MAC of a STA receiving an MLME-SCAN.request shall use the regulatory domain information it has to process the request and shall return a result code of NOT_SUPPORTED to a request for an active scan if regulatory domain information indicates an active scan is illegal.

Upon receipt of the MLME-SCAN.request primitive, a STA shall perform scanning. The SSID parameter indicates the SSID for which to scan. To become a member of a particular ESS using passive scanning, a STA shall scan for Beacon frames containing that ESS's SSID, returning all Beacon frames matching the desired SSID in the BSSDescriptionSet parameter of the corresponding MLME-SCAN.confirm primitive with the appropriate bits in the Capabilities Information field indicating whether the Beacon frame came from an infrastructure BSS or IBSS. To actively scan, the STA shall transmit Probe request frames containing the desired SSID. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

Upon receipt of an MLME-JOIN.request, the STA shall use the syncrhonization procedure described in 11.1.3.4.

Upon receipt of an MLME-SCAN.request with the SSID parameter set to the wildcard SSID, the STA shall passively scan for any Beacon frames, or actively transmit Probe request frames containing the wildcard SSID, as appropriate depending upon the value of ScanMode. Upon completion of scanning, an MLME-SCAN.confirm is issued by the MLME indicating all of the BSS information received.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 469 of 1232

If a STA's scanning does not result in finding a BSS with the desired SSID and of the desired type, or does not result in finding any BSS, the STA may start an IBSS upon receipt of the MLME-START.request. The MAC of a STA receiving an MLME-START.request shall use the regulatory domain information it has to process the request and shall return a result code of NOT_SUPPORTED to the request if regulatory domain information indicates starting the IBSS is illegal.

When a STA starts a BSS, that STA shall determine the BSSID of the BSS. If the BSSType indicates an infrastructure BSS, then the STA shall start an infrastructure BSS and the BSSID shall be equal to the STA's dot11StationID. The value of the BSSID shall remain unchanged, even if the value of dot11StationID is changed after the completion of the MLME-START.request. If the BSSType indicates an IBSS, the STA shall start an IBSS, and the BSSID shall be an individual locally administered IEEE MAC address as defined in 5.2 of IEEE Std 802-1990. The remaining 46 bits of that MAC address shall be a number selected in a manner that minimizes the probability of STAs generating the same number, even when those STAs are subjected to the same initial conditions. The value SSID parameter shall be used as the SSID of the new BSS. It is important that designers recognize the need for statistical independence among the random number streams among STAs.

#### 11.1.3.1 Passive scanning

If the ScanType parameter indicates a passive scan, the STA shall listen to each channel scanned for no longer than a maximum duration defined by the MaxChannelTime parameter.

#### 11.1.3.2 Active scanning

Active scanning involves the generation of Probe request frames and the subsequent processing of received Probe Response frames. The details of the active scanning procedures are as specified in the following subclauses.

##### 11.1.3.2.1 Sending a probe response

STAs, subject to criteria below, receiving Probe Request frames shall respond with a probe response only if

a) The SSID in the probe request is the wildcard SSID or the specific SSID of the STA,
b) The BSSID field in the probe request is the wildcard BSSID or the BSSID of the STA, and
c) The DA field in the probe request is the broadcast address or the specific MAC address of the STA.

Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The probe response shall be sent using normal frame transmission rules. An AP shall respond to all probe requests meeting the above criteria. In an IBSS, the STA that generated the last Beacon frame shall be the STA that responds to a probe request.

Only APs and STAs in an IBSS respond to probe requests. The procedures defined in this subclause ensure that in each BSS there is at least one STA that is awake at any given time to receive and respond to probe requests. A STA that sent a Beacon frame shall remain in the Awake state and shall respond to probe requests, subject to criteria in the next paragraph, until a Beacon frame with the current BSSID is received. If the STA is an AP, it shall always remain in the Awake state and always respond to probe requests, subject to criteria in the next paragraph. There may be more than one STA in an IBSS that responds to any given probe request, particularly in cases where more than one STA transmitted a Beacon frame following the most recent TBTT, either due to not receiving successfully a previous Beacon frame or due to collisions between beacon transmissions.

STAs receiving Probe Request frames shall respond with a probe response when the SSID in the probe request is the wildcard SSID or matches the specific SSID of the STA. Probe Response frames shall be sent as directed frames to the address of the STA that generated the probe request. The probe response shall be


Copyright © 2007 IEEE. All rights reserved.

sent using normal frame transmission rules. An AP shall respond to all probe requests meeting the above criteria. In an IBSS, the STA that generated the last Beacon frame shall be the STA that responds to a probe request.

#### 11.1.3.2.2 Active scanning procedure

Upon receipt of the MLME-SCAN.request with ScanType indicating an active scan, a STA shall use the following procedure:

For each channel to be scanned,

a) Wait until the ProbeDelay time has expired or a PHYRxStart.indication has been received;

b) Perform the Basic Access procedure as defined in 9.2.5.1;

c) Send a probe request to the broadcast destination address, with the SSID and BSSID from the MLME-SCAN.request primitive;

d) Clear and start a ProbeTimer;

e) If PHY-CCA.indication (busy) has not been detected before the ProbeTimer reaches MinChannelTime, then clear NAV and scan the next channel, else when ProbeTimer reaches MaxChannelTime, process all received probe responses;

f) Clear NAV and scan the next channel.

See Figure 11-3.

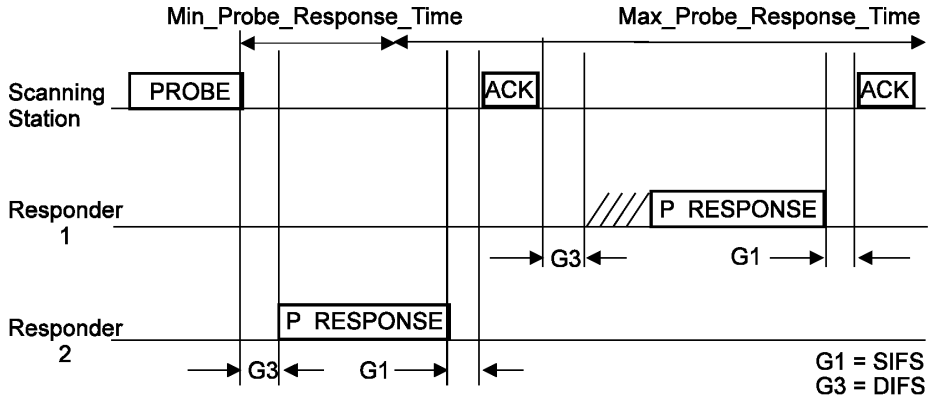



**Figure 11-3—Probe response**

When all channels in the ChannelList have been scanned, the MLME shall issue an MLME-SCAN.confirm with the BSSDescriptionSet containing all of the information gathered during the scan.

#### 11.1.3.3 Initializing a BSS

Upon receipt of an MLME-START.request, a STA shall determine the BSS's BSSID (as described in 11.1.3), select channel synchronization information, select a beacon period, initialize and start its TSF timer, and begin transmitting Beacon frames. If the dot11MultiDomainCapabilityEnabled attribute is true, a STA shall not start a BSS, neither an infrastructure BSS nor an IBSS, unless a properly formed Beacon frame including a Country information element can be constructed, and the dot11CountryString attribute has been set.

Copyright © 2007 IEEE. All rights reserved.

#### 11.1.3.4 Synchronizing with a BSS

Upon receipt of an MLME-JOIN.request, a STA shall adopt the BSSID in the request. Upon receipt of a Beacon frame from the BSS, a STA shall adopt the channel synchronization information (applicable only if the STA contains an FH PHY), and TSF timer value of the parameters in the Beacon frame using the algorithm described in 11.1.2.4, and the MLME shall issue an MLME-JOIN.confirm indicating the operation was successful.

In addition to these synchronization parameters, a STA joining an infrastructure BSS will adopt each of the parameters found in the BssDescription of the MLME-JOIN.request except Local time, Capability Information, and BSSBasicRateSet parameters. Local time is not adopted but is used as a local variable in adopting the TSF as described in 11.1.2.4. The Capability Information reflects the capabilities of the sender and is not adopted but may be used to determine local configuration or behavior. The BSSBasicRateSet parameter is not adopted but may determine if the STA can join the BSS. A STA joining an IBSS will adopt the same parameters except the CF parameter set (since contention free period is not permitted in an IBSS).

If the JoinFailureTimeout timer expires prior to the receipt of a Beacon frame from the BSS, the MLME shall issue an MLME-JOIN.confirm indicating the operation was unsuccessful.

If the dot11MultiDomainCapabilityEnabled attribute is true, a STA receiving a Beacon or Probe Response frame containing a Country information element shall adopt the all parameters included in that Country information element when joining a BSS and the dot11RegDomainsSupportEntry shall be set to Other.

If a Hopping Pattern Parameters element is present in the Beacon or Probe Response frame, and if the dot11MultiDomainCapability-Enabled attribute is true, a STA shall adopt the pattern parameters in the element and calculate the hopping patterns using one of the algorithms defined in 7.3.2.10 or 7.3.2.11. Using the appropriate pattern, set, and index values from the FH Parameter Set element, the STA shall adopt the values in use by the BSS when joining. The dot11RegDomainsSupportedValue shall be set to Other when the STA is operating using Country information element settings

### 11.1.4 Adjusting STA timers

In an infrastructure BSS, STAs shall always adopt the TSF timer value in a Beacon frame or probe response coming from the AP in their BSS by using the algorithm in 11.1.2.4.

In response to an MLME-JOIN.request, a STA joining an IBSS shall initialize its TSF timer to 0 and shall not transmit a Beacon frame or probe response until it hears a Beacon frame or probe response from a member of the IBSS with a matching SSID. This ensures that such a STA will adopt the timer from the next Beacon frame or probe response from its IBSS.

All Beacon and Probe Response frames carry a Timestamp field. A STA receiving such a frame from another STA in an IBSS with the same SSID shall compare the Timestamp field with its own TSF time. If the Timestamp field of the received frame is later than its own TSF timer, the STA in the IBSS shall adopt all parameters contained in the Beacon frame, except the Capability bits, Supported Rates information element, and Extended Supported Rates information element.

### 11.1.5 Timing synchronization for FH PHYs

NOTE—This subclause pertains only to STAs using an FH PHY.

The TSF described here provides a mechanism for STAs in an FH system to synchronize their transitions from one channel to another (their “hops”). Every STA shall maintain a table of all of the hopping sequences that are used in the system. All of the STAs in a BSS shall use the same hopping sequence. Each Beacon


Copyright © 2007 IEEE. All rights reserved.

frame and probe response includes the channel synchronization information necessary to determine the hop pattern and timing for the BSS.

STAs shall use their TSF timer to time the aCurrentDwellTime. The aCurrentDwellTime is the length of time that STAs shall stay on each frequency in their hopping sequence. Once STAs are synchronized, they have the same TSF timer value.

STAs in the BSS shall issue an appropriate PLME service primitive for the PHY in use to tune to the next frequency in the hopping sequence when

TSF timer MOD aCurrentDwellTime = 0

## 11.2 Power management

### 11.2.1 Power management in an infrastructure network

STAs changing Power Management mode shall inform the AP of this fact using the Power Management bits within the Frame Control field of transmitted frames. The AP shall not arbitrarily transmit MSDUs to STAs operating in a PS mode, but shall buffer MSDUs and only transmit them at designated times.

The STAs that currently have buffered MSDUs within the AP are identified in a TIM, which shall be included as an element within all Beacon frames generated by the AP. A STA shall determine that an MSDU is buffered for it by receiving and interpreting a TIM.

STAs operating in PS modes shall periodically listen for Beacon frames, as determined by the STA's ListenInterval and the ReceiveDTIMs parameter in the MLME-POWERMGT.request primitive.

In a BSS operating under the DCF, or during the CP of a BSS using the PCF, upon determining that an MSDU is currently buffered in the AP, a STA operating in the *PS mode* shall transmit a short PS-Poll frame to the AP, which shall respond with the corresponding buffered MSDU immediately, or acknowledge the PS-Poll and respond with the corresponding MSDU at a later time. If the TIM indicating the buffered MSDU is sent during a CFP, a CF-Pollable STA operating in the PS mode does not send a PS-Poll frame, but remains active until the buffered MSDU is received (or the CFP ends). If any STA in its BSS is in PS mode, the AP shall buffer all broadcast and multicast MSDUs and deliver them to all STAs immediately following the next Beacon frame containing a DTIM transmission.

A STA shall remain in its current Power Management mode until it informs the AP of a Power Management mode change via a frame exchange that includes an acknowledgment from the AP. Power Management mode shall not change during any single frame exchange sequence, as described in 9.12.

A non-AP QoS STA may be in PS mode before the setup of DLS or Block Ack. Once DLS is set up with another non-AP STA, the non-AP STA 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 STA continue to operate normally.

#### 11.2.1.1 STA Power Management modes

A STA may be in one of two different power states:

— *Awake:* STA is fully powered.

— *Doze:* STA is not able to transmit or receive and consumes very low power.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 473 of 1232

The manner in which a STA transitions between these two power states shall be determined by the STA's Power Management mode. These modes are summarized in Table 11-1.

**Table 11-1—Power Management modes**

| | |
|---|---|
| Active mode or AM | STA may receive frames at any time. In Active mode, a STA shall be in the Awake state. A STA on the polling list of a PCF shall be in Active mode for the duration of the CFP. |
| PS | STA listens to selected Beacon frames (based upon the ListenInterval parameter of the MLME-ASSOCIATE.request primitive) and sends PS-Poll frames to the AP if the TIM element in the most recent Beacon frame indicates a directed MSDU buffered for that STA. The AP shall transmit buffered directed MSDUs to a PS STA only in response to a PS-Poll from that STA, or during the CFP in the case of a CF-Pollable PS STA. In PS mode, a STA shall be in the Doze state and shall enter the Awake state to receive selected Beacon frames, to receive broadcast and multicast transmissions following certain received Beacon frames, to transmit, and to await responses to transmitted PS-Poll frames or (for CF-Pollable STAs) to receive CF transmissions of buffered MSDUs. |

The Power Management mode of a STA is selected by the PowerManagementMode parameter of the MLME-POWERMGT.request. Once the STA updates its Power Management mode, the MLME shall issue an MLME-POWERMGT.confirm indicating the success of the operation.

To change Power Management modes, a STA shall inform the AP through a successful frame exchange initiated by the STA. The Power Management bit in the Frame Control field of the frame sent by the STA in this exchange indicates the Power Management mode that the STA shall adopt upon successful completion of the entire frame exchange. The Power Managment bit shall not be set in any management frame, except an Action frame.

A STA that is changing from Doze to Awake in order to transmit shall perform CCA until a frame sequence is detected by which it can correctly set its NAV, or until a period of time equal to the ProbeDelay has transpired.

#### 11.2.1.2 AP TIM transmissions

The TIM shall identify the STAs for which traffic is pending and buffered in the AP. This information is coded in a *partial virtual bitmap*, as described in 7.3.2.6. In addition, the TIM contains an indication whether broadcast/multicast traffic is pending. Every STA is assigned an AID by the AP as part of the association process. AID 0 (zero) is reserved to indicate the presence of buffered broadcast/multicast MSDUs. The AP shall identify those STAs for which it is prepared to deliver buffered MSDUs by setting bits in the TIM's partial virtual bitmap that correspond to the appropriate AIDs.

#### 11.2.1.3 TIM types

Two different TIM types are distinguished: TIM and DTIM. After a DTIM, the AP shall send out the buffered broadcast/multicast MSDUs using normal frame transmission rules, before transmitting any unicast frames.

The AP shall transmit a TIM with every Beacon frame. Every DTIMPeriod, a TIM of type *DTIM* is transmitted within a Beacon frame, rather than an ordinary TIM.

Figure 11-4 illustrates the AP and STA activity under the assumptions that no PCF is operating and that a DTIM is transmitted once every three TIMs. The top line in Figure 11-4 represents the time axis, with the beacon interval shown together with a DTIM Interval of three beacon intervals. The second line depicts AP


Copyright © 2007 IEEE. All rights reserved.

activity. The AP schedules Beacon frames for transmission every beacon interval, but the Beacon frames may be delayed if there is traffic at the TBTT. This is indicated as "busy medium" on the second line. For the purposes of this figure, the important fact about Beacon frames is that they contain TIMs, some of which are DTIMs. Note that the second STA with ReceiveDTIMs set to false does not power-on its receiver for all DTIMs.

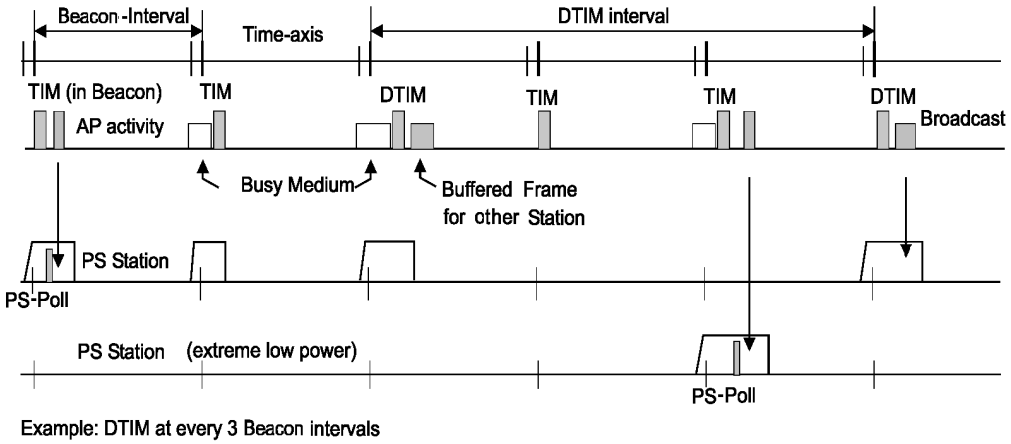



**Figure 11-4—Infrastructure power management operation (no PCF operating)**

The third and fourth lines in Figure 11-4 depict the activity of two STAs operating with different power management requirements. Both STAs power-on their receivers when they need to listen for a TIM. This is indicated as a ramp-up of the receiver power prior to the TBTT. The first STA, for example, powers up its receiver and receives a TIM in the first Beacon frame; that TIM indicates the presence of a buffered MSDU for the receiving STA. The receiving STA then generates a PS-Poll frame, which elicits the transmission of the buffered data MSDU from the AP. Broadcast and multicast MSDUs are sent by the AP subsequent to the transmission of a Beacon frame containing a DTIM. The DTIM is indicated by the DTIM count field of the TIM element having a value of 0.

### 11.2.1.4 Power management with APSD

QoS APs 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 QoS STAs 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 STA using APSD shall have the Power Management bit in the Frame Control field set to 1 for buffering to take place at the AP.

APSD defines two delivery mechanisms, namely *unscheduled APSD* (U-APSD) and *scheduled APSD* (S-APSD). Non-AP STAs may use U-APSD to have some or all of their frames delivered during unscheduled SPs. Non-AP STAs 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 AP receives a trigger frame from a non-AP STA, 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 AP has attempted to transmit at least one MSDU or MMPDU associated with a delivery-enabled AC and destined for the non-AP STA, but no more than the number indicated in the Max SP Length field if the field has a nonzero value.

Copyright © 2007 IEEE. All rights reserved.

In order to configure an AP to deliver frames during an unscheduled SP, the non-AP STA 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 STA may configure an AP to use U-APSD using two methods. First, a non-AP STA 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 STA 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 STA is trigger- or delivery-enabled during (re)association.

Alternatively, a non-AP STA 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 AP 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 STA 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, a non-AP STA shall send an ADDTS Request frame to the AP with the APSD subfield of the TS Info field in the TSPEC element set to 1. To use a scheduled SP for a TS for a AC when the access policy is contention-based channel access, a non-AP STA shall send an ADDTS Request frame to the AP 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 AP and the AP accepts the corresponding ADDTS Request frame from the non-AP STA, the AP shall respond to the ADDTS Request frame with a response containing the Schedule element indicating that the requested service can be accommodated by the AP. The first scheduled SP starts when the lower order 4 octets of the TSF timer equals the value specified in the Service Start Time field. A non-AP STA 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 AP 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 BSS, a STA may use both unscheduled and scheduled APSD on different ACs at the same time. When a non-AP STA establishes scheduled delivery for an AC, that AC shall be considered delivery-enabled. However, the AP 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 AP 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 STA. Broadcast/multicast frame delivery shall follow the frame delivery rules defined for broadcast/multicast frames as defined in 11.2.1.6.


Copyright © 2007 IEEE. All rights reserved.

#### 11.2.1.5 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. APs that implement and signal their support of APSD shall maintain an APSD and an access policy status for each currently associated non-AP STA that indicates whether the non-AP STA is presently using APSD and shall maintain the schedule (if any) for the non-AP STA. An AP shall, depending on the Power Management mode of the STA, temporarily buffer the MSDU or management frame destined to the STA. An AP implementing APSD shall, if a non-AP STA is using APSD and is in PS mode, temporarily buffer the MSDU or management frames destined to that non-AP STA. No MSDUs or management frames addressed directly to 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 STAs using APSD shall be temporarily buffered in the APSD-capable AP. 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 frame. At every beacon interval, the APSD-capable AP 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 STAs 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 AP shall assemble the partial virtual bitmap containing the buffer status for all ACs per destination for non-AP STAs.

d) If a non-AP STA has set up a scheduled SP, it shall automatically wake up at each SP. Therefore, the APSD-capable AP shall transmit frames associated with admitted traffic with the APSD subfield set to 1 in the TSPECs buffered for the non-AP STA during a scheduled SP. If the non-AP STA has set up to use unscheduled SPs, the AP 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 STA, which indicates the start of an unscheduled SP. A trigger frame received by the AP from a non-AP STA that already has an unscheduled SP underway shall not trigger the start of a new unscheduled SP. The AP transmits frames destined for the non-AP STA and associated with delivery-enabled ACs during an unscheduled SP. The bit for AID 0 (zero) in the bit map control field of the TIM IE shall be set to 1 when broadcast or multicast traffic is buffered, according to 7.3.2.6.

e) 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) 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) in the bit map control field of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.

g) 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 STAs using U-APSD, the AP transmits one frame destined for the non-AP STA from any AC that is not delivery-enabled in response to PS-Poll from the non-AP STA. When all ACs associated with the non-AP STA are delivery-enabled, AP transmits one frame from the highest priority AC. The AP can respond with either an immediate Data frame or with an ACK, while delaying the responding Data frame.

   For a STA in PS mode and not using U-APSD, the More Data field of the response Data frame shall be set to indicate the presence of further buffered MSDUs or management frames for the polling

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 477 of 1232

STA. For a non-AP STA 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 STA 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 AP 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. An AP 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 STA, the APSD-capable AP 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 STA. At each unscheduled SP for a non-AP STA, the AP 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 STA.

   The More Data bit of the directed data or management frame associated with delivery-enabled ACs and destined for that non-AP STA 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 STA 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. An AP 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 AP considers APSD STA 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 STA. If necessary, the AP may generate an extra QoS Null frame, with the EOSP set to 1. When the AP has transmitted a directed frame to the non-AP STA with the EOSP subfield set to 1 during the SP except for retransmissions of that frame, the AP shall not transmit any more frames using this mechanism until the next SP. The AP shall set the EOSP subfield to 1 to indicate the end of the SP in APSD.

i) If the AP does not receive an acknowledgment to a directed MSDU or management frame sent to a non-AP STA in PS mode following receipt of a PS-Poll from that non-AP STA, 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 frame, but it shall retransmit that frame at least once before the next Beacon frame, 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 AP 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.


Copyright © 2007 IEEE. All rights reserved.

k) An AP can delete buffered frames for implementation-dependent reasons, including the use of an aging function and availability of buffers. The AP may base the aging function on the listen interval specified by the non-AP STA in the (Re)Association Request frame.

l) 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 an AP is informed that an APSD-capable non-AP STA is not using APSD, then the AP shall send buffered MSDUs and management frames (if any exist) to that non-AP STA according to the rules corresponding to the current PS mode of the non-AP STA.

#### 11.2.1.6 AP operation during the CFP

APs shall maintain a Power Management status for each currently associated CF-Pollable STA that indicates in which Power Management mode the STA is currently operating. An AP shall, for STAs in PS mode, temporarily buffer the MSDU destined to the STA.

a) MSDUs destined for PS STAs shall be temporarily buffered in the AP. The algorithm to manage this buffering is beyond the scope of this standard.

b) MSDUs destined to STAs in the Active mode shall be transmitted as defined in Clause 9.

c) Prior to every CFP, and at each beacon interval within the CFP, the AP shall assemble the partial virtual bitmap containing the buffer status per destination for STAs in the PS mode, set the bits in the partial virtual bitmap for STAs the PC is intending to poll during this CFP, and shall send this out in the TIM field of the DTIM. The bit for AID 0 (zero) in the bit map control field of the TIM IE shall be set when broadcast or multicast traffic is buffered, according to 7.3.2.6.

d) All broadcast and multicast MSDUs, with the Order bit in the Frame Control field clear, shall be buffered if any associated STAs are in the PS mode, whether or not those STAs are CF-Pollable.

e) Immediately after every DTIM (Beacon frame with DTIM Count field of the TIM element equal to zero), the AP shall transmit all buffered broadcast and multicast frames. The More Data field shall be set in the headers of all but the final such frame 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) in the bit map control field of the TIM element of every Beacon frame, until all buffered broadcast/multicast frames have been transmitted.

f) Buffered MSDUs or MMPDUs for STAs in the PS mode shall be forwarded to the CF-Pollable STAs under control of the PC. Transmission of these buffered MSDUs or management frames as well as CF-Polls to STAs in the PS mode that were indicated in the DTIM in accordance with paragraph c) of this subclause shall begin immediately after transmission of buffered broadcast and multicast frames (if any), and shall occur in order by increasing AID of CF-Pollable STAs. A CF-Pollable STA for which the TIM element of the most recent Beacon frame indicated buffered MSDUs or management frames shall be in the Awake state at least until the receipt of a directed frame from the AP in which the Frame Control field does not indicate the existence of more buffered MSDUs or management frames. After acknowledging the last of the buffered MSDUs or management frames, the CF-Pollable STA operating in the PS mode may enter the Doze state until the next DTIM is expected.

g) An AP shall have an aging function to delete pending traffic buffered for an excessive time period. The exact specification of the aging function is beyond the scope of this standard.

h) When an AP detects that a CF-Pollable STA has changed from the PS mode to the Active mode, then the AP shall queue any buffered frames addressed to that STA for transmission to that CF-Pollable STA as directed by the AP's PC.

Copyright © 2007 IEEE. All rights reserved.

#### 11.2.1.7 Receive operation for STAs in PS mode during the CP

STAs in PS mode shall operate as follows to receive an MSDU or management frame from the AP when no PC is operating and during the CP when a PC is operating.

a) STAs shall wake up early enough to be able to receive the first Beacon frame scheduled for transmission after the time corresponding to the last TBTT plus the STA's current ListenInterval.

b) When a STA detects that the bit corresponding to its AID is set in the TIM, the STA shall issue a PS-Poll to retrieve the buffered MSDU or management frame. The PS-Poll shall be transmitted after a random delay uniformly distributed between zero and aCWmin slots following a DIFS.

c) The STA shall remain in the Awake state until it receives the data or management frame in response to its poll or it receives another Beacon frame 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 STA 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 STA shall issue a trigger frame or a PS-Poll frame to retrieve the buffered MSDU or management frames.

d) If the More Data field in the received MSDU or management frame indicates that more traffic for that STA is buffered, the STA, at its convenience, shall Poll until no more MSDUs or management frames are buffered for that STA.

e) When ReceiveDTIMs is true, the STA shall wake up early enough to be able to receive every DTIM sent by the AP of the BSS. A STA that stays awake to receive 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 STA receives a QoS +CF-Ack frame from its AP with the More Data bit set to 1, then the STA shall operate exactly as if it received a TIM with its AID bit set. If a non-AP STA has set the More Data Ack subfield in QoS Capability information element to 1, then if it receives an ACK frame from its AP with the More Data bit set to 1, the STA shall operate exactly as if it received a TIM with its AID bit set. For example, a STA that is using the PS-Poll delivery method shall issue a PS-Poll frame to retrieve a buffered frame.

#### 11.2.1.8 Receive operation for STAs in PS mode during the CFP

STAs in PS mode that are associated as CF-Pollable shall operate as follows in a BSS with an active PC to receive MSDUs or management frames from the AP during the CFP:

a) STAs shall enter the Awake state so as to receive the Beacon frame (which contains a DTIM) at the start of each CFP.

b) To receive broadcast/multicast MSDUs, the STA shall wake up early enough to be able to receive every DTIM that may be sent during the CFP. 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 broadcast/multicast MSDUs buffered.

c) When a STA detects that the bit corresponding to its AID is set in the DTIM at the start of the CFP (or in a subsequent TIM during the CFP), the STA shall remain in the Awake state for at least that portion of the CFP through the time that the STA receives a directed MSDU or management frame from the AP with the More Data field in the Frame Control field indicating that no further traffic is buffered.

d) If the More Data field in the Frame Control field of the last MSDU or management frame received from the AP indicates that more traffic for the STA is buffered, then, when the CFP ends, the STA may remain in the Awake state and transmit PS-Poll frames during the CP to request the delivery of additional buffered MSDU or management frames, or may enter the Doze state during the CP (except at TBTTs for DTIMs expected during the CP), awaiting the start of the next CFP.


Copyright © 2007 IEEE. All rights reserved.

#### 11.2.1.9 Receive operation for non-AP STAs using APSD

A non-AP STA using APSD shall operate as follows to receive an MSDU or management frame from the AP:

a) If a scheduled SP has been set up, the non-AP STA wakes up at its scheduled start time. (The non-AP STA shall wake up early enough to receive transmissions at the scheduled SP.)

b) If the non-AP STA is initiating an unscheduled SP, the non-AP STA wakes up and transmits a trigger frame to the AP. When one or more ACs are not delivery-enabled, the non-AP STA may retrieve MSDUs and MMPDUs belonging to those ACs by sending PS-Poll frames to the AP.

c) The non-AP STA 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 STA 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 STA 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.

#### 11.2.1.10 STAs operating in the Active mode

A STA operating in this mode shall have its receiver activated continuously; such STAs do not need to interpret the TIM information elements in Beacon frames.

#### 11.2.1.11 AP aging function

Any AP aging function shall not cause the buffered traffic to be discarded after any period that is shorter than the ListenInterval of the STA for which the traffic is buffered. The exact specification of the aging function is beyond the scope of this standard.

### 11.2.2 Power management in an IBSS

This subclause specifies the power management mechanism for use within an IBSS.

#### 11.2.2.1 Basic approach

The basic approach is similar to the infrastructure case in that the STAs are synchronized, and multicast MSDUs and those MSDUs that are to be transmitted to a power-conserving STA are first announced during a period when all STAs are awake. The announcement is done via an ad hoc ATIM sent in an ATIM Window. A STA in the PS mode shall listen for these announcements to determine if it needs to remain in the Awake state. The presence of the ATIM window in the IBSS indicates if the STA may use PS Mode. To maintain correct information on the power save state of other STAs in an IBSS, a STA needs to remain awake during the ATIM window. At other times the STA may enter the Doze state except as indicated in the following procedures

The basic approach is similar to the infrastructure case in that the STAs are synchronized, and multicast MSDUs and those MSDUs that are to be transmitted to a power-conserving STA are first announced during a period when all STAs are awake. The announcement is done via an ad hoc ATIM sent in an ATIM Window. A STA in the PS mode shall listen for these announcements to determine if it needs to remain in the Awake state. The presence of the ATIM window in the IBSS indicates if the STA may use PS Mode. To maintain correct information on the power save state of other STAs in an IBSS, a STA needs to remain awake during the ATIM window. At other times the STA may enter the Doze state except as indicated in the following procedures.

When an MSDU is to be transmitted to a destination STA that is in a PS mode, the transmitting STA first transmits an ATIM frame during the ATIM Window, in which all the STAs including those operating in a

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 481 of 1232

PS mode are awake. The ATIM Window is defined as a specific period of time, defined by the value of the ATIM Window parameter in the IBSS Parameter Set supplied to the MLME-START.request primitive, following a TBTT, during which only Beacon or ATIM frames shall be transmitted. ATIM transmission times are randomized, after a Beacon frame is either transmitted or received by the STA, using the backoff procedure with the CW equal to aCWmin. Directed ATIMs shall be acknowledged. If a STA transmitting a directed ATIM does not receive an acknowledgment, the STA shall execute the backoff procedure for retransmission of the ATIM. Multicast ATIMs shall not be acknowledged.

If a STA receives a directed ATIM frame during the ATIM Window, it shall acknowledge the directed ATIM and stay awake for the entire beacon interval waiting for the announced MSDU(s) to be received. If a STA does not receive an ATIM, it may enter the Doze state at the end of the ATIM Window. Transmissions of MSDUs announced by ATIMs are randomized after the ATIM Window, using the backoff procedure described in Clause 9.

It is possible that an ATIM may be received from more than one STA, and that a STA that receives an ATIM may receive more than a single MSDU from the transmitting STA. ATIM frames are only addressed to the destination STA of the MSDU.

An ATIM for a broadcast or multicast MSDU shall have a destination address identical to that of the MSDU.

After the ATIM interval, only those directed MSDUs that have been successfully announced with an acknowledged ATIM, and broadcast/multicast MSDUs that have been announced with an ATIM, shall be transmitted to STAs in the PS mode. Transmission of these frames shall be done using the normal DCF access procedure.

Figure 11-5 illustrates the basic PS operation.

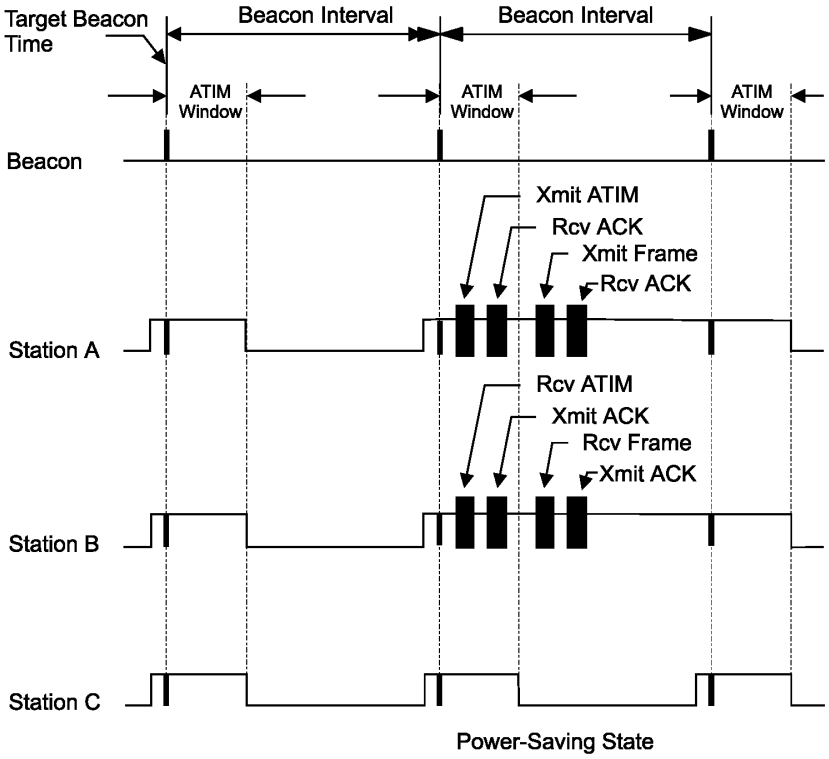



**Figure 11-5—Power management in an IBSS—basic operation**


Copyright © 2007 IEEE. All rights reserved.

The estimated power-saving state of another STA may be based on the power management information transmitted by that STA and on additional information available locally, such as a history of failed transmission attempts. The use of RTS/CTS in an IBSS may reduce the number of transmissions to a STA that is in PS mode. If an RTS is sent and a CTS is not received, the transmitting STA may assume that the destination STA is in PS mode. The method of estimating the power management state of other STAs in the IBSS is outside the scope of this standard.

#### 11.2.2.2 Initialization of power management within an IBSS

The following procedure shall be used to initialize power management within a new IBSS, or to learn about the power management being used within an existing IBSS.

a) A STA joining an existing IBSS by the procedure in 11.1.3.3 shall update its ATIM Window with the value contained in the ATIM Window field of the IBSS Parameter Set element within the Beacon or Probe Response management frame received during the scan procedure.

b) A STA creating a new IBSS by the procedure in 11.1.3.3 shall set the value of the ATIM Window field of the IBSS Parameter Set element within the Beacon management frames transmitted to the value of its ATIM Window.

c) The start of the ATIM Window shall be the TBTT, defined in 11.1.2.2. The end of the ATIM Window shall be defined as

   TSF timer MOD dot11BeaconInterval = ATIMWindow, where ATIMWindow is the value of the ATIM Window parameter of the IBSS Parameter Set from the MLME-Start.request or MLME-JOIN.request primitives.

d) The ATIM Window period shall be static during the lifetime of the IBSS.

e) An ATIM Window value of zero shall indicate that power management is not usable within the IBSS.

#### 11.2.2.3 STA power state transitions

A STA may enter PS mode if and only if the value of the ATIM Window in use within the IBSS is greater than zero. A STA shall set the Power Management subfield in the Frame Control field of MSDUs that it transmits using the rules in 7.1.3.1.6.

A STA in PS mode shall transition between Awake and Doze states according to the following rules:

a) If a STA is operating in PS mode, it shall enter the Awake state prior to each TBTT.

b) If a STA receives a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window it shall remain in the Awake state until the end of the next ATIM Window.

c) If a STA transmits a Beacon or an ATIM management frame, it shall remain in the Awake state until the end of the next ATIM Window regardless of whether an acknowledgment is received for the ATIM.

d) If the STA has not transmitted an ATIM and does not receive either a directed ATIM management frame containing its individual address, or a multicast ATIM management frame during the ATIM Window, it may return to the Doze state following the end of the current ATIM Window.

#### 11.2.2.4 ATIM and frame transmission

If power management is in use within an IBSS, all STAs shall buffer MSDUs for STAs that are known to be in PS mode. The algorithm used for the estimation of the power management state of STAs within the IBSS is outside the scope of this standard. MSDUs may be sent to STAs in Active mode at any valid time.

a) Following the reception or transmission of the Beacon frame, during the ATIM Window, the STA shall transmit a directed ATIM management frame to each STA for which it has one or more

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 483 of 1232

buffered unicast MSDUs. If the STA has one or more buffered multicast MSDUs, with the Strictly Ordered bit clear, it shall transmit an appropriately addressed multicast ATIM frame. A STA transmitting an ATIM management frame shall remain awake for the entire current beacon interval.

b) All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first ATIM following the Beacon frame. All remaining ATIMs shall be transmitted using the conventional DCF access procedure.

c) ATIM management frames shall only be transmitted during the ATIM Window.

d) A STA shall transmit no frame types other than RTS, CTS, and ACK Control frames and Beacon and ATIM management frames during the ATIM Window.

e) Directed ATIM management frames shall be acknowledged. If no acknowledgment is received, the ATIM shall be retransmitted using the conventional DCF access procedure. Multicast ATIM management frames shall not be acknowledged.

f) If a STA is unable to transmit an ATIM during the ATIM Window, for example due to contention with other STAs, the STA shall retain the buffered MSDU(s) and attempt to transmit the ATIM during the next ATIM Window.

g) Immediately following the ATIM Window, a STA shall begin transmission of buffered broadcast/multicast frames for which an ATIM was previously transmitted. Following the transmission of any broadcast/multicast frames, any MSDUs and management frames addressed to STAs for which an acknowledgment for a previously transmitted ATIM frame was received shall be transmitted. All STAs shall use the backoff procedure defined in 9.2.5.2 for transmission of the first frame following the ATIM Window. All remaining frames shall be transmitted using the conventional DCF access procedure.

h) A buffered MSDU may be transmitted using fragmentation. If an MSDU has been partially transmitted when the next Beacon frame is sent, the STA shall retain the buffered MSDU and announce the remaining fragments by transmitting an ATIM during the next ATIM Window.

i) If a STA is unable to transmit a buffered MSDU during the beacon interval in which it was announced, for example due to contention with other STAs, the STA shall retain the buffered MSDU and announce the MSDU again by transmitting an ATIM during the next ATIM Window.

j) Following the transmission of all buffered MSDUs, a STA may transmit MSDUs without announcement to STAs that are known to be in the Awake state for the current beacon interval due to an appropriate ATIM management or Beacon frame having been transmitted or received.

k) A STA may discard frames buffered for later transmission to power-saving STAs if the STA determines that the frame has been buffered for an excessive amount of time or if other conditions internal to the STA implementation make it desirable to discard buffered frames (e.g., buffer starvation). In no case shall a frame be discarded that has been buffered for less than dot11BeaconPeriod. The algorithm to manage this buffering is beyond the scope of this standard.

## 11.3 STA authentication and association

A STA keeps two state variables for each STA with which direct communication via the WM is needed:

— *Authentication state:* The values are unauthenticated and authenticated.

— *Association state:* The values are unassociated and associated.

These two variables create three local states for each remote STA:

— *State 1:* Initial start state, unauthenticated, unassociated.

— *State 2:* Authenticated, not associated.

— *State 3:* Authenticated and associated.


Copyright © 2007 IEEE. All rights reserved.

The relationships between these STA state variables and the services are given in Figure 11-6.

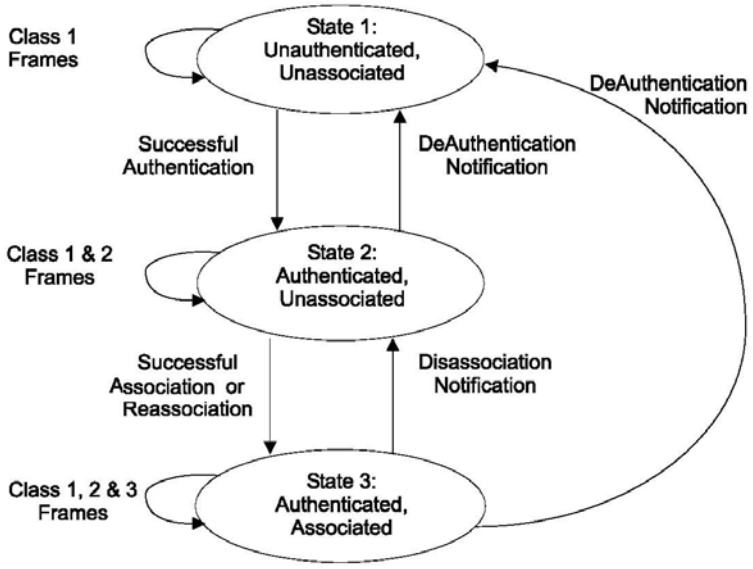



**Figure 11-6—Relationship between state variables and services**

The current state existing between the source and destination STAs determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs (see Clause 7). The state of the sending STA given by Figure 11-6 is with respect to the intended receiving STA. The allowed frame types are grouped into classes and the classes correspond to the STA state. In State 1, only Class 1 frames are allowed. In State 2, either Class 1 or Class 2 frames are allowed. In State 3, all frames are allowed (Classes 1, 2, and 3). The frame classes are defined as follows:

a) Class 1 frames (permitted from within States 1, 2, and 3)
   1) Control frames
      i) Request to send (RTS)
      ii) Clear to send (CTS)
      iii) Acknowledgment (ACK)
      iv) Contention-Free (CF)-End+ACK
      v) CF-End
   2) Management frames
      i) Probe request/response
      ii) Beacon
      iii) Authentication: Successful authentication enables a STA to exchange Class 2 frames. Unsuccessful authentication leaves the STA in State 1.
      iv) Deauthentication: Deauthentication notification when in State 2 or State 3 changes the STA's state to State 1. The STA shall become authenticated again prior to sending Class 2 frames. Deauthentication notification when in State 3 implies disassociation as well.
      v) Announcement traffic indication message (ATIM)
      vi) Spectrum Management Action: Within an IBSS, action frames are Class 1.
   3) Data frames
      i) Data: Data frames between STAs in an IBSS with frame control (FC) bits "To DS" and "From DS" both false.

Copyright © 2007 IEEE. All rights reserved.

b) Class 2 frames (if and only if authenticated; allowed from within States 2 and 3 only)
   1) Management frames
      i) Association request/response: Successful association enables Class 3 frames. Unsuccessful association leaves STA in State 2.
      ii) Reassociation request/response: Successful reassociation enables Class 3 frames. Unsuccessful reassociation leaves the STA in State 2 (with respect to the STA that was sent the reassociation message). Reassociation frames shall only be sent if the sending STA is already associated in the same ESS.
      iii) Disassociation: Disassociation notification when in State 3 changes a STA's state to State 2. This STA shall become associated again if it wishes to utilize the DS.

   If STA A receives a Class 2 frame with a unicast address in the Address 1 field from STA B that is not authenticated with STA A, STA A shall disallow the received Class 2 frame and send a deauthentication frame to STA B.

c) Class 3 frames (if and only if associated; allowed only from within State 3)
   1) Data frames
      i) Data subtypes: Data frames allowed, i.e., either the "To DS" or "From DS" FC bits may be set to true to utilize the DSS.
      ii) QoS data subtypes allowed to/from non-AP STA(s) that are associated with AP(s).
      iii) Data frames between STAs in a BSS with FC bits "To DS" and "From DS" both false.
   2) Management frames
      i) QoS, DLS, and Block Ack Action
   3) Control frames
      i) Power save (PS)-Poll
      ii) Action: Within an infrastucture BSS, action frames are Class 3.
      iii) Block Ack (BlockAck)
      iv) Block Ack Request (BlockAckReq)

   If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is authenticated but not associated with STA A, STA A shall disallow the received Class 3 frame and send a disassociation frame to STA B.

   If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is not authenticated with STA A, STA A shall disallow the received Class 3 frame and send a deauthentication frame to STA B.

   (The use of the word "receive" in this subclause refers to a frame that meets all of the filtering criteria specified in Clause 8 and Clause 9.)

### 11.3.1 Authentication and deauthentication

This subclause describes the procedures used for IEEE 802.11 authentication and deauthentication. The states used in this description are defined in 11.3.

#### 11.3.1.1 Authentication—originating STA

Upon receipt of an MLME-AUTHENTICATE.request primitive, the originating STA shall authenticate with the indicated STA using the following procedure:

a) In an ESS, or optionally in an IBSS, the STA shall execute the authentication mechanism described in 8.2.2.2.


Copyright © 2007 IEEE. All rights reserved.

b) If the authentication was successful, the state variable for the indicated STA shall be set to State 2.

c) The STA shall issue an MLME-AUTHENTICATE.confirm primitive to inform the SME of the result of the authentication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-AUTHENTICATE.request primitive.

#### 11.3.1.2 Authentication—destination STA

Upon receipt of an Authentication frame with authentication transaction sequence number equal to 1, the destination STA shall authenticate with the indicated STA using the following procedure:

a) The STA shall execute the authentication mechanism described in 8.2.2.2.

b) If the authentication was successful, the state variable for the indicated STA shall be set to State 2.

c) The STA shall issue an MLME-AUTHENTICATE.indication primitive to inform the SME of the authentication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-AUTHENTICATE.indication primitive.

If the STA is in an IBSS, if the SME decides to initiate an RSNA, and if the SME does not know the security policy of the peer, it may issue a unicast Probe Request frame to the peer by invoking an MLME-SCAN.request to discover the peer's security policy.

#### 11.3.1.3 Deauthentication—originating STA

Upon receipt of an MLME-DEAUTHENTICATE.request primitive, the originating STA shall deauthenticate with the indicated STA using the following procedure:

a) If the state variable for the indicated STA is in State 2 or State 3, the STA shall send a Deauthentication frame to the indicated STA.

b) The state variable for the indicated STA shall be set to State 1.

c) The STA shall issue an MLME-DEAUTHENTICATE.confirm primitive to inform the SME of the completion of the deauthentication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) before invoking the MLME-DEAUTHENTICATE.request primitive.

#### 11.3.1.4 Deauthentication—destination STA

Upon receipt of a Deauthentication frame, the destination STA shall deauthenticate with the indicated STA using the following procedure:

a) The state variable for the indicated STA shall be set to State 1.

b) The STA shall issue an MLME-DEAUTHENTICATE.indication primitive to inform the SME of the deauthentication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) upon receiving an MLME-DEAUTHENTICATE.indication primitive.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 487 of 1232

### 11.3.2 Association, reassociation, and disassociation

This subclause defines how a STA associates and reassociates with an AP and how it disassociates from it.

The states used in this description are defined in 11.3.

#### 11.3.2.1 STA association procedures

Upon receipt of an MLME-ASSOCIATE.request primitive, a STA shall associate with an AP via the following procedure:

a) The STA shall transmit an Association Request frame to an AP with which that STA is authenticated. If the MLME-ASSOCIATE.request primitive contained an RSN information element with only one pairwise cipher suite and only one authenticated key suite, this RSN information element shall be included in the Association Request frame.

b) If an Association Response frame is received with a status value of "successful," the STA is now associated with the AP. The state variable shall be set to State 3, and the MLME shall issue an MLME-ASSOCIATE.confirm primitive indicating the successful completion of the operation.

c) If an Association Response frame is received with a status value other than "successful" or the AssociateFailureTimeout expires, the STA is not associated with the AP. The MLME shall issue an MLME-ASSOCIATE.confirm primitive indicating the failure of the operation. The status code returned in the Association Response frame indicates the cause of the failed association attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA's Supported Rates information element, shall be corrected before the SME issues an MLME-ASSOCIATE.request for the same AP. If the status code indicates the association failed because of a reason that is not related to configuration, e.g., the AP is unable to support additional associations, the SME shall not issue an MLME-ASSOCIATE.request for the same AP until a period of at least 2 s has elapsed.

d) The SME shall establish an RSNA, or it shall enable WEP by calling MLME-SETPROTECTION.request primitive with ProtectType set to "Rx_Tx," or it shall do nothing if it does not wish to secure communication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-ASSOCIATE.request primitive.

#### 11.3.2.2 AP association procedures

When an Association Request frame is received from a STA, the AP shall associate with the STA using the following procedure:

a) If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA and terminate the association procedure.

b) In an RSNA, the AP shall check the values received in the RSN information element to see whether the values received match the AP's security policy. If not, the association shall not be accepted.

c) Upon receipt of an MLME-Associate.response service primitive, the AP shall transmit an Association Response with a status code as defined in 7.3.1.9. If the status value is "successful," the association identifier assigned to the STA shall be included in the response.

d) When the status value of the association is not successful, the AP shall indicate a specific reason for the failure to associate in the status code of the Association Response frame. If any status code value from Table 7-23 in 7.3.1.9 is an appropriate reason for the failure to associate, the AP shall indicate that status code value. The use of the unspecified reason value of the status code shall indicate the association failed for a reason that is unrelated to every other defined status code value in Table 7-23.


Copyright © 2007 IEEE. All rights reserved.

e) When the Association Response frame with a status value of "successful" is acknowledged by the STA, the STA is considered to be associated with this AP. The state variable for the STA shall be set to State 3.

f) The SME shall establish an RSNA, or it shall enable WEP by calling MLME-SETPROTECTION.request primitive with ProtectType set to "Rx_Tx," or it shall do nothing if it does not wish to secure communication.

g) The SME will inform the DS of the new association.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-ASSOCIATE.indication primitive.

#### 11.3.2.3 STA reassociation procedures

Upon receipt of an MLME-REASSOCIATE.request primitive, a STA shall reassociate with an AP via the following procedure:

a) If the state variable is in State 1, the STA shall inform the SME of the failure of the reassociation by issuing an MLME-REASSOCIATE.confirm primitive.

b) The STA shall transmit a Reassociation Request frame to the new AP. If the MLME-REASSOCIATE.request primitive contained an RSN information element with only one pairwise cipher suite and only one authenticated key suite, this RSN information element shall be included in the Reassociation Request frame.

c) If a Reassociation Response frame is received with a status value of "successful," the STA is now associated with the new AP. The state variable shall be set to State 3, and the MLME shall issue an MLME-REASSOCIATE.confirm primitive indicating the successful completion of the operation.

d) If a Reassociation Response frame is received with a status value other than "successful" or the AssociateFailureTimeout expires, the STA is not associated with the AP. The MLME shall issue an MLME-REASSOCIATE.confirm primitive indicating the failure of the operation. The status code returned in the Reassociation Response frame indicates the cause of the failed reassociation attempt. Any misconfiguration or parameter mismatch, e.g., data rates required as basic rates that the STA did not indicate as supported in the STA's Supported Rates information element, shall be corrected before the SME issues an MLME-REASSOCIATE.request for the same AP. If the status code indicates the reassociation failed because of a reason that is not related to configuration, e.g., the AP is unable to support additional associations, the SME shall not issue an MLME-REASSOCIATE.request for the same AP until a period of at least 2 s has elapsed.

e) The SME shall establish an RSNA, or it shall enable WEP by calling MLME-SETPROTECTION.request primitive with ProtectType set to "Rx_Tx," or it shall do nothing if it does not wish to secure communication.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-REASSOCIATE.request primitive.

#### 11.3.2.4 AP reassociation procedures

Whenever a Reassociation Request frame is received from a STA, the AP uses the following procedure to support reassociation:

a) If the STA is not authenticated, the AP shall transmit a Deauthentication frame to the STA and terminate the reassociation procedure.

b) In an RSNA, the AP shall check the values received in the RSN information element to see whether the values received match the AP's security policy. If not, the association shall not be accepted.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 489 of 1232

c) Upon receipt of an MLME-Associate.response service primitive, the AP shall transmit a Reassociation Response frame with a status code as defined in 7.3.1.9. If the status value is "successful," the association identifier assigned to the STA shall be included in the response.

d) When the Reassociation Response frame with a status value of "successful" is acknowledged by the STA, the STA is considered to be associated with this AP. The state variable for the STA shall be set to State 3.

e) When the status value of the reassociation is not successful, the AP shall indicate a specific reason for the failure to reassociate in the status code of the Reassociation Response frame. If any status code value from Table 7-23 in 7.3.1.9 is an appropriate reason for the failure to reassociate, the AP shall indicate that status code value. The use of the unspecified reason value of the status code shall indicate the reassociation failed for a reason that is unrelated to every other defined status code value in Table 7-23.

f) The SME shall establish an RSNA, or it shall enable WEP by calling MLME-SETPROTECTION.request primitive with ProtectType set to "Rx_Tx," or it shall do nothing if it does not wish to secure communication.

g) The SME will inform the DS of the new association.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-REASSOCIATE.indication primitive.

#### 11.3.2.5 STA disassociation procedures

Upon receipt of an MLME-DISASSOCIATE.request primitive, an associated STA shall disassociate from an AP using the following procedure:

a) The STA shall transmit a Disassociation frame to the AP with which that STA is associated.

b) The state variable for the AP shall be set to State 2 if and only if it was not State 1.

c) The MLME shall issue an MLME-DISASSOCIATE.confirm primitive indicating the successful completion of the operation.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) before invoking the MLME-DISASSOCIATE.request primitive.

#### 11.3.2.6 Non-AP STA disassociation receipt procedure

Upon receipt of a Disassociation frame, a STA shall operate as follows:

a) The MLME shall issue an MLME-DISASSOCIATE.indication with the ReasonCode parameter set to the value of the reason code received in the Disassociation frame.

b) The state variable for the AP shall be set to State 2 if and only if it was not State 1.

c) If the reason code indicates a configuration or parameter mismatch as the cause of the disassociation, the STA shall not attempt to associate or reassociate with the AP sending the Disassociation frame until the configuration or parameter mismatch has been corrected.

d) If the reason code indicates the STA was disassociated for a reason other than configuration or parameter mismatch, the STA shall not attempt to associate or reassociate with the AP sending the Disassociation frame until a period of 2 s has elapsed.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) before invoking the MLME-DISASSOCIATE.request primitive.


Copyright © 2007 IEEE. All rights reserved.

#### 11.3.2.7 AP disassociation initiation procedure

Upon receipt of an MLME-DISASSOCIATE.request, an AP shall use the following procedure when disassociating a STA:

a) The AP shall send a Disassociation frame to the STA being disassociated.

b) The AP shall indicate a specific reason for the disassociation in the Reason Code field of the Disassociation frame. If any reason code value other than the unspecified reason code from Table 7-22 of 7.3.1.7 is appropriate for indicating the reason for the disassociation, the AP shall indicate that reason code value. The use of the unspecified reason value shall indicate the STA was disassociated for a reason unrelated to all defined reason code values defined in Table 7-22.

c) The state variable for the STA shall be set to State 2.

d) The SME shall update the DS.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) upon receiving a MLME-DISASSOCIATE.indication primitive.

#### 11.3.2.8 AP disassociation receipt procedure

Upon receipt of a Disassociation frame from an associated STA, the AP shall disassociate the STA via the following procedure:

a) The state variable for the STA shall be set to State 2.

b) The MLME shall issue an MLME-DISASSOCIATE.indication primitive to inform the SME of the disassociation.

c) The SME will update the DS.

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) and by invoking MLME-SETPROTECTION.request(None) upon receiving a MLME-DISASSOCIATE.indication primitive.

## 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 STA that sets up the stream) associated with it. The AP 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 STA by its TSID and direction, and identified within the HC by a combination of TSID, direction, and non-AP STA address.

It is always the responsibility of the non-AP STA to initiate the creation of a TS regardless of its direction.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 491 of 1232

In the direct-link case, it is the responsibility of the non-AP STA that is going to send the data to create the TS. In this case, the non-AP STA 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 STAs concerning the TS: the originator can discover the capabilities of the recipient (rates, BlockAck) using the DLS.

In the case of traffic relayed by an AP, the sending and receiving non-AP STAs 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 STA may simultaneously support up to eight TSs from the HC to itself and up to eight TSs from itself to other STAs, 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 STA. 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 11-7 summarizes the TS lifecycle (using the HMSC syntax defined in ITU-T Recommendation Z.120 (2004).

Initially a TS is inactive. A STA shall not transmit any QoS data frames using an inactive TS.

Following a successful TS setup initiated by the non-AP STA, the TS becomes active, and either the non-AP STA 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 STA. 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 STA 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.

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.


Copyright © 2007 IEEE. All rights reserved.

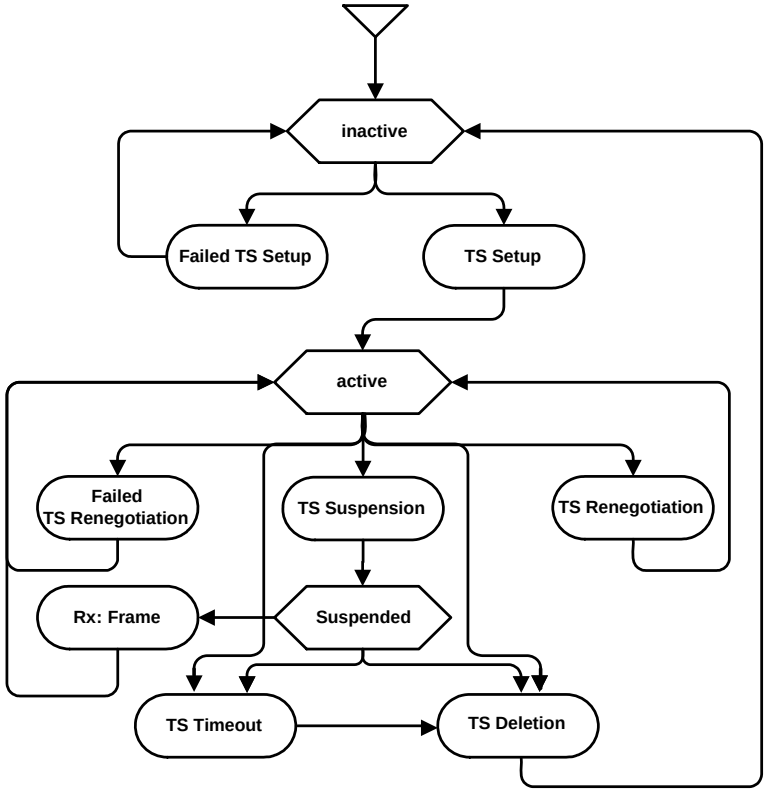



**Figure 11-7—TS lifecycle**

### 11.4.4 TS setup

Figure 11-8 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.

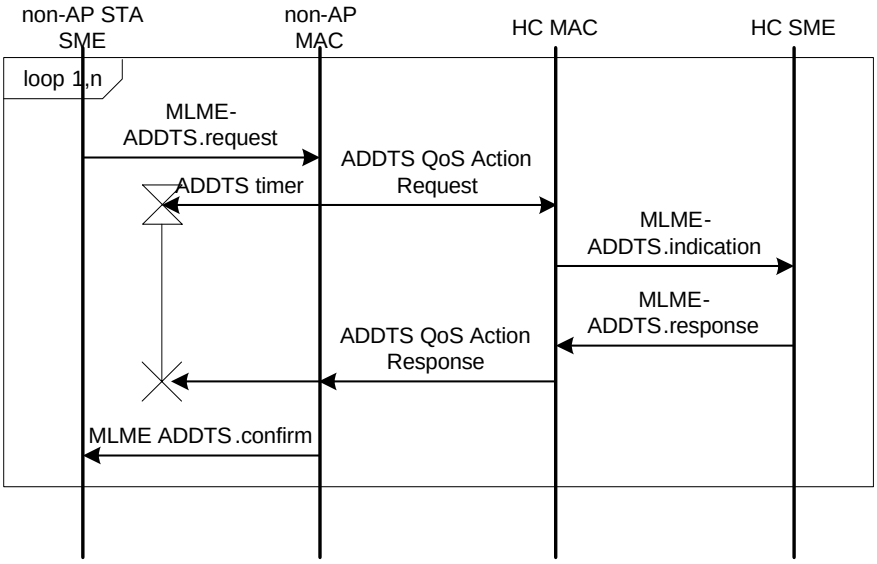



**Figure 11-8—TS setup**

Copyright © 2007 IEEE. All rights reserved.

The non-AP STA 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 standard. 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 STA 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 10.3.24.4.2.

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 11-2.

**Table 11-2—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 STA 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 STA SME decides whether the response meets its needs. How it does this is beyond the scope of this standard. 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 STA 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 STA that originated the ADDTS Request frame. If a non-AP STA 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.


Copyright © 2007 IEEE. All rights reserved.

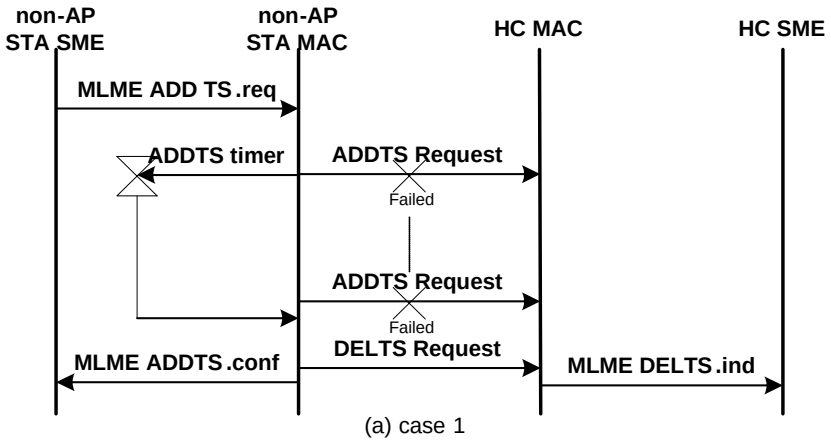
### 11.4.5 Failed TS setup

There are two possible types of failed TS setup:

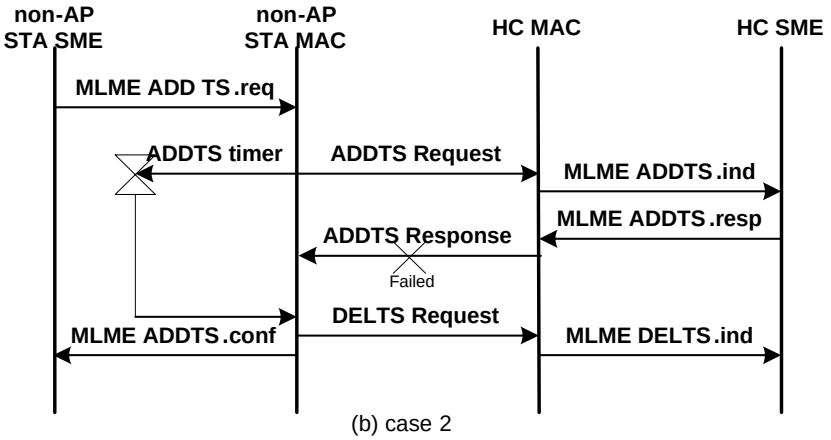
a) The transmission of ADDTS Request frame failed.

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 11-9 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 STA 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.




(a) case 1




(b) case 2

**Figure 11-9—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 STA, the MSDUs are transmitted using QoS data frames, when the non-AP STA is polled by the HC.

Copyright © 2007 IEEE. All rights reserved.

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 STA-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 11-3.

**Table 11-3—Encoding of ReasonCode to Reason Code field value for DELTS**

| ReasonCode | Reason Code field |
|---|---|
| STA_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 11-10 shows the case of TS deletion initiated by the non-AP STA and the case of TS deletion initiated by the HC.

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 STA and AP to clear their state, and the non-AP STA will have to reinitiate the setup.


Copyright © 2007 IEEE. All rights reserved.

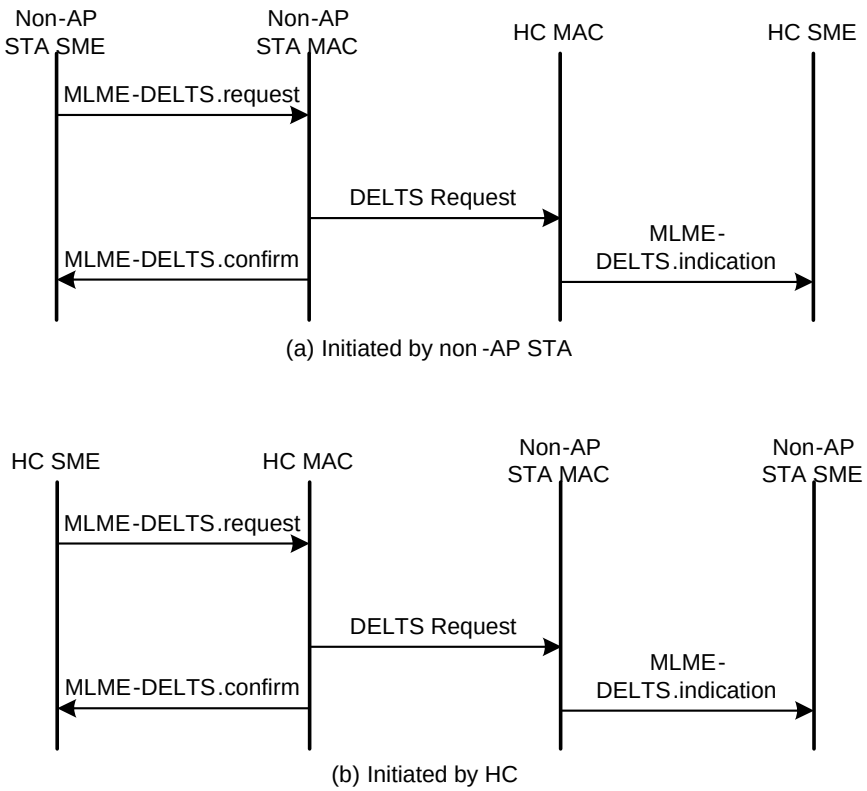



**Figure 11-10—TS deletion**

### 11.4.8 TS timeout

TS timeout is detected within the HC MAC when no traffic is detected on the TS within the inactivity timeout 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 STA is actually using the assigned TXOP for the given TSID.

Any other use of a polled TXOP delivered to the non-AP STA is considered to be activity on all direct-link TS associated with that non-AP STA. Detection of inactivity of this type is optional.

Copyright © 2007 IEEE. All rights reserved.

In response to an inactivity timeout, the HC shall send a DELTS frame to the non-AP STA 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 11-11.

#### 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 STA 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 AP 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 11-12.

##### 11.5.1.1 Procedure at the originator

Upon receipt of an MLME-ADDBA.request primitive, an initiating STA that intends to send QoS data frames under the Block Ack mechanism shall set up the Block Ack using the following procedure:

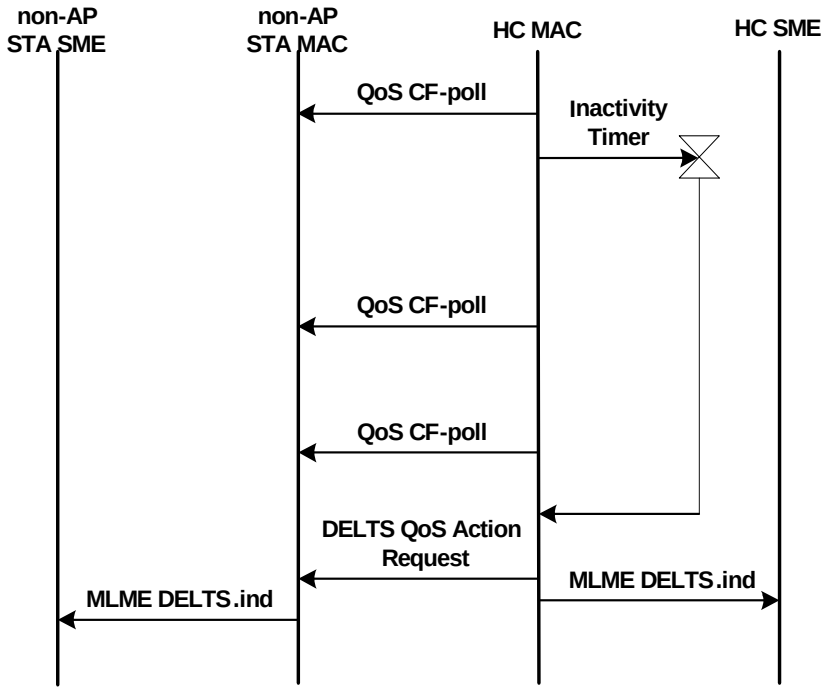
a) Check whether the intended peer STA is capable of participating in the Block Ack mechanism by discovering and examining its “Delayed Block Ack” and “Immediate Block Ack” capability bits. If the recipient is capable of participating, the originator sends an ADDBA frame indicating the TID and the buffer 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 STA has established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the successful completion of the Block Ack setup.

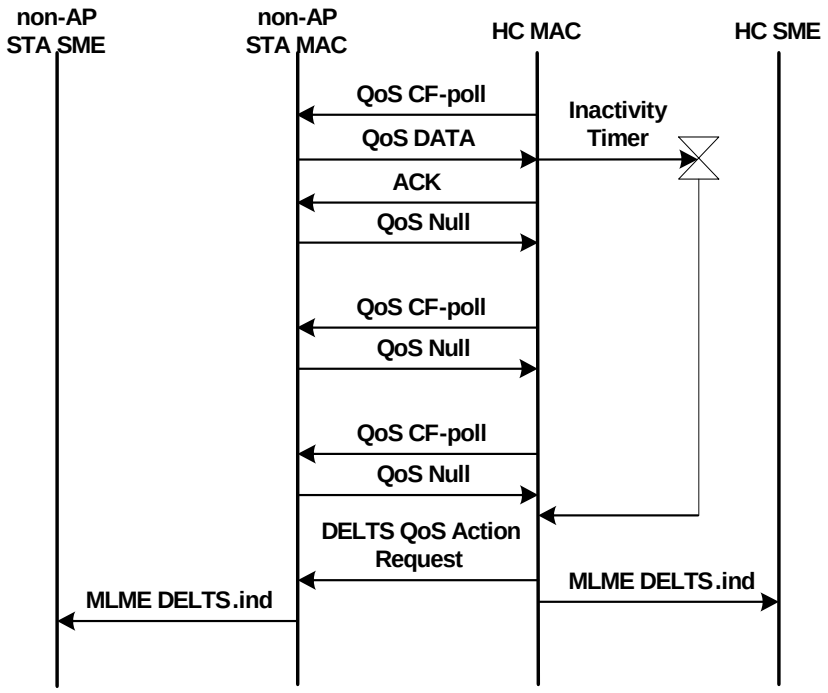
c) If an ADDBA Response frame is received with the matching dialog token and the TID and with a status code set to a value other than 0, the STA has not established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive indicating the failure of the Block Ack setup.

d) If there is no response from the recipient within dot11ADDBAFailureTimeout, the STA has not established a Block Ack mechanism with the recipient STA; and the MLME shall issue an MLME-ADDBA.confirm primitive with a result code of TIMEOUT.


Copyright © 2007 IEEE. All rights reserved.




(a) No response from STA




(b) No Data frames from STA

**Figure 11-11—TS timeout**

Copyright © 2007 IEEE. All rights reserved.

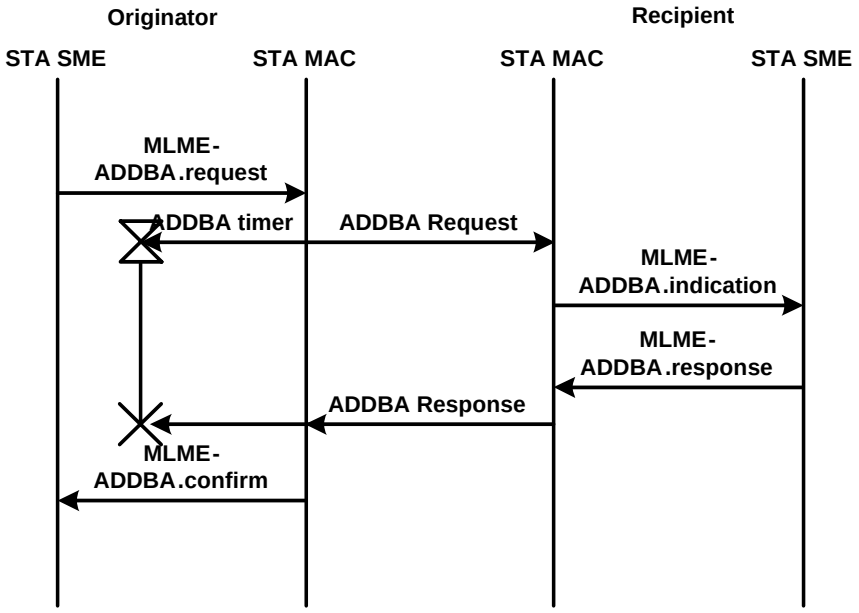



**Figure 11-12—Block Ack setup**

#### 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 STA, the MLME shall issue an MLME-ADDBA.indication primitive.

b) Upon receipt of the MLME-ADDBA.response primitive, the STA 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 11-4.

**Table 11-4—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 STAs is described in 11.5.2.1 and 11.5.2.2 and illustrated in Figure 11-13.


Copyright © 2007 IEEE. All rights reserved.

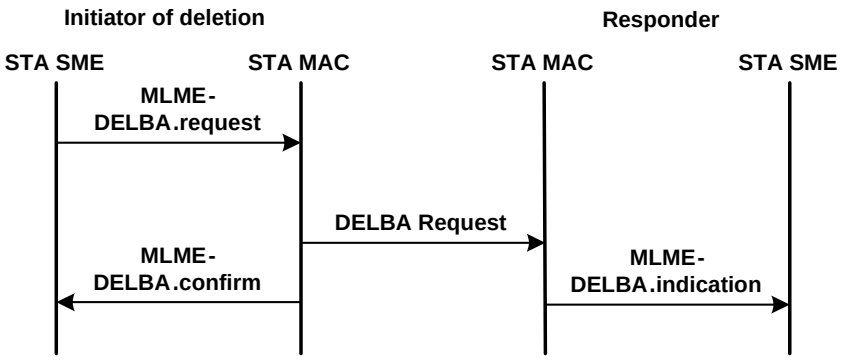



**Figure 11-13—Block Ack deletion**

#### 11.5.2.1 Procedure at the initiator of the Block Ack teardown

Upon receipt of an MLME-DELBA.request primitive, the STA MAC/MLME shall tear down the Block Ack using the following procedure:

a) The STA 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 11-5.

**Table 11-5—Encoding of ReasonCode to Reason Code field value for DELBA**

| ReasonCode | Reason Code field |
|---|---|
| STA_LEAVING | 36 |
| END_BA | 37 |
| UNKNOWN_BA | 38 |
| TIMEOUT | 39 |

#### 11.5.2.2 Procedure at the recipient of the DELBA frame

A STA 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 STA shall send a DELBA frame to the peer STA with the Reason Code field set to TIMEOUT

Copyright © 2007 IEEE. All rights reserved.

and shall issue a MLME-DELBA.indication primitive with the ReasonCode parameter having a value of TIMEOUT. The procedure is illustrated in Figure 11-14.

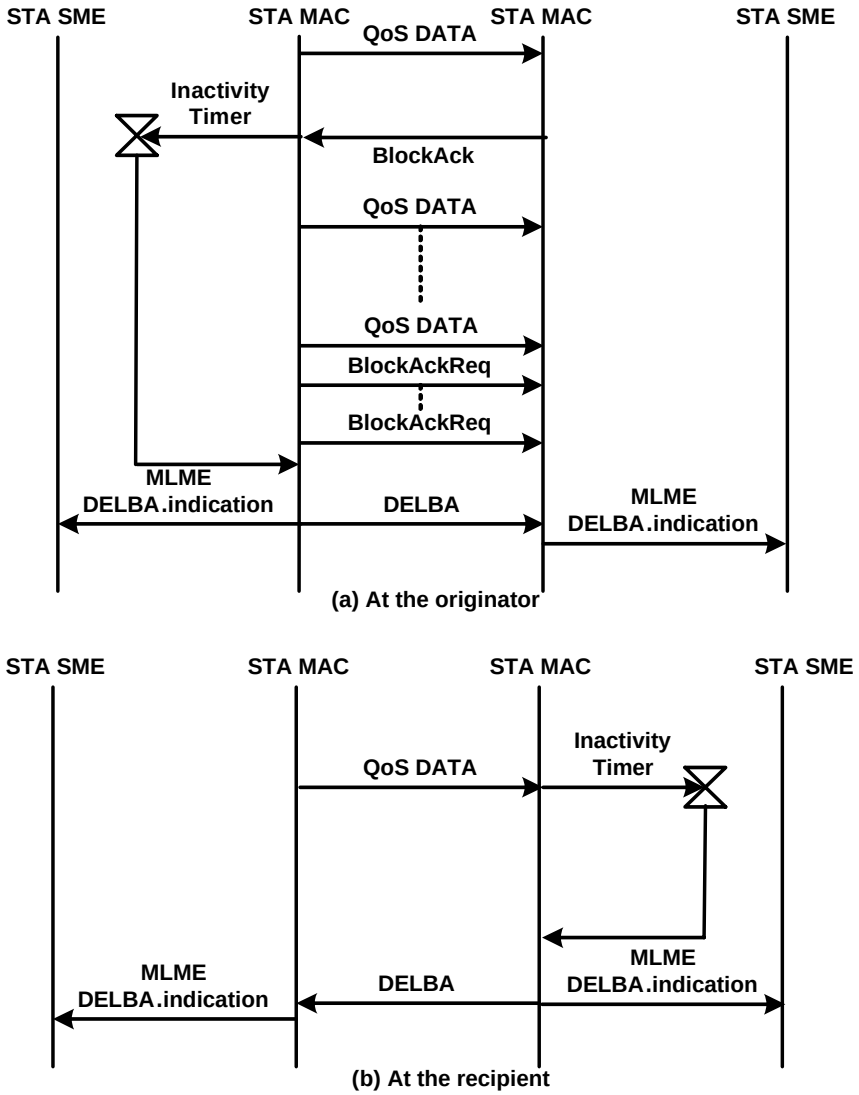



**Figure 11-14—Error recovery by the receiver upon a peer failure**

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 STA 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.


Copyright © 2007 IEEE. All rights reserved.

The originator STA may send an ADDBA Request frame in order to update Block ACK Timeout Value. If the updated ADDBA Request frame is accepted, both STAs initialize the timer to detect Block ACK timeout. 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 standard. However, explanation on how the constructs in this standard 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 STA to know exactly when each Sync packet is transmitted. A receiving STA can similarly note the time when each Sync packet is received as well as the sequence number for that frame. The receiving STA 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 STA 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 STAs 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.

This is an example implementation using the higher layer timer synchronization feature. Other implementations are possible.

Copyright © 2007 IEEE. All rights reserved.

### 11.6.2 Procedure at the STA

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 may transmit frames directly to another STA 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 AP. 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 an IBSS, where frames are always sent directly from one STA to another.

The handshake involved in the setup is illustrated in Figure 11-15 and involves the four steps listed after the figure.

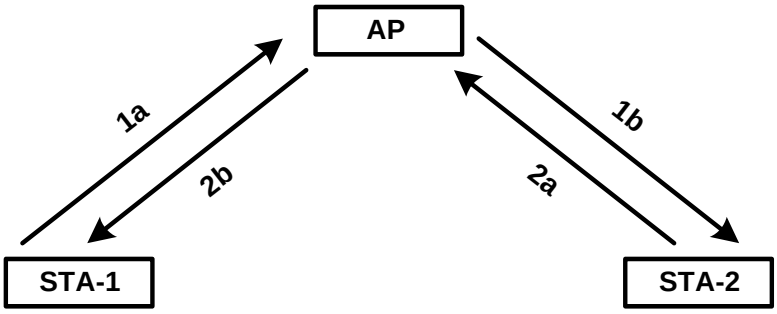



**Figure 11-15—The four steps involved in direct-link handshake**

a) A STA, STA-1, that intends to exchange frames directly with another non-AP STA, STA-2, invokes DLS and sends a DLS Request frame to the AP (step 1a in Figure 11-15). This request contains the rate set, capabilities of STA-1, and the MAC addresses of STA-1 and STA-2.

b) If STA-2 is associated in the BSS, direct streams are allowed in the policy of the BSS, and STA-2 is indeed a QoS STA, then the AP forwards the DLS Request frame to the recipient, STA-2 (step 1b in Figure 11-15).


Copyright © 2007 IEEE. All rights reserved.

c) If STA-2 accepts the direct stream, it sends a DLS Response frame to the AP (step 2a in Figure 11-15), which contains the rate set, (extended) capabilities of STA-2, and the MAC addresses of STA-1 and STA-2.

d) The AP forwards the DLS Response frame to STA-1 (step 2b in Figure 11-15), after which the direct link becomes active and frames can be sent from STA-1 to STA-2 and from STA-2 to STA-1.

When a STA transitions to a differnent AP after a DLS is set up, the DLS shall be torn down as described in 11.7.4.

### 11.7.1 DLS procedures

The DLS message flow is illustrated in Figure 11-16.

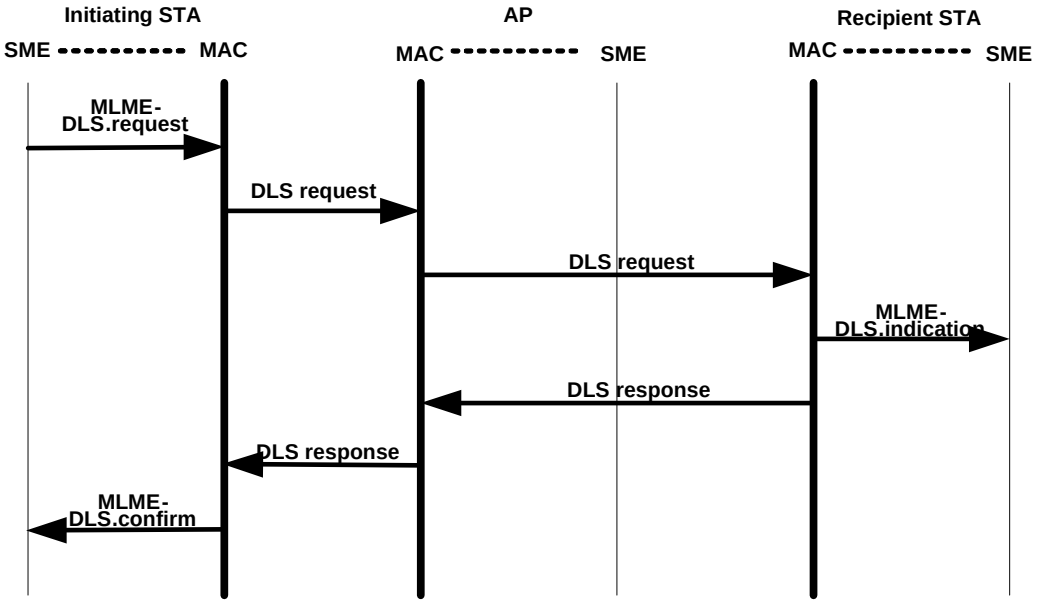



**Figure 11-16—DLS message flow**

#### 11.7.1.1 Setup procedure at the QoS STA

A STA shall maintain a list of all STAs with which a direct link has been established.

Upon receipt of an MLME-DLS.request primitive from the SME, the STA 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 STAs 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 11-15). If the STA does not receive a DLS Response frame within DLSResponseTimeout after sending the DLS Request frame, the STA shall issue an MLME-DLS.confirm primitive with the result code of TIMEOUT.

Upon receipt of the DLS Request frame from the AP (step 1b in Figure 11-15), the STA MAC shall send to the AP a DLS Response frame (step 2a in Figure 11-15) with a result code of

Copyright © 2007 IEEE. All rights reserved.

— SUCCESS if the STA is willing to participate in the direct link with the source STA. The STA shall also issue an MLME-DLS.indication primitive to the SME and shall add the source STA to the list of the STAs, if not already present, with which direct link has been established.

— REFUSED if the STA is not willing to participate in the direct link.

Upon receipt of the DLS Response frame from the AP (step 2b in Figure 11-15), the STA shall issue an MLME-DLS.confirm primitive.

— If the result code is SUCCESS, the direct link is considered to be established with the destination STA in the DLS Response frame, and the STA MAC shall add the destination STA to the list of STAs 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 AP

Upon receipt of the DLS Request frame (step 1a in Figure 11-15), the AP shall

— Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not Allowed in the BSS, if direct links are not allowed in the BSS (step 2b in Figure 11-15).

— Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not Present, if the destination STA is not present in the BSS (step 2b in Figure 11-15).

— Send DLS Response frame to the STA that sent the DLS Request frame with a result code of Not a STA, if the destination STA does not have QoS facility (step 2b in Figure 11-15).

— Send the DLS Request frame, with all fields having the same value as the DLS Request frame received by the AP, to the destination STA (step 1b in Figure 11-15).

Upon receipt of the DLS Response frame from a STA (step 2a in Figure 11-15), the AP shall send DLS Response frame to the source STA (step 2b in Figure 11-15).

The mapping of Status Code field values to ResultCode parameter values in an MLME-DLS.confirm primitive is defined in Table 11-6.

**Table 11-6—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_QOS_STA | 50 |

### 11.7.2 Data transfer after setup

Both the STAs may use direct link for data transfers using any of the access mechanisms defined in this standard. STAs may also set up Block Ack if needed. If needed, the STAs 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.


Copyright © 2007 IEEE. All rights reserved.

### 11.7.3 DLS teardown

The DLS teardown procedure is divided into STA-initiated and AP-initiated DLS teardown. The STA-initiated DLS teardown message flow is illustrated in Figure 11-17.

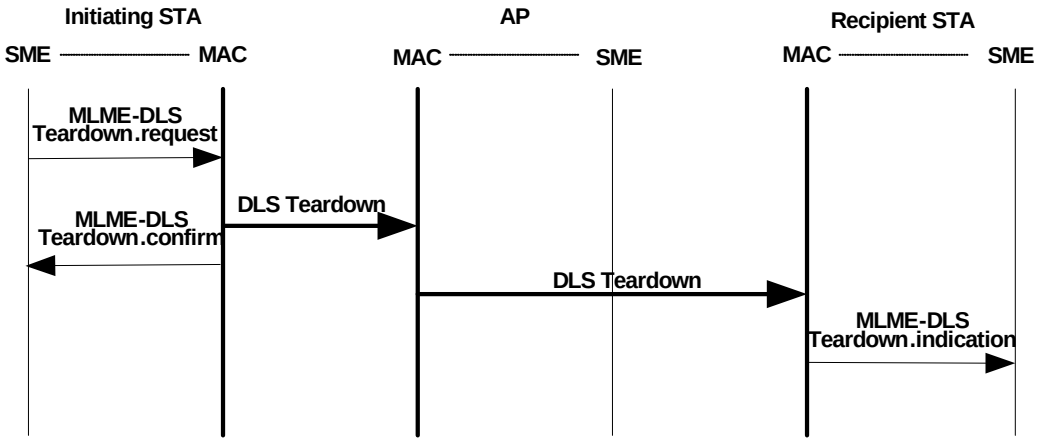



**Figure 11-17—STA-initiated DLS teardown message flow**

#### 11.7.3.1 STA-initiated DLS teardown procedure

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the STA shall initiate the teardown of the direct link by sending the DLS Teardown frame to the AP. The applicable values of ReasonCode and their encoding to the Reason Code field (see 7.3.1.7) values are defined in Table 11-7.

**Table 11-7—Encoding of ReasonCode to Reason Code field value for DLS teardown**

| ReasonCode | Reason Code field | Applicable at |
|---|---|---|
| STA_LEAVING | 36 | Non-AP STA |
| END_DLS | 37 | Non-AP STA |
| UNKNOWN_DLS | 38 | Non-AP STA |
| TIMEOUT | 39 | Non-AP STA |
| PEERKEY_MISMATCH | 45 | AP |
| PEER_INITIATED | 46 | AP and Non-AP STA |
| AP_INITIATED | 47 | Non-AP STA |

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 AP), the STA shall issue an MLME-DLSTeardown.indication primitive to the SME and shall delete the STA from the list of the STAs with which direct link has been established.

The DLS teardown procedure shall apply to a specific DLS session, as each STA may have multiple simultaneous DLS sessions with other STAs.

Copyright © 2007 IEEE. All rights reserved.

Prior to disassociation/deauthentication from the AP, the STA (STA1) shall initiate the teardown of any direct links it has by sending a DLS Teardown frame to the AP. If the DLS Teardown frame's Max Retry Limit was reached with no response from the AP, the STA shall send a DLS Teardown frame to its peer DLS STA (STA2).

A recipient STA (STA2), either on expiry of its DLSResponseTimeout or on receipt of a DLS Teardown frame with ReasonCode set to PEER_INITIATED (from STA1), shall send a DLS Teardown frame to the AP with ReasonCode set to PEER_INITIATED.

An AP that receives a DLS Teardown frame with ReasonCode set to PEER_INITIATED should send a DLS Teardown frame to any STAs that have a DLS link established with STA1.

NOTE—The failed STA may reestablish its DLS link according to 11.7.

#### 11.7.3.2 Teardown procedure at the AP

Upon receipt of the DLS Teardown frame from a STA, the AP shall send a DLS Teardown frame to the destination STA.

Upon receipt of MLME-DLSTeardown.request primitive from the SME, the AP shall announce the tearing down of the direct link by sending the DLS Teardown frame to the two STAs using the direct link. The only applicable values of the ReasonCode are PeerKey_MISMATCH and STA_LEAVING. The encoding to Reason Code field values is defined in Table 11-7.

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.

#### 11.7.3.3 AP-initiated DLS teardown procedure

The AP-initiated DLS teardown procedure may be used in cases where the STA is unable to initiate the DLS teardown. The AP should initiate a DLS teardown procedure when the AP detects that either end of a DLS link has left the BSS without teardown of the DLS link, e.g., through receipt of a deauthentication frame or receipt of a deassociation frame.

If there are one or more STAs with open DLS connections with the STA being removed, the AP shall send a DLS Teardown frame to each such STA with ReasonCode AP_INITIATED.

NOTE—The AP can also initiate DLS teardown for implementation-dependent reasons.

### 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 nonzero 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 AP, 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 AP. The procedure is illustrated in Figure 14-9.

If there has been a TS setup for the data transfer, it is the responsibility of the STAs to renegotiate the parameters of the TSPEC with the HC.


Copyright © 2007 IEEE. All rights reserved.

When two STAs have set up a direct link, either STA may send DLS Request frames in order to update the DLS timeout value. If the updated DLS Request frame is accepted, both STAs 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

PeerKey Handshake, defined in 8.5.8, is used to establish the keys needed to enable secure DLS. The PeerKey message exchange shall be commenced after the DLS establishment and completed prior to initiation of the DLS data frame exchange.

The STKSA remains valid even if the STA disassociates from the originating AP, but the STKSA shall be deleted before a STA attempts another association or reassociation. If an AP transmits a Deauthenticate or Disassociate message to a STA, the AP shall also initiate teardowns for any existing DLS. The DLS STKs shall be deleted when the DLS teardown messages is sent or received.

## 11.8 TPC procedures

Regulations that apply to the 5GHz band in most regulatory domains require RLANs operating in the 5 GHz band to use transmitter power control, involving specification of a regulatory maximum transmit power and a mitigation requirement for each allowed channel, to reduce interference with satellite services. This standard describes such a mechanism, referred to as transmit power control (TPC).

This subclause describes TPC procedures that may satisfy needs in many regulatory domains and other frequency bands and may be useful for other purposes (e.g., reduction of interference, range control, reduction of power consumption).

STAs shall use the TPC procedures defined in this subclause if dot11SpectrumManagementRequired is true. dot11SpectrumManagementRequired shall be set to TRUE when regulatory authorities require TPC. It may also be set to TRUE in other circumstances. The TPC procedures provide for the following:

— Association of STAs with an AP in a BSS based on the STAs' power capability (see 11.8.1).

— Specification of regulatory and local maximum transmit power levels for the current channel (see 11.8.2).

— Selection of a transmit power for each transmission in a channel within constraints imposed by regulatory and local requirements (see 11.8.3).

— Adaptation of transmit power based on a range of information, including path loss and link margin estimates (see 11.8.4).

For the purposes of TPC, the following statements apply:

— A STA with dot11SpectrumManagementRequired set to TRUE shall not operate in a BSS or IBSS unless the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from other STAs in the BSS or IBSS, *with the following exception*.

— A STA may operate when the Spectrum Management bit is set to 0 if the STA can determine that it is in a regulatory domain that does not require TPC or can ensure that it will meet regulatory requirements even if TPC is not employed. Potential methods for determining the regulatory domain include receiving a country indication in the Beacon frame, user confirmation, or configuration information within the device. Potential methods to ensure regulations are met even if TPC is not employed include using a transmit power that is below the legal maximum (including any mitigation factor).

— A STA shall set dot11SpectrumManagementRequired to TRUE before associating with a BSS or IBSS in which the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from the BSS or IBSS.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 509 of 1232

— APs may allow association of devices that do not have the Spectrum Management bit set to 1 in the Capability Information field in Association Request frames and Reassociation Request frames received from the STA to account for the existence of legacy devices that do not support TPC, but do meet regulatory requirements.

### 11.8.1 Association based on transmit power capability

A STA shall provide an AP with its minimum and maximum transmit power capability for the current channel when associating or reassociating, using a Power Capability element in Association Request frames or Reassociation Request frames.

An AP may use the minimum and maximum transmit power capability of associated STAs as an input into the algorithm used to determine the local transmit power constraint for any BSS it maintains. The specification of the algorithm is beyond the scope of this standard.

An AP may reject an association or reassociation request from a STA if it considers the STA's minimum or maximum transmit power capability is unacceptable. For example, a STA's power capability might be unacceptable if it violates local regulatory constraints or increases the probability of hidden STAs by a significant degree. The criteria for accepting or rejecting an association or reassociation on the basis of transmit power capability are beyond the scope of this standard.

### 11.8.2 Specification of regulatory and local maximum transmit power levels

A STA shall determine a regulatory maximum transmit power for the current channel. The STA shall use the minimum of the following:

— Any regulatory maximum transmit power received in a Country element from the AP in its BSS or another STA in its IBSS and

— Any regulatory maximum transmit power for the channel in the current regulatory domain known by the STA from other sources.

A STA shall determine a local maximum transmit power for the current channel. The STA shall use the minimum of the following:

— Any local maximum transmit power received in the combination of a Country element and a Power Constraint element from the AP in its BSS or another STA in its IBSS and

— Any local maximum transmit power for the channel regulatory domain known by the STA from other sources.

Any calculation of the local maximum transmit power for the channel shall ensure the mitigation requirements for the channel in the current regulatory domain can be satisfied. The conservative approach is to set the local maximum transmit power level equal to the regulatory maximum transmit power level minus the mitigation requirement. However, it may be possible to satisfy the mitigation requirement using a higher local maximum transmit power level. A lower local maximum transmit power level may be used for other purposes (e.g., range control, reduction of interference).

The regulatory and local maximum transmit powers may change in a STA during the life of a BSS. However, network stability should be considered when deciding how often or by how much these maximums are changed. The regulatory and local maximum transmit powers shall not change during the life of an IBSS.

An AP in a BSS and a STA in an IBSS shall advertise the regulatory maximum transmit power for the current channel in Beacon frames and Probe Response frames using a Country element. An AP in a BSS and a STA in an IBSS shall advertise the local maximum transmit power for the current channel in Beacon


Copyright © 2007 IEEE. All rights reserved.

frames and Probe Response frames using the combination of a Country element and a Power Constraint element.

### 11.8.3 Selection of a transmit power

A STA may select any transmit power for transmissions in a channel within the following constraints:

— A STA shall determine a regulatory maximum transmit power and a local maximum transmit power for a channel in the current regulatory domain before transmitting in the channel.

— An AP shall use a transmit power less than or equal to the regulatory maximum transmit power level for the channel. However, the AP shall also ensure the regulatory mitigation requirement is met.

— A STA that is not an AP shall use a transmit power less than or equal to the local maximum transmit power level for the channel.

### 11.8.4 Adaptation of the transmit power

A STA may use any criteria, and in particular any path loss and link margin estimates, to dynamically adapt the transmit power for transmissions of an MPDU to another STA. The adaptation methods or criteria are beyond the scope of this standard.

A STA may use a TPC Request frame to request another STA to respond with a TPC Report frame containing link margin and transmit power information. A STA receiving a TPC Request frame shall respond with a TPC Report frame containing the power used to transmit the response in the Transmit Power field and the estimated link margin in a Link Margin field.

An AP in a BSS or a STA in an IBSS shall autonomously include a TPC Report element with the Link Margin field set to 0 and containing transmit power information in the Transmit Power field in any Beacon frame or Probe Response frame it transmits.

## 11.9 DFS procedures

Regulations that apply to the 5GHz band in most regulatory domains require RLANs operating in the 5 GHz band to implement a mechanism to avoid co-channel operation with radar systems and to ensure uniform utilization of available channels. This standard describes such a mechanism, referred to as dynamic frequency selection (DFS).

This subclause describes DFS procedures that can be used to satisfy these and similar future regulatory requirements. The procedures may also satisfy comparable needs in other frequency bands and may be useful for other purposes.

STAs shall use the DFS procedures defined in this subclause if dot11SpectrumManagementRequired is true. dot11SpectrumManagementRequired shall be set to TRUE when regulatory authorities require DFS. It may also be set to TRUE in other circumstances. The DFS procedures provide for the following:

— Associating STAs with an AP in a BSS based on the STAs' supported channels (see 11.9.1).

— Quieting the current channel so it can be tested for the presence of radar with less interference from other STAs (see 11.9.2).

— Testing channels for radar before using a channel and while operating in a channel (see 11.9.3).

— Discontinuing operations after detecting radar in the current channel to avoid further interfering with the radar (see 11.9.4).

— Detecting radar in the current and other channels based on regulatory requirements (see 11.9.5).

— Requesting and reporting measurements in the current and other channels (see 11.9.6).

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 511 of 1232

— Selecting and advertising a new channel to assist the migration of a BSS or IBSS after radar is detected (see 11.9.7).

For the purposes of DFS, the following statements apply:

— A STA with dot11SpectrumManagementRequired set to TRUE shall not operate in a BSS or IBSS unless the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from other STAs in the BSS or IBSS, *with the following exception*.

— A STA may operate when the Spectrum Management bit is set to 0 if the STA can determine that it is in a regulatory domain that does not require DFS or can ensure that it will meet regulatory requirements even if DFS is not employed. Potential methods for determining the regulatory domain include receiving a country indication in the Beacon frame, user confirmation, or configuration information within the device. Potential methods to ensure regulations are met even if DFS is not employed include independently detecting radar and ceasing operation on channels on which radar is detected.

— A STA shall set dot11SpectrumManagementRequired to TRUE before associating with a BSS or IBSS in which the Spectrum Management bit is set to 1 in the Capability Information field in Beacon frames and Probe Response frames received from the BSS or IBSS.

— APs may allow association of devices that do not have the Spectrum Management bit set to 1 in the Capability Information field in Association Request frames and Reassociation Request frames received from a STA to account for the existence of legacy devices that do not support DFS, but do meet regulatory requirements.

### 11.9.1 Association based on supported channels

A STA shall provide an AP with a list of the channels in which it can operate when associating or reassociating using a Supported Channels element in Association Request frames or Reassociation Request frames.

An AP may use the supported channels list for associated STAs as an input into an algorithm used to select a new channel for the BSS. The specification of the algorithm is beyond the scope of this standard.

An AP may reject an association or reassociation request from a STA if it considers the STA's supported channel list is unacceptable. For example, a STA's supported channel list might be unacceptable if it can operate only in a limited number of channels. The criteria for accepting or rejecting associations or reassociations are beyond the scope of this standard.

### 11.9.2 Quieting channels for testing

An AP in a BSS may schedule quiet intervals by transmitting one or more Quiet elements in Beacon frames and Probe Response frames. The AP may stop scheduling quiet intervals or change the value of the Quiet Period field, the Quiet Duration field, and the Quiet Offset field in Quiet elements as required. Only the most recently received Beacon frame or Probe Response frame defines all future quiet intervals; therefore, quiet intervals based on older Beacon frames or Probe Response frames shall be discarded.

Only the STA starting an IBSS may specify a schedule of quiet intervals, by transmitting one or more Quiet elements in the first Beacon frame establishing the IBSS. All STAs in an IBSS shall continue these quiet interval schedules by including appropriate Quiet elements in any transmitted Beacon frames or Probe Response frames.

Multiple independent quiet intervals may be scheduled, to ensure that not all quiet intervals have the same timing relationship to TBTT, by including multiple Quiet elements in Beacon frames or Probe Response frames.


Copyright © 2007 IEEE. All rights reserved.

Control of the channel is lost at the start of a quiet interval, and the NAV is set by all the STAs in the BSS or IBSS for the length of the quiet interval. Transmission of any MPDU and any associated acknowledgment shall be complete before the start of the quiet interval. If, before starting transmission of an MPDU, there is not enough time remaining to allow the transmission to complete before the quiet interval starts, the STA shall defer the transmission by selecting a random backoff time, using the present CW (without advancing to the next value in the series). The short retry counter and long retry counter for the MSDU are not affected.

### 11.9.3 Testing channels for radars

A STA shall not transmit in a channel unless it has been tested for the presence of radars according to regulatory requirements.

### 11.9.4 Discontinuing operations after detecting radars

If a STA is operating in a channel and detects radar operating in the channel or accepts that another STA has detected radar operating in the channel then the STA shall discontinue transmissions according to regulatory requirements.

### 11.9.5 Detecting radars

The methods to detect radars operating in a channel that satisfy regulatory requirements are beyond the scope of this standard.

### 11.9.6 Requesting and reporting of measurements

A STA may measure one or more channels itself or a STA may request other STAs in the same BSS or IBSS to measure one or more channels on its behalf, either in a quiet interval or during normal operation.

When requesting other STAs to measure one or more channels, a STA shall use a Measurement Request frame containing one or more Measurement Request elements. The measurement request may be sent to an individual or group destination address. Addressing requests to multiple STAs should be used with care to avoid a reply storm.

The measurement requests effectively allowed by these rules are shown in Table 11-8.

**Table 11-8—Allowed measurement requests**

| Service set | Source of request | Destination of request | Type of measurement request allowed |
|---|---|---|---|
| BSS | AP | STA | Individual or group |
| | STA | AP | Individual only |
| | STA | STA | None |
| IBSS | STA | STA | Individual or group |

A STA that successfully requests another STA to perform a measurement on another channel should not transmit MSDUs or MMPDUs to that STA during the interval defined for the measurement plus any required channel switch intervals. In determining this period, a STA shall assume that any required channel switches take less than dot11ChannelSwitchTime per switch.

Copyright © 2007 IEEE. All rights reserved.

A STA that receives a Measurement Request frame from a STA in its BSS or IBSS shall parse the frame's Measurement Request elements in order, with measurements starting at the times specified by the Measurement Request elements. A STA may ignore any group addressed Measurement Request frames.

Any result of a measurement request shall be returned without undue delay to the requesting STA in Measurement Report elements using one or more Measurement Report frames. The result may be the completed measurement or an indication that the STA is unable to complete the measurement request.

A STA shall report it is too late to undertake a measurement request if it receives the request after the specified starting time for the measurement.

A STA shall report it is refusing a measurement request if all of the following conditions exist:

— The STA is capable of undertaking a measurement request,

— The STA does not want to undertake the measurement request at this time, and

— The measurement request is not mandatory (NOTE: measurements are specified as mandatory or optional in 7.3.2.21).

A STA shall report it is incapable a measurement request if any of the following conditions exists:

— The STA is incapable of undertaking an optional measurement request, or

— The STA does not support the channel specified in a mandatory measurement request, or

— The STA does not support any requested parallel measurements in the same or different channels.

The Measurement Report frames shall contain the same Dialog Token field as the corresponding Measurement Request frame, and each Measurement Report element shall contain the same Measurement Token field as the corresponding Measurement Request element.

A STA may autonomously report measurements to another STA in its BSS or IBSS using a Measurement Report frame with a Dialog Token field set to 0 with one or more Measurement Report elements. A STA in an IBSS may also autonomously report measurements to other STAs in the IBSS using the Channel Map field in the IBSS DFS element in a Beacon frame or Probe Response frame.

A STA may enable or disable measurement requests or autonomous measurement reports from another STA by transmitting Measurement Request elements with the Enable bit set to 1 and the Request bit and Report bit set to 0 or 1, as appropriate. These elements do not require a corresponding Measurement Report element in a Measurement Report frame. All measurement requests and reports are enabled by default. An AP may ignore a request to disable a mandatory measurement request. All others requests shall be honored.

### 11.9.7 Selecting and advertising a new channel

An attempt may be made to move a BSS to a new operating channel. It is an objective that disruption to the BSS is minimized in this process, although it should be recognized that a channel switch might not successfully move all STAs. It should also be stressed that the channel switch process is distinct from the regulatory requirement to cease transmission on a particular channel in the presence of radar.

#### 11.9.7.1 Selecting and advertising a new channel in an infrastructure BSS

The decision to switch to a new operating channel in an infrastructure BSS shall be made only by the AP. An AP may make use of the information in Supported Channel elements and the results of measurements undertaken by the AP and other STAs in the BSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy applicable regulatory requirements, including uniform spreading rules and channel testing rules. The AP shall attempt to select a


Copyright © 2007 IEEE. All rights reserved.

new channel that is supported by all associated STAs, although it should be noted that this might not always be possible.

An AP shall inform associated STAs that the AP is moving to a new channel and maintain the association by advertising the switch using Channel Switch Announcement elements in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time. The AP may force STAs in the BSS to stop transmissions until the channel switch takes place using the Channel Switch Mode field in the Channel Switch Announcement element. If possible, the channel switch should be scheduled so that all STAs in the BSS, including STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch. The AP may send the Channel Switch Announcement frame in a BSS without performing a backoff, after determining the WM is idle for one PIFS period.

A STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take alternative action. For example, it may choose to move to a different BSS.

A STA in a BSS that is not the AP shall not transmit the Channel Switch Announcement element.

#### 11.9.7.2 Selecting and advertising a new channel in an IBSS

DFS in an IBSS is complicated by the following:

- There is no central AP function for collating measurements or coordinating a channel switch. If STAs make independent decisions to switch channel in the presence of radar, there is a danger that all STAs will announce a switch to differing channels if several of them detect the radar.
- There is no association protocol that can be used to
  - Exchange supported channel information and
  - Determine membership of the IBSS at a given instant for requesting measurements.
- Beaconing is a shared process; therefore, it cannot be guaranteed that a STA that has something to send (e.g., a channel switch message) will be the next STA to Beacon frame.

The DFS owner service, IBSS DFS element, and Channel Switch Announcement frame address these complications.

- The DFS owner service provides a central point of coordination for a channel switch. It attempts to minimize the probability that multiple STAs concurrently decide to switch to different channels. The DFS Owner field and DFS Recovery Interval field within the IBSS DFS element support the DFS owner service.
- Each STA shall include a Channel Map field within the IBSS DFS elements that it transmits. The channel map communicates the STA-supported channel set and basic measurement reports for that STA.
- The ability to send a Channel Switch Announcement element within a management frame other than a Beacon frame or Probe Response frame is essential.

The potential for hidden nodes within an IBSS means that the IBSS channel switch protocol is best effort. All members of an IBSS shall have an individual responsibility to cease transmission on a particular channel in the presence of radar.

A STA at which an IBSS is started shall be a DFS owner in that IBSS. That STA shall include its MAC address in the DFS Owner field of the IBSS DFS element and DFS Recovery Interval field from the MLME.START.request parameter. The purpose of the DFS owner is to coordinate a channel switch when required. All STAs within a spectrum-managed IBSS shall have the ability to become DFS owner.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 515 of 1232

Each STA in an IBSS shall adopt the DFS owner and the DFS owner recovery interval from any valid IBSS DFS element when the frame contained a matching SSID and the value of the timestamp is later than the STA's TSF timer. The STA shall include the adopted values within the IBSS DFS elements it transmits. Because all STAs in an IBSS participate in sending Beacon frames, this process will always, over a number of beacon intervals, result in a unified view of one DFS owner throughout the IBSS.

If a STA detects radar and wants to attempt a channel switch using the DFS owner, the STA shall broadcast one or more Measurement Report frames indicating the presence of the radar.

A DFS owner receiving a Measurement Report frame indicating the presence of radar in the current channel shall select and advertise a new operating channel (including the possibility of no change). The DFS owner may make use of information received in Channel Map fields and from measurements undertaken by other members of the IBSS to assist the selection of the new channel. The algorithm to choose a new channel is beyond the scope of this standard, but shall satisfy any regulatory requirements, including uniform spreading rules and channel testing rules. The DFS owner shall attempt to select a new channel that is supported by all members of the IBSS. It should be noted that this process might be imperfect in that the DFS owner may have incomplete knowledge and there may be no suitable channel.

The DFS owner shall attempt to make the members of the IBSS aware of the new operating channel by broadcasting at least one Channel Switch Announcement frame. The DFS owner shall also include the Channel Switch Announcement element in all Beacon frames, Probe Response frames, or Channel Switch Announcement frames until the intended channel switch time. A STA that receives a valid Channel Switch Announcement element shall repeat this element in all Beacon frames and Probe Response frames that it transmits. The DFS owner may attempt to silence STAs in the IBSS until the channel switch takes place using the Channel Switch Mode field in the Channel Switch Announcement element. If possible, the channel switch should be scheduled so that all STAs in the IBSS, including STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

If a STA does not receive a valid Channel Switch Announcement element from the DFS owner within DFS recovery time measured from the end of the frame within which radar notification was first transmitted by the STA or received from another STA, then it shall enter a DFS owner recovery mode. In DFS owner recovery mode, the STA shall assume the role of DFS owner, shall select a new operating channel, and shall advertise the new channel by transmitting a Channel Switch Announcement frame using the contention resolution algorithm defined for beacon transmission at TBTT in 11.1.2.2. The STA shall also include the Channel Switch Announcement element in all Beacon frames and Probe Response frames until the intended channel switch time. A STA that is not the DFS owner shall not initiate a channel switch.

If the STA receives a valid Channel Switch Announcement element from another member of the IBSS, the STA shall leave DFS owner recovery mode prior to the channel switch and adopt the received channel switch information. If the Channel Switch Announcement element was within a Beacon frame or Probe Response frame, the STA shall also adopt the DFS owner address from the IBSS DFS element. If the Channel Switch Announcement element was within a Channel Switch Announcement frame, the STA shall adopt the DFS owner from the transmitter address of the received frame.

There are several circumstances when DFS owner recovery is required (e.g., if the original DFS owner has left the network or if the original measurement report was not received by the initial DFS owner). It should be noted that DFS owner recovery might temporarily give rise to more than one DFS owner within the IBSS. This risk is mitigated by the random nature of the IBSS DFS recovery mechanism. However, because all STAs in an IBSS participate in sending Beacon frames, over a number of beacon periods, there will be convergence from multiple DFS owners to one DFS owner.


Copyright © 2007 IEEE. All rights reserved.

# 12. PHY service specification

## 12.1 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC are described in this clause. Different PHYs are defined as part of this standard. Each PHY can consist of two protocol functions as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs.

Each PMD sublayer may require the definition of a unique PLCP. If the PMD sublayer already provides the defined PHY services, the PHY convergence function might be null.

## 12.2 PHY functions

The protocol reference model for the IEEE 802.11 architecture is shown in Figure 5-10 (in 5.7). Most PHY definitions contain three functional entities: the PMD function, the PHY convergence function, and the layer management function.

The PHY service is provided to the MAC entity at the STA through a SAP, called the PHY-SAP, as shown in Figure 5-10. A set of primitives might also be defined to describe the interface between the PLCP sublayer and the PMD sublayer, called the PMD_SAP.

## 12.3 Detailed PHY service specifications

### 12.3.1 Scope and field of application

The services provided by the PHY to the IEEE 802.11 MAC are specified in this subclause. These services are described in an abstract way and do not imply any particular implementation or exposed interface.

### 12.3.2 Overview of the service

The PHY function as shown in Figure 5-10 is separated into two sublayers: the PLCP sublayer and the PMD sublayer. The function of the PLCP sublayer is to provide a mechanism for transferring MPDUs between two or more STAs over the PMD sublayer.

### 12.3.3 Overview of interactions

The primitives associated with communication between the IEEE 802.11 MAC sublayer and the IEEE 802.11 PHY fall into two basic categories:

a) Service primitives that support MAC peer-to-peer interactions;

b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

### 12.3.4 Basic service and options

All of the service primitives described here are considered mandatory unless otherwise specified.

Copyright © 2007 IEEE. All rights reserved.

#### 12.3.4.1 PHY-SAP peer-to-peer service primitives

Table 12-1 indicates the primitives for peer-to-peer interactions.

**Table 12-1—PHY-SAP peer-to-peer service primitives**

| Primitive | Request | Indicate | Confirm |
|---|---|---|---|
| PHY-DATA | X | X | X |

#### 12.3.4.2 PHY-SAP sublayer-to-sublayer service primitives

Table 12-2 indicates the primitives for sublayer-to-sublayer interactions.

**Table 12-2—PHY-SAP sublayer-to-sublayer service primitives**

| Primitive | Request | Indicate | Confirm |
|---|---|---|---|
| PHY-TXSTART | X | | X |
| PHY-TXEND | X | | X |
| PHY-CCARESET | X | | X |
| PHY-CCA | | X | |
| PHY-RXSTART | | X | |
| PHY-RXEND | | X | |

#### 12.3.4.3 PHY-SAP service primitives parameters

Table 12-3 shows the parameters used by one or more of the PHY-SAP service primitives.

**Table 12-3—PHY-SAP service primitive parameters**

| Parameter | Associated primitive | Value |
|---|---|---|
| DATA | PHY-DATA.request<br>PHY-DATA.indication | Octet value X'00'–X'FF' |
| TXVECTOR | PHY-TXSTART.request | A set of parameters |
| STATUS | PHY-CCA.indication | BUSY, IDLE |
| RXVECTOR | PHY-RXSTART.indication | A set of parameters |
| RXERROR | PHY-RXEND.indication | NoError, FormatViolation, Carrier-Lost, UnsupportedRate |


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.4.4 Vector descriptions

Several service primitives include a parameter vector. This vector is a list of parameters that may vary depending on the PHY type. Table 12-4 lists the parameter values required by the MAC or PHY in each of the parameter vectors. Parameters in the vectors that are management rather than MAC may be specific to the PHY and are listed in the clause covering that PHY.

**Table 12-4—Vector descriptions**

| Parameter | Associate vector | Value |
|---|---|---|
| DATARATE | TXVECTOR, RXVECTOR | PHY dependent. The name of the field used to specify the Tx data rate and report the Rx data rate may vary for different PHYs. |
| LENGTH | TXVECTOR, RXVECTOR | PHY dependent |

Copyright © 2007 IEEE. All rights reserved.

### 12.3.5 PHY-SAP detailed service specification

The following subclause describes the services provided by each PHY primitive.

#### 12.3.5.1 PHY-DATA.request

##### 12.3.5.1.1 Function

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity.

##### 12.3.5.1.2 Semantics of the service primitive

The primitive provides the following parameters:
PHY-DATA.request(DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

##### 12.3.5.1.3 When generated

This primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive can only be issued following a transmit initialization response (PHY-TXSTART.confirm) from the PHY.

##### 12.3.5.1.4 Effect of receipt

The receipt of this primitive by the PHY entity causes the PLCP transmit state machine to transmit an octet of data. When the PHY entity receives the octet, it will issue a PHY-DATA.confirm to the MAC sublayer.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.2 PHY-DATA.indication

##### 12.3.5.2.1 Function

This primitive indicates the transfer of data from the PHY to the local MAC entity.

##### 12.3.5.2.2 Semantics of the service primitive

The primitive provides the following parameters:
PHY-DATA.indication (DATA)

The DATA parameter is an octet of value X'00' through X'FF'.

##### 12.3.5.2.3 When generated

The PHY-DATA.indication is generated by a receiving PHY entity to transfer the received octet of data to the local MAC entity. The time between receipt of the last bit of the provided octet from the WM and the receipt of this primitive by the MAC entity will be the sum of aRXRFDelay + aRxPLCPDelay.

##### 12.3.5.2.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.3 PHY-DATA.confirm

##### 12.3.5.3.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the transfer of data from the MAC entity to the PHY.

##### 12.3.5.3.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PHY-DATA.confirm

This primitive has no parameters.

##### 12.3.5.3.3 When generated

This primitive will be issued by the PHY to the MAC entity when the PLCP has completed the transfer of data from the MAC entity to the PHY. The PHY will issue this primitive in response to every PHY-DATA.request primitive issued by the MAC sublayer.

##### 12.3.5.3.4 Effect of receipt

The receipt of this primitive by the MAC will cause the MAC to start the next MAC entity request.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.4 PHY-TXSTART.request

##### 12.3.5.4.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to start the transmission of an MPDU.

##### 12.3.5.4.2 Semantics of the service primitive

The primitive provides the following parameters:
PHY-TXSTART.request(TXVECTOR)

The TXVECTOR represents a list of parameters that the MAC sublayer provides to the local PHY entity in order to transmit an MPDU. This vector contains both PLCP and PHY management parameters. The required PHY parameters are listed in 12.3.4.4.

##### 12.3.5.4.3 When generated

This primitive will be issued by the MAC sublayer to the PHY entity when the MAC sublayer needs to begin the transmission of an MPDU.

##### 12.3.5.4.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity will be to start the local transmit state machine.

The behavior expected by the MAC pursuant to the issuance of PHY-TXSTART.Request is shown in Figure 9-12 (in 9.2.10).

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 523 of 1232

#### 12.3.5.5 PHY-TXSTART.confirm

##### 12.3.5.5.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the start of a transmission. The PHY will issue this primitive in response to every PHY-TXSTART.request primitive issued by the MAC sublayer.

##### 12.3.5.5.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PHY-TXSTART.confirm

This primitive has no parameters.

##### 12.3.5.5.3 When generated

This primitive will be issued by the PHY to the MAC entity when the PHY has received a PHY-TXSTART.request from the MAC entity and is ready to begin accepting outgoing data octets from the MAC.

##### 12.3.5.5.4 Effect of receipt

The receipt of this primitive by the MAC entity will cause the MAC to start the transfer of data octets.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.6 PHY-TXEND.request

##### 12.3.5.6.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity that the current transmission of the MPDU be completed.

##### 12.3.5.6.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PHY-TXEND.request

This primitive has no parameters.

##### 12.3.5.6.3 When generated

This primitive will be generated when the MAC sublayer has received the last PHY-DATA.confirm from the local PHY entity for the MPDU currently being transferred.

##### 12.3.5.6.4 Effect of receipt

The effect of receipt of this primitive by the local PHY entity will be to stop the transmit state machine.

Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.7 PHY-TXEND.confirm

##### 12.3.5.7.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm the completion of a transmission. The PHY issues this primitive in response to every PHY-TXEND.request primitive issued by the MAC sublayer.

##### 12.3.5.7.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PHY-TXEND.confirm

This primitive has no parameters.

##### 12.3.5.7.3 When generated

This primitive will be issued by the PHY to the MAC entity when the PHY has received a PHY-TXEND.request immediately after transmitting the end of the last bit of the last data octet indicating that the symbol containing the last data octet has been transferred.

##### 12.3.5.7.4 Effect of receipt

The receipt of this primitive by the MAC entity provides the time reference for the contention backoff protocol.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.8 PHY-CCARESET.request

##### 12.3.5.8.1 Function

This primitive is a request by the MAC sublayer to the local PHY entity to reset the CCA state machine.

##### 12.3.5.8.2 Semantics of the service primitive

The semantics of the primitives are as follows:
PHY-CCARESET.request

This primitive has no parameters.

##### 12.3.5.8.3 When generated

This primitive is generated by the MAC sublayer for the local PHY entity at the end of a NAV timer. This request can be used by some PHY implementations that may synchronize antenna diversity with slot timings.

##### 12.3.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to reset the PLCP CS/CCA timers to the state appropriate for the end of a received frame.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 527 of 1232

#### 12.3.5.9 PHY-CCARESET.confirm

##### 12.3.5.9.1 Function

This primitive is issued by the PHY to the local MAC entity to confirm that the PHY has reset the CCA state machine.

##### 12.3.5.9.2 Semantics of the service primitive

The semantics of the primitives are as follows:
PHY-CCARESET.confirm

This primitive has no parameters.

##### 12.3.5.9.3 When generated

This primitive is issued by the PHY to the MAC entity when the PHY has received a PHY-CCA-RESET.request.

##### 12.3.5.9.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.10 PHY-CCA.indication

##### 12.3.5.10.1 Function

This primitive is an indication by the PHY to the local MAC entity of the current state of the medium.

##### 12.3.5.10.2 Semantics of the service primitive

The primitive provides the following parameter:
PHY-CCA.indication (STATE)

The STATE parameter can be one of two values: BUSY or IDLE. The parameter value is BUSY if the channel assessment by the PHY determines that the channel is not available. Otherwise, the value of the parameter is IDLE.

##### 12.3.5.10.3 When generated

This primitive is generated within aCCATime of the occurrence of a change in the status of the channel changes from channel idle to channel busy or from channel busy to channel idle. This includes the period of time when the PHY is receiving data. The PHY maintains the channel busy indication until the period indicated by the LENGTH field in a valid PLCP header has expired.

##### 12.3.5.10.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 529 of 1232

#### 12.3.5.11 PHY-RXSTART.indication

##### 12.3.5.11.1 Function

This primitive is an indication by the PHY to the local MAC entity that the PLCP has received a valid start frame delimiter (SFD) and PLCP header.

##### 12.3.5.11.2 Semantics of the service primitive

The primitive provides the following parameter:
PHY-RXSTART.indication (RXVECTOR)

The RXVECTOR represents a list of parameters that the PHY provides the local MAC entity upon receipt of a valid PLCP header. This vector may contain both MAC and MAC management parameters. The required parameters are listed in 12.3.4.4.

##### 12.3.5.11.3 When generated

This primitive is generated by the local PHY entity to the MAC sublayer when the PHY has successfully validated the PLCP header at the start of a new PPDU.

After generating a PHYRXSTART.indication, the PHY is expected to maintain physical medium busy status (not generating PHY-CCA.indication(IDLE)) during the period required by that PHY to transfer a frame of the indicated LENGTH at the indicated DATARATE. This physical medium busy condition should be maintained even if a PHY-RXEND.indication(CarrierLost) or a PHYRXEND.indication(Format-Violation) is generated by the PHY prior to the end of this period.

##### 12.3.5.11.4 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.


Copyright © 2007 IEEE. All rights reserved.

#### 12.3.5.12 PHY-RXEND.indication

##### 12.3.5.12.1 Function

This primitive is an indication by the PHY to the local MAC entity that the MPDU currently being received is complete.

##### 12.3.5.12.2 Semantics of the service primitive

The primitive provides the following parameter:
PHY-RXEND.indication (RXERROR)

The RXERROR parameter can convey one or more of the following values: NoError, FormatViolation, CarrierLost, or UnsupportedRate. A number of error conditions may occur after the PLCP's receive state machine has detected what appears to be a valid preamble and SFD. The following describes the parameter returned for each of those error conditions.

— *NoError*. This value is used to indicate that no error occurred during the receive process in the PLCP.

— *FormatViolation.* This value is used to indicate that the format of the received PPDU was in error.

— *CarrierLost.* This value is used to indicate that during the reception of the incoming MPDU, the carrier was lost and no further processing of the MPDU can be accomplished.

— *UnsupportedRate.* This value is used to indicate that during the reception of the incoming PPDU, a nonsupported date rate was detected.

##### 12.3.5.12.3 When generated

This primitive is generated by the PHY for the local MAC entity to indicate that the receive state machine has completed a reception with or without errors.

In the case of an RXERROR value of NoError, the MAC uses the PHY-RXEND.Indication as reference for channel access timing, as shown in Figure 9-12 (in 9.2.10).

##### 12.3.5.12.4 Effect of receipt

The effect of receipt of this primitive is for the MAC to begin inter-frame space processing, as described in 9.2.10.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 531 of 1232

Copyright © 2007 IEEE. All rights reserved.
**Samsung Ex. 1044, Page 532 of 1232**

# 13. PHY management

The MIB comprises the managed objects, attributes, actions, and notifications required to manage a STA. The definition of these managed objects, attributes, actions, and notifications, as well as their structure, is presented in Annex D.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.
**Samsung Ex. 1044, Page 534 of 1232**

# 14. Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz industrial, scientific, and medical (ISM) band

## 14.1 Overview

### 14.1.1 Overview of FHSS PHY

The PHY services provided to the IEEE 802.11 WLAN MAC for the 2.4 GHz frequency-hopping spread spectrum (FHSS) system are described in this clause. The FHSS PHY consists of the following two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs.

### 14.1.2 FHSS PHY functions

The 2.4 GHz FHSS PHY architecture is shown in Figure 5-10 (in 5.7). The FHSS PHY contains three functional entities: the PMD function, the PHY convergence function, and the PHY management function. Each of these functions is described in detail in 14.1.2.1 through 14.1.2.3.

The FHSS PHY service is provided to the MAC entity at the STA through a SAP, called the PHY-SAP, as shown in Figure 5-10. A set of primitives might also be defined that describe the interface between the PLCP sublayer and the PMD sublayer, called the PMD_SAP.

#### 14.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies provision of a PHY service interface to the IEEE 802.11 MAC services.

#### 14.1.2.2 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

#### 14.1.2.3 PMD sublayer

The PMD sublayer provides a transmission interface used to send and receive data between two or more STAs.

### 14.1.3 Service specification method and notation

The models represented by state diagrams in the following subclauses are intended as the primary specifications of the functions provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology.

Copyright © 2007 IEEE. All rights reserved.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition of service is independent of any particular implementation.

## 14.2 FHSS PHY-specific service parameter lists

### 14.2.1 Overview

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY-dependent MAC state machines reside in a sublayer defined as the MLME. The MLME in certain PMD implementations may need to interact with the PLME as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY Service Primitives as TXVECTOR and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR and RXVECTOR for the FHSS PHY.

All of the values included in the TXVECTOR or RXVECTOR described in this subclause are considered mandatory unless otherwise specified. The 1 Mb/s and 2 Mb/s data rates are the only rates currently supported. Other indicated data rates are for possible future use.

### 14.2.2 TXVECTOR parameters

The parameters in Table 14-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request service primitive.

**Table 14-1—TXVECTOR parameters**

| Parameter | Associate primitive | Value |
|---|---|---|
| LENGTH | PHY-TXSTART.request(TXVECTOR) | 1–4095 |
| DATARATE | PHY-TXSTART.request(TXVECTOR) | 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 |

#### 14.2.2.1 TXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start a transmission.

#### 14.2.2.2 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP should transmit the PLCP service data unit (PSDU). Its value can be any of the rates as defined in Table 14-1, and supported by the conformant FH PHY.

### 14.2.3 RXVECTOR parameters

The parameters in Table 14-2 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indicate service primitive.


Copyright © 2007 IEEE. All rights reserved.

**Table 14-2—RXVECTOR parameters**

| Parameter | Associate primitive | Value |
|---|---|---|
| LENGTH | PHY-RXSTART.indicate(RXVECTOR) | 1–4095 |
| Receive signal strength indicator (RSSI) | PHY-RXSTART.indicate(RXVECTOR) | 0–RSSI Max |
| DATARATE | PHY-RXSTART.request(RXVECTOR) | 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 |

#### 14.2.3.1 TRXVECTOR LENGTH

The LENGTH parameter has the value of 1 to 4095. This parameter is used to indicate the value contained in the LENGTH field that the PLCP has received in the PLCP header. The MAC and PLCP will use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

#### 14.2.3.2 RXVECTOR RSSI

The RSSI is an optional parameter that has a value of 0 through RSSI Max. This parameter is a measure by the PHY of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured between the beginning of the SFD and the end of the PLCP HEC. RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.

## 14.3 FHSS PLCP sublayer

### 14.3.1 Overview

This subclause provides a convergence procedure to map MPDUs into a frame format designed for FHSS radio transceivers. The procedures for transmission, CS, and reception are defined for single and multiple antenna diversity radios.

#### 14.3.1.1 State diagram notation

The operation of the procedures can be described by state diagrams. Each diagram represents the domain and consists of a group of connected, mutually exclusive states. Only one state is active at any given time. Each state is represented by a rectangle as shown in Figure 14-1. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name. The lower part contains the name of any signal that is generated. Actions described by short phrases are enclosed in brackets.

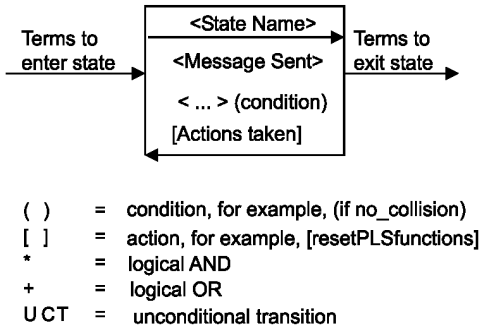



**Figure 14-1—State diagram notation example**

Copyright © 2007 IEEE. All rights reserved.

Each permissible transition between the states is represented graphically by an arrow from the initial to the terminal state. A transition that is global in nature (e.g., an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow. Labels on transitions are qualifiers that must be fulfilled before the transition will be taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.

State transitions and sending and receiving of messages occur instantaneously. When a state is entered and the condition to leave that state is not immediately fulfilled, the state executes continuously, sending the messages and executing the actions contained in the state in a continuous manner.

Some devices described in this standard are allowed to have two or more ports. State diagrams capable of describing the operation of devices with an unspecified number of ports require qualifier notation that allows testing for conditions at multiple ports. The notation used is a term that includes a description in parentheses of which ports must meet the term for the qualifier to be satisfied (e.g., ANY and ALL). It is also necessary to provide for term-assignment statements that assign a name to a port that satisfies a qualifier. The following convention is used to describe a term-assignment statement that is associated with a transition:

a) The character ":" (colon) is a delimiter used to denote that a term assignment statement follows.

b) The character "<" (left arrow) denotes assignment of the value following the arrow to the term preceding the arrow.

The state diagrams contain the authoritative statement of the procedures they depict; when apparent conflicts between descriptive text and state diagrams arise, the state diagrams are to take precedence. This does not, however, override any explicit description in the text that has no parallel in the state diagrams.

The models presented by state diagrams are intended as the primary specifications to be provided. It is important to distinguish, however, between a model and a real implementation. The models are optimized for simplicity and clarity of presentation, while any realistic implementation may place heavier emphasis on efficiency and suitability to a particular implementation technology. It is the functional behavior of any unit that must match the standard, not its internal structure. The internal details of the model are useful only to the extent that they specify the external behavior clearly and precisely.

### 14.3.2 PLCP frame format

The PPDU frame format provides for the asynchronous transfer of MAC sublayer MPDUs from any transmitting STA to all receiving STAs within the WLAN's BSS. The PPDU illustrated in Figure 14-2 consists of three parts: a PLCP preamble, a PLCP header, and a PSDU. The PLCP preamble provides a period of time for several receiver functions. These functions include antenna diversity, clock and data recovery, and field delineation of the PLCP header and the PSDU. The PLCP header is used to specify the length of the whitened PSDU field and support any PLCP management information. The PPDU contains the PLCP preamble, the PLCP header, and the PSDU modified by the PPDU data whitener.

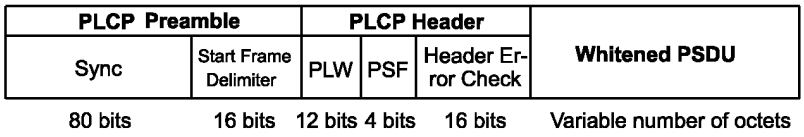



**Figure 14-2—PLCP frame format**


Copyright © 2007 IEEE. All rights reserved.

#### 14.3.2.1 PLCP Preamble field

The PLCP Preamble field contains two separate subfields, the Preamble Synchronization (SYNC) field and the SFD, to allow the PHY circuitry to reach steady-state demodulation and synchronization of bit clock and frame start.

##### 14.3.2.1.1 Preamble SYNC field

The Preamble SYNC field is an 80-bit field containing an alternating zero-one pattern, transmitted starting with zero and ending with one, to be used by the PHY to detect a potentially receivable signal, select an antenna if diversity is utilized, and reach steady-state frequency offset correction and synchronization with the received packet timing.

##### 14.3.2.1.2 SFD

The SFD consists of the 16-bit binary pattern 0000 1100 1011 1101 (transmitted leftmost bit first). The first bit of the SFD follows the last bit of the sync pattern. The SFD defines the frame timing.

#### 14.3.2.2 PLCP Header field

The PLCP Header field contains three separate subfields: a 12-bit PSDU Length Word (PLW), a 4-bit PLCP Signaling field (PSF), and a 16-bit PLCP HEC field.

##### 14.3.2.2.1 PLW

The PLW is passed from the MAC as a parameter within the PHY-TXSTART.request primitive. The PLW specifies the number of octets contained in the PSDU. Its valid values are X'001'–X'FFF', representing counts of one to 4095 octets. The PLW is transmitted LSB first and MSB last. The PLW is used by the receiving STA, in combination with the 32/33 coding algorithm specified in this clause, to determine the last bit in the packet.

##### 14.3.2.2.2 PSF

The 4-bit PSF is defined in Table 14-3. The PSF is transmitted bit 0 first and bit 3 last.

**Table 14-3—PSF bit descriptions**

| Bit | Parameter name | Parameter values | Description |
|---|---|---|---|
| 0 | Reserved | Default = 0 | Reserved |
| 1:3 | PLCP_BITRATE | b1 b2 b3 = Data Rate<br>0 0 0 = 1.0 Mb/s,<br>0 0 1 = 1.5 Mb/s,<br>0 1 0 = 2.0 Mb/s,<br>0 1 1 = 2.5 Mb/s,<br>1 0 0 = 3.0 Mb/s,<br>1 0 1 = 3.5 Mb/s,<br>1 1 0 = 4.0 Mb/s,<br>1 1 1 = 4.5 Mb/s | This field indicates the data rate of the whitened PSDU from 1 Mb/s to 4.5 Mb/s in 0.5 Mb/s increments. |

Copyright © 2007 IEEE. All rights reserved.

#### 14.3.2.2.3 HEC field

The HEC field is a 16-bit CRC-16 error detection field. The HEC uses the CRC-16 generator polynomial $G(x)$ as follows:

$$G(x) = x^{16}+x^{12}+x^{5}+1$$

The HEC shall be the ones complement of the sum (modulo 2) of the following:

a) The remainder of $x^k \times (x^{15} + x^{14} + \ldots + x^2 + x^1 + 1)$ divided (modulo 2) by $G(x)$, where $k$ is the number of bits in the PSF and PLW fields of the PLCP header;

b) The remainder after multiplication by $x^{16}$ and then division (modulo 2) by $G(x)$ of the content (treated as a polynomial) of the PSF and PLW fields.

The HEC shall be transmitted with the coefficient of the highest term first.

As a typical implementation, at the transmitter, the initial remainder of the division is preset to all ones and is then modified by division of the PSF and PLW fields by the generator polynomial, $G(x)$. The ones complement of this remainder is inserted in the HEC field with the MSB transmitted first.

At the receiver, the initial remainder of the division is again preset to all ones. The division of the received PSF, PLW, and HEC fields by the generator polynomial, $G(x)$, results, in the absence of transmission errors, in a unique nonzero value, which is the following polynomial $R(x)$:

$$R(x) = x^{12} + x^{11} + x^{10} + x^{8} + x^{3} + x^{2} + x^{1} + 1$$

#### 14.3.2.3 PLCP data whitener

The PLCP data whitener uses a length-127 frame-synchronous scrambler followed by a 32/33 bias-suppression encoding to randomize the data and to minimize the data dc bias and maximum run lengths. Data octets are placed in the transmit serial bit stream LSB first and MSB last. The frame synchronous scrambler uses the generator polynomial $S(x)$ as follows:

$$S(x) = x^{7}+x^{4}+1$$

and is illustrated in Figure 14-3. The 127-bit sequence generated repeatedly by the scrambler is (leftmost bit used first) 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111. The same scrambler is used to scramble transmit data and to descramble receive data. The data whitening starts with the first bit of the PSDU, which follows the last bit of the PLCP header. The specific bias suppression encoding and decoding method used is defined in Figure 14-7 and Figure 14-12. The format of the packet after data whitening is as shown in Figure 14-4.

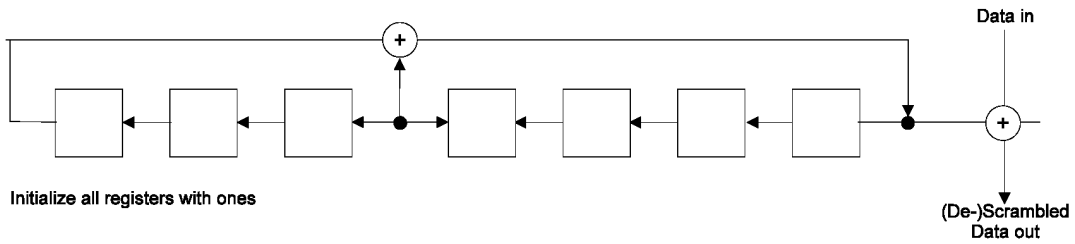



**Figure 14-3—Frame synchronous scrambler/descrambler**


Copyright © 2007 IEEE. All rights reserved.

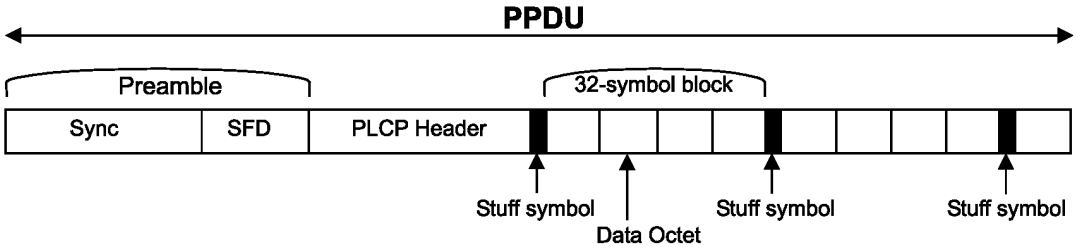



**Figure 14-4—PLCP data whitener format**

### 14.3.3 PLCP state machines

The PLCP consists of three state machines, as illustrated in the overview diagram of Figure 14-5: the TX, the CS/CCA, and the RX state machines. The three PLCP state machines are defined in the subclauses below; Figure 14-5 is not a state diagram itself. Execution of the PLCP state machines normally is initiated by the FH PLME state machine and begins at the CS/CCA state machine. The PLCP returns to the FH PLME state machine upon interrupt to service a PLME service request, e.g., PLME-SET, PLME-RESET.

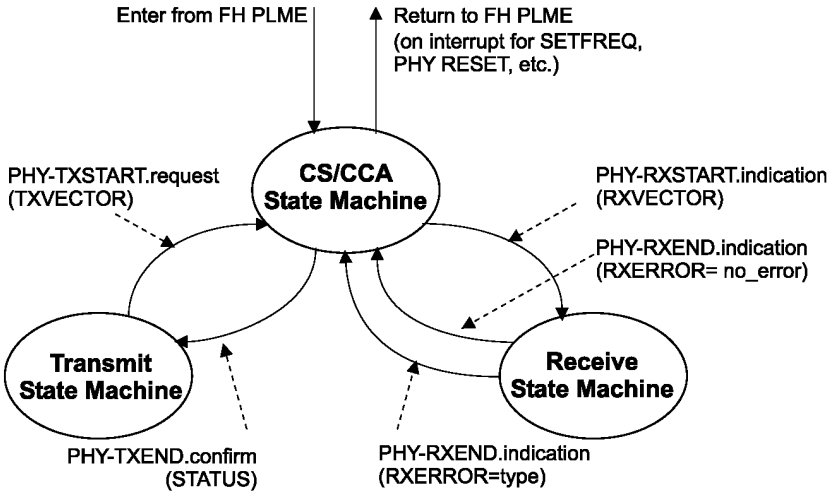



**Figure 14-5—PLCP top-level state diagram**

#### 14.3.3.1 Transmit PLCP

The transmit PLCP is invoked by the CS/CCA procedure immediately upon receiving a PHY-TXSTART.request(TXVECTOR) from the MAC sublayer. The CSMA/CA protocol is performed by the MAC with the PHY PLCP in the CS/CCA procedure prior to executing the transmit procedure.

##### 14.3.3.1.1 Transmit state machine

The PLCP transmit state machine illustrated in Figure 14-6 includes functions that must be performed prior to, during, and after PPDU data transmission. Upon entering the transmit procedure in response to a PHY-TXSTART.request(TXVECTOR) from the MAC, the PLCP shall switch the PHY PMD circuitry from receive to transmit state; ramp on the transmit power amplifier in the manner prescribed in 14.6; and transmit the preamble sync pattern and SFD. The PLCP shall generate the PLCP header as defined in

Copyright © 2007 IEEE. All rights reserved.

14.3.2.2 in sufficient time to send the bits at their designated bit slot time. The PLCP shall add the PLCP header to the start of the PSDU data.

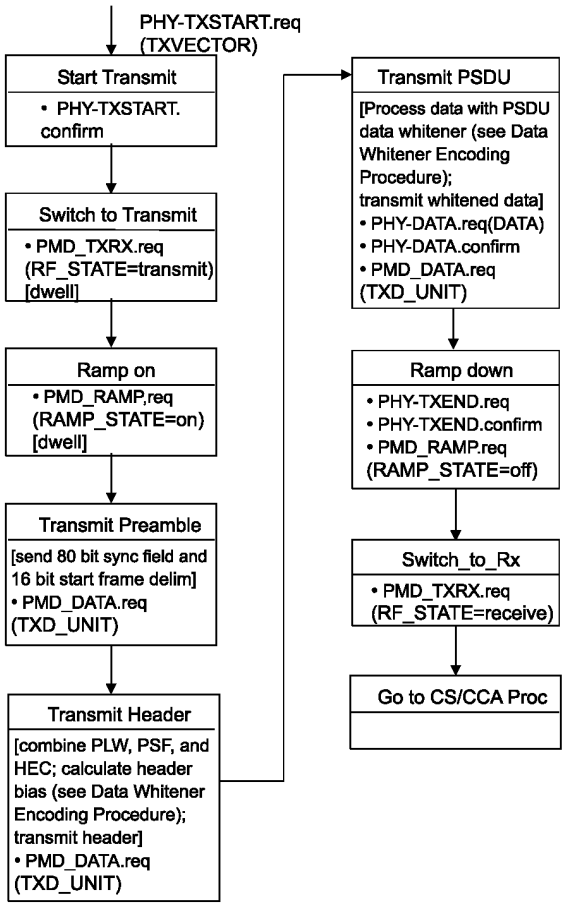



**Figure 14-6—Transmit state machine**

Prior to transmitting the first PSDU data bit, the PLCP shall send a PHY-TXSTART.confirm message to the MAC indicating that the PLCP is ready to receive an MPDU data octet. The MAC will pass an MPDU data octet to the PHY with a PHY-DATA.request(DATA), which the PHY will respond to with a PHY-DATA.confirm. This sequence of PHY-DATA.request(DATA) and PHY-DATA.confirm shall be executed until the last data octet is passed to the PLCP. During transmission of the PSDU data, each bit of the PSDU shall be processed by the data whitener algorithm defined in Figure 14-7 and described in 14.3.2.3. Each PSDU data octet is processed and transmitted LSB first and MSB last.

After the last MPDU octet is passed to the PLCP, the MAC will indicate the end of the frame with a *PHY-TXEND.request*. After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall complete the transmit procedure by sending a *PHY-TXEND.confirm* to the MAC sublayer, ramp off the power amplifier in the manner prescribed in 14.6, and switch the PHY PMD circuitry from transmit to receive state. The execution shall then return to the CS/CCA procedure.


Copyright © 2007 IEEE. All rights reserved.

**Data whitener encoding algorithm:**

```
/*      If MSB of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*      Accumulate PLCP header; begin stuffing on first bit of the PSDU */

/********* Calculate number of 32-symbol BSE blocks required to send PSDU;
            no padding is necessary when the number of symbols is not a multiple of 32 *********/
Input parameter: number_of_PSDU_octets, rate;                   /* rate is 1 or 2*/
number_of_symbols= (number_of_PSDU_octets *8) /rate;
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32)};

/********* Accumulate the bias in the header to use in calculating the inversion state of the first
            block of PSDU data *********/
Read in header {b(1),...,b(32)};            /* b(1) is first bit in */
header_bias = Sum{weight(b(1)),...,weight(b(32))};
                        /* calculate bias in header; weights are defined in Table 14-4*/
Transmit {b(1),...,b(32)};                  /* no stuffing on header */
accum=header_bias;                          /* initialize accum */
Initialize scrambler to all ones;

/********* Whiten the PSDU data with scrambler and BSE encoder *********/
For n = 1 to number_of_blocks_in_packet
{
        b(0) = 0 for 1 Mb/s; b(0)=00 for 2 Mb/s;                    /* b(0) is the stuff symbol */
        N = min(32, number_of_symbols);                             /* N= block size in symbols */
        Read in next symbol block {b(1),...,b(N)};                  /* b(n) = {0,1} or {0,1,2,3};
        1 - 8 octets, use PHY-DATA.req(DATA), PHY-DATA.confirm for each octet*/
        Scramble {b(1),...,b(N)};                                       /* see 14.3.2.3*/
        bias_next_block = Sum{weight(b(0)),...,weight(b(N))};       /* calculate bias with b(0)=0 */

        /***** if accum and bias of next block has the same sign, then invert block;
                if accum=0 or bias_next_block=0, do not invert *****/
        If {[accum * bias_next_block > 0] then
         {
                Invert {b(0),...,b(N)};     /* Invert deviation, or, negate MSB of symbol */
                bias_next_block = - bias_next_block;
         }

        accum = accum + bias_next_block;
        transmit {b(0),...,b(N)};                       /* b(0) is first symbol out */
        number_of_symbols = number_of_symbols - N
}
```

**Figure 14-7—Data whitener encoding procedure**

The weights assigned to each value of the symbols are defined in Table 14-4 for the 1 Mb/s [two-level Gaussian frequency shift keying (2GFSK)] and 2 Mb/s [four-level GFSK (4GFSK)] symbols.

**Table 14-4—PLCP field bit descriptions**

| 2GFSK | 4GFSK | Weight |
|---|---|---|
| — | 10 | 3 |
| 1 | — | 2 |
| — | 11 | 1 |
| Center | Center | 0 |
| — | 01 | –1 |
| 0 | — | –2 |
| — | 00 | –3 |

Copyright © 2007 IEEE. All rights reserved.

#### 14.3.3.1.2 Transmit state timing

The transmit timing illustrated in Figure 14-8 is defined from the instant that the *PHY-TXSTART.request(TXVECTOR)* is received from the MAC sublayer. The PLCP shall switch the PMD circuitry from receive to transmit, turn on and settle the transmitter, and begin transmitting the first bit of the preamble at the antenna within a maximum of 20 μs of receiving the *PHY-TXSTART.request(TXVECTOR)*. The PLCP preamble shall be transmitted at 1 Mb/s and be completed in 96 μs. The PLCP header shall be transmitted at 1 Mb/s and be completed in 32 μs. The variable length PSDU shall be transmitted at the selected data rate. After the last bit of the PSDU data has completed propagation through the radio and been transmitted on the air, the PLCP shall send the *PHY-TXEND.confirm* to the MAC sublayer. The PLCP shall turn off the transmitter, reducing the output energy to less than the specified off-mode transmit power within the time specified in 14.6. At the end of the power amplifier ramp down period, the PLCP shall switch the PMD circuitry from transmit to receive.

### 14.3.3.2 CS/CCA procedure

The CS/CCA procedure is executed while the receiver is turned on and the STA is not currently receiving or transmitting a packet. The CS/CCA procedure is used for two purposes: to detect the start of a network signal that can be received (CS) and to determine whether the channel is clear prior to transmitting a packet (CCA).

#### 14.3.3.2.1 CS/CCA state machine

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA windows is defined relative to the end of the last bit of the last packet on the air. The CS/CCA state machine is shown in Figure 14-9. The PLCP shall perform a CS/CCA on a minimum of one antenna within a MAC contention backoff slot time of 50 μs. The PLCP shall be capable of detecting within the slot time an FH-PHY-conformant signal that is received at the selected antenna up to 22 μs after the start of the slot time with the synchronous detection performance specified in 14.6.15.3. Detection performance with zero-one sync patterns and with random data patterns is specified in 14.6.15.3. If a start of a transmission is asynchronous with the BSS and arrives after the start of the slot but at least 16 μs prior to the end of the slot, the PLCP shall indicate a busy channel prior to the end of the slot time with the asynchronous detection performance specified in 14.6.15.3. The CCA indication immediately prior to transmission shall be performed on an antenna with essentially the same free space gain and gain pattern as the antenna to be used for transmission. The method of determining CS/CCA is unspecified except for the detection performance of a conformant method as specified in 14.6.15.3.

If a *PHY-TXSTART.request(TXVECTOR)* is received, the CS/CCA procedure shall exit to the transmit procedure within 1 μs. If a *PHY-CCARESET.request* is received, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame. This service primitive is generated by the MAC at the end of a NAV period. The PHY shall indicate completion of the request by sending a *PHY-CCARESET.confirm* to the MAC.

If a CS/CCA returns a channel idle result, the PHY shall send a *PHY-CCA.indicate(STATUS=idle)* to the MAC.

If a CS/CCA returns a channel busy result, the PHY shall send a *PHY-CCA.indicate(STATUS=busy)* to the MAC. Upon a channel busy assessment, the PLCP shall stop any antenna switching prior to the earliest possible arrival time of the SFD and detect a valid SFD and PLCP header if received. A valid PLCP header is defined as containing valid PLW and PSF values and a valid HEC field. If a valid SFD/PLCP header is detected, the CS/CCA procedure shall send a PHY-RXSTART.indicate(RXVECTOR) message to the MAC sublayer and exit to the receive procedure. The PLCP shall dwell and search for the SFD/PLCP header for a minimum period longer than the latest possible arrival time of the SFD/PLCP header. Indication of a busy channel does not necessarily lead to the successful reception of a frame.


Copyright © 2007 IEEE. All rights reserved.

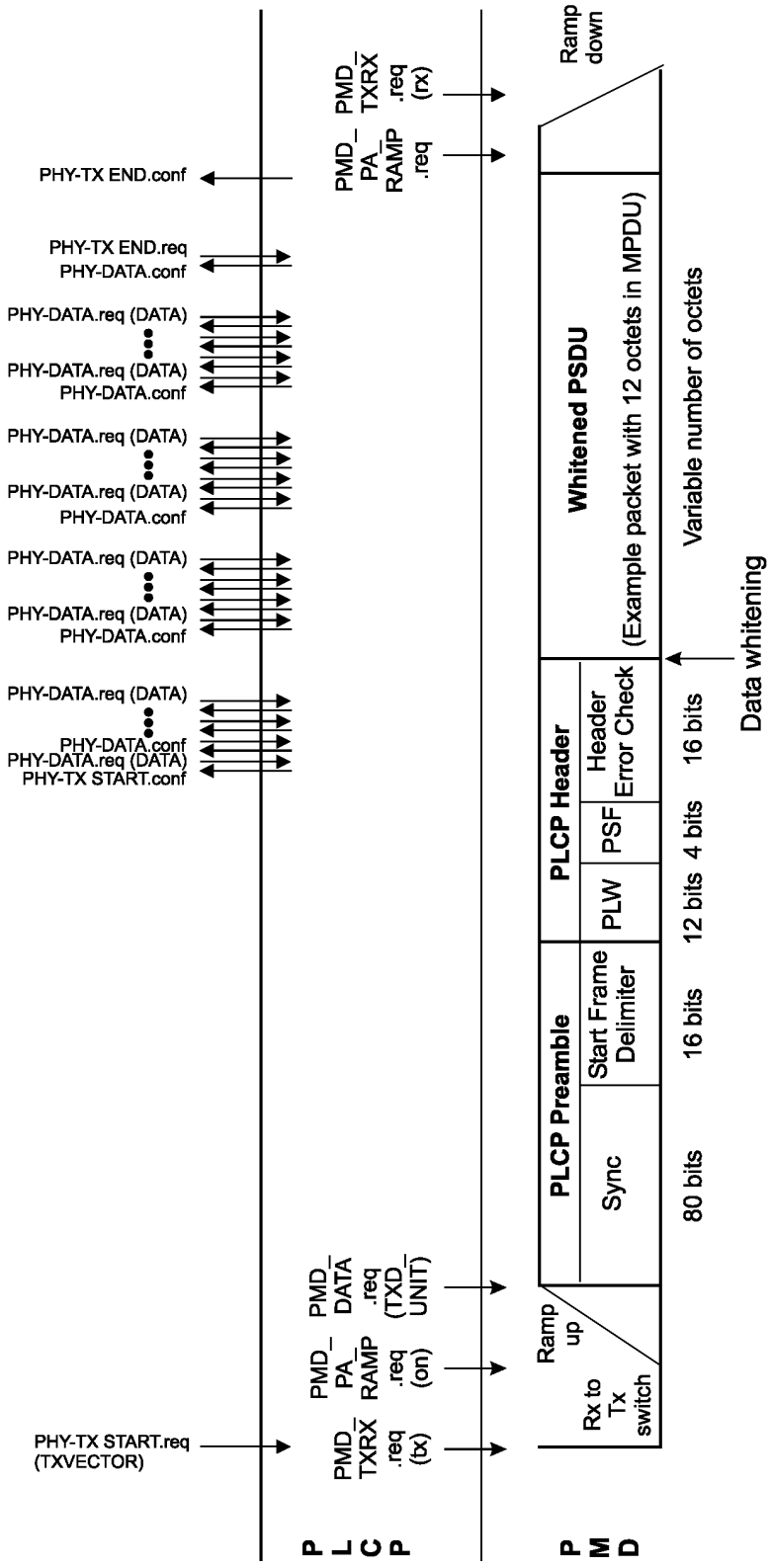



**Figure 14-8—Transmit state timing**

Copyright © 2007 IEEE. All rights reserved.

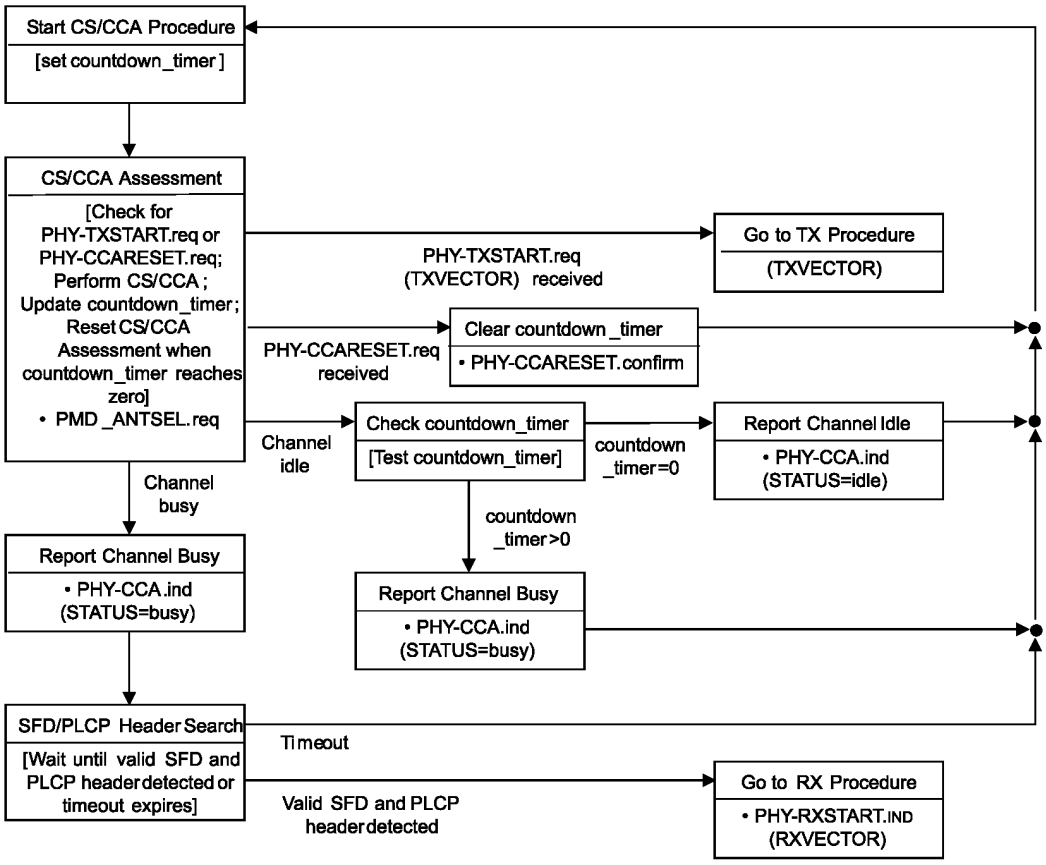



**Figure 14-9—CS/CCA state machine**

The octet/bit count remaining may be a nonzero value when returning from the receive procedure if a signal in the process of being received was lost prior to the end as determined from the LENGTH field of a valid PLCP header. The countdown timer shall be set to the octet/bit count and used to force the CS/CCA indication to remain in the BUSY state until the predicted end of the frame regardless of actual CS/CCA indications.

However, if the CS/CCA procedure indicates the start of a new frame within the countdown timer period, it is possible to transition to the receive procedure prior to the end of the countdown timer period. If the PHY transitions to receive under these conditions, the countdown timer shall be reset to the longer of

— The remaining time of the current frame or

— The length of the new frame.

When a nonzero countdown timer reaches zero, the PLCP shall reset the CS/CCA state machine to the state appropriate for the end of a complete received frame and the CS/CCA indication shall reflect the state of the channel.

If the receive procedure encountered an unsupported rate error, the PLCP shall keep the CS/CCA state at Busy for the duration of the frame by setting the countdown timer to the value corresponding to the calculated time based on the information in the PLCP header and the 33/32 expansion factor.


Copyright © 2007 IEEE. All rights reserved.

#### 14.3.3.2.2 CS/CCA state timing

Timing for priority (PIFS, DIFS), contention backoff (slot times), and CS/CCA windows is defined relative to the end of the last bit of the last packet on the air. The PLCP shall perform a CS/CCA on a minimum of one antenna within a slot time. The appropriate CS/CCA indication shall be available prior to the end of each 50 μs slot time with the performance specified in 14.6. See Figure 14-10.

If a STA has not successfully received the previous packet, the perceived packet end time and slot boundary times will have a higher uncertainty for that STA.

### 14.3.3.3 Receive PLCP

The receive PLCP is invoked by the CS/CCA procedure upon detecting a portion of the preamble sync pattern followed by a valid SFD and PLCP header.

#### 14.3.3.3.1 Receive state machine

The receive PLCP shown in Figure 14-11 includes functions that must be performed while the PPDU is being received. The receive PLCP begins upon detection of a valid SFD and PLCP header in the CS/CCA procedure. The PLCP shall set a PPDU octet/bit counter to indicate the last bit of the packet, receive the PPDU bits, and perform the data whitening decoding procedure shown in Figure 14-12 on each PPDU bit. The PLCP shall pass correctly received data octets to the MAC with a series of *PHY-DATA.indicate(DATA)*. After the last PPDU bit is received and the last octet is passed to the MAC, the PLCP shall send a *PHY-RXEND.indicate(RXERROR=no_error)* to the MAC sublayer. Upon error-free completion of a packet reception, the PLCP shall exit the receive procedure and return to the CS/CCA procedure with the octet/bit count set to 0.

If the PLCP header was decoded without a CRC error but encountered an unsupported rate, then the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate(RXERROR = unsupported_rate)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP header.

If an error was detected during the reception of the PPDU, the PLCP shall immediately complete the receive procedure with a *PHY-RXEND.indicate(RXERROR=carrier_lost)* to the MAC, and return to the CS/CCA procedure with the octet/bit count remaining and the data rate value contained in the PLCP header.

.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 547 of 1232

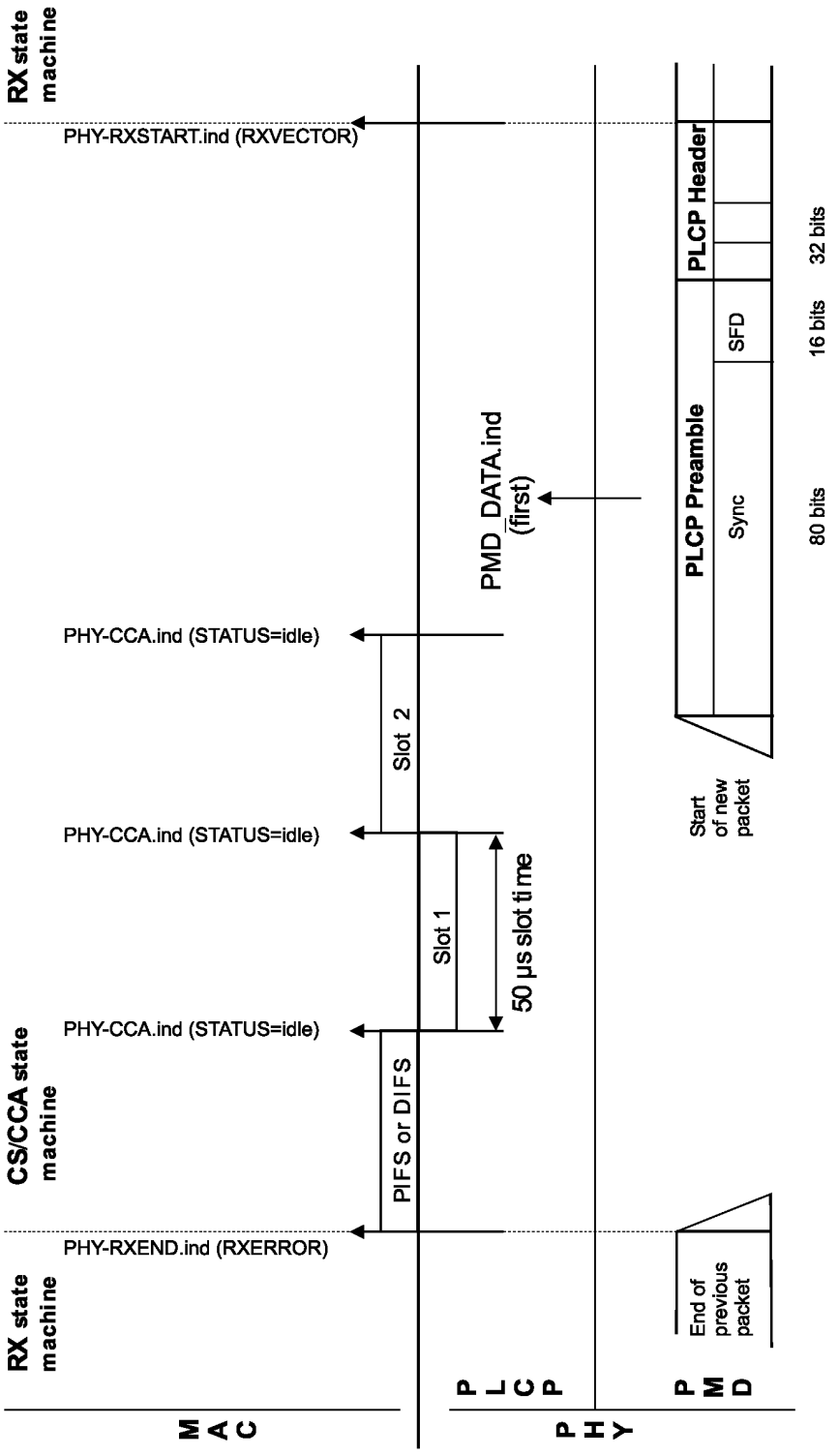



Figure 14-10—CS/CCA state timing


Copyright © 2007 IEEE. All rights reserved.

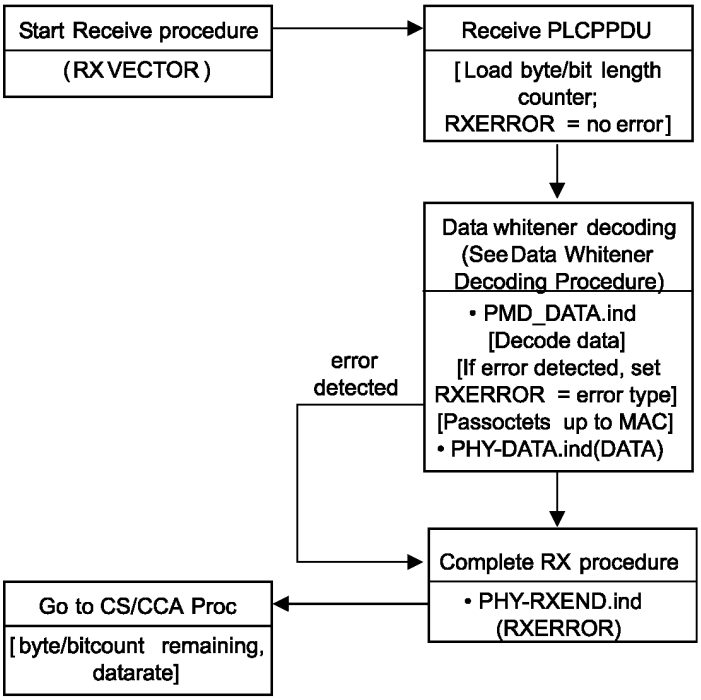



**Figure 14-11—Receive state machine**

**Data whitener decoding algorithm:**

```
/*      If MSB of stuff symbol = 1 then the next block is inverted; 0 = not inverted */
/*      Stuffing begins on first symbol of PLCP header following the SFD */
/*      Algorithm begins after verifying validity of header with HEC */

/********* Read header *********/
Read in header {b(1),...,b(32)};                        /* b(1) is first bit in */

Get number_of_PSDU_octets, rate from header;            /* rate is 1 or 2 */
number_of_symbols = (number_of_PSDU_octets*8)/rate
number_of_blocks_in_packet = truncate{(number_of_symbols + 31) / 32};
Initialize scrambler to all ones;

/********* De-whiten the PPDU data with BSE decoder and de-scrambler *********/
For n = 1 to number_of_blocks_in_packet
{
        N = min(32, # of symbols remaining);        /* N= block size in symbols */
        Read in next block {b(0),...,b(N)};/                * b(n) = {0,1} or {0,1,2,3} */

        If {[MSB of b(0)=1] then Invert {b(1),...,b(N)};    /* if invert bit=true */
        Descramble {b(1),...,b(N)};                 /* see 14.3.2.3*/
        Send {b(1),...,b(N)} to MAC
                                    /* 1 - 8 octets; use PHY-DATA.ind(DATA) for each octet. */
}
```

**Figure 14-12—Data whitener decoding procedure**

Copyright © 2007 IEEE. All rights reserved.

#### 14.3.3.3.2 Receive state timing

The receive state timing shown in Figure 14-13 is defined to begin upon detection of a valid SFD and PLCP header in the CS/CCA procedure. The PLCP shall begin receiving the variable length whitened PSDU immediately after the end of the last bit of the PLCP header. The PLCP shall send a *PHY-RXEND.indicate(RXERROR)* after receiving the last PPDU data bit.

If any error was detected during the reception of the PPDU, the PLCP may send a PHY-RXEND.indicate(RXERROR) and terminate the receive procedure before the last bit arrives.

## 14.4 PLME SAP layer management

### 14.4.1 Overview

This subclause describes the services provided by the FHSS PLME to the upper LMEs. The PLME/PMD services are defined in terms of service primitives. These primitives are abstract representations of the services and are not intended to restrict implementations.

### 14.4.2 FH PHY specific MLME procedures

#### 14.4.2.1 Overview

The specific MLME procedures required for operating the FHSS PHY are specified in this portion of the subclause. The relationship between the MLME and FH PLME procedures is also described.

#### 14.4.2.2 FH synchronization

The MLME of a compliant FH PHY STA shall perform the FH time synchronization procedure as defined in 11.1.5. This procedure provides for synchronized FH for all compliant FH PHY STAs within a single BSS or ad hoc network. The FH PLME accepts PLME-SET.request commands from the MLME to change the tune frequency at the time determined by the MLME. The tune frequency is changed by updating any combination of the Set, Pattern, and Index PHY MIB parameters.

### 14.4.3 FH PLME state machines

#### 14.4.3.1 Overview

This portion of this subclause describes the FH PLME state machines to turn the PMD on/off, reset the PLCP state machine, and change the frequency hop channel.

#### 14.4.3.2 PLME state machine

The PLME state machine in Figure 14-14 begins with a *PLME-SET.request(dot11CurrentPowerState= ON)*, which turns on the PHY circuitry, resets the PLME and PLCP state machines, and sends a *PLME-SET.confirm*. The MAC then sends a series of three *PLME-SET.request* primitives to update the dot11CurrentSet, dot11CurrentPattern, and dot11CurrentIndex PHY MIB parameters, which together tune the PMD to the selected channel. The PLME then transfers execution to the PLCP state machine as defined in 14.3.3.


Copyright © 2007 IEEE. All rights reserved.

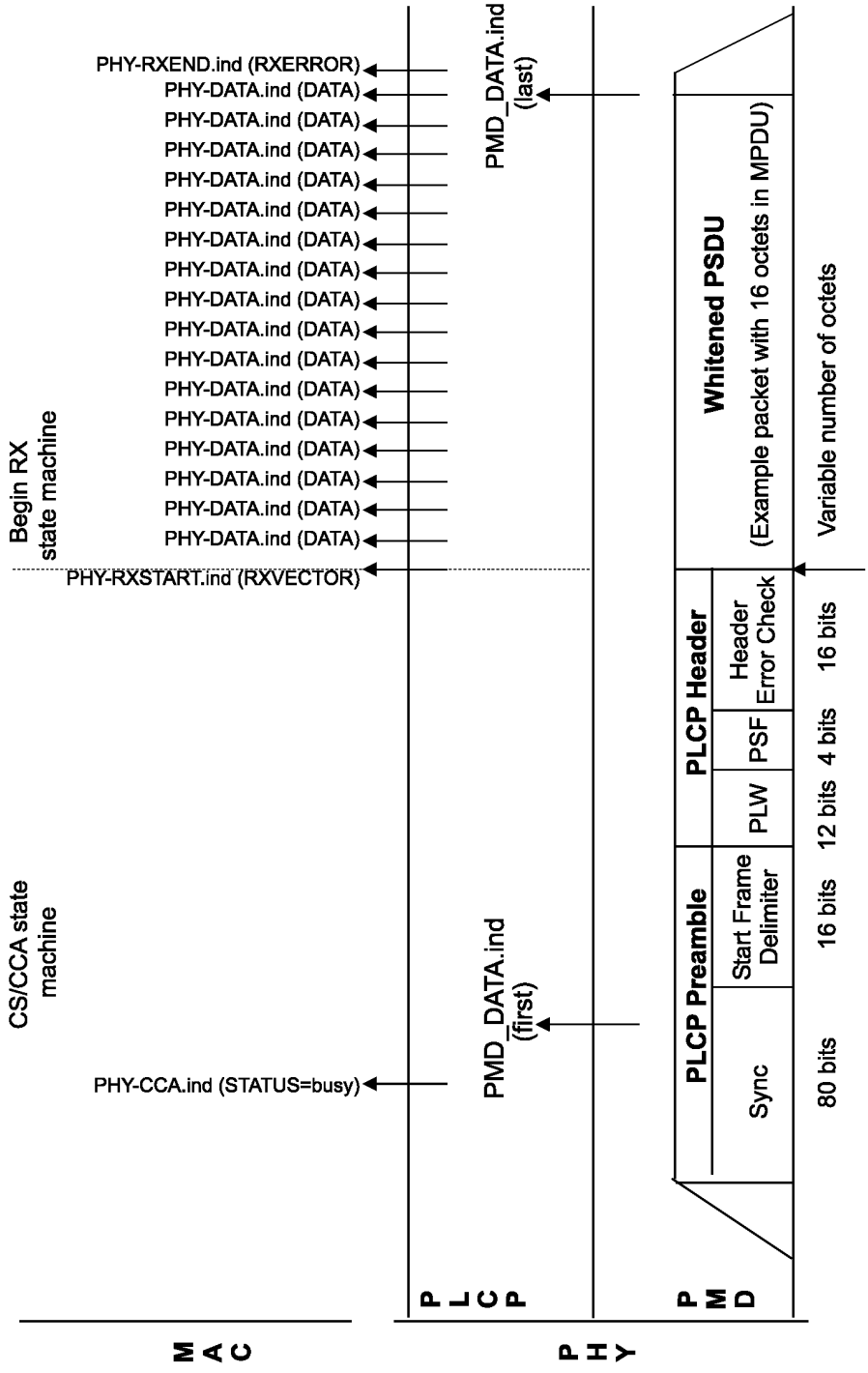



Figure 14-13—Receive timing

Copyright © 2007 IEEE. All rights reserved.

Upon receiving a PLME request from a higher level LME, the PLCP shall return execution to the PLME state machine and process the request. A *PLME-RESET.request* shall cause a reset to the PLME and PLCP state machines. A *PLME-SET.request* updating the dot11CurrentIndex or a combination of the dot11CurrentSet, dot11CurrentPattern, and dot11CurrentIndex shall cause the PLCP to terminate a receive or CS/CCA process and change frequency before returning to the PLCP state machine. A *PLME-SET.request(dot11CurrentPowerState=OFF)* shall cause the PLCP to terminate a receive or CS/CCA process, power-down the PMD circuitry, and return the PLME state machine to the idle state. *PLME-SET.request*s to any parameter other than the ones identified within this paragraph shall be executed and control shall be returned to the PLCP state machine. The MAC should not send a PLME request while the PLCP is in the transmit state.

All *PLME-GET.request*s shall be processed in parallel and with no interruption to the execution of any state machine in process.

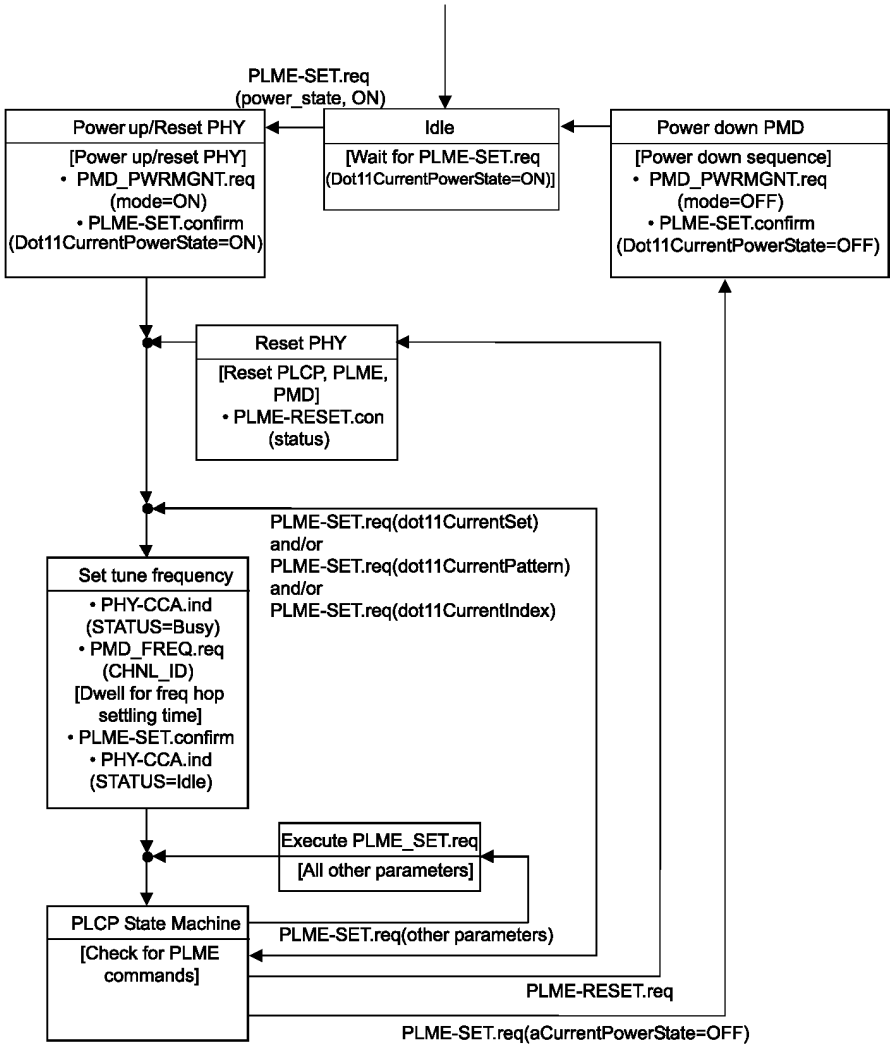



**Figure 14-14—PLME state machine**

### 14.4.3.3 PLME management primitives

The FH PLME uses the generic management primitives defined in 10.2 to manage all FH PHY parameters.


Copyright © 2007 IEEE. All rights reserved.

## 14.5 FHSS PMD sublayer services

### 14.5.1 Scope and field of application

The PMD services provided to the PLCP for the FHSS PHY are described in this subclause. Also defined in this subclause are the functional, electrical, and radio frequency (RF) characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire FHSS PHY is shown in Figure 14-15.

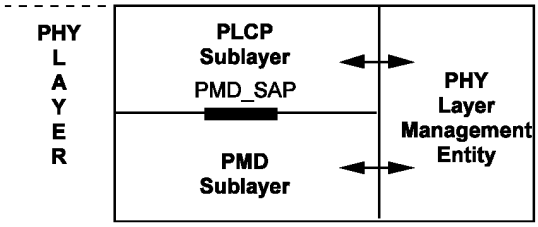



**Figure 14-15—PMD layer reference model**

### 14.5.2 Overview of services

In general, the FHSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which the signals required by these primitives are imposed onto the medium. In the FHSS PMD sublayer at the receiver the process is reversed. The combined function of the transmitting and receiving FHSS PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving PLCP sublayer.

### 14.5.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the FHSS PMD sublayer fall into the following two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions;

b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

### 14.5.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory unless otherwise specified.

#### 14.5.4.1 PMD_SAP peer-to-peer service primitives

Table 14-5 indicates the primitives for peer-to-peer interactions.

**Table 14-5—PMD_SAP peer-to-peer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_DATA | X | X | — | — |

Copyright © 2007 IEEE. All rights reserved.

#### 14.5.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 14-6 indicates the primitives for sublayer-to-sublayer interactions.

**Table 14-6—PMD_SAP sublayer-to-sublayer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_TXRX | X | — | — | — |
| PMD_PA_RAMP | X | — | — | — |
| PMD_ANTSEL | X | — | — | — |
| PMD_TXPWRLVL | X | — | — | — |
| PMD_FREQ | X | — | — | — |
| PMD_RSSI | — | X | — | — |
| PMD_PWRMGMT | X | — | — | — |

#### 14.5.4.3 PMD_SAP service primitives parameters

Table 14-7 shows the parameters used by one or more of the PMD_SAP service primitives.

**Table 14-7—List of parameters for PMD primitives**

| Parameter | Associate primitive | Value |
|---|---|---|
| TXD_UNIT | PMD_DATA.request | 1 Mb/s: 0, 1<br>2 Mb/s: 0, 1, 2, 3 |
| RXD_UNIT | PMD_DATA.indicate | 1 Mb/s: 0, 1<br>2 Mb/s: 0, 1, 2, 3 |
| RF_STATE | PMD_TXRX.request | TRANSMIT, RECEIVE |
| RAMP_STATE | PMD_PA_RAMP.request | ON, OFF |
| ANTENNA_STATE | PMD_ANTSEL.request | 1 to 255 |
| TXPWR_LEVEL | PMD_TXPWRLVL.request | LEVEL1, LEVEL2, LEVEL3, LEVEL4 |
| CHNL_ID | PMD_FREQ.request | 2–80 inclusive |
| STRENGTH | PMD_RSSI.indicate | 0 to RSSI Max |
| MODE | PMD_PWRMGMT.request | ON, OFF |


Copyright © 2007 IEEE. All rights reserved.

### 14.5.5 PMD_SAP detailed service specification

This subclause describes the services provided by each PMD primitive.

#### 14.5.5.1 PMD_DATA.request

##### 14.5.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

##### 14.5.5.1.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.request(TXD_UNIT)

The TXD_UNIT parameter can take on one of two values: one or zero. This parameter represents a single data bit. The effect of this parameter is that the PMD will properly modulate the medium to represent ones or zeros as defined in the FHSS PMD modulation specifications for a given data rate.

##### 14.5.5.1.3 When generated

This primitive is generated by the PLCP sublayer to request the transmission of a single data bit on the PMD sublayer. The bit clock is assumed to be resident or part of the PLCP and this primitive is issued at every clock edge once the PLCP has begun transmitting data.

##### 14.5.5.1.4 Effect of receipt

The receipt of this primitive will cause the PMD entity to encode and transmit a single data bit.

Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.2 PMD_DATA.indicate

##### 14.5.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

##### 14.5.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.indicate(RXD_UNIT)

The RXD_UNIT parameter can take on one of two values: one or zero. This parameter represents the current state of the medium as determined by the FHSS PMD modulation specifications for a given data rate.

##### 14.5.5.2.3 When generated

The PMD_DATA.indicate is generated to all receiving PLCP entities in the network after a PMD_DATA.request is issued.

##### 14.5.5.2.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.


Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.3 PMD_TXRX.request

##### 14.5.5.3.1 Function

This primitive is used to place the PMD entity into the transmit or receive function.

##### 14.5.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXRX.request(RF_STATE)

The RF_STATE parameter can take on one of two values: TRANSMIT or RECEIVE. When the value of the primitive is TRANSMIT, the RF state of the radio is transmit. If the value of the primitive is RECEIVE, the RF state of the radio is receive.

##### 14.5.5.3.3 When generated

This primitive is generated when the mode of the radio needs to be set or when changing from transmit to receive or receive to transmit.

##### 14.5.5.3.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the mode of the radio to be in either transmit or receive.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 557 of 1232

#### 14.5.5.4 PMD_PA_RAMP.request

##### 14.5.5.4.1 Function

This primitive defines the start of the ramp up or ramp down of the radio transmitter's power amplifier.

##### 14.5.5.4.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_PA_RAMP.request(RAMP_STATE)

The RAMP_STATE parameter can take on one of two values: ON or OFF. When the value of the primitive is ON, the state of the transmit power amplifier is "on." If the value of the primitive is OFF, the state of the transmit power amplifier is "off."

##### 14.5.5.4.3 When generated

This primitive is issued only during transmit and to establish the initial state. It is generated by the PLCP at the start of the transmit function to turn the transmitter's power amplifier "on." A power amplifier ramp-up period follows the change of state from "off" to "on." After the PLCP has transferred all required data to the PMD entity, this primitive again will be issued by the PLCP to place the transmit power amplifier back into the "off" state. A power amplifier ramp-down period follows the change of state from "on" to "off."

##### 14.5.5.4.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power amplifier to turn on or off.


Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.5 PMD_ANTSEL.request

##### 14.5.5.5.1 Function

This primitive is used to select which antenna the PMD entity will use to transmit or receive data.

##### 14.5.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_ANTSEL.request(ANTENNA_STATE)

The ANTENNA_STATE parameter can take on values from one to *N* (where *N* is the number of antennas supported). When the value of the primitive is a ONE, the PMD will switch to antenna 1 for receive or transmit; if the value of the primitive is TWO, the PMD entity will switch to antenna 2 for receive or transmit, etc.

##### 14.5.5.5.3 When generated

This primitive is generated at various times by the PLCP entity to select an antenna. During receive, this primitive can be used to manage antenna diversity. During transmit, this primitive can be use to select a transmit antenna. This primitive will also be used during CCA.

##### 14.5.5.5.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to select the antenna specified.

Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.6 PMD_TXPWRLVL.request

##### 14.5.5.6.1 Function

This primitive defines the power level the PMD entity will use to transmit data.

##### 14.5.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXPWRLVL.request(TXPOWER_LEVEL)

The TXPOWER_LEVEL parameter can be one of the values listed in Table 14-8.

**Table 14-8—Transmit power levels**

| TXPWR_LEVEL | Level description |
|---|---|
| LEVEL1 | Defined as TxPowerLevel1 in MIB |
| LEVEL2 | Defined as TxPowerLevel2 in MIB |
| LEVEL3 | Defined as TxPowerLevel3 in MIB |
| LEVEL4 | Defined as TxPowerLevel4 in MIB |
| LEVEL5 | Defined as TxPowerLevel5 in MIB |
| LEVEL6 | Defined as TxPowerLevel6 in MIB |
| LEVEL7 | Defined as TxPowerLevel7 in MIB |
| LEVEL8 | Defined as TxPowerLevel8 in MIB |

##### 14.5.5.6.3 When generated

This primitive is generated as part of the transmit sequence.

##### 14.5.5.6.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the transmit power level to be modify.


Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.7 PMD_FREQ.request

##### 14.5.5.7.1 Function

This primitive defines the frequency the PMD entity will use to receive or transmit data. Because changing the RF is not an immediate function, this primitive serves also as an indication of the start of this process. The completion of this process is dictated by other PMD specifications.

##### 14.5.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_FREQ.request(CHANNEL_ID)

The CHANNEL_ID parameter can be one of the values listed in Table 14-11, Table 14-12, Table 14-13, or Table 14-14 (in 14.6.5).

##### 14.5.5.7.3 When generated

This primitive is generated by the PLCP when a change to a new frequency is required.

##### 14.5.5.7.4 Effect of receipt

The receipt of this primitive by the PMD entity will cause the radio to change to a new frequency defined by the value of the CHNL_ID.

Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.8 PMD_RSSI.indicate

##### 14.5.5.8.1 Function

This primitive transfers a receiver signal strength indication of the physical medium from the PMD sublayer to the PLCP sublayer. This value will be used by the PLCP to perform any diversity or CCA functions required by the PLCP or other sublayers.

##### 14.5.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_RSSI.indicate(STRENGTH)

The STRENGTH parameter can be a value from 0 to 15. This parameter is an indication by the PMD sublayer of the magnitude of the energy observed at the selected antenna. This reported value is used to generate the RSSI term in the PHY-RXSTART.ind(RXVECTOR) primitive and might also be used by any diversity function. Because RSSI is only used in a relative manner by the MAC sublayer, this parameter is defined to have no more than 16 values, ranging from 0 through RSSI_Max. The value zero is the weakest signal strength, while RSSI_Max is the strongest signal strength.

##### 14.5.5.8.3 When generated

This primitive is generated continually by the PMD entity to transfer a RSSI to the PLCP.

##### 14.5.5.8.4 Effect of receipt

The effect of receipt of this primitive by the PLCP is unspecified in this standard.


Copyright © 2007 IEEE. All rights reserved.

#### 14.5.5.9 PMD_PWRMGMT.request

##### 14.5.5.9.1 Function

This primitive is used by the higher layer entities to manage or control the power consumption of the PMD when not in use. This allows higher layer entities to put the radio into a sleep or standby mode when receipt or sending of any data is not expected.

##### 14.5.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_PWRMGMT.request(MODE)

The MODE parameter can have one of two values: ON or OFF. When the value of the parameter is ON, the PMD entity will enter into a fully functional mode that allows it to send or receive data. When the value of the parameter is OFF, the PMD entity will place itself in a standby or power-saving mode. In the low-power mode, the PMD entity is not expected to be able to perform any request by the PLCP, nor is it expected to indicate any change in PMD state or status.

##### 14.5.5.9.3 When generated

This primitive is delivered by the PLCP but actually is generated by a higher level LME.

##### 14.5.5.9.4 Effect of receipt

Upon receipt of this primitive, the PMD entity will enter a fully functional or low power consumption state depending on the value of the primitive's parameter.

Copyright © 2007 IEEE. All rights reserved.

## 14.6 FHSS PMD sublayer, 1.0 Mb/s

### 14.6.1 1 Mb/s PMD operating specifications, general

In general, the PMD accepts convergence layer service primitives and provides the actual means by which the signals required by these primitives are imposed on the medium. In the PMD sublayer at the receiver, the process is reversed. The combined function of the transmitting and receiving PMD sublayers results in a data stream, timing information, and receive parameter information being delivered to the receiving convergence sublayer.

### 14.6.2 Regulatory requirements

WLANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed within this standard. Implementers are referred to the appropriate regulatory sources for further information. Table 14-9 specify the current regulatory requirements for various geographic areas at the time this standard was developed. They are provided for information only and are subject to change or revision at any time.

### 14.6.3 Operating frequency range

A conformant PMD implementation shall be able to select the carrier frequency ($F_C$) from the full geographic-specific set of available carrier frequencies. Table 14-9 summarizes these frequencies for a number of geographic locations.

**Table 14-9—Operating frequency range**

| Lower Limit | Upper limit | Regulatory range | Geography |
|---|---|---|---|
| 2.402 GHz | 2.480 GHz | 2.400–2.4835 GHz | China |
| 2.402 GHz | 2.480 GHz | 2.400–2.4835 GHz | North America |
| 2.402 GHz | 2.480 GHz | 2.400–2.4835 GHz | Europe[a] |
| 2.473 GHz | 2.495 GHz | 2.471–2.497 GHz | Japan |
| 2.447 GHz | 2.473 GHz | 2.445–2.475 GHz | Spain |
| 2.448 GHz | 2.482 GHz | 2.4465–2.4835 GHz | France |
| NOTE—The frequency ranges in this table are subject to the geographic-specific regulatory authorities. | | | |

[a]Excluding Spain and France.

### 14.6.4 Number of operating channels

The number of transmit and receive frequency channels used for operating the PMD entity is 79 for the United States and Europe, and 23 for Japan. Table 14-10 summarizes these frequencies for a number of geographic locations. This is more fully defined in Table 14-11 through Table 14-14.


Copyright © 2007 IEEE. All rights reserved.

**Table 14-10—Number of operating channels**

| Minimum | Hopping set | Geography |
|---|---|---|
| 75 | 79 | China |
| 75 | 79 | North America |
| 20 | 79 | Europe[a] |
| N/A | 23 | Japan |
| 20 | 27 | Spain |
| 20 | 35 | France |
| NOTE—The number of required hopping channels is subject to the geographic-specific regulatory authorities. | | |

[a]Excluding Spain and France.

### 14.6.5 Operating channel center frequency

The channel center frequency is defined in sequential 1.0 MHz steps beginning with the first channel, channel 2.402 GHz for China, the United States, and Europe excluding Spain and France, as listed in Table 14-11. The channel centers for Japan, starting at 2.473 GHz with 1 MHz increments, are listed in Table 14-12. The channel centers for Spain and France are listed in Table 14-13 and Table 14-14, respectively.

**Table 14-11—Requirements in China, North America and Europe (excluding Spain and France; values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 2 | 2.402 | 28 | 2.428 | 54 | 2.454 |
| 3 | 2.403 | 29 | 2.429 | 55 | 2.455 |
| 4 | 2.404 | 30 | 2.430 | 56 | 2.456 |
| 5 | 2.405 | 31 | 2.431 | 57 | 2.457 |
| 6 | 2.406 | 32 | 2.432 | 58 | 2.458 |
| 7 | 2.407 | 33 | 2.433 | 59 | 2.459 |
| 8 | 2.408 | 34 | 2.434 | 60 | 2.460 |
| 9 | 2.409 | 35 | 2.435 | 61 | 2.461 |
| 10 | 2.410 | 36 | 2.436 | 62 | 2.462 |
| 11 | 2.411 | 37 | 2.437 | 63 | 2.463 |
| 12 | 2.412 | 38 | 2.438 | 64 | 2.464 |
| 13 | 2.413 | 39 | 2.439 | 65 | 2.465 |
| 14 | 2.414 | 40 | 2.440 | 66 | 2.466 |
| 15 | 2.415 | 41 | 2.441 | 67 | 2.467 |
| 16 | 2.416 | 42 | 2.442 | 68 | 2.468 |

Copyright © 2007 IEEE. All rights reserved.

**Table 14-11—Requirements in China, North America and Europe *(continued)***
**(excluding Spain and France; values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 17 | 2.417 | 43 | 2.443 | 69 | 2.469 |
| 18 | 2.418 | 44 | 2.444 | 70 | 2.470 |
| 19 | 2.419 | 45 | 2.445 | 71 | 2.471 |
| 20 | 2.420 | 46 | 2.446 | 72 | 2.472 |
| 21 | 2.421 | 47 | 2.447 | 73 | 2.473 |
| 22 | 2.422 | 48 | 2.448 | 74 | 2.474 |
| 23 | 2.423 | 49 | 2.449 | 75 | 2.475 |
| 24 | 2.424 | 50 | 2.450 | 76 | 2.476 |
| 25 | 2.425 | 51 | 2.451 | 77 | 2.477 |
| 26 | 2.426 | 52 | 2.452 | 78 | 2.478 |
| 27 | 2.427 | 53 | 2.453 | 79 | 2.479 |
| — | — | — | — | 80 | 2.480 |

**Table 14-12—Requirements in Japan**
**(values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 73 | 2.473 | 81 | 2.481 | 89 | 2.489 |
| 74 | 2.474 | 82 | 2.482 | 90 | 2.490 |
| 75 | 2.475 | 83 | 2.483 | 91 | 2.491 |
| 76 | 2.476 | 84 | 2.484 | 92 | 2.492 |
| 77 | 2.477 | 85 | 2.485 | 93 | 2.493 |
| 78 | 2.478 | 86 | 2.486 | 94 | 2.494 |
| 79 | 2.479 | 87 | 2.487 | 95 | 2.495 |
| 80 | 2.480 | 88 | 2.488 | — | — |

**Table 14-13—Requirements in Spain**
**(values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 47 | 2.447 | 56 | 2.456 | 65 | 2.465 |
| 48 | 2.448 | 57 | 2.457 | 66 | 2.466 |
| 49 | 2.449 | 58 | 2.458 | 67 | 2.467 |
| 50 | 2.450 | 59 | 2.459 | 68 | 2.468 |


Copyright © 2007 IEEE. All rights reserved.

**Table 14-13—Requirements in Spain *(continued)***
**(values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 51 | 2.451 | 60 | 2.460 | 69 | 2.469 |
| 52 | 2.452 | 61 | 2.461 | 70 | 2.470 |
| 53 | 2.453 | 62 | 2.462 | 71 | 2.471 |
| 54 | 2.454 | 63 | 2.463 | 72 | 2.472 |
| 55 | 2.455 | 64 | 2.464 | 73 | 2.473 |

**Table 14-14—Requirements in France**
**(values specified in GHz)**

| Channel # | Value | Channel # | Value | Channel # | Value |
|---|---|---|---|---|---|
| 48 | 2.448 | 60 | 2.460 | 72 | 2.472 |
| 49 | 2.449 | 61 | 2.461 | 73 | 2.473 |
| 50 | 2.450 | 62 | 2.462 | 74 | 2.474 |
| 51 | 2.451 | 63 | 2.463 | 75 | 2.475 |
| 52 | 2.452 | 64 | 2.464 | 76 | 2.476 |
| 53 | 2.453 | 65 | 2.465 | 77 | 2.477 |
| 54 | 2.454 | 66 | 2.466 | 78 | 2.478 |
| 55 | 2.455 | 67 | 2.467 | 79 | 2.479 |
| 56 | 2.456 | 68 | 2.468 | 80 | 2.480 |
| 57 | 2.457 | 69 | 2.469 | 81 | 2.481 |
| 58 | 2.458 | 70 | 2.470 | 82 | 2.482 |
| 59 | 2.459 | 71 | 2.471 | — | — |

### 14.6.6 Occupied channel bandwidth

Occupied channel bandwidth shall meet all applicable local geographic regulations for 1 MHz channel spacing. The rate at which the PMD entity will hop is governed by the MAC. The hop rate is an attribute with a maximum dwell time subject to local geographic regulations.

### 14.6.7 Minimum hop rate

The minimum hop rate shall be governed by the regulatory authorities.

### 14.6.8 Hop sequences

The hopping sequence of an individual PMD entity is used to create a pseudo-random hopping pattern utilizing uniformly the designated frequency band. Sets of hopping sequences are used to co-locate multiple

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 567 of 1232

PMD entities in similar networks in the same geographic area and to enhance the overall efficiency and throughput capacity of each individual network.

An FH pattern, $F_x$, consists of a permutation of all frequency channels defined in Table 14-11 and Table 14-12. For a given pattern number, $x$, the hopping sequence can be written as follows:

$$F_x = \{f_x(1), f_x(2), \ldots f_x(p)\} \quad (14\text{-}1)$$

where

$f_x(i)$ is the channel number (as defined in 14.6.4) for $i^{th}$ frequency in $x^{th}$ hopping pattern;

$p$ is the number of frequency channels in hopping pattern (79 for China, North America and most of Europe, 23 for Japan, 35 for France, 27 for Spain).

Given the hopping pattern number, $x$, and the index for the next frequency, $i$ (in the range 1 to $p$), the channel number shall be defined to be as follows:

$f_x(i)$ = $[b(i) + x] \bmod (79) + 2$ in China, North America and most of Europe, with $b(i)$ defined in Table 14-15.

= $[(i - 1) \times x] \bmod (23) + 73$ in Japan.

= $[b(i) + x] \bmod (27) + 47$ in Spain with $b(i)$ defined in Table 14-16.

= $[b(i) + x] \bmod (35) + 48$ in France with $b(i)$ defined in Table 14-17.

**Table 14-15—Base-Hopping sequence *b(i)* for China, North America and most of Europe**

| *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 11 | 76 | 21 | 18 | 31 | 34 | 41 | 14 | 51 | 20 | 61 | 48 | 71 | 55 |
| 2 | 23 | 12 | 29 | 22 | 11 | 32 | 66 | 42 | 57 | 52 | 73 | 62 | 15 | 72 | 35 |
| 3 | 62 | 13 | 59 | 23 | 36 | 33 | 7 | 43 | 41 | 53 | 64 | 63 | 5 | 73 | 53 |
| 4 | 8 | 14 | 22 | 24 | 71 | 34 | 68 | 44 | 74 | 54 | 39 | 64 | 17 | 74 | 24 |
| 5 | 43 | 15 | 52 | 25 | 54 | 35 | 75 | 45 | 32 | 55 | 13 | 65 | 6 | 75 | 44 |
| 6 | 16 | 16 | 63 | 26 | 69 | 36 | 4 | 46 | 70 | 56 | 33 | 66 | 67 | 76 | 51 |
| 7 | 71 | 17 | 26 | 27 | 21 | 37 | 60 | 47 | 9 | 57 | 65 | 67 | 49 | 77 | 38 |
| 8 | 47 | 18 | 77 | 28 | 3 | 38 | 27 | 48 | 58 | 58 | 50 | 68 | 40 | 78 | 30 |
| 9 | 19 | 19 | 31 | 29 | 37 | 39 | 12 | 49 | 78 | 59 | 56 | 69 | 1 | 79 | 46 |
| 10 | 61 | 20 | 2 | 30 | 10 | 40 | 25 | 50 | 45 | 60 | 42 | 70 | 28 | — | — |

**Table 14-16—Base-Hopping sequence *b(i)* for Spain**

| *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* |
|---|---|---|---|---|---|
| 1 | 13 | 10 | 19 | 19 | 14 |
| 2 | 4 | 11 | 8 | 20 | 1 |
| 3 | 24 | 12 | 23 | 21 | 20 |
| 4 | 18 | 13 | 15 | 22 | 7 |


Copyright © 2007 IEEE. All rights reserved.

**Table 14-16—Base-Hopping sequence *b(i)* for Spain *(continued)***

| *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* |
|---|---|---|---|---|---|
| 5 | 5 | 14 | 22 | 23 | 16 |
| 6 | 12 | 15 | 9 | 24 | 2 |
| 7 | 3 | 16 | 21 | 25 | 11 |
| 8 | 10 | 17 | 0 | 26 | 17 |
| 9 | 25 | 18 | 6 | 27 | 26 |

**Table 14-17—Base-Hopping sequence *b(i)* for France**

| *i* | *b(i)* | *i* | *b(i)* | *i* | *b(i)* |
|---|---|---|---|---|---|
| 1 | 17 | 13 | 31 | 25 | 15 |
| 2 | 5 | 14 | 20 | 26 | 3 |
| 3 | 18 | 15 | 29 | 27 | 11 |
| 4 | 32 | 16 | 22 | 28 | 30 |
| 5 | 23 | 17 | 12 | 29 | 24 |
| 6 | 7 | 18 | 6 | 30 | 9 |
| 7 | 16 | 19 | 28 | 31 | 27 |
| 8 | 4 | 20 | 14 | 32 | 19 |
| 9 | 13 | 21 | 25 | 33 | 2 |
| 10 | 33 | 22 | 0 | 34 | 21 |
| 11 | 26 | 23 | 8 | 35 | 34 |
| 12 | 10 | 24 | 1 | — | — |

The sequences are designed to ensure some minimum distance in frequency between contiguous hops. The minimum hop size is 6 MHz for China, North America and Europe, including Spain and France, and 5 MHz for Japan.

The hopping pattern numbers $x$ are divided into three sets. The sets are designed to avoid prolonged collision periods between different hopping sequences in a set. Hopping sequence sets contain 26 sequences for China, North America and Europe, and 4 sequences per set for Japan:

For China, North America and most of Europe:

$x$ = {0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,60,63,66,69,72,75} Set 1
$x$ = {1,4,7,10,13,16,19,22,25,28,31,34,37,40,43,46,49,52,55,58,61,64,67,70,73,76} Set 2
$x$ = {2,5,8,11,14,17,20,23,26,29,32,35,38,41,44,47,50,53,56,59,62,65,68,71,74,77} Set 3

For Japan:

$x$ = {6,9,12,15} Set 1
$x$ = {7,10,13,16} Set 2
$x$ = {8,11,14,17} Set 3

Copyright © 2007 IEEE. All rights reserved.

For Spain:

$x$ = {0,3,6,9,12,15,18,21,24} Set 1
$x$ = {1,4,7,10,13,16,19,22,25} Set 2
$x$ = {2,5,8,11,14,17,20,23,26} Set 3

For France:

$x$ = {0,3,6,9,12,15,18,21,24,27,30} Set 1
$x$ = {1,4,7,10,13,16,19,22,25,28,31} Set 2
$x$ = {2,5,8,11,14,17,20,23,26,29,32} Set 3

The three sets of hopping sequences for China, North America and most of Europe, of 26 patterns each, are listed in Table B.1, Table B.2, and Table B.3. Similarly, there are three sets for Japan of four patterns each. The three sets for Spain have nine patterns each. The three sets for France have 11 patterns each. The channel numbers listed under each pattern refer to the actual frequency values listed in Table 14-11 and Table 14-12.

### 14.6.9 Unwanted emissions

Conformant PMD implementations shall limit the emissions that fall outside of the operating frequency range, defined in Table 14-9, to the geographically applicable limits**.**

### 14.6.10 Modulation

The minimum set of requirements for a PMD to be compliant with the IEEE 802.11 FHSS PHY shall be as follows.

The PMD shall be capable of operating using 2GFSK modulation with a nominal bandwidth bit time (BT) = 0.5. The PMD shall accept symbols from the set {{1},{0}} from the PLCP. The symbol {1} shall be encoded with a peak deviation of ($+f_d$), giving a peak transmit frequency of ($F_c+f_d$), which is greater than the carrier center frequency ($F_c$). The symbol {0} shall be encoded with a peak frequency deviation of ($-f_d$), giving a peak transmit frequency of ($F_c-f_d$).

An incoming bit stream at 1 Mb/s will be converted to symbols at *Fclk* = 1 *M*symbols/s, as shown in Table 14-18.

**Table 14-18—Symbol encoding into carrier deviation (1 Mb/s, 2GFSK)**

| Symbol | Carrier deviation |
|---|---|
| 1 | $1/2 \times h2 \times Fclk$ |
| 0 | $-1/2 \times h2 \times Fclk$ |
| NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping. | |

The deviation factor h2 for 2GFSK (measured as difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) will nominally be 0.32.

The minimum frequency deviation, as shown in Figure 14-16, shall be greater than 110 kHz relative to the nominal center frequency $F_c$. $F_d$ is the average center frequency of the last 8 bits of the Preamble SYNC field, measured as the deviation at the midsymbol. Midsymbol is defined as the point that is midway between the zero crossings derived from a best fit to the last 8 bits of the Sync field. Maximum deviation is not specified, but modulation is subject to the occupied bandwidth limits of 14.6.5.


Copyright © 2007 IEEE. All rights reserved.

The zero crossing error shall be less than ±1/8 of a symbol period. The zero crossing error is the time difference between the ideal symbol periods and measured crossings of $F_c$. This is illustrated in Figure 14-16.

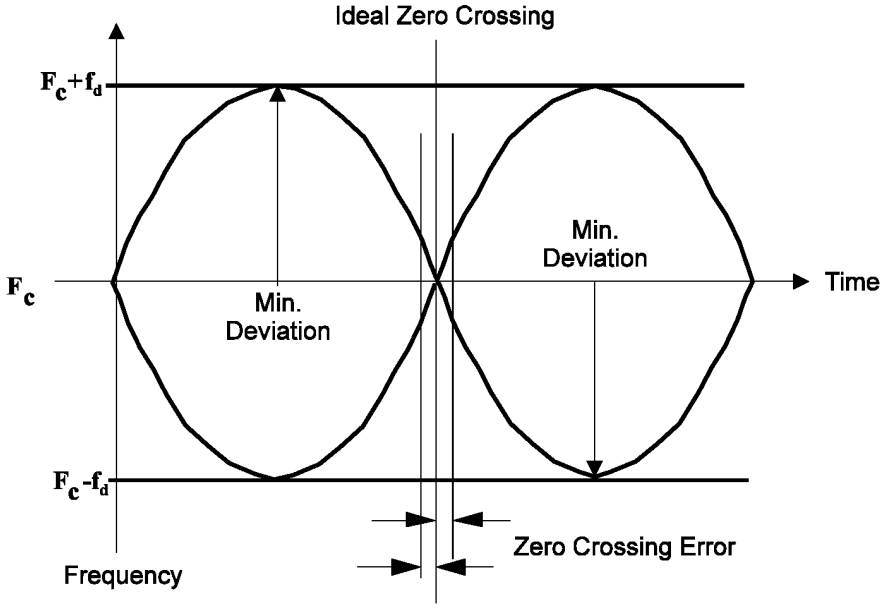



**Figure 14-16—Transmit modulation mask**

### 14.6.11 Channel data rate

A compliant IEEE 802.11 FHSS PMD shall be capable of transmitting and receiving at a nominal data rate of 1.0 Mb/s ±50 ppm.

### 14.6.12 Channel switching/settling time

The time to change from one operating channel frequency, as specified in 14.6.3, is defined as 224 µs. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within ±60 kHz of the nominal channel center frequency as outlined in 14.6.3.

### 14.6.13 Receive to transmit switch time

The maximum time for a conformant PMD to switch the radio from the receive state to the transmit state and place the start of the first bit on the air shall be 19 µs. At the end of this 19 µs, the RF carrier shall be within the nominal transmit power level range, and within the described modulation specifications.

### 14.6.14 PMD transmit specifications

The following portion of this subclause describes the transmit functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP, and the transmit PMD entity provides the actual means by which the signals required by the PLCP primitives are imposed onto the medium.

#### 14.6.14.1 Nominal transmit power

The nominal transmit power of a frame is defined as the power averaged between the start of the first symbol in the PLCP header to the end of the last symbol in the PLCP header. When in the transmit state, the transmit

Copyright © 2007 IEEE. All rights reserved.

power shall be within 2 dB of the nominal transmit power from the start of the Preamble SYNC field to the last symbol at the end of the frame.

#### 14.6.14.2 Transmit power levels

Unless governed by more stringent local geographic regulations, the radiated emissions from compliant devices shall meet IEEE Std C95.1-1991 limits for controlled or uncontrolled environments, in accordance with their intended usage. In addition, all conformant PMD implementations shall support at least one power level with a minimum equivalent isotropically radiated power (EIRP) of 10 mW.

#### 14.6.14.3 Transmit power level control

If a conformant PMD implementation has the ability to transmit in a manner that results in the EIRP of the transmit signal exceeding the level of 100 mW, at least one level of TPC shall be implemented. This TPC shall be such that the level of the emission is reduced to a level at or below 100 mW under the influence of said power control.

#### 14.6.14.4 Transmit spectrum shape

Within the operational frequency band the transmitter shall pass a spectrum mask test. The duty cycle between Tx and Rx is nominally 50% and the transmit frame length is nominally 400 μs. The adjacent channel power is defined as the sum of the power measured in a 1 MHz band. For a pseudo-random data pattern, the adjacent channel power shall be a function of the offset between channel number *N* and the assigned transmitter channel *M*, where *M* is the actual transmitted center frequency and *N* is a channel separated from it by an integer number of megahertz.

Channel offset:

$|N-M|=2$ –20 dBm or –40 dBc, whichever is the lower power.

$|N-M|\geq 3$ –40 dBm or –60 dBc, whichever is the lower power.

The levels given in dBc are measured relative to the transmitter power measured in a 1 MHz channel centered on the transmitter center frequency. The adjacent channel power and the transmitter power for this subclause of the specification shall be measured with a resolution bandwidth of 100 kHz, a video bandwidth of 300 kHz, and a peak detector, and with the measurement device set to maximum hold.

For any transmit center frequency M, two exceptions to the spectrum mask requirements are permitted within the operational frequency band, provided the exceptions are less than –50 dBc, where each offset channel exceeded counts as a separate exception. An exception occurs when the total energy within a given 1 MHz channel as defined in 14.6.5 exceeds the levels specified in 14.6.14.1 through 14.6.14.4.

#### 14.6.14.5 Transmit center frequency tolerance

The PMD transmit center frequency shall be within ±60 kHz of the nominal center frequency as specified in 14.6.5.

#### 14.6.14.6 Transmitter ramp periods

The transmitter shall go from off to within 2 dB of the nominal transmit power in 8 μs or less. The transmitter shall go from within 2 dB of the nominal transmit power to off (less than –50 dBm) in 8 μs or less.


Copyright © 2007 IEEE. All rights reserved.

### 14.6.15 PMD receiver specifications

The following portion of this subclause describes the receive functions and parameters associated with the PMD sublayer. In general, these are specified by primitives from the PLCP. The Receive PMD entity provides the actual means by which the signals required by the PLCP primitives are recovered from the medium. The PMD sublayer monitors signals on the medium and will return symbols from the set {{1},{0}} to the PLCP sublayer.

#### 14.6.15.1 Input signal range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with a frame error ratio (FER) ≤ 3% for PSDUs of 400 octets generated with pseudo-random data, for receiver input signal levels in the range from –20 dBm to the receiver sensitivity (as specified in 14.6.15.4), across the frequency band of operation.

#### 14.6.15.2 Receive center frequency acceptance range

An IEEE 802.11 FHSS-compliant PMD shall meet all specifications with an input signal having a center frequency range of ±60 kHz from nominal.

#### 14.6.15.3 CCA power threshold

In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –85 dBm that starts synchronously with respect to slot times as specified in 14.3.3.2.1, the PHY shall signal busy, with a 90% probability of detection, during the preamble within the CCA window. In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –85 dBm that starts asynchronously with respect to slot times as specified in 14.3.3.2.1, the PHY shall signal busy, with a 70% probability of detection, during the preamble within the CCA window. In the presence of any IEEE 802.11-compliant 1 Mb/s FH PMD signal above –65 dBm, the PHY shall signal busy, with a 70% probability of detection, during random data within the CCA window. This specification applies to a PMD operating with a nominal EIRP of < 100 mW. A compliant PMD operating at a nominal output power greater than 100 mW shall use the following equation to define the CCA threshold, where $P_t$ represents transmit power.

$$\text{CCA threshold (preamble)} = -85\ \text{dBm} - \left[5 \times \log_{10}\left(\frac{P_t}{100\ \text{mW}}\right)\right]\ \text{dBm}$$

CCA threshold (random data) = CCA threshold (preamble) + 20 dB

#### 14.6.15.4 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudo-random data. The sensitivity shall be less than or equal to –80 dBm. The reference sensitivity is defined as –80 dBm for the 1 Mb/s FH PHY specifications.

#### 14.6.15.5 Intermodulation

Intermodulation protection (IMp) is defined as the ratio of the minimum amplitude of one of two equal interfering signals to the desired signal amplitude, where the interfering signals are spaced 4 MHz and 8 MHz removed from the center frequency of the desired signal, both on the same side of center frequency. The IMp ratio is established at the interfering signal level that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudo-random data, when the desired signal is –77 dBm. Each interfering signal is modulated with the FH PMD modulation uncorrelated in time to each other or the desired signal. The PMD shall have the IMp for the interfering signal at 4 MHz and 8 MHz be ≥ 30 dB.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 573 of 1232

#### 14.6.15.6 Desensitization (Dp)

Desensitization (Dp) is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER at the output of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudo-random data, when the desired signal is –77 dBm. The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 14-19. The spectral purity of the interferer shall be sufficient to ensure that the measurement is limited by the receiver performance.

**Table 14-19—1 Mb/s Dp**

| Interferer frequency[a] | Dp minimum |
|---|---|
| $M = N \pm 2$ | 30 dB |
| $M = N \pm 3$ or more | 40 dB |

[a]Where $M$ is the interferer frequency and $N$ is the desired channel frequency.

#### 14.6.15.7 Receiver radiation

The signal leakage when receiving shall not exceed –50 dBm EIRP in the operating frequency range. The FHSS PHY shall conform with out-of-band spurious emissions by regulatory bodies.

#### 14.6.16 Operating temperature range

Two temperature ranges for full operation compliance to the FH PHY are specified. Type 1 is defined as 0 °C to 40 °C and is designated for office environments. Type 2 is defined as –30 °C to +70 °C and is designated for industrial environments.

### 14.7 FHSS PMD sublayer, 2.0 Mb/s

#### 14.7.1 Overview

This subclause details the RF specification differences of the optional 2 Mb/s operation from the baseline 1 Mb/s PMD as contained in 14.6. Unless otherwise specified in this subclause, the compliant PMD shall also meet all requirements of 14.6 when transmitting at 2 Mb/s. When implementing the 2 Mb/s option, the preamble and PHY header shall be transmitted at 1 Mb/s. STAs implementing the 2 Mb/s option shall also be capable of transmitting and receiving PPDUs at 1 Mb/s.

#### 14.7.2 4GFSK modulation

For an FHSS 2 Mb/s PMD, the modulation scheme shall be 4GFSK, with a nominal symbol-period bandwidth product (BT) of 0.5. The four-level deviation factor, defined as the frequency separation of adjacent symbols divided by symbol rate, h4, shall be related to the deviation factor of the 2GFSK modulation, h2, by the following equation:

$$h4/h2 = 20055 \pm 0.01$$

An incoming bit stream at 2 Mb/s will be converted to 2-bit words or symbols, with a rate of Fclk = 1 Msymbol/s. The first received bit will be encoded as the LMB of the symbol in Table 14-20. The bits will be encoded into symbols as shown in Table 14-20.


Copyright © 2007 IEEE. All rights reserved.

**Table 14-20—Symbol encoding into carrier deviation**

| 1 Mb/s, 2GFSK | |
|---|---|
| **Symbol** | **Carrier deviation** |
| 1 | $1/2 \times h2 \times Fclk$ |
| 0 | $-1/2 \times h2 \times Fclk$ |
| **2 Mb/s, 4GFSK** | |
| **Symbol** | **Carrier deviation** |
| 10 | $3/2 \times h4 \times Fclk$ |
| 11 | $1/2 \times h4 \times Fclk$ |
| 01 | $-1/2 \times h4 \times Fclk$ |
| 00 | $-3/2 \times h4 \times Fclk$ |
| NOTE—These deviation values are measured using the center symbol of 7 consecutive symbols of the same value. The instantaneous deviation will vary due to Gaussian pulse shaping. | |

The deviation factor h2 for 2GFSK (measured as the difference between frequencies measured in the middle of 0000 and 1111 patterns encountered in the SFD, divided by 1 MHz) will nominally be 0.32. The deviation factor h2 will be no less than 0.30 (with maximum dictated by regulatory bandwidth requirement). Accordingly, h4 (measured as a difference between the outermost frequencies, divided by 3, divided by 1 MHz) is nominally $0.45 \times 0.32 = 0.144$, and it will be no less than $0.45 \times 0.3 = 0.135$.

The modulation error shall be less than ±15 kHz at the midsymbol time for 4GFSK, from the frequency deviations specified above, for a symbol surrounded by identical symbols, and less than ±25 kHz for any symbol. The deviation is relative to the actual center frequency of the RF carrier. For definition purposes, the actual center frequency is the midfrequency between symbols 11 and 01. The actual center frequency shall be within ±60 kHz of the nominal channel center frequency defined in 14.6.5 and shall not vary by more than ±10 kHz/ms, from the start to end of the PPDU. The peak-to-peak variation of the actual center frequency over the PPDU shall not exceed 15 kHz. Symbols and terms used within this subclause are illustrated in Figure 14-17.

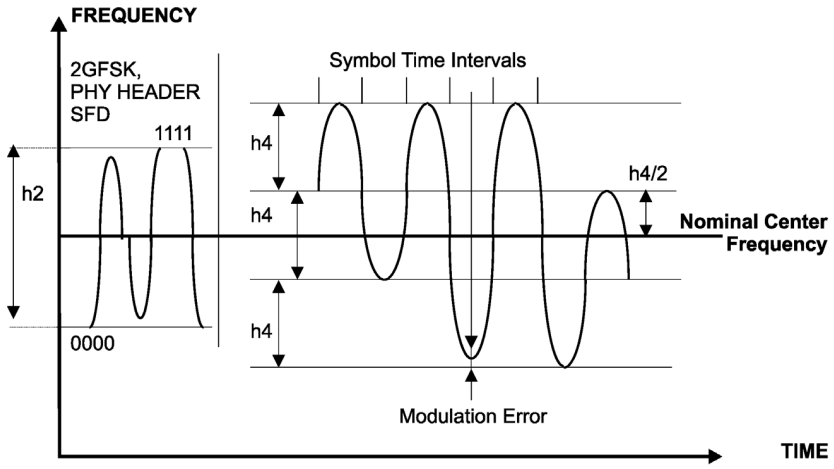



**Figure 14-17—4GFSK transmit modulation**

Copyright © 2007 IEEE. All rights reserved.

#### 14.7.2.1 Frame structure for HS FHSS PHY

The high rate FHSS PPDU consists of PLCP preamble, PLCP header, and whitened PSDU. The PLCP preamble and PLCP header format are identical to the 1 Mb/s PHY, as described in 14.3.2. The whitened PSDU is transmitted in 2GFSK, 4GFSK, or potentially a higher-rate format, according to the rate chosen. The rate is indicated in a 3-bit field in a PLCP header, having a value of 1 or 2 bits per symbol (or Mb/s).

The PPDU is transmitted as four-level symbols, with the amount determined by number_of_symbols = (number_of_PSDU_octets × 8)/rate.

The input bits are scrambled according to the method in 14.3.2.3.

The scrambled bit stream is divided into groups of rate (1 or 2) consecutive bits. The bits are mapped into symbols according to Table 14-20.

A bias suppression algorithm is applied to the resulting symbol stream. The bias suppression algorithm is defined in 14.3.2.3, Figure 14-4, and Figure 14-7. A polarity control symbol is inserted prior to each block of 32 symbols (or less for the last block). The polarity control signals are 4GFSK symbols 10 or 00. The algorithm is equivalent to the case of 2GFSK, with the polarity symbol 2GFSK “1” replaced with 4GFSK symbol “10,” and the 2GFSK polarity symbol “0” replaced with a 4GFSK symbol “00.”

### 14.7.3 Channel data rate

The data rate for the whitened PSDU at the optional rate shall be 2.0 Mb/s ±50 ppm.

#### 14.7.3.1 Input dynamic range

The PMD shall be capable of recovering a conformant PMD signal from the medium, as described in related subclauses, with an FER ≤ 3% for PSDUs of 400 octets generated with pseudo-random data, for receiver input signal levels in the range from –20 dBm to the receiver sensitivity (as specified in 14.7.3.2), across the frequency band of operation.

#### 14.7.3.2 Receiver sensitivity

The sensitivity is defined as the minimum signal level required for an FER of 3% for PSDUs of 400 octets generated with pseudo-random data. The sensitivity shall be less than or equal to –75 dBm. The reference sensitivity is defined as –75 dBm for the 2 Mb/s FH PHY specifications.

#### 14.7.3.3 IMp

IMp is defined as the ratio to –77 dBm of the minimum amplitude of one of the two equal-level interfering signals at 4 MHz and 8 MHz removed from center frequency, both on the same side of center frequency, that cause the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudo-random data, when the desired signal is –72 dBm (3 dB above the specified sensitivity specified in 14.7.3.2). Each interfering signal is modulated with the FH 1 Mb/s PMD modulation uncorrelated in time to each other or the desired signal. The FHSS optional 2 Mb/s rate IMp shall be ≥ 25 dB.

#### 14.7.3.4 Dp

Dp is defined as the ratio to measured sensitivity of the minimum amplitude of an interfering signal that causes the FER of the receiver to be increased to 3% for PSDUs of 400 octets generated with pseudo-random data, when the desired signal is –72 dB (3 dB above sensitivity specified in 14.7.3.2). The interfering signal shall be modulated with the FHSS PMD modulation uncorrelated in time to the desired signal. The minimum Dp shall be as given in Table 14-21.


Copyright © 2007 IEEE. All rights reserved.

**Table 14-21—2 Mb/s Dp**

| Interferer frequency[a] | DP minimum |
|---|---|
| $M = N \pm 2$ | 20 dB |
| $M = N \pm 3$ or more | 30 dB |

[a]Where $M$ is the interferer frequency and $N$ is the desired channel frequency.

## 14.8 FHSS PHY MIB

### 14.8.1 FH PHY attributes

This subclause defines the attributes for the FHSS MIB. Table 14-22 lists these attributes and the default values. A description of each attribute is given in 14.8.2.

**Table 14-22—FHSS PHY attributes**

| Attribute | Default value | Operational semantics | Operational behavior |
|---|---|---|---|
| dot11PHYType | FHSS = X'01' | Static | Identical for all FH PHYs |
| dot11RegDomainsSupported | FCC = X'10'<br>IC = X'20'<br>ETSI = X'30'<br>Spain = X'31'<br>France = X'32'<br>Japan = X'40'<br>China = X'50'<br>Other = X'00' | Static | Implementation dependent |
| dot11CurrentRegDomain | X'00' | Dynamic LME | Implementation dependent |
| dot11TempType | Type 1 = X'01'<br>Type 2 = X'02'<br>Type 3 = X'03' | Static | Implementation dependent |
| dot11SupportedDataRatesTX | 1 Mb/s = X'02' mandatory<br>2 Mb/s = X'04' optional | Static | Identical for all FH PHYs |
| dot11SupportedDataRatesRX | 1 Mb/s = X'02' mandatory<br>2 Mb/s = X'04' optional | Static | Identical for all FH PHYs |
| dot11SupportedTxAntennas | Ant 1 = X'01'<br>Ant 2 = X'02'<br>Ant 3 = X'03'<br>Ant n = n | Static | Implementation dependent |
| dot11CurrentTxAntenna | Ant 1 = default | Dynamic LME | Implementation dependent |
| dot11SupportedRxAntennas | Ant 1 = X'01'<br>Ant 2 = X'02'<br>Ant 3 = X'03'<br>Ant n = n | Static | Implementation dependent |
| dot11DiversitySupport | Available = X'01'<br>Not avail. = X'02'<br>Control avail. = X'03' | Static | Implementation dependent |

Copyright © 2007 IEEE. All rights reserved.

**Table 14-22—FHSS PHY attributes *(continued)***

| Attribute | Default value | Operational semantics | Operational behavior |
|---|---|---|---|
| dot11DiversitySelectionRx | Ant 1 = X'01'<br>Ant 2 = X'02'<br>Ant 3 = X'03'<br>Ant 4 = X'04'<br>Ant 5 = X'05'<br>Ant 6 = X'06'<br>Ant 7 = X'07'<br>Ant 8 = X'08' | Dynamic LME | Implementation dependent |
| dot11NumberSupportedPowerLevels | Lvl1 = X'01'<br>Lvl2 = X'02'<br>Lvl3 = X'03'<br>Lvl4 = X'04'<br>Lvl5 = X'05'<br>Lvl6 = X'06'<br>Lvl7 = X'07'<br>Lvl8 = X'08' | Static | Implementation dependent |
| dot11TxPowerLevel1 | Factory default | Static | Implementation dependent |
| dot11TxPowerLevel2 | Factory default | Static | Implementation dependent |
| dot11TxPowerLevel3 | Factory default | Static | Implementation dependent |
| dot11TxPowerLevel4 | Factory def. | Static | Implementation dependent |
| dot11TxPowerLevel5 | Factory def. | Static | Implementation dependent |
| dot11TxPowerLevel6 | Factory def. | Static | Implementation dependent |
| dot11TxPowerLevel7 | Factory def. | Static | Implementation dependent |
| dot11TxPowerLevel8 | Factory def. | Static | Implementation dependent |
| dot11CurrentTxPowerLevel | TxPowerLevel1 | Dynamic LME | Implementation dependent |
| dot11HopTime | 224 µs | Static | Identical for all FH PHYs |
| dot11CurrentChannelNumber | X'00' | Dynamic PLME | |
| dot11MaxDwellTime | 390 TU | Static | Regulatory domain dependent |
| dot11CurrentSet | X'00' | Dynamic PLME | |
| dot11CurrentPattern | X'00' | Dynamic PLME | |
| dot11CurrentIndex | X'00' | Dynamic PLME | |
| dot11CurrentPowerState | X'01' off<br>X'02' on | Dynamic LME | |

NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entity. When an attribute is defined as dynamic, the column also shows which entity has control over the attribute. LME refers to the MLME, while PHY refers to the PLME.

### 14.8.2 FH PHY attribute definitions

#### 14.8.2.1 dot11PHYType

The dot11PHYType is FHSS. The LME uses this attribute to determine what PLCP and PMD are providing services to the MAC. It also is used by the MAC to determine what MAC sublayer management state machines must be invoked to support the PHY. The value of this attribute is defined as the integer 01 to indicate the FHSS PHY.


Copyright © 2007 IEEE. All rights reserved.

#### 14.8.2.2 dot11RegDomainsSupported

Operational requirements for FHSS PHY are defined by agencies representing certain geographical regulatory domains. These regulatory agencies may define limits on various parameters that differ from region to region. This parameters may include dot11TxPowerLevels, and dot11MaxDwellTime, as well as the total number of frequencies in the hopping pattern. The values shown in Table 14-23 indicate regulatory agencies supported by this document.

**Table 14-23—Regulatory domain codes**

| Code point | Regulatory agency | Region |
|---|---|---|
| X'10' | FCC | United States |
| X'20' | IC | Canada |
| X'30' | ETSI | Most of Europe |
| X'31' | Spain | Spain |
| X'32' | France | France |
| X'40' | Japan | Japan |
| X'50' | Radio Administration of Information Industry Ministry | China |

Because a PLCP and PMD might be designed to support operation in more than one regulatory domain, this attribute can actually represent a list of agencies. This list can be one or more of the above agencies and must be terminated using the null terminator. Upon activation of the PLCP and PMD, the information in this list must be used to set the value of the dot11CurrentRegDomain attribute.

#### 14.8.2.3 dot11CurrentRegDomain

The dot11CurrentRegDomain attribute for the FHSS PHY is defined as the regulatory domain under which the PMD is currently operating. This value must be one of the values listed in the dot11RegDomainsSupported list. This MIB attribute is managed by the LME.

#### 14.8.2.4 dot11TempType

The parameter dot11TempType defines the temperature range supported by the PHY. Type 1 equipment (X'01') supports a temperature range of 0 °C to 40 °C. Type 2 equipment (X'02') supports a temperature range of –20 °C to +55 °C. Type 3 equipment (X'03') supports a temperature range of –30 °C to +70 °C.

#### 14.8.2.5 dot11CurrentPowerState

The dot11CurrentPowerState attribute for the FHSS PHY allows the MLME to control the power state of the PHY. This attribute can be updated using the PLMESET.request. The permissible values are ON and OFF.

#### 14.8.2.6 dot11SupportedDataRatesTX

The dot11SupportedDataRatesTX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the transmit mode for this implementation. Table 14-24 shows the possible values appearing in the list.

Copyright © 2007 IEEE. All rights reserved.

**Table 14-24—Supported data rate codes (dot11SupportedDataRatesTX)**

| Code point | Data rate |
|---|---|
| X'02' | 1 Mb/s |
| X'04' | 2 Mb/s |
| X'00' | Null terminator |

#### 14.8.2.7 dot11SupportedDataRatesRX

The dot11SupportedDataRatesRX attribute for the FHSS PHY is defined as a null terminated list of supported data rates in the receive mode for this implementation. Table 14-25 shows the possible values appearing in the list.

**Table 14-25—Supported data rate codes (dot11SupportedDataRatesRX)**

| Code point | Data rate |
|---|---|
| X'02' | 1 Mb/s |
| X'04' | 2 Mb/s |
| X'00' | Null terminator |

#### 14.8.2.8 aMPDUMaxLength

The aMPDUMaximumLength attribute for the FHSS PHY is defined as the maximum PSDU, in octets, that the PHY shall ever be capable of accepting. This value for the FHSS PHY is set at 4095 octets. The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mb/s and 800 octets at 2 Mb/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band.

#### 14.8.2.9 dot11SupportedTxAntennas

The dot11SupportedTxAntennas attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to transmit data. Table 14-26 shows the possible values appearing in the list, where $N \leq 255$.

**Table 14-26—Number of transmit antennas**

| Code point | Antenna number |
|---|---|
| X'01' | Tx Antenna 1 |
| X'02' | Tx Antenna 2 |
| X'03' | Tx Antenna 3 |
| … | … |
| *N* | Tx Antenna *N* |
| X'00' | Null terminator |


Copyright © 2007 IEEE. All rights reserved.

#### 14.8.2.10 dot11CurrentTxAntenna

The dot11CurrentTxAntenna attribute for the FHSS PHY is used to describe the current antenna the implementation is using for transmission. This value should represent one of the antennas appearing in the dot11SupportedTxAntennas list.

#### 14.8.2.11 dot11SupportedRxAntenna

The dot11SupportedRxAntennas attribute for the FHSS PHY is defined as a null terminated list of antennas that this implementation can use to receive data. In the FHSS PHY primitives, one of these values is passed as part of the PHY-RXSTART.indicate to the MAC sublayer for every received packet. Table 14-27 shows the possible values appearing in the list, where $N \leq 255$.

**Table 14-27—Number of receive antennas**

| Code point | Antenna number |
|---|---|
| X'01' | Rx Antenna 1 |
| X'02' | Rx Antenna 2 |
| X'03' | Rx Antenna 3 |
| … | … |
| *N* | Rx Antenna *N* |
| X'00' | Null terminator |

#### 14.8.2.12 dot11DiversitySupport

The dot11DiversitySupport attribute for the FHSS PHY is used to describe the implementation's diversity support. Table 14-28 shows the possible values appearing in the list.

**Table 14-28—Diversity support codes**

| Code point | Diversity support |
|---|---|
| X'01' | Diversity available |
| X'02' | No diversity |
| X'03' | Control available |

The value X'01' indicates that this implementation uses two or more antennas for diversity. The value X'02' indicates that the implementation has no diversity support. The value X'03' indicates that the choice of antennas used during diversity is programmable. (See 14.8.2.13.)

#### 14.8.2.13 dot11DiversitySelectionRx

The dot11DiversitySelectionRx attribute for the FHSS PHY is a null terminated list describing the receive antenna or antennas currently in use during diversity and packet reception. Table 14-29 shows the possible values appearing in the list, where $N \leq 255$.

The null terminated list can consist of one or more of the receive antennas listed in the dot11SupportedRxAntennas attribute. This attribute can be changed dynamically by the LME.

Copyright © 2007 IEEE. All rights reserved.

**Table 14-29—Diversity select antenna codes**

| Code point | Antenna number |
|---|---|
| X'01' | Rx Antenna 1 |
| X'02' | Rx Antenna 2 |
| X'03' | Rx Antenna 3 |
| … | … |
| *N* | Rx Antenna *N* |
| X'00' | Null terminator |

#### 14.8.2.14 dot11NumberSupportedPowerLevels

The dot11NumberSupportedPowerLevels attribute for the FHSS PHY describes the number of power levels this implementation supports. This attribute can be an integer of value 1 through 8, inclusive.

#### 14.8.2.15 dot11TxPowerLevel1-8

Some implementations may provide up to eight different transmit power levels. The dot11TxPowerLevels attribute for the FHSS PHY is a list of up to eight power levels supported. Table 14-30 describes the list.

**Table 14-30—Transmit power levels**

| Attribute | Power level |
|---|---|
| TxPowerLevel1 | Default setting |
| TxPowerLevel2 | Level 2 |
| TxPowerLevel3 | Level 3 |
| TxPowerLevel4 | Level 4 |
| TxPowerLevel5 | Level 5 |
| TxPowerLevel6 | Level 6 |
| TxPowerLevel7 | Level 7 |
| TxPowerLevel8 | Level 8 |

#### 14.8.2.16 dot11CurrentTxPowerLevel

The dot11CurrentTxPowerLevel attribute for the FHSS PHY is defined as the current transmit output power level. This level shall be one of the levels implemented in the list of attributes called dot11TxPowerLevel*N* (where *N* is 1–8). This MIB attribute is also used to define the sensitivity of the CCA mechanism when the output power exceeds 100 mW. This MIB attribute is managed by the LME.

#### 14.8.2.17 dot11HopTime

The dot11HopTime attribute for the FHSS PHY describes the time allocated for the PHY to change to a new frequency. For the FHSS PHY, this time period is 224 μs.


Copyright © 2007 IEEE. All rights reserved.

#### 14.8.2.18 dot11CurrentChannelNumber

The dot11CurrentChannelNumber attribute for the FHSS PHY is defined as the current operating channel number of the PMD. The values of this attribute correspond to the values shown in Table 14-11. This MIB attribute is managed by the PLME and is updated as the result of a PLMESET.request to dot11CurrentSet, dot11CurrentPattern, or dot11CurrentIndex.

#### 14.8.2.19 dot11MaxDwellTime

The dot11MaxDwellTime attribute for the FHSS PHY is defined as the maximum time the PMD can dwell on a channel and meet the requirements of the current regulatory domain. For the FCC regulatory domain, this number is 390 TU (FCC = 400 ms). The recommended dwell time for the FHSS PHY is 19 TU.

#### 14.8.2.20 dot11CurrentSet

The FHSS PHY contains three sets of hopping patterns. The dot11CurrentSet attribute for the FHSS PHY defines what set the STA is using to determine the hopping pattern. Its value can be 1, 2, or 3. This attribute is managed by the PLME. When dot11MultiDomainCapabilityImplemented is true, this value may also be 0. A value of 0 indicates that the hopping pattern is to be obtained from the Hopping Pattern Table information element most recently received in a Beacon or Probe Response frame.

#### 14.8.2.21 dot11CurrentPattern

There are up to 78 patterns in each hopping set used by the FHSS PHY. The dot11CurrentPattern attribute for the FHSS PHY defines the *x* value used in Equation (14-1) in 14.6.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 0 to 77, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

#### 14.8.2.22 dot11CurrentIndex

The FHSS PHY addresses each channel in the selected hopping pattern through an index. The dot11CurrentIndex attribute for the FHSS PHY defines the *i* value used in the equation for $f_x(i)$ in 14.6.8 to calculate the current channel number. Its value has various ranges, always within the overall range of 1 to 79, depending on the dot11CurrentRegDomain. This attribute is managed by the PLME.

#### 14.8.2.23 dot11CurrentPowerState

The parameter dot11CurrentPowerState defines the operational state of the FHSS PHY. When this attribute has a value of X'01', the PHY is “OFF.” When this attribute has a value of X'02', the PHY is “ON.” This attribute is managed by the PLME.

Copyright © 2007 IEEE. All rights reserved.

## 14.9 FH PHY characteristics

Table 14-31 gives the static FH PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive. The definitions of these characteristics are in 10.4.3.

**Table 14-31—FH PHY characteristics**

| Characteristic | Value | Notes |
|---|---|---|
| aSlotTime | 50 μs | — |
| aSIFSTime | 28 μs | In order to account for variations between implementations, this value has a tolerance as specified in 9.2.3.1. |
| aCCATime | 27 μs | This period includes the aRxRFDelay and the aRxPLCPDelay. |
| aPHY-RX-START-Delay | 128 μs | The delay from the start of the preamble to the issuance of the RX-START.indicate by the PHY. |
| aRxTxTurnaroundTime | 20 μs | — |
| aTxPLCPDelay | 1 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxPLCPDelay | 2 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aRxTxSwitchTime | 10 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aTxRampOnTime | 8 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aTxRampOffTime | 8 μs | — |
| aTxRFDelay | 1 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxRFDelay | 4 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aAirPropagationTime | 1 μs | Variations in the actual propagation time are accounted for in the allowable range of aSIFSTime. |
| aMACProcessingDelay | 2 μs | Implementers may choose to increase or decrease this delay as long as the requirements of aSIFSTime are met. |
| aPreambleLength | 96 μs | — |
| aPLCPHeaderLength | 32 μs | — |
| aMPDUDurationFactor | 31250000 | This factor is calculated as $[(33/32) - 1] \times 10^9$ to account for the expansion due to the data whitener encoding algorithm. |
| aMPDUMaxLength | 4095 | The recommended value for maximum PSDU length in an FHSS PHY system is 400 octets at 1 Mb/s and 800 octets at 2 Mb/s, which corresponds to a frame duration less than 3.5 ms. These values are optimized to achieve high performance in a variety of RF channel conditions, particularly with respect to indoor multipath, channel stability for moving STAs, and interference in the 2.4 GHz band. |
| aCWmin | 15 | — |
| aCWmax | 1023 | — |


Copyright © 2007 IEEE. All rights reserved.

# 15. DSSS PHY specification for the 2.4 GHz band designated for ISM applications

## 15.1 Overview

The PHY for the DSSS system is described in this clause. The RF LAN system is aimed for the 2.4 GHz band designated for ISM applications.

The DSSS system provides a WLAN with both a 1 Mb/s and a 2 Mb/s data payload communication capability. The DSSS system uses baseband modulations of differential binary phase shift keying (DBPSK) and differential quadrature phase shift keying (DQPSK) to provide the 1 Mb/s and 2 Mb/s data rates, respectively.

### 15.1.1 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC by the 2.4 GHz DSSS system are described in this clause. The DSSS PHY consists of two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function shall be supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs each using the DSSS system.

### 15.1.2 DSSS PHY functions

The 2.4 GHz DSSS PHY architecture is depicted in the reference model shown in Figure 5-10 (in 5.7). The DSSS PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail in the following subclauses.

The DSSS PHY service shall be provided to the MAC through the PHY service primitives described in Clause 12.

#### 15.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services.

#### 15.1.2.2 PMD sublayer

The PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with the 2.4 GHz ISM bands using direct sequence modulation.

#### 15.1.2.3 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

### 15.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 585 of 1232

simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 DSSS-PHY-compliant developer.

The service of a layer or sublayer is a set of capabilities that it offers to a user in the next-higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

## 15.2 DSSS PLCP sublayer

### 15.2.1 Overview

This subclause provides a convergence procedure in which MPDUs are converted to and from PPDUs. During transmission, the MPDU shall be prepended with a PLCP preamble and header to create the PPDU. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the MPDU.

### 15.2.2 PLCP frame format

Figure 15-1 shows the format for the PPDU including the DSSS PLCP preamble, the DSSS PLCP header, and the MPDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: IEEE 802.11 Signaling (SIGNAL), IEEE 802.11 Service (SERVICE), length (LENGTH), and CRC-16 (CRC). Each of these fields is described in detail in 15.2.3.

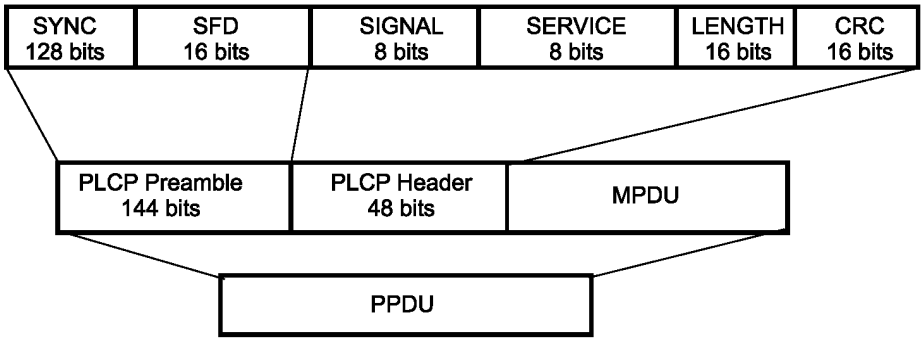



**Figure 15-1—PLCP frame format**

### 15.2.3 PLCP field definitions

The entire PLCP preamble and header shall be transmitted using the 1 Mb/s DBPSK modulation described in 15.4.7. All transmitted bits shall be scrambled using the feedthrough scrambler described in 15.2.4.

#### 15.2.3.1 PLCP SYNC field

The SYNC field shall consist of 128 bits of scrambled ones. This field shall be provided so that the receiver can perform the necessary operations for synchronization.

#### 15.2.3.2 PLCP SFD

The SFD shall be provided to indicate the start of PHY-dependent parameters within the PLCP preamble. The SFD shall be a 16-bit field, X'F3A0' (MSB to LSB). The LSB shall be transmitted first in time.


Copyright © 2007 IEEE. All rights reserved.

#### 15.2.3.3 PLCP IEEE 802.11 SIGNAL field

The 8-bit IEEE 802.11 SIGNAL field indicates to the PHY the modulation that shall be used for transmission (and reception) of the MPDU. The data rate shall be equal to the signal field value multiplied by 100 kbit/s. The DSSS PHY currently supports two mandatory modulation services given by the following 8-bit words, where the LSB shall be transmitted first in time:

a) X'0A' (MSB to LSB) for 1 Mb/s DBPSK

b) X'14' (MSB to LSB) for 2 Mb/s DQPSK

The DSSS PHY rate change capability is described in 15.2.5. This field shall be protected by the CRC-16 FCS described in 15.2.3.6.

#### 15.2.3.4 PLCP IEEE 802.11 SERVICE field

The 8-bit IEEE 802.11 SERVICE field shall be reserved for future use. The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 15.2.3.6.

#### 15.2.3.5 PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 16-bit integer that indicates the number of microseconds (16 to $2^{16}$–1 as defined by aMPDUMaxLength) required to transmit the MPDU. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 12.3.5.4. The LENGTH field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 15.2.3.6.

#### 15.2.3.6 PLCP CRC field

The IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE, and LENGTH fields shall be protected with a CRC-16 FCS. The CRC-16 FCS shall be the ones complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling.

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a packet length of 192 μs (24 octets) would be given by the following:

0101 0000 0000 0000 0000 0011 0000 0000 (leftmost bit transmitted first in time)

The ones complement FCS for these protected PLCP preamble bits would be the following:

0101 1011 0101 0111 (leftmost bit transmitted first in time)

Figure 15-2 depicts this example.

An illustrative example of the CRC-16 FCS using the information from Figure 15-2 follows in Figure 15-3.

Copyright © 2007 IEEE. All rights reserved.

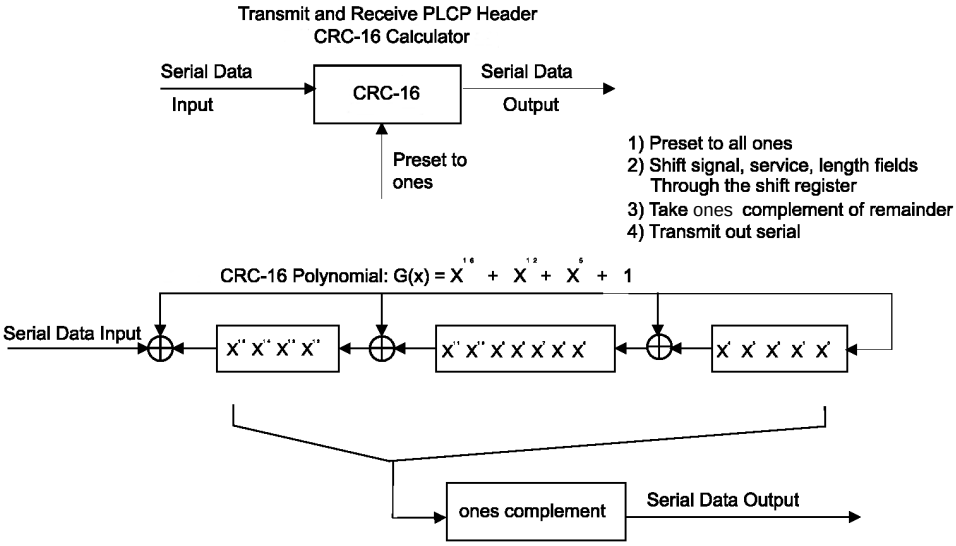



**Figure 15-2—CRC-16 implementation**

| Data | CRC registers MSB LSB | |
|---|---|---|
| | 1111111111111111 | ; initialize preset to ones |
| 0 | 1110111111011111 | |
| 1 | 1101111110111110 | |
| 0 | 1010111101011101 | |
| 1 | 0101111010111010 | |
| 0 | 1011110101110100 | |
| 0 | 0110101011001001 | |
| 0 | 1101010110010010 | |
| 0 | 1011101100000101 | |
| 0 | 0110011000101011 | |
| 0 | 1100110001010110 | |
| 0 | 1000100010001101 | |
| 0 | 0000000100111011 | |
| 0 | 0000001001110110 | |
| 0 | 0000010011101100 | |
| 0 | 0000100111011000 | |
| 0 | 0001001110110000 | |
| 0 | 0010011101100000 | |
| 0 | 0100111011000000 | |
| 0 | 1001110110000000 | |
| 0 | 0010101100100001 | |
| 0 | 0101011001000010 | |
| 0 | 1010110010000100 | |
| 1 | 0101100100001000 | |
| 1 | 1010001000110001 | |
| 0 | 0101010001000011 | |
| 0 | 1010100010000110 | |
| 0 | 0100000100101101 | |
| 0 | 1000001001011010 | |
| 0 | 0001010010010101 | |
| 0 | 0010100100101010 | |
| 0 | 0101001001010100 | |
| 0 | 1010010010101000 | |
| | 0101101101010111 | ; ones complement, result = CRC FCS parity |

**Figure 15-3—Example CRC calculation**


Copyright © 2007 IEEE. All rights reserved.

### 15.2.4 PLCP/DSSS PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted by the DSSS PHY. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 15-4 and Figure 15-5 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

The scrambler should be initialized to any state except all ones when transmitting.

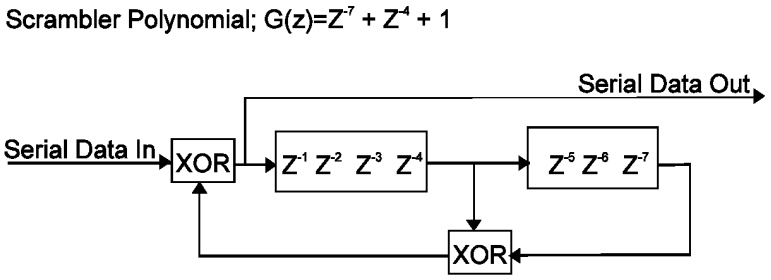



**Figure 15-4—Data scrambler**

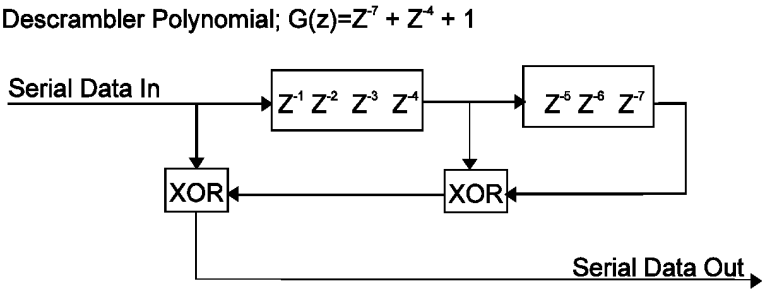



**Figure 15-5—Data descrambler**

### 15.2.5 PLCP data modulation and modulation rate change

The PLCP preamble shall be transmitted using the 1 Mb/s DBPSK modulation. The IEEE 802.11 SIGNAL field shall indicate the modulation that shall be used to transmit the MPDU. The transmitter and receiver shall initiate the modulation indicated by the IEEE 802.11 SIGNAL field starting with the first symbol (1 bit for DBPSK or 2 bits for DQPSK) of the MPDU. The MPDU transmission rate shall be set by the DATA-RATE parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 15.4.4.1.

### 15.2.6 Transmit PLCP

The transmit PLCP is shown in Figure 15-6.

In order to transmit data, PHY-TXSTART.request shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate channel through STA management via the PLME. Other transmit parameters such as DATARATE, TX antenna, and TX power are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR) as described in 15.4.4.2.

Copyright © 2007 IEEE. All rights reserved.

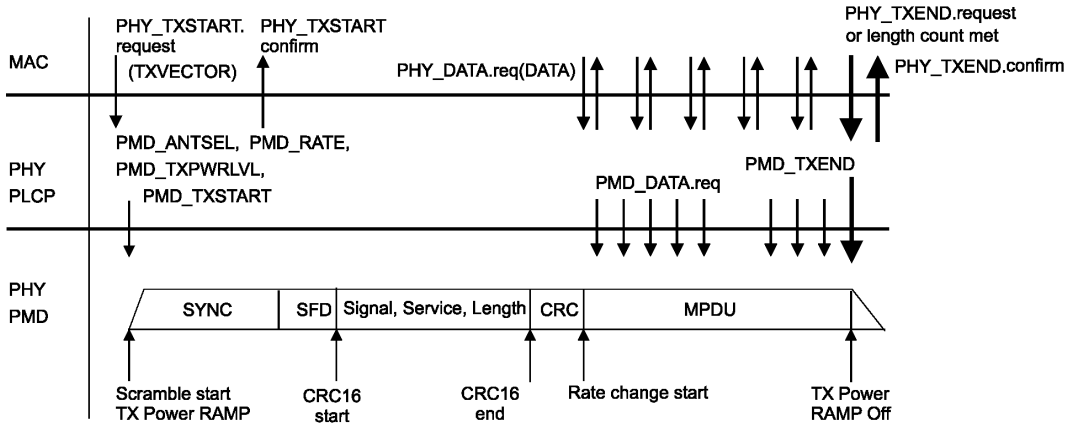



**Figure 15-6—Transmit PLCP**

Based on the status of CCA indicated by PHY-CCA.indicate, the MAC will assess that the channel is clear. A clear channel shall be indicated by PHY-CCA.indicate(IDLE). If the channel is clear, transmission of the PPDU shall be initiated by issuing the PHY-TXSTART.request(TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request are the PLCP header parameters SIGNAL (DATARATE), SERVICE, and LENGTH, and the PMD parameters of TX_ANTENNA and TXPWR_LEVEL. The PLCP header parameter LENGTH is calculated from the TXVECTOR element by multiplying by 8 for 1 Mb/s and by 4 for 2 Mb/s.

The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for transmit power-on ramp described in 15.4.7.7 shall be included in the PLCP SYNC field. Once the PLCP preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation rate change, if any, shall be initiated with the first data symbol of the MPDU as described in 15.2.5. The PHY proceeds with MPDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in LSB-to-MSB order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request. Normal termination occurs after the transmission of the final bit of the last MPDU octet according to the number supplied in the TXVECTOR LENGTH field. The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). It is recommended that chipping continue during power-down. Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

A typical state machine implementation of the transmit PLCP is provided in Figure 15-7.

### 15.2.7 Receive PLCP

The receive PLCP is shown in Figure 15-8.


Copyright © 2007 IEEE. All rights reserved.

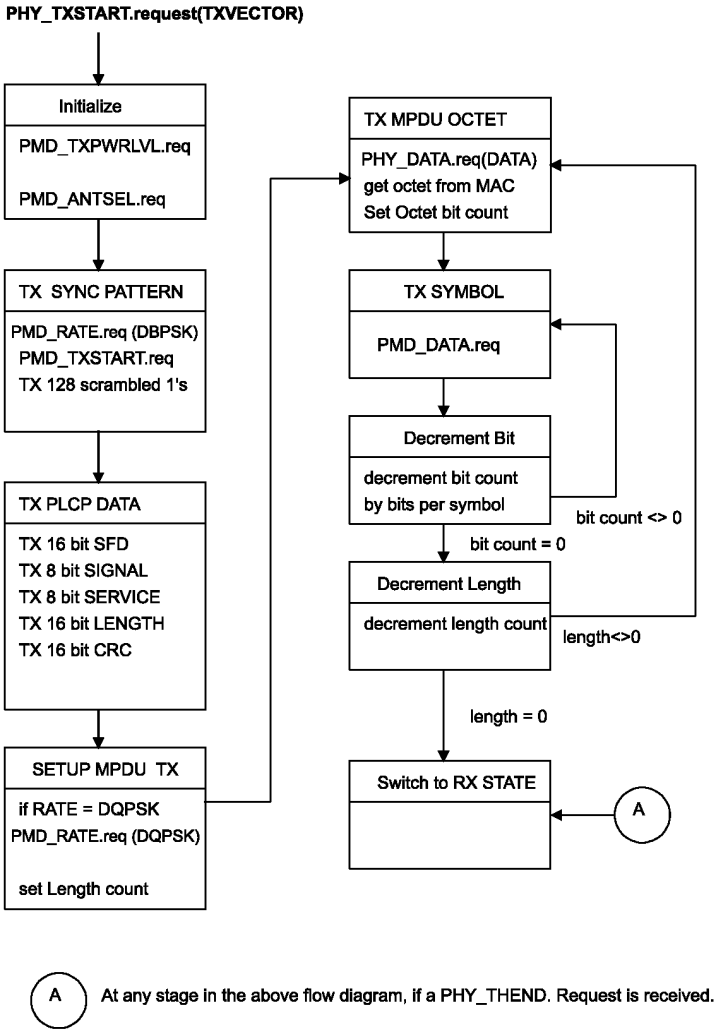



**Figure 15-7—PLCP transmit state machine**

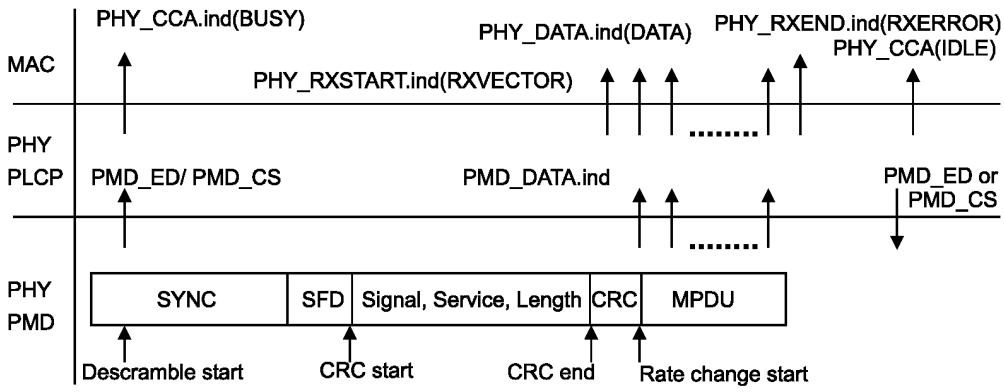



**Figure 15-8—Receive PLCP**

Copyright © 2007 IEEE. All rights reserved.

In order to receive data, PHY-TXSTART.request shall be disabled so that the PHY entity is in the receive state. Further, through STA management via the PLME, the PHY is set to the appropriate channel and the CCA method is chosen. Other receive parameters such as RSSI, signal quality (SQ), and indicated DATARATE may be accessed via the PHY-SAP.

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 15.4.8.4) as the RSSI reaches the ED_THRESHOLD and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via PHY-CCA.indicate according to 15.4.8.4. PHY-CCA.indicate(BUSY) shall be issued for energy detection (ED) and/or code lock prior to correct reception of the PLCP frame. The PMD primitives PMD_SQ and PMD_RSSI are issued to update the RSSI and SQ parameters reported to the MAC.

After PHY-CCA.indicate is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CRC-16 processing shall be initiated and the PLCP IEEE 802.11 SIGNAL, IEEE 802.11 SERVICE and LENGTH fields are received. The CRC-16 FCS shall be processed. If the CRC-16 FCS check fails, the PHY receiver shall return to the RX IDLE state as depicted in Figure 15-9. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indicate(RXVECTOR) shall be issued. The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the MPDU length in octets (calculated from the LENGTH field in microseconds), the antenna used for receive (RX_ANTENNA), RSSI, and SQ.

The received MPDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indicate(DATA) primitive exchanges. The rate change indicated in the IEEE 802.11 SIGNAL field shall be initiated with the first symbol of the MPDU as described in 15.2.5. The PHY proceeds with MPDU reception. After the reception of the final bit of the last MPDU octet indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state as shown in Figure 15-9. A PHY-RXEND.indicate(NoError) primitive shall be issued. A PHY-CCA.indicate(IDLE) primitive shall be issued following a change in PHY carrier sense (PHYCS) and/or PHY energy detection (PHYED) according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the MPDU as indicated by the PLCP LENGTH field, the error condition PHY-RXEND.indicate(CarrierLost) shall be reported to the MAC. The DSSS PHY will ensure that the CCA will indicate a busy medium for the intended duration of the transmitted packet.

If the PLCP header is successful, but the indicated rate in the SIGNAL field is not receivable, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(UnsupportedRate). If the PLCP header is successful, but the SERVICE field is out of IEEE 802.11 DSSS specification, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(FormatViolation). Also, in both cases, the DSSS PHY will ensure that the CCA shall indicate a busy medium for the intended duration of the transmitted frame as indicated by the LENGTH field. The intended duration is indicated by the LENGTH field (length $\times 1$ μs).

A typical state machine implementation of the receive PLCP is provided in Figure 15-9.


Copyright © 2007 IEEE. All rights reserved.

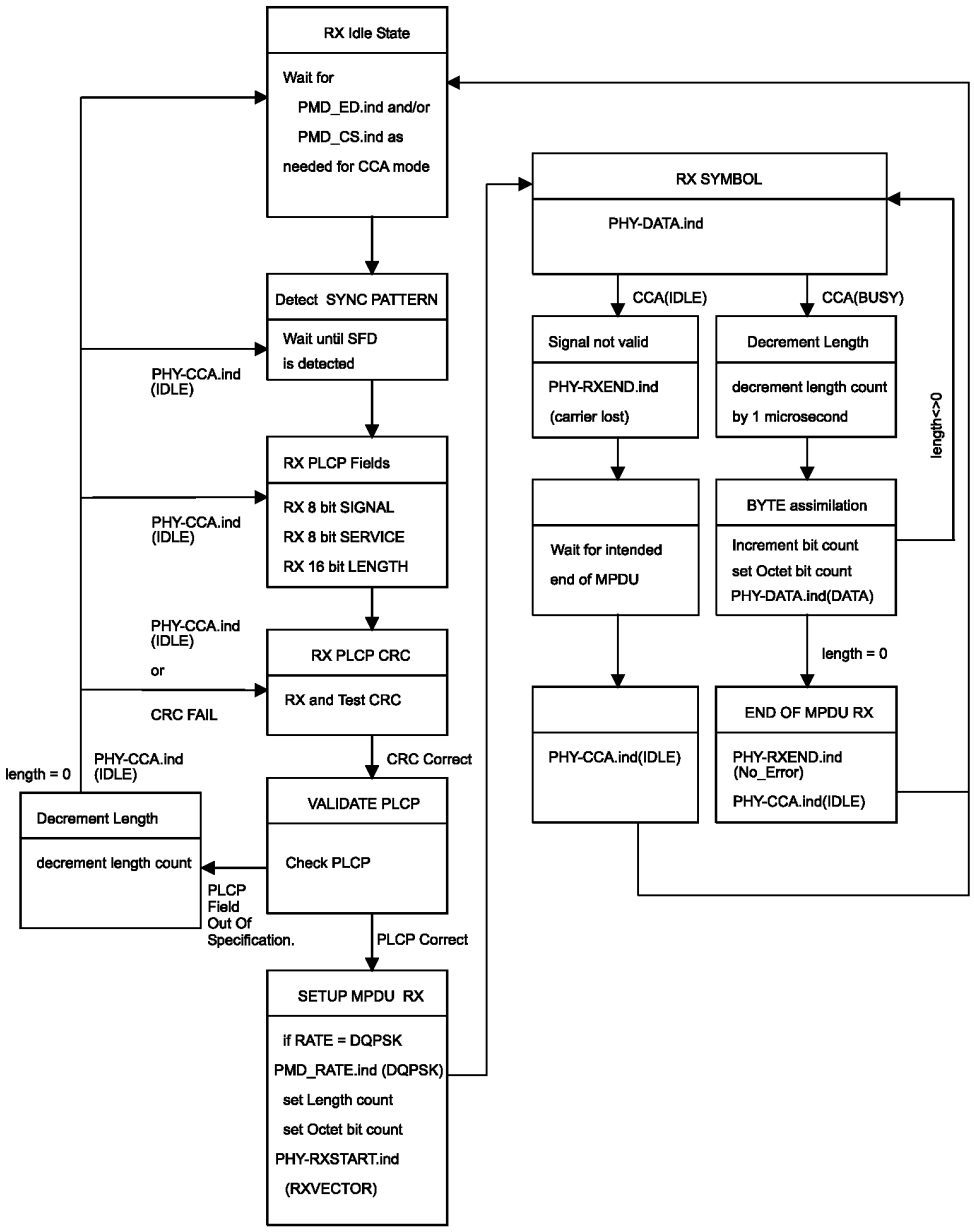



**Figure 15-9—PLCP receive state machine**

## 15.3 DSSS PLME

### 15.3.1 PLME_SAP sublayer management primitives

Table 15-1 lists the MIB attributes that may be accessed by the PHY entities and intralayer of higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 10.

Copyright © 2007 IEEE. All rights reserved.

### 15.3.2 DSSS PHY MIB

All DSSS PHY MIB attributes are defined in Clause 12, with specific values defined in Table 15-1.

**Table 15-1—MIB attribute default values/ranges**

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11PhyOperationComplianceGroup** | | |
| dot11PHYdot11TempType | DSSS–2.4 (02) | Static |
| dot11TempType | Implementation dependent | Static |
| dot11RegDomainsSupported | Implementation dependent | Static |
| dot11CurrentRegDomain | Implementation dependent | Static |
| **dot11PhyRateGroup** | | |
| dot11SupportedDataRatesTx | X'02', X'04' | Static |
| dot11SupportedDataRatesRx | X'02', X'04' | Static |
| **dot11PhyAntennaComplianceGroup** | | |
| dot11CurrentTxAntenna | Implementation dependent | Dynamic |
| dot11DiversitySupport | Implementation dependent | Static |
| dot11CurrentRxAntenna | Implementation dependent | Dynamic |
| **dot11PhyTxPowerComplianceGroup** | | |
| dot11NumberSupportedPowerLevels | Implementation dependent | Static |
| dot11TxPowerLevel1 | Implementation dependent | Static |
| dot11TxPowerLevel2 | Implementation dependent | Static |
| dot11TxPowerLevel3 | Implementation dependent | Static |
| dot11TxPowerLevel4 | Implementation dependent | Static |
| dot11TxPowerLevel5 | Implementation dependent | Static |
| dot11TxPowerLevel6 | Implementation dependent | Static |
| dot11TxPowerLevel7 | implementation dependent | Static |
| dot11TxPowerLevel8 | Implementation dependent | Static |
| dot11CurrentTxPowerLevel | Implementation dependent | Dynamic |
| **dot11PhyDSSSComplianceGroup** | | |
| dot11CurrentChannel | Implementation dependent | Dynamic |
| dot11CCAModeSupported | Implementation dependent | Static |
| dot11CurrentCCAMode | Implementation dependent | Dynamic |
| dot11EDThreshold | Implementation dependent | Dynamic |
| **dot11AntennasListGroup** | | |
| dot11SupportedTxAntenna | Implementation dependent | Static |
| dot11SupportedRxAntenna | Implementation dependent | Static |
| dot11DiversitySelectionRx | Implementation dependent | Dynamic |
| NOTE—The column titled “Operational semantics” contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. MIB attributes defined as dynamic can be modified by some management entities. | | |


Copyright © 2007 IEEE. All rights reserved.

### 15.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 15-2. The definitions of these characteristics are in 10.4.3.

**Table 15-2—DS PHY characteristics**

| Characteristic | Value |
|---|---|
| aSlotTime | 20 μs |
| aSIFSTime | 10 μs |
| aCCATime | ≤ 15 μs |
| aPHY-RX-START-Delay | 192 μs |
| aRxTxTurnaroundTime | ≤ 5 μs |
| aTxPLCPDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxPLCPDelay | Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aRxTxSwitchTime | ≤ 5 μs |
| aTxRampOnTime | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aTxRampOffTime | Implementers may choose any value for this delay as long as the requirements of aSIFSTime are met. |
| aTxRFDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxRFDelay | Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aAirPropagationTime | 1 μs |
| aMACProcessingDelay | ≤ 2 μs |
| aPreambleLength | 144 μs |
| aPLCPHeaderLength | 48 μs |
| aMPDUDurationFactor | 0 |
| aMPDUMaxLength | $4 \le x \le (2^{13} - 1)$ |
| aCWmin | 31 |
| aCWmax | 1023 |

## 15.4 DSSS PMD sublayer

### 15.4.1 Scope and field of application

This subclause describes the PMD services provided to the PLCP for the DSSS PHY. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of

Copyright © 2007 IEEE. All rights reserved.

implementations conforming to this standard. The relationship of this standard to the entire DSSS PHY is shown in Figure 15-10.

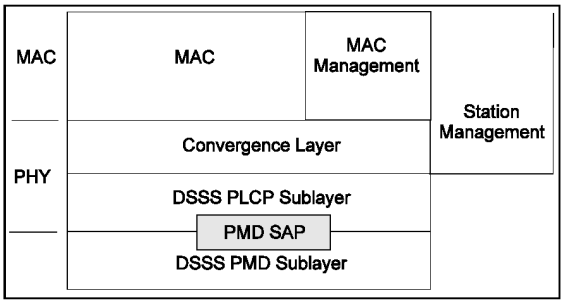



**Figure 15-10—PMD layer reference model**

### 15.4.2 Overview of service

The DSSS PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data shall be transmitted or received from the medium. The combined function of DSSS PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality shall be provided for data transmission.

### 15.4.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the DSSS PMD fall into two basic categories:

a) Service primitives that support PLCP peer-to-peer interactions, and

b) Service primitives that have local significance and that support sublayer-to-sublayer interactions.

### 15.4.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory unless otherwise specified.

#### 15.4.4.1 PMD_SAP peer-to-peer service primitives

Table 15-3 indicates the primitives for peer-to-peer interactions.

**Table 15-3—PMD_SAP peer-to-peer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PHY-RXSTART | | X | | — |
| PHY-RXEND | | X | | — |
| PHY-CCA | | X | | — |
| PHY-TXSTART | X | | X | — |
| PHY-TXEND | X | | X | — |
| PHY-DATA | X | X | X | — |


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.4.2 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 15-4 indicates the parameters required by the MAC or DSSS PHY in each of the parameter vectors used for peer-to-peer interactions.

**Table 15-4—DSSS PMD_SAP peer-to-peer service primitives**

| Parameter | Associated primitive | Value |
|---|---|---|
| LENGTH | RXVECTOR, TXVECTOR | 0 to $2^{13} - 1$ |
| DATARATE | RXVECTOR, TXVECTOR | 1, 2 Mb/s |
| SERVICE | RXVECTOR, TXVECTOR | 1, 2 Mb/s |
| TXPWR_LEVEL | TXVECTOR | Level1, Level2, Level3, Level4 |
| TX_ANTENNA | TXVECTOR | 1-256 |
| RSSI | RXVECTOR | 0-255 |
| SQ | RXVECTOR | 0-255 |
| RX_ANTENNA | RXVECTOR | 1-256 |

#### 15.4.4.3 PMD_SAP sublayer-to-sublayer service primitives

Table 15-5 indicates the primitives for sublayer-to-sublayer interactions.

**Table 15-5—PMD_SAP sublayer-to-sublayer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_TXSTART | X | | — | — |
| PMD_TXEND | X | | — | — |
| PMD_ANTSEL | X | X | — | — |
| PMD_TXPWRLVL | X | | — | — |
| PMD_RATE | X | X | — | — |
| PMD_RSSI | | X | — | — |
| PMD_SQ | | X | — | — |
| PMD_CS | | X | — | — |
| PMD_ED | X | X | — | — |

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.4.4 PMD_SAP service primitive parameters

Table 15-6 indicates the parameters for the PMD primitives.

**Table 15-6—List of parameters for the PMD primitives**

| Parameter | Associate primitive | Value |
|---|---|---|
| DATA | PHY-DATA.request<br>PHY-DATA.indicate | Octet value: X'00'–X'FF' |
| TXVECTOR | PHY-DATA.request | A set of parameters |
| RXVECTOR | PHY-DATA.indicate | A set of parameters |
| TXD_UNIT | PMD_DATA.request | 1, 0: DBPSK<br>dibit combinations<br>00,01,11,10: DQPSK |
| RXD_UNIT | PMD_DATA.indicate | 1, 0: DBPSK<br>dibit combinations<br>00,01,11,10: DQPSK |
| RF_STATE | PMD_TXE.request | Receive, Transmit |
| ANT_STATE | PMD_ANTSEL.indicate<br>PMD_ANTSEL.request | 1 to 256 |
| TXPWR_LEVEL | PHY-TXSTART | 0, 1, 2, 3 (max of 4 levels) |
| RATE | PMD_RATE.indicate<br>PMD_RATE.request | X'0A' for 1 Mb/s DBPSK<br>X'14' for 2 Mb/s DQPSK |
| RSSI | PMD_RSSI.indicate | 0–8 bits of RSSI |
| SQ | PMD_SQ.indicate | 0–8 bits of SQ |


Copyright © 2007 IEEE. All rights reserved.

### 15.4.5 PMD_SAP detailed service specification

The services provided by each PMD primitive are described in 15.4.5.1 through 15.4.5.15.

#### 15.4.5.1 PMD_DATA.request

##### 15.4.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

##### 15.4.5.1.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.request(TXD_UNIT)

The TXD_UNIT parameter takes on the value of either 1 or 0 for DBPSK modulation or the dibit combination 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single block of data, which, in turn, shall be used by the PHY to be differentially encoded into a DBPSK or DQPSK transmitted symbol. The symbol itself shall be spread by the PN code prior to transmission.

##### 15.4.5.1.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

##### 15.4.5.1.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation, and transmission of the data.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.2 PMD_DATA.indicate

##### 15.4.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

##### 15.4.5.2.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_DATA.indicate(RXD_UNIT)

The RXD_UNIT parameter takes on the value of 1 or 0 for DBPSK modulation or as the dibit 00, 01, 11, or 10 for DQPSK modulation. This parameter represents a single symbol that has been demodulated by the PMD entity.

##### 15.4.5.2.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive shall be supplied by the PMD layer based on the PN code repetition.

##### 15.4.5.2.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the MPDU.


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.3 PMD_TXSTART.request

##### 15.4.5.3.1 Function

This primitive, which is generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

##### 15.4.5.3.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXSTART.request

##### 15.4.5.3.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

##### 15.4.5.3.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.4 PMD_TXEND.request

##### 15.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

##### 15.4.5.4.2 Semantics of the service primitive

The semantics of the primitive are as follows:
PMD_TXEND.request

##### 15.4.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

##### 15.4.5.4.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.


Copyright © 2007 IEEE. All rights reserved.

### 15.4.5.5 PMD_ANTSEL.request

#### 15.4.5.5.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

#### 15.4.5.5.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_ANTSEL.request(ANT_STATE)

ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas shall be determined from the MIB table parameters aSuprtRxAntennas and aSuprtTxAntennas.

#### 15.4.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific antenna for transmission or reception (when diversity is disabled).

#### 15.4.5.5.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.6 PMD_ANTSEL.indicate

##### 15.4.5.6.1 Function

This primitive, which is generated by the PHY PLCP sublayer, reports the antenna used by the PHY for reception of the most recent packet.

##### 15.4.5.6.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_ANTSEL.indicate(ANT_STATE)

ANT_STATE reports which of the available antennas was used for reception of the most recent packet.

##### 15.4.5.6.3 When generated

This primitive shall be generated by the PLCP sublayer to report the antenna used for the most recent packet reception.

##### 15.4.5.6.4 Effect of receipt

PMD_ANTSEL immediately reports the antenna specified by ANT_STATE.


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.7 PMD_TXPWRLVL.request

##### 15.4.5.7.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

##### 15.4.5.7.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_TXPWRLVL.request(TXPWR_LEVEL)

TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter dot11NumberSupportedPowerLevels. See 15.4.7.3 for further information on the optional DSSS PHY power-level-control capabilities.

##### 15.4.5.7.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART to the transmit state.

##### 15.4.5.7.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.8 PMD_RATE.request

##### 15.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation rate that shall be used by the DSSS PHY for transmission.

##### 15.4.5.8.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_RATE.request(RATE)

The RATE parameter selects which of the DSSS PHY data rates shall be used for MPDU transmission. Subclause 15.4.6.4 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 15.2.

##### 15.4.5.8.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY modulation rate used for the MPDU portion of a PPDU.

##### 15.4.5.8.4 Effect of receipt

The receipt of PMD_RATE selects the rate that shall be used for all subsequent MPDU transmissions. This rate shall be used for transmission only. The DSSS PHY shall still be capable of receiving all the required DSSS PHY modulation rates.


Copyright © 2007 IEEE. All rights reserved.

### 15.4.5.9 PMD_RATE.indicate

#### 15.4.5.9.1 Function

This primitive, which is generated by the PMD sublayer, indicates which modulation rate was used to receive the MPDU portion of the PPDU. The modulation shall be indicated in the PSF.

#### 15.4.5.9.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_RATE.indicate(RATE)

In receive mode, the RATE parameter informs the PLCP layer which of the DSSS PHY data rates was used to process the MPDU portion of the PPDU. Subclause 15.4.6.4 provides further information on the DSSS PHY modulation rates. The DSSS PHY rate change capability is fully described in 15.2.

#### 15.4.5.9.3 When generated

This primitive shall be generated by the PMD sublayer when the PSF has been properly detected.

#### 15.4.5.9.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.10 PMD_RSSI.indicate

##### 15.4.5.10.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the receive signal strength.

##### 15.4.5.10.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_RSSI.indicate(RSSI)

The RSSI shall be a measure of the RF energy received by the DSSS PHY. RSSI indications of up to 8 bits (256 levels) are supported.

##### 15.4.5.10.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

##### 15.4.5.10.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used in conjunction with SQ as part of a CCA scheme.


Copyright © 2007 IEEE. All rights reserved.

### 15.4.5.11 PMD_SQ.indicate

#### 15.4.5.11.1 Function

This optional primitive, which is generated by the PMD sublayer, provides to the PLCP and MAC entity the SQ of the DSSS PHY PN code correlation. The SQ shall be sampled when the DSSS PHY achieves code lock and shall be held until the next code lock acquisition.

#### 15.4.5.11.2 Semantics of the service primitive

The primitive shall provide the following parameter:
PMD_SQ.indicate(SQ)

The SQ shall be a measure of the PN code correlation quality received by the DSSS PHY. SQ indications of up to 8 bits (256 levels) are supported.

#### 15.4.5.11.3 When generated

This primitive shall be generated by the PMD when the DSSS PHY is in the receive state and code lock is achieved. It shall be continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

#### 15.4.5.11.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.12 PMD_CS.indicate

##### 15.4.5.12.1 Function

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the PN code and data are being demodulated.

##### 15.4.5.12.2 Semantics of the service primitive

The PMD_CS (carrier sense) primitive in conjunction with PMD_ED provides CCA status through the PLCP layer PHY-CCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS shall be ENABLED when the correlator SQ indicated in PMD_SQ is greater than the CS_THRESHOLD parameter. PMD_CS shall be DISABLED when the PMD_SQ falls below the correlation threshold.

##### 15.4.5.12.3 When generated

This primitive shall be generated by the PHY when the DSSS PHY is receiving a PPDU and the PN code has been acquired.

##### 15.4.5.12.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a DSSS PHY signal. The DSSS PHY should not be placed into the transmit state when PMD_CS is ENABLED.


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.13 PMD_ED.indicate

##### 15.4.5.13.1 Function

This optional primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

##### 15.4.5.13.2 Semantics of the service primitive

The PMD_ED (energy detect) primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED shall be ENABLED when the RSSI indicated in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED shall be DISABLED when the PMD_RSSI falls below the energy detect threshold.

##### 15.4.5.13.3 When generated

This primitive shall be generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

##### 15.4.5.13.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.14 PMD_ED.request

##### 15.4.5.14.1 Function

This optional primitive, which is generated by the PHY PLCP, sets the energy detect ED_THRESHOLD value.

##### 15.4.5.14.2 Semantics of the service primitive

The primitive shall provide the following parameters:
PMD_ED.request(ED_THRESHOLD)

ED_THRESHOLD is the value that the RSSI indicated shall exceed for PMD_ED to be enabled.

##### 15.4.5.14.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current DSSS PHY energy detect threshold.

##### 15.4.5.14.4 Effect of receipt

The receipt of PMD_ED immediately changes the ED threshold as set by the ED_THRESHOLD parameter.


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.5.15 PHY-CCA.indicate

##### 15.4.5.15.1 Function

This primitive, which is generated by the PMD, indicates to the PLCP layer that the receiver has detected RF energy that adheres to the CCA algorithm.

##### 15.4.5.15.2 Semantics of the service primitive

The PHY-CCA primitive provides CCA status at the PLCP layer to the MAC.

##### 15.4.5.15.3 When generated

This primitive shall be generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter (PMD_ED is active), and optionally is a valid correlated DSSS PHY signal whereby PMD_CS would also be active.

##### 15.4.5.15.4 Effect of receipt

This indicator shall be provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter indicates that the RF medium may be busy with an RF energy source that may or may not be DSSS PHY compliant. If a DSSS PHY source is being received, the PMD_CS function shall be enabled shortly after the PMD_ED function is enabled.

Copyright © 2007 IEEE. All rights reserved.

### 15.4.6 PMD operating specifications, general

The following subclauses provide general specifications for the DSSS PMD sublayer. These specifications apply to both the Receive and the Transmit functions and general operation of a DSSS PHY.

#### 15.4.6.1 Operating frequency range

The DSSS PHY shall operate in the frequency range of 2.4 GHz to 2.4835 GHz as allocated by regulatory bodies in the China, United States and Europe or in the 2.471 GHz to 2.497 GHz frequency band as allocated by regulatory authority in Japan.

#### 15.4.6.2 Number of operating channels

The channel center frequencies and CHNL_ID numbers shall be as shown in Table 15-7. See the applicable regulations for the countries in which the implementation will operate. For each supported regulatory domain, all channels in Table 15-7 marked with "X" shall be supported .

**Table 15-7—DSSS PHY frequency channel plan**

| CHNL_ID | Frequency | Regulatory domains | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | X'10' FCC | X'20' IC | X'30' ETSI | X'31' Spain | X'32' France | X'40' Japan | X'50' China |
| 1 | 2412 MHz | X | X | X | — | — | — | X |
| 2 | 2417 MHz | X | X | X | — | — | — | X |
| 3 | 2422 MHz | X | X | X | — | — | — | X |
| 4 | 2427 MHz | X | X | X | — | — | — | X |
| 5 | 2432 MHz | X | X | X | — | — | — | X |
| 6 | 2437 MHz | X | X | X | — | — | — | X |
| 7 | 2442 MHz | X | X | X | — | — | — | X |
| 8 | 2447 MHz | X | X | X | — | — | — | X |
| 9 | 2452 MHz | X | X | X | — | — | — | X |
| 10 | 2457 MHz | X | X | X | X | X | — | X |
| 11 | 2462 MHz | X | X | X | X | X | — | X |
| 12 | 2467 MHz | — | — | X | — | X | — | X |
| 13 | 2472 MHz | — | — | X | — | X | — | X |
| 14 | 2484 MHz | — | — | — | — | — | X | — |

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 30 MHz. Channel 14 shall be designated specifically for operation in Japan.


Copyright © 2007 IEEE. All rights reserved.

#### 15.4.6.3 Spreading sequence

The following 11-chip Barker sequence shall be used as the PN code sequence:

+1, –1, +1, +1, –1, +1, +1, +1, –1, –1, –1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

#### 15.4.6.4 Modulation and channel data rates

Two modulation formats and data rates are specified for the DSSS PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate shall be based on 1 Mb/s DBPSK modulation. The DBPSK encoder is specified in Table 15-8. The enhanced access rate shall be based on 2 Mb/s DQPSK. The DQPSK encoder is specified in Table 15-9. (In the tables, $+j\omega$ shall be defined as counterclockwise rotation.)

**Table 15-8—1 Mb/s DBPSK encoding table**

| Bit input | Phase change ($+j\omega$) |
|---|---|
| 0 | 0 |
| 1 | $\pi$ |

**Table 15-9—2 Mb/s DQPSK encoding table**

| Dibit pattern (d0,d1)<br>d0 is first in time | Phase change ($+j\omega$) |
|---|---|
| 00 | 0 |
| 01 | $\pi/2$ |
| 11 | $\pi$ |
| 10 | $3\pi/2$ ($-\pi/2$) |

#### 15.4.6.5 Transmit and receive in-band and out-of-band spurious emissions

The DSSS PHY shall conform with in-band and out-of-band spurious emissions as set by the appropriate regulatory bodies.

#### 15.4.6.6 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than 10 μs, including the power-down ramp specified in 15.4.7.7.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to valid CCA detection of the incoming signal. The CCA should occur within 25 μs (10 μs for turnaround time plus 15 μs for energy detect) or by the next slot boundary occurring after 25 μs has elapsed (refer to 15.4.8.4). A receiver input signal 3 dB above the ED threshold described in 15.4.8.4 shall be present at the receiver.

Copyright © 2007 IEEE. All rights reserved.

#### 15.4.6.7 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface, using PHYTXSTART.request and shall be ≤ 5 μs. This includes the transmit power-on ramp described in 15.4.7.7.

#### 15.4.6.8 Slot time

The slot time for the DSSS PHY shall be the sum of the RX-to-TX turnaround time (5 μs) and the energy detect time (15 μs specified in 15.4.8.4). The propagation delay shall be regarded as being included in the energy detect time.

#### 15.4.6.9 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

#### 15.4.6.10 Transmit and receive operating temperature range

Three temperature ranges for full operation compliance to the DSSS PHY are specified in Clause 13. Type 1 shall be defined as 0 °C to 40 °C, and is designated for office environments. Type 2 shall be defined as –20 °C to +50 °C, and Type 3 shall be defined as –30 °C to +70 °C. Types 2 and 3 are designated for industrial environments.

### 15.4.7 PMD transmit specifications

The transmit functions and parameters associated with the PMD sublayer are described in 15.4.7.1 through 15.4.7.9.

#### 15.4.7.1 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.

#### 15.4.7.2 Minimum transmitted power level

The minimum transmitted power shall be no less than 1 mW.

#### 15.4.7.3 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. At a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

#### 15.4.7.4 Transmit spectrum mask

The transmitted spectral products shall be less than –30 dBr (decibel relative to the SINx/x peak) for $f_c$ – 22 MHz < $f$ < $f_c$ –11 MHz, $f_c$ +11 MHz < $f$ < $f_c$ + 22 MHz, –50 dBr for $f$ < $f_c$ –22 MHz, and $f$ > $f_c$ + 22 MHz, where $f_c$ is the channel center frequency. The transmit spectral mask is shown in Figure 15-11. The measurements shall be made using 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

Channel 14 is unique. The Japanese standard ARIB RCR-STD 33 (5.0) [B3] states that B90/2pi normalized to the 'transmission speed of modulation signal' shall be > 10. Therefore, for channel 14, B90/2pi > 13.75 MHz for CCK spreading and >10.0 MHz for Barker spreading.


Copyright © 2007 IEEE. All rights reserved.

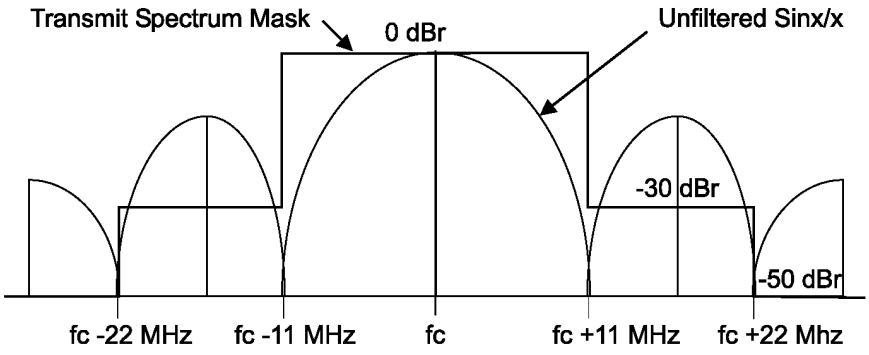



**Figure 15-11—Transmit spectrum mask**

### 15.4.7.5 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

### 15.4.7.6 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ±25 ppm maximum.

### 15.4.7.7 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 µs. The transmit power-on ramp is shown in Figure 15-12.

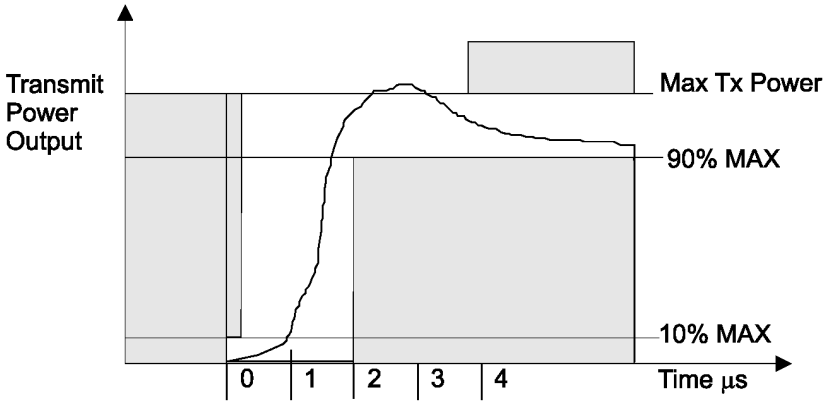



**Figure 15-12—Transmit power-on ramp**

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 µs. The transmit power-down ramp is shown in Figure 15-13.

The transmit power ramps shall be constructed such that the DSSS PHY emissions conform with the spurious frequency product specification defined in 15.4.6.5.

Copyright © 2007 IEEE. All rights reserved.

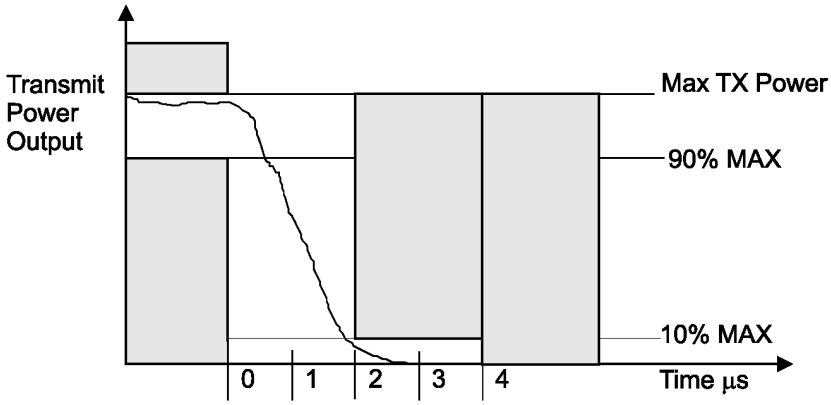



**Figure 15-13—Transmit power-down ramp**

### 15.4.7.8 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak SIN(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

### 15.4.7.9 Transmit modulation accuracy

The transmit modulation accuracy requirement for the DSSS PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude measured during each chip period. Worst-case vector error magnitude shall not exceed 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation (0.707, 0.707), (0.707, –0.707), (–0.707, 0.707), (–0.707, –0.707) shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 15-14 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

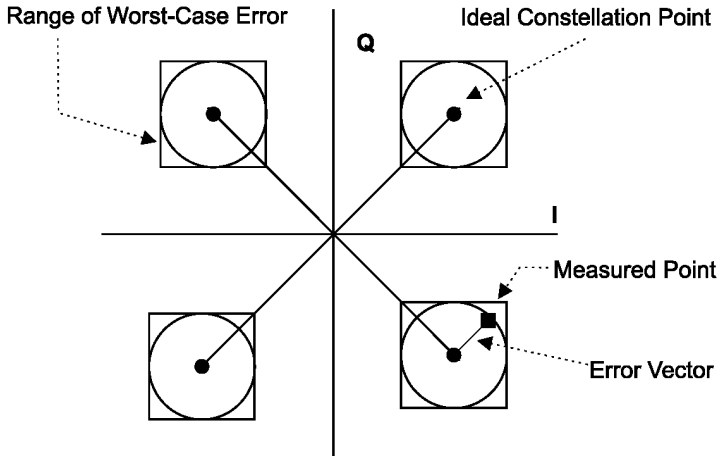



**Figure 15-14—Modulation accuracy measurement example**


Copyright © 2007 IEEE. All rights reserved.

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11 vendor compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all ones.

The eye pattern of the I channel shall be used to determine the *I* and *Q* sampling point. The chip clock provided by the vendor radio shall be time delayed such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 15-15). This is the ideal center of the eye and may not be the point of maximum eye opening.

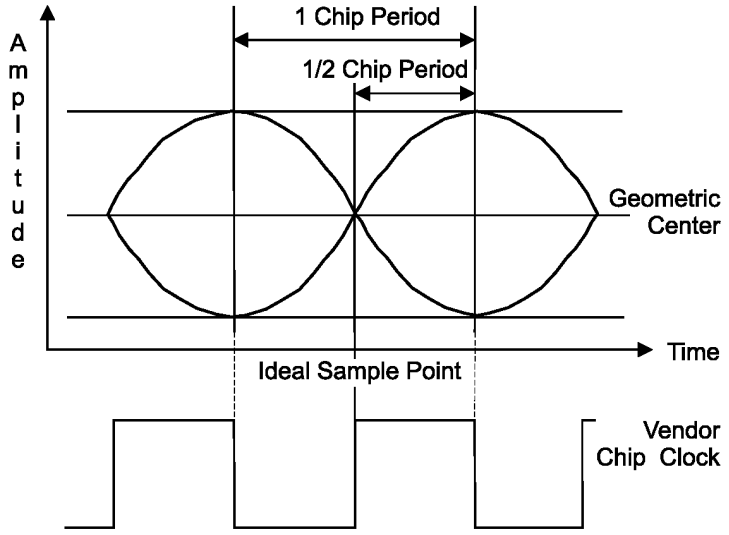



**Figure 15-15—Chip clock alignment with baseband eye pattern**

Using the aligned chip clock, 1000 samples of the *I* and *Q* baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for *I* and *Q* samples.

$$I_{\text{mean}} = \sum_{n=1}^{1000} |I(n)| / 1000$$

$$Q_{\text{mean}} = \sum_{n=1}^{1000} |Q(n)| / 1000$$

Calculate the dc corrected *I* and *Q* samples for all $n$ =1000 sample pairs.

$$I_{\text{dc}}(n) = I(n) - I_{\text{mean}}$$

Copyright © 2007 IEEE. All rights reserved.

$$Q_{dc}(n) = Q(n) - Q_{mean}$$

Calculate the average magnitude of $I$ and $Q$ samples.

$$I_{mag} = \sum_{n=1}^{1000} |I_{dc}(n)| / 1000$$

$$Q_{mag} = \sum_{n=1}^{1000} |Q_{dc}(n)| / 1000$$

Calculate the normalized error vector magnitude for the $I_{dc}(n)/Q_{dc}(n)$ pairs.

$$V_{err}(n) = [\{|I_{dc}(n)| / I_{mag} - 1\}^2 + \{|Q_{dc}(n)| / Q_{mag} - 1\}^2]^{\frac{1}{2}} - V_{correction}$$

with $V_{correction}$ = error induced by the reference receiver system.

A vendor DSSS PHY implementation shall be compliant if for all $n$ =1000 samples the following condition is met:

$$V_{err}(n) < 0.35$$

### 15.4.8 PMD receiver specifications

The receive functions and parameters associated with the PMD sublayer are described in 15.4.8.1 through 15.4.8.4.

#### 15.4.8.1 Receiver minimum input level sensitivity

The FER shall be less than $8\times10^{-2}$ at an MPDU length of 1024 octets for an input level of –80 dBm measured at the antenna connector. This FER shall be specified for 2 Mb/s DQPSK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set ≤ –80 dBm.

#### 15.4.8.2 Receiver maximum input level

The receiver shall provide a maximum FER of $8\times10^{-2}$ at an MPDU length of 1024 octets for a maximum input level of –4 dBm measured at the antenna. This FER shall be specified for 2 Mb/s DQPSK modulation.

#### 15.4.8.3 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥ 30 MHz separation in each channel group defined in 15.4.6.2.

The adjacent channel rejection shall be ≥ 35 dB with an FER of $8\times10^{-2}$ using 2 Mb/s DQPSK modulation described in 15.4.6.4 and an MPDU length of 1024 octets.

The adjacent channel rejection shall be measured using the following method:


Copyright © 2007 IEEE. All rights reserved.

Input a 2 Mb/s DQPSK modulated signal at a level 6 dB greater than specified in 15.4.8.1. In an adjacent channel (≥ 30 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion that adheres to the transmit mask specified in 15.4.7.4 to a level 41 dB above the level specified in 15.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than $8\times10^{-2}$.

#### 15.4.8.4 CCA

The DSSS PHY shall provide the capability to perform CCA according to at least one of the following three methods:

— *CCA Mode 1:* Energy above threshold. CCA shall report a busy medium upon detection of any energy above the ED threshold.

— *CCA Mode 2:* CS only. CCA shall report a busy medium only upon detection of a DSSS signal. This signal may be above or below the ED threshold.

— *CCA Mode 3:* CS with energy above threshold. CCA shall report a busy medium upon detection of a DSSS signal with energy above the ED threshold.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP to indicate activity to the MAC through the PHY interface primitive PHY-CCA.indicate.

A busy channel shall be indicated by PHY-CCA.indicate of class BUSY.

A clear channel shall be indicated by PHY-CCA.indicate of class IDLE.

The PHY MIB attribute dot11CCAModeSupported shall indicate the appropriate operation modes. The PHY shall be configured through the PHY MIB attribute dot11CurrentCCAMode.

The CCA shall be TRUE if there is no energy detect or CS. The CCA parameters are subject to the following criteria:

a) The ED threshold shall be ≤ –80 dBm for TX power > 100 mW, –76 dBm for 50 mW < TX power ≤ 100 mW, and –70 dBm for TX power ≤ 50 mW.

b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 μs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to Figure 9-12 (in 9.2.10) for a definition of slot time boundary.

c) In the event that a correct PLCP header is received, the DSSS PHY shall hold the CCA signal inactive (channel busy) for the full duration as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted packet.

Conformance to DSSS PHY CCA shall be demonstrated by applying a DSSS-compliant signal, above the appropriate ED threshold (item a), so that all conditions described in item b and item c are demonstrated.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 621 of 1232

Copyright © 2007 IEEE. All rights reserved.
**Samsung Ex. 1044, Page 622 of 1232**

# 16. Infrared (IR) PHY specification

This clause is no longer maintained and may not be compatible with all features of this standard.

## 16.1 Overview

The PHY for the infrared (IR) system is specified in this clause. The IR PHY uses near-visible light in the 850 nm to 950 nm range for signaling. This is similar to the spectral usage of both common consumer devices such as IR remote controls, as well as other data communications equipment, such as IR data association (IrDA) devices.

Unlike many other IR devices, however, the IR PHY is not directed. That is, the receiver and transmitter do not have to be aimed at each other and do not need a clear line-of-sight. This permits the construction of a true LAN system, whereas with an aimed system, it would be difficult or impossible to install a LAN because of physical constraints.

A pair of conformant IR devices would be able to communicate in a typical environment at a range up to about 10 m. This standard allows conformant devices to have more sensitive receivers, and this may increase range up to about 20 m.

The IR PHY relies on both reflected IR energy as well as line-of-sight IR energy for communications. Most designs anticipate that *all* of the energy at the receiver is reflected energy. This reliance on reflected IR energy is called *diffuse IR* transmission.

This standard specifies the transmitter and receiver in such a way that a conformant design will operate well in most environments where there is no line-of-sight path from the transmitter to the receiver. However, in an environment that has few or no reflecting surfaces, and where there is no line-of-sight, an IR PHY system may suffer reduced range.

The IR PHY will operate only in indoor environments. IR radiation does not pass through walls, and is significantly attenuated passing through most exterior windows. This characteristic can be used to “contain” an IR PHY in a single physical room, like a classroom or conference room. Different LANs using the IR PHY can operate in adjacent rooms separated only by a wall without interference, and without the possibility of eavesdropping.

At the time of this standard’s preparation, the only known regulatory standards that apply to the use of IR radiation are safety regulations, such as IEC 60825-1:1993 [B10] and ANSI Z136.1-1993 [B1]. While a conformant IR PHY device can be designed to also comply with these safety standards, conformance with this standard does not ensure conformance with other standards.

Worldwide, no frequency allocation or bandwidth allocation regulatory restrictions currently exist on IR emissions.

Emitter (typically LED) and detector (typically PIN diode) devices for IR communications are relatively inexpensive at the IR wavelengths specified in the IR PHY, and at the electrical operating frequencies required by this PHY.

While many other devices in common use also use IR emissions in the same optical band, these devices usually transmit IR intermittently and do not interfere with the proper operation of a compliant IR PHY. If such a device does interfere, by transmitting continuously and with a very strong signal, it can be physically isolated (placing it in a different room) from the IEEE 802.11 LAN.

Copyright © 2007 IEEE. All rights reserved.

### 16.1.1 Scope

The PHY services provided to the IEEE 802.11 WLAN MAC by the IR system are described in this clause. The IR PHY consists of two protocol functions as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, the WM between two or more STAs.

### 16.1.2 IR PHY functions

The IR PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail below.

The IR PHY service is provided to the MAC entity at the STA through a SAP as described in Clause 12. For a visual guide to the relationship of the IR PHY to the remainder of the system, refer to Figure 5-10 (in 5.7).

#### 16.1.2.1 PLCP sublayer

To allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services. The PHY-specific preamble is normally associated with this convergence layer.

#### 16.1.2.2 PMD sublayer

The PMD sublayer provides a CCA mechanism, transmission mechanism, and reception mechanism that are used by the MAC via the PLCP to send or receive data between two or more STAs.

#### 16.1.2.3 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME. See 16.4 for the MIB variables that may be accessed by the PHY entities and intralayer of higher level LMEs. These variables are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 10.

### 16.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended as illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 IR-PHY-compliant developer. Conformance to this standard is not dependent on following the model, and an implementation that follows the model closely may not be conformant.

Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation. In particular, the PHY-SAP operations are defined and described as instantaneous; however, this may be difficult to achieve in an implementation.


Copyright © 2007 IEEE. All rights reserved.

## 16.2 IR PLCP sublayer

While the PLCP sublayer and the PMD sublayer are described separately, the separation and distinction between these sublayers is artificial, and is not meant to imply that the implementation must separate these functions. This distinction is made primarily to provide a point of reference from which to describe certain functional components and aspects of the PMD. The functions of the PLCP can be subsumed by a PMD sublayer; in this case, the PMD will incorporate the PHY-SAP as its interface, and will not offer a PMD_SAP.

### 16.2.1 Overview

A convergence procedure is provided by which MPDUs are converted to and from PPDUs. During transmission, the MPDU (PSDU) is prepended with a PLCP preamble and PLCP header to create the PPDU. At the receiver, the PLCP preamble is processed and the internal data fields are processed to aid in demodulation and delivery of the MPDU (PSDU).

### 16.2.2 PLCP frame format

Figure 16-1 shows the format for the PPDU including the PLCP preamble, the PLCP header, and the PSDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: data rate (DR), dc level adjustment (DCLA), length (LENGTH), and CRC. Each of these fields is described in detail in 16.2.4.

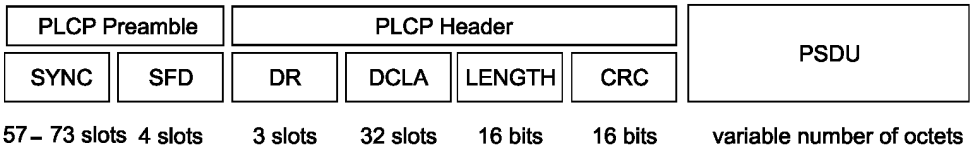



**Figure 16-1—PPDU frame format**

### 16.2.3 PLCP modulation and rate change

The PLCP preamble shall be transmitted using the basic pulse defined in 16.3.3.2. The PSDU, LENGTH, and CRC fields shall be transmitted using pulse position modulation (PPM). PPM maps bits in the octet into symbols: 16-PPM maps four bits into a 16-position symbol, and 4-PPM maps two bits into a 4-position symbol. The basic L-PPM TU is the slot. A slot corresponds to one of the L positions of a symbol and has a 250 ns duration. The PSDU, LENGTH, and CRC fields are transmitted at one of two bit rates: 1 Mb/s or 2 Mb/s. The DR field indicates the data rate that will be used to transmit the PSDU, LENGTH, and CRC fields. The 1 Mb/s data rate uses 16-PPM (basic access rate), and the 2 Mb/s data rate uses 4-PPM (enhanced access rate). The transmitter and receiver will initiate the modulation or demodulation indicated by the DR field starting with the first 4 bits (in 16-PPM) or 2 bits (in 4-PPM) of the LENGTH field. The PSDU transmission rate is set by the DATARATE parameter in the PHY-TXSTART.request primitive. Any conformant IR PHY shall be capable of receiving at 1 Mb/s and 2 Mb/s. Transmission at 2 Mb/s is optional.

A PHY-TXSTART.request that specifies a data rate that is not supported by a PHY instance will cause the PHY to indicate an error to its MAC instance. A PHY is not permitted under any circumstance to transmit at a different rate than the requested rate.

Copyright © 2007 IEEE. All rights reserved.

### 16.2.4 PLCP field definitions

#### 16.2.4.1 PLCP SYNC field

The SYNC field consists of a sequence of alternated presence and absence of a pulse in consecutive slots. The SYNC field has a minimum length of 57 L-PPM slots and a maximum length of 73 L-PPM slots and shall terminate with the absence of a pulse in the last slot. This field is provided so that the receiver can perform clock recovery (slot synchronization), automatic gain control (optional), signal-to-noise ratio estimation (optional), and diversity selection (optional).

The SYNC field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol. See 16.3.2.1 for legal symbols.

#### 16.2.4.2 PLCP SFD field

The SFD field length is four L-PPM slots and consists of the binary sequence 1001, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot. The leftmost bit shall be transmitted first. The SFD field is provided to indicate the start of the PLCP preamble and to perform bit and symbol synchronization.

The SFD field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

#### 16.2.4.3 PLCP DR field

The DR field indicates to the PHY the data rate that shall be used for the transmission or reception of the PSDU, LENGTH, and CRC fields. The transmitted value shall be provided by the PHY-TXSTART.request primitive as described in Clause 12. The DR field has a length of three L-PPM slots. The leftmost bit, as shown below, shall be transmitted first. The IR PHY currently supports two data rates defined by the slot pattern shown for the three L-PPM slots following the SFD, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mb/s: 000
2 Mb/s: 001

The DR field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

#### 16.2.4.4 PLCP DCLA field

The DCLA field is required to allow the receiver to stabilize the dc level after the SYNC, SFD, and DR fields. The leftmost bit, as shown below, shall be transmitted first. The length of the DCLA field is 32 L-PPM slots and consists of the contents shown, where 1 indicates a pulse in the L-PPM slot and 0 indicates no pulse in the L-PPM slot:

1 Mb/s: 00000000100000000000000010000000
2 Mb/s: 00100010001000100010001000100010

The DCLA field is not modulated using L-PPM, but instead consists of transitions in L-PPM slots that would otherwise constitute an illegal symbol.

#### 16.2.4.5 PLCP LENGTH field

The LENGTH field is an unsigned 16-bit integer that indicates the number of octets to be transmitted in the PSDU. The transmitted value shall be provided by the PHYTXSTART.request primitive as described in


Copyright © 2007 IEEE. All rights reserved.

Clause 12. The LSB shall be transmitted first. This field is modulated and sent in L-PPM format. This field is protected by the CRC described in 16.2.4.6.

#### 16.2.4.6 PLCP CRC field

The LENGTH field shall be protected by a 16-bit CRC. The CRC is the ones complement of the remainder generated by the modulo 2 division of the LENGTH field by the polynomial:

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits will be processed in transmit order. The MSB of the 16-bit CRC shall be transmitted first. This field shall be modulated and sent in L-PPM format. All CRC calculations shall be made prior to L-PPM encoding on transmission and after L-PPM decoding on reception.

#### 16.2.4.7 PSDU field

This field is composed of a variable number of octets. The minimum is 0 (zero) and the maximum is 2500. The LSB of each octet shall be transmitted first. All the octets of this field shall be modulated and sent in L-PPM format.

### 16.2.5 PLCPs

#### 16.2.5.1 Transmit PLCP

All commands issued by the MAC require that a confirmation primitive be issued by the PHY. The confirmation primitives provide flow control between the MAC and the PHY.

The transmit PLCP is as follows:

a) Based on the status of CCA, the MAC shall determine whether the channel is clear.

b) If the channel is clear, transmission of the PSDU shall be initiated by a PHY-TXSTART.request with parameters LENGTH and DATARATE.

c) The PHY entity shall immediately initiate transmission of the PLCP preamble and PLCP header based on the LENGTH and DATARATE parameters passed in the PHY-TXSTART.request. Once the PLCP preamble and PLCP header transmission is completed, the PHY entity shall issue a PHY-TXSTART.confirm.

d) Each octet of the PSDU is passed from the MAC to the PHY by a single PHY-DATA.request primitive. Each PHY-DATA.request shall be confirmed by the PHY with a PHY-DATA.confirm before the next request can be made.

e) At the PHY each PSDU octet shall be divided into symbols of 2 bits or 4 bits each. The symbols shall be modulated using L-PPM and transmitted into the medium.

f) Transmission is terminated by the MAC through the primitive PHY-TXEND.request. The PHY shall confirm the resulting end of transmission with a PHY-TXEND.confirm.

#### 16.2.5.2 Receive PLCP

The receive PLCP is as follows:

a) CCA is provided to the MAC via the PHY-CCA.indicate primitive. When the PHY senses activity on the medium, it shall indicate that the medium is busy with a PHY-CCA.indicate with a value of BUSY. This will normally occur during the SYNC field of the PLCP preamble.

b) The PHY entity shall begin searching for the SFD field. Once the SFD field is detected, the PHY entity shall attempt to receive the PLCP header. After receiving the DR and DCLA fields, the PHY

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 627 of 1232

shall initiate processing of the received CRC and LENGTH fields. The data rate indicated in the DR field applies to all symbols in the latter part of the received PSDU, commencing with the first symbol of the LENGTH field. The CRC shall be checked for correctness immediately after its reception.

c) If the CRC check fails, or the value received in the DR field is not one supported by the PHY, then a PHY-RXSTART.indicate shall not be issued to the MAC. When the medium is again free, the PHY shall issue a PHY-CCA.indicate with a value of IDLE.

d) If the PLCP preamble and PLCP header reception is successful, the PHY shall send a PHY-RXSTART.indicate to the MAC; this includes the parameters DATARATE and LENGTH.

   In the absence of errors, the receiving PHY shall report the same length to its local MAC, in the RXVECTOR parameter of the PHY-RXSTART.indicate primitive, that the peer MAC presented to its local PHY entity in the TXVECTOR parameter of its respective PHY-TXSTART.request.

e) The received PSDU L-PPM symbols shall be assembled into octets and presented to the MAC using a series of PHY-DATA.indicate primitives, one per octet.

f) Reception shall be terminated after the reception of the final symbol of the last PSDU octet indicated by the PLCP header's LENGTH field. After the PHY-DATA.indicate for that octet is issued, the PHY shall issue a PHY-RXEND.indicate primitive to its MAC.

g) After issuing the PHY-RXEND.indicate primitive, and when the medium is no longer busy, the PHY shall issue a PHY-CCA.indicate primitive with a value of IDLE.

#### 16.2.5.3 CCA procedure

CCA is provided to the MAC via the PHY-CCA.indicate primitive.

The CCA procedure is as follows:

a) When the PHY senses activity on the medium, a PHY-CCA.indicate primitive with a value of BUSY shall be issued. This will normally occur during reception of the SYNC field of the PLCP preamble.

b) When the PHY senses that the medium is free, a PHY-CCA.indicate primitive with a value of IDLE shall be issued.

c) At any time, the MAC may issue a PHY-CCARESET.request primitive, which will reset the PHY's internal CCA detection mechanism to the medium not-busy (IDLE) state. This primitive will be acknowledged with a PHY-CCARESET.confirm primitive.

#### 16.2.5.4 PMD_SAP peer-to-peer service primitive parameters

Several service primitives include a parameter vector. This vector shall be a list of parameters that may vary depending on PHY type. Table 16-1 indicates the parameters required by the MAC or IR PHY in each of the parameter vectors used for peer-to-peer interactions.

**Table 16-1—IR PMD_SAP peer-to-peer service primitives**

| Parameter | Associated primitive | Value |
|---|---|---|
| LENGTH | RXVECTOR, TXVECTOR | 4 to $2^{16} - 1$ |
| DATARATE | RXVECTOR, TXVECTOR | PHY dependent |


Copyright © 2007 IEEE. All rights reserved.

## 16.3 IR PMD sublayer

The IR PMD sublayer does not define PMD SAPs. The mechanism for communications between the PLCP and PMD sublayers, as well as the distinction between these two sublayers, if any, is left to implementers. In particular, it is possible to design and implement, in a conformant way, a single sublayer that subsumes the functions of both the PLCP and PMD, presenting only the PHY-SAP.

### 16.3.1 Overview

The PMD functional, electrical, and optical characteristics required for interoperability of implementations conforming to this specification are described in this subclause. The relationship of this specification to the entire IR PHY is shown in Figure 5-10 (in 5.7).

### 16.3.2 PMD operating specifications, general

General specifications for the IR PMD sublayer are provided in this subclause. These specifications apply to both the receive and transmit functions and general operation of a compliant IR PHY.

#### 16.3.2.1 Modulation and channel data rates

Two modulation formats and data rates are specified for the IR PHY: a *basic access rate* and an *enhanced access rate*. The basic access rate is based on 1 Mb/s 16-PPM modulation. The 16-PPM encoding is specified in Table 16-2. Each group of 4 data bits is mapped to one of the 16-PPM symbols. The enhanced access rate is based on 2 Mb/s 4-PPM. The 4-PPM encoding is specified in Table 16-3. Each group of 2 data bits is mapped to one of the 4-PPM symbols. Transmission order of the symbol slots is from left to right, as shown in the table, where a one indicates in-band energy in the slot, and a zero indicates the absence of in-band energy in the slot.

The data in Table 16-2 and Table 16-3 have been arranged (Gray coded) so that a single out-of-position-by-one error in the medium, caused, for example, by intersymbol interference, results in only a single bit error in the received data, rather than in a multiple bit error.

**Table 16-2—Sixteen-PPM basic rate mapping**

| Data | 16-PPM symbol |
|---|---|
| 0000 | 0000000000000001 |
| 0001 | 0000000000000010 |
| 0011 | 0000000000000100 |
| 0010 | 0000000000001000 |
| 0110 | 0000000000010000 |
| 0111 | 0000000000100000 |
| 0101 | 0000000001000000 |
| 0100 | 0000000010000000 |
| 1100 | 0000000100000000 |
| 1101 | 0000001000000000 |
| 1111 | 0000010000000000 |

Copyright © 2007 IEEE. All rights reserved.

**Table 16-2—Sixteen-PPM basic rate mapping *(continued)***

| Data | 16-PPM symbol |
|---|---|
| 1110 | 0000100000000000 |
| 1010 | 0001000000000000 |
| 1011 | 0010000000000000 |
| 1001 | 0100000000000000 |
| 1000 | 1000000000000000 |

**Table 16-3—Four-PPM enhanced rate mapping**

| Data | 4-PPM symbol |
|---|---|
| 00 | 0001 |
| 01 | 0010 |
| 11 | 0100 |
| 10 | 1000 |

#### 16.3.2.2 Octet partition and PPM symbol generation procedure

Because PPM is a block modulation method, with the block size less than a full octet, octets have to be partitioned prior to modulation (mapping into PPM symbols).

Octet partition depends on the PPM order being used.

Assume an octet is formed by eight bits numbered 7 6 5 4 3 2 1 0, where bit 0 is the LSB. Partition the octet as follows:

For 16-PPM, create two PPM symbols:

— The symbol using bits 3 2 1 0 shall be transmitted onto the medium first.

— The symbol using bits 7 6 5 4 shall be transmitted onto the medium last.

For 4-PPM, create four PPM symbols:

— The symbol using bits 1 0 shall be transmitted onto the medium first.

— The symbol using bits 3 2 shall be transmitted onto the medium second.

— The symbol using bits 5 4 shall be transmitted onto the medium third.

— The symbol using bits 7 6 shall be transmitted onto the medium last.

#### 16.3.2.3 Operating environment

The IR PHY will operate only in indoor environments. IR PHY interfaces cannot be exposed to direct sunlight. The IR PHY relies on reflected IR energy and does not require a line-of-sight between emitter and receiver in order to work properly. The range and bit error rate of the system may vary with the geometry of the environment and with natural and artificial illumination conditions.

#### 16.3.2.4 Operating temperature range

The temperature range for full operation compliance with the IR PHY is specified as 0 °C to 40 °C.


Copyright © 2007 IEEE. All rights reserved.

### 16.3.3 PMD transmit specifications

The following subclauses describe the transmit functions and parameters associated with the PMD sublayer.

#### 16.3.3.1 Transmitted peak optical power

The peak optical power of an emitted pulse shall be as specified in Table 16-4.

**Table 16-4—Peak optical power as a function of emitter radiation pattern mask**

| Emitter radiation pattern mask | Peak optical power |
|---|---|
| Mask 1 | 2 W ±20% |
| Mask 2 | 0.55 W ±20% |

#### 16.3.3.2 Basic pulse shape and parameters

The basic pulse width, measured between the 50% amplitude points, shall be 250 ±10 ns. The pulse rise time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The pulse fall time, measured between the 10% and 90% amplitude points, shall be no more than 40 ns. The edge jitter, defined as the absolute deviation of the edge from its correct position, shall be no more than 10 ns. The basic pulse shape is shown in Figure 16-2.

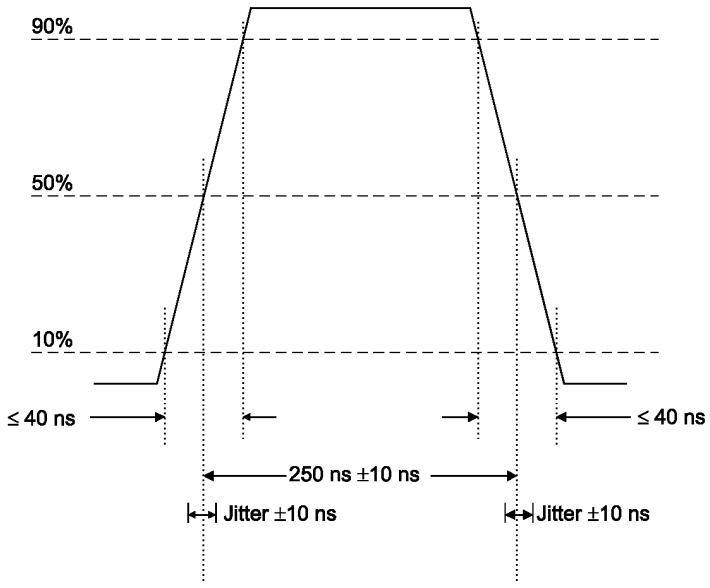



**Figure 16-2—Basic pulse shape**

#### 16.3.3.3 Emitter radiation pattern mask

The standard contains two emitter radiation pattern masks. Mask 1 is defined in Table 16-5 and illustrated in Figure 16-3. Mask 2 is defined in Table 16-6 and illustrated in Figure 16-4.

Copyright © 2007 IEEE. All rights reserved.

**Table 16-5—Definition of the emitter radiation pattern Mask 1**

| Declination angle | Normalized irradiance |
|---|---|
| $\alpha \leq 60°$ | $> 3.5 \times 10^{-6}$ |
| $\alpha \leq 29°$ | $\leq 2.2 \times 10^{-5}$ |
| $29° < \alpha \leq 43°$ | $\leq -1.06 \times 10^{-4} + (0.44 \times 10^{-5})\ \alpha$ |
| $43° < \alpha \leq 57°$ | $\leq 1.15 \times 10^{-4} - (7.1 \times 10^{-7})\ \alpha$ |
| $57° < \alpha \leq 74°$ | $\leq 2.98 \times 10^{-4} - (3.9 \times 10^{-6})\ \alpha$ |
| $74° < \alpha \leq 90°$ | $\leq 4.05 \times 10^{-5} - (4.5 \times 10^{-7})\ \alpha$ |

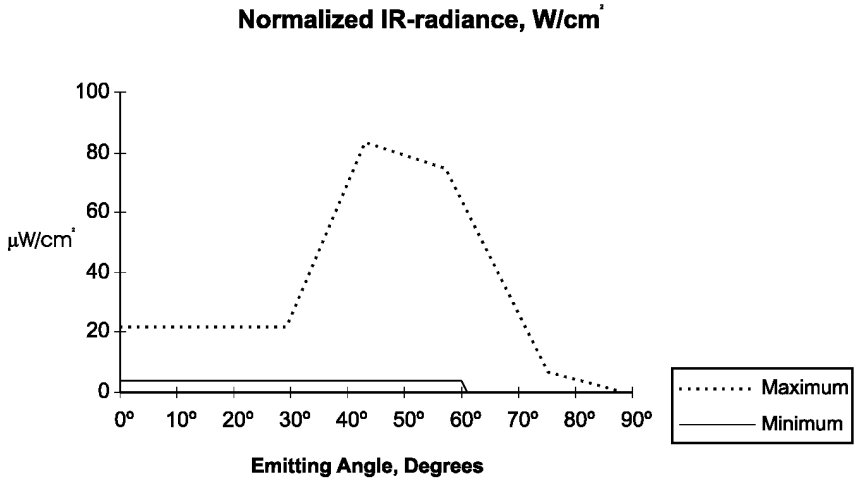



**Figure 16-3—Emitter radiation pattern Mask 1**

Following is a description of how to interpret the Mask 1 table and figure. Position the conformant Mask 1 device in its recommended attitude. Define the conformant Mask 1 device axis as the axis passing through the emitter center and having the direction perpendicular to the floor. The mask represents the irradiance normalized to the total peak emitted power, as a function of the angle between the conformant Mask 1 device axis and the axis from the emitter center to the test receiver center (declination angle). The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The azimuth angle is a rotation angle on the conformant device axis.

A device is conformant if for any azimuth angle its radiation pattern as a function of declination angle falls within the pattern mask.

**Table 16-6—Definition of emitter radiation pattern Mask 2**

| Declination angle | Pitch angle | Normalized irradiance |
|---|---|---|
| $\alpha \leq 60$ | $\alpha = 0$ | $0.05 \pm 15\%$ |
| $\alpha \leq 90$ | $\alpha = 0$ | $0.025 \pm 15\%$ |
| $\alpha \geq 100$ | $\alpha = 0$ | $\leq 0.015$ |
| $0 \leq \alpha \leq 60$ | $0 \leq \alpha \leq 10$ | $0.035 \leq I \leq 0.055$ |
| $0 \leq \alpha \leq 60$ | $10 \leq \alpha \leq 20$ | $0.0225 \leq I \leq 0.05$ |
| $0 \leq \alpha \leq 60$ | $\alpha \geq 30$ | $\leq 0.015$ |


Copyright © 2007 IEEE. All rights reserved.

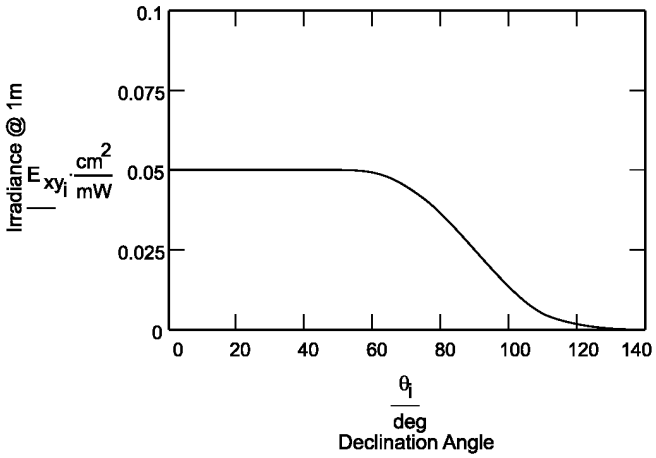



**Figure 16-4—Emitter radiation pattern Mask 2**

Figure 16-5 is a description of how to interpret the Mask 2 table with reference to Figure 16-4.

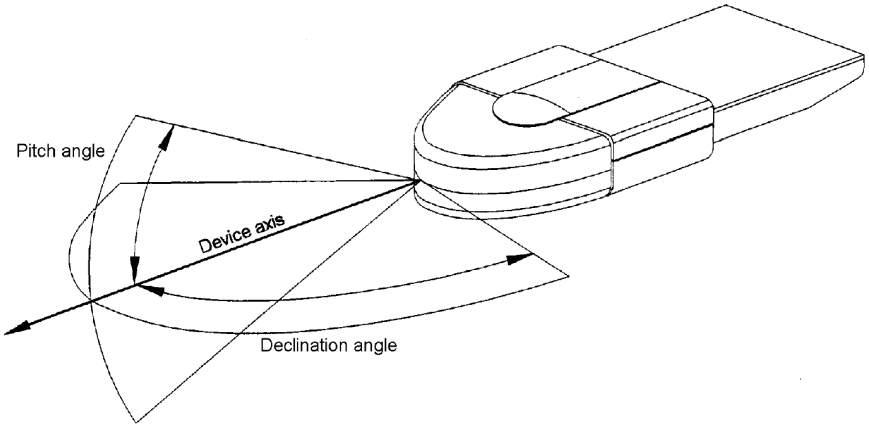



**Figure 16-5—Mask 2 device orientation drawing**

Position the conformant Mask 2 device in its recommended attitude. Define the conformant Mask 2 device axis as passing through the emitter center and having the direction relative to the device as defined by the manufacturer. The declination angle plane is as defined by the manufacturer. The mask represents the irradiance normalized to the peak emitted power on the conformant Mask 2 device axis, as a function of the angle between the conformant device axis and the axis from the emitter center to the test receiver center (declination angle) in the declination plane. The distance between emitter and test receiver is 1 m. The test receiver normal is always aimed at the emitter center. The pitch angle is an angle relative to the conformant device axis which is perpendicular to the declination plane.

The device is conformant if, for a pitch angle of 0 degrees, at any declination angle from 0 to 100 degrees, and if, for any declination angle from 0 to 60 degrees, at any pitch angle from 0 to 20 degrees, its radiation pattern as a function of angle falls within the pattern mask.

Other radiation patterns are for future study.

Copyright © 2007 IEEE. All rights reserved.

#### 16.3.3.4 Optical emitter peak wavelength

The optical emitter peak wavelength shall be between 850 nm and 950 nm.

#### 16.3.3.5 Transmit spectrum mask

Define the transmit spectrum of a transmitter as the Fourier Transform, or equivalent, of a voltage (or current) signal whose amplitude, as a function of time, is proportional to the transmitted optical power.

The transmit spectrum of a conformant transmitter shall be 20 dB below its maximum for all frequencies above 15 MHz. The transmit spectrum mask is shown in Figure 16-6.

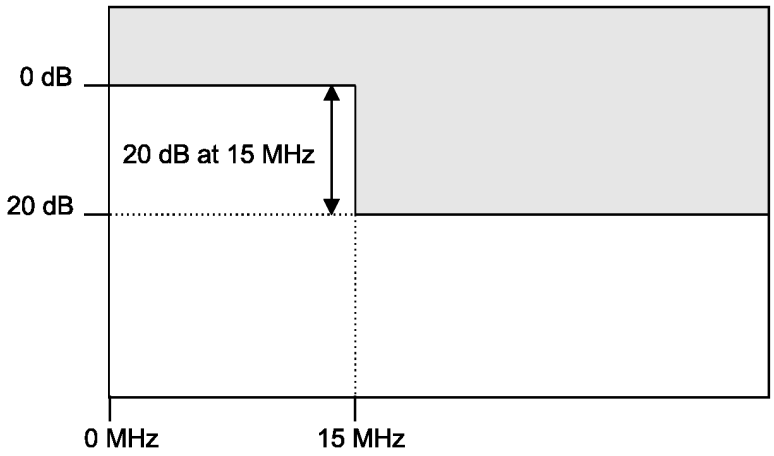



**Figure 16-6—Transmit spectrum mask**

### 16.3.4 PMD receiver specifications

The following subclauses describe the receive functions and parameters associated with the PMD sublayer.

#### 16.3.4.1 Receiver sensitivity

The receiver sensitivity, defined as the minimum irradiance (in $mW/cm^2$) at the photodetector plane required for a FER of $4 \times 10^{-5}$ with a PSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 $mW/cm^2$, shall be

1 Mb/s: $2 \times 10^{-5} mW/cm^2$
2 Mb/s: $8 \times 10^{-5} mW/cm^2$

#### 16.3.4.2 Receiver dynamic range

The receiver dynamic range, defined as the ratio between the maximum and minimum irradiance at the plane normal to the receiver axis that assures an FER lower than or equal to $4 \times 10^{-5}$ with a PSDU of 512 octets and with an unmodulated background IR source between 800 nm and 1000 nm with a level of 0.1 $mW/cm^2$, shall be ≥ 30 dB.

#### 16.3.4.3 Receiver field of view (FOV)

The receiver axis is defined as the direction of incidence of the optical signal at which the received optical power is maximum.


Copyright © 2007 IEEE. All rights reserved.

The received optical power shall be greater than the values given in Table 16-7, at the angles indicated, where "angle of incidence" is the angle of the optical signal relative to the receiver axis, and "received power" is the received optical power as a percentage of that measured at the receiver axis.

**Table 16-7—Definition of the receiver FOV**

| Angle of incidence | Received power |
|---|---|
| $\alpha \leq 20°$ | ≥ 65% |
| $\alpha \leq 40°$ | ≥ 55% |
| $\alpha \leq 60°$ | ≥ 35% |
| $\alpha \leq 80°$ | ≥ 10% |

### 16.3.5 ED, CS, and CCA definitions

#### 16.3.5.1 ED signal

The ED signal shall be set true when IR energy variations in the band between 1 MHz and 10 MHz exceed 0.001 mW/cm$^2$.

The ED shall operate independently of the CS. The ED shall not be asserted at the minimum signal level specified in 16.3.4.1, which is below the level specified in this subclause.

This signal is not directly available to the MAC.

#### 16.3.5.2 CS signal

The CS shall be asserted by the PHY when it detects and locks onto an incoming PLCP preamble signal. Conforming PHYs shall assert this condition within the first 12 μs of signal reception, at the minimum signal level equal to the receiver sensitivity specified in 16.3.4.1, with a background IR level as specified in 16.3.4.1.

The CS shall be deasserted by the PHY when the receiving conformant device loses carrier lock.

NOTE—The 12 μs specification is somewhat less than the minimum length of the PLCP SYNC interval, which is 14.25 μs.

The CS shall operate independently of the ED and shall not require a prior ED before the acquisition and assertion of CS. This permits reception of signals at the minimum signal level specified in 16.3.4.1, even though these signals fall below the ED level.

This signal is not directly available to the MAC.

#### 16.3.5.3 CCA

CCA shall be asserted IDLE by the PHY when the CS and the ED are both false, or when ED has been continuously asserted for a period of time defined by the product of dot11CCAWatchdogTimerMax and dot11CCAWatchdogCountMax without CS becoming active. When either CS or ED go true, CCA is indicated as BUSY to the MAC via the primitive PHY-CCA.indicate. CS and ED behavior are defined in 16.3.5.2.

Normally, CCA will be held BUSY throughout the period of the PLCP header. After receiving the last PLCP bit and the first data octet, the PHY shall signal PHY-RXSTART.indicate with the parameters LENGTH and

Copyright © 2007 IEEE. All rights reserved.

RATE. CCA shall be held BUSY until the number of octets specified in the decoded PLCP header are received. At that time the PHY shall signal PHY-RXEND.indicate. The CCA may remain BUSY after the end of data if some form of energy is still being detected. The PHY will signal PHY-CCA.indicate with a value of IDLE only when the CCA goes CLEAR.

The transition of CCA from BUSY to IDLE is indicated to the MAC via the primitive PHY-CCA.indicate.

If CS and ED go false before the PHY signals PHY-RXSTART.indicate, CCA is set to IDLE and *immediately* signaled to the MAC via PHY-CCA.indicate with a value of IDLE. If CS and ED go false after the PHY has signaled PHY-RXSTART.indicate, implying that the PLCP header has been properly decoded, then the PHY shall not signal a change in state of CCA until the proper interval has passed for the number of octets indicated by the received PLCP LENGTH field. At that time, the PHY shall signal PHY-RXEND.indicate with an RXERROR parameter of CarrierLost followed by PHY-CCA.indicate with a value of IDLE.

The transition of CCA from CLEAR to BUSY resets the CCA watchdog timer and CCA watchdog counter. dot11CCAWatchdogTimerMax and dot11CCAWatchdogCountMax are parameters available via MIB entries and can be read and set via the LME.

Rise and fall times of CCA relative to the OR'ing of the CS and ED signals shall be less than 30 ns. CS and ED are both internal signals to the PHY and are not available directly to the MAC, nor are they defined at any exposed interface.

#### 16.3.5.4 CHNL_ID

For the IR PHY, CHNL_ID = X'01' is defined as the baseband modulation method. All other values are not defined.

## 16.4 PHY attributes

PHY attributes have allowed values and default values that are PHY dependent. Table 16-8 and Table 16-9 describe those values, and further specify whether they are permitted to vary from implementation to implementation.

Table 16-8 does not provide the definition of the attributes, but only provides the IR PHY-specific values for the attributes whose definitions are in Clause 13.

**Table 16-8—IR PHY MIB attributes**

| PHY MIB object | Default value | Operational semantics | Operational behavior |
|---|---|---|---|
| dot11CCAWatchdogTimerMax | Implementation dependent | Dynamic | A conformant PHY may set this via the LME |
| dot11CCAWatchdogCountMax | Implementation dependent | Dynamic | A conformant PHY may set this via the LME |
| dot11CCAWatchdogTimerMin | 22 μs | Static | Identical for all conformant PHYs |
| dot11CCAWatchdogCountMin | 1 | Static | Identical for all conformant PHYs |


Copyright © 2007 IEEE. All rights reserved.

**Table 16-8—IR PHY MIB attributes *(continued)***

| PHY MIB object | Default value | Operational semantics | Operational behavior |
|---|---|---|---|
| dot11SupportedDataRatesTx | Implementation dependent | Static | All conformant PHYs must include the value X'02' (1 Mb/s). |
| dot11SupportedDataRatesRx | Implementation dependent | Static | All conformant PHYs must include the values X'02' (1 Mb/s) and X'04' (2 Mb/s). |
| dot11PhyType | 03 | Static | Identical for all conformant PHYs |
| dot11PhyTempType | X'01' | Static | Identical for all conformant PHYs |

The static IR PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 16-9. The definitions of these characteristics are in 10.4.3.

**Table 16-9—IR PHY characteristics**

| Characteristic | Value |
|---|---|
| aSlotTime | 8 μs |
| aSIFSTime | 10 μs |
| aCCATime | 5 μs |
| aPHY-RX-START-Delay | 57 μs |
| aRxTxTurnaroundTime | 0 μs |
| aTxPLCPDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxPLCPDelay | 1 μs |
| aRxTxSwitchTime | 0 μs |
| aTxRampOnTime | 0 μs |
| aTxRampOffTime | 0 μs |
| aTxRFDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxRFDelay | Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aAirPropagationTime | 1 μs |
| aMACProcessingDelay | 2 μs |
| aPreambleLength | 16 μs (1 Mb/s)<br>20 μs (2 Mb/s) |
| aPLCPHeaderLength | 41 μs (1 Mb/s)<br>25 μs (2 Mb/s) |
| aMPDUDurationFactor | 0 |
| aMPDUMaxLength | 2500 |
| aCWmin | 63 |
| aCWmax | 1023 |

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# 17. Orthogonal frequency division multiplexing (OFDM) PHY specification for the 5 GHz band

## 17.1 Introduction

This clause specifies the PHY entity for an orthogonal frequency division multiplexing (OFDM) system. The OFDM system provides a WLAN with data payload communication capabilities of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. The support of transmitting and receiving at data rates of 6, 12, and 24 Mb/s is mandatory. The system uses 52 subcarriers that are modulated using binary or quadrature phase shift keying (BPSK or QPSK) or using 16- or 64-quadrature amplitude modulation (16-QAM or 64-QAM). Forward error correction coding (convolutional coding) is used with a coding rate of 1/2, 2/3, or 3/4.

The OFDM system also provides a "half-clocked" operation using 10 MHz channel spacings with data communications capabilities of 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s. The support of transmitting and receiving at data rates of 3, 6, and 12 Mb/s is mandatory when using 10 MHz channel spacing. The half-clocked operation doubles symbol times and clear channel assessment (CCA) times when using 10 MHz channel spacing. The regulatory requirements and information regarding use of this OFDM system in 4.9 GHz and 5 GHz bands is in Annex I and Annex J.

The OFDM system also provides a "quarter-clocked" operation using 5 MHz channel spacing with data communication capabilities of 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s. The support of transmitting and receiving at data rates of 1.5, 3, and 6 Mb/s is mandatory when using 5 MHz channel spacing. The quarter-clocked operation quadruples symbol times and CCA times when using 5 MHz channel spacing. The regulatory requirements and information regarding use of this OFDM system in the 4.9 GHz band is in Annex I and Annex J.

### 17.1.1 Scope

This subclause describes the PHY services provided to the IEEE 802.11 WLAN MAC by the 5 GHz (bands) OFDM system. The OFDM PHY consists of two protocol functions, as follows:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method of mapping the IEEE 802.11 PSDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system.

b) A PMD system whose function defines the characteristics and method of transmitting and receiving data through a WM between two or more STAs, each using the OFDM system.

### 17.1.2 OFDM PHY functions

The 5 GHz OFDM PHY architecture is depicted in the reference model shown in Figure 5-10 (in 5.7). The OFDM PHY contains three functional entities: the PMD function, the PHY convergence function, and the layer management function. Each of these functions is described in detail in 17.1.2.1 through 17.1.2.4.

The OFDM PHY service is provided to the MAC through the PHY service primitives described in Clause 12.

#### 17.1.2.1 PLCP sublayer

In order to allow the IEEE 802.11 MAC to operate with minimum dependence on the PMD sublayer, a PHY convergence sublayer is defined. This function simplifies the PHY service interface to the IEEE 802.11 MAC services.

Copyright © 2007 IEEE. All rights reserved.

#### 17.1.2.2 PMD sublayer

The PMD sublayer provides a means to send and receive data between two or more STAs. This clause is concerned with the 5 GHz band using OFDM modulation.

#### 17.1.2.3 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

#### 17.1.2.4 Service specification method

The models represented by figures and state diagrams are intended to be illustrations of the functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the IEEE 802.11 OFDM-PHY-compliant developer.

The service of a layer or sublayer is the set of capabilities that it offers to a user in the next higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

## 17.2 OFDM PHY specific service parameter list

### 17.2.1 Introduction

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require medium management state machines running in the MAC sublayer in order to meet certain PMD requirements. These PHY-dependent MAC state machines reside in a sublayer defined as the MLME. In certain PMD implementations, the MLME may need to interact with the PLME as part of the normal PHY-SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY service primitives as TXVECTOR and RXVECTOR. The list of these parameters, and the values they may represent, are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR and RXVECTOR for the OFDM PHY.

### 17.2.2 TXVECTOR parameters

The parameters in Table 17-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request service primitive.

#### 17.2.2.1 TXVECTOR LENGTH

The allowed values for the LENGTH parameter are in the range of 1 to 4095. This parameter is used to indicate the number of octets in the MPDU which the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start the transmission.

#### 17.2.2.2 TXVECTOR DATARATE

The DATARATE parameter describes the bit rate at which the PLCP shall transmit the PSDU. Its value can be any of the rates defined in Table 17-1. Data rates of 6, 12, and 24 Mb/s shall be supported for 20 MHz channel spacing, data rates of 3, 6, and 12 Mb/s shall be supported for 10 MHz channel spacing, and data rates of 1.5, 3, and 6 Mb/s shall be supported for 5 MHz channel spacing; other rates may also be supported.


Copyright © 2007 IEEE. All rights reserved.

**Table 17-1—TXVECTOR parameters**

| Parameter | Associate primitive | Value |
|---|---|---|
| LENGTH | PHY-TXSTART.request (TXVECTOR) | 1–4095 |
| DATATRATE | PHY-TXSTART.request (TXVECTOR) | 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Support of 6, 12, and 24 Mb/s data rates is mandatory.)<br><br>3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Support of 3, 6, and 12 Mb/s data rates is mandatory.)<br><br>1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Support of 1.5, 3, and 6 Mb/s data rates is mandatory.) |
| SERVICE | PHY-TXSTART.request (TXVECTOR) | Scrambler initialization; 7 null bits + 9 reserved null bits |
| TXPWR_LEVEL | PHY-TXSTART.request (TXVECTOR) | 1–8 |

#### 17.2.2.3 TXVECTOR SERVICE

The SERVICE parameter consists of 7 null bits used for the scrambler initialization and 9 null bits reserved for future use.

#### 17.2.2.4 TXVECTOR TXPWR_LEVEL

The allowed values for the TXPWR_LEVEL parameter are in the range from 1 to 8. This parameter is used to indicate which of the available TxPowerLevel attributes defined in the MIB shall be used for the current transmission.

### 17.2.3 RXVECTOR parameters

The parameters listed in Table 17-2 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indicate service primitive.

**Table 17-2—RXVECTOR parameters**

| Parameter | Associate primitive | Value |
|---|---|---|
| LENGTH | PHY-RXSTART.indicate | 1–4095 |
| RSSI | PHY-RXSTART.indicate (RXVECTOR) | 0–RSSI maximum |

Copyright © 2007 IEEE. All rights reserved.

**Table 17-2—RXVECTOR parameters *(continued)***

| Parameter | Associate primitive | Value |
|---|---|---|
| DATARATE | PHY-RXSTART.request (RXVECTOR) | 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Support of 6, 12, and 24 Mb/s data rates is mandatory.)<br><br>3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Support of 3, 6, and 12 Mb/s data rates is mandatory.)<br><br>1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Support of 1.5, 3, and 6 Mb/s data rates is mandatory.) |
| SERVICE | PHY-RXSTART.request (RXVECTOR) | Null |

#### 17.2.3.1 RXVECTOR LENGTH

The allowed values for the LENGTH parameter are in the range from 1–4095. This parameter is used to indicate the value contained in the LENGTH field which the PLCP has received in the PLCP header. The MAC and PLCP will use this value to determine the number of octet transfers that will occur between the two sublayers during the transfer of the received PSDU.

#### 17.2.3.2 RXVECTOR RSSI

The allowed values for the RSSI parameter are in the range from 0 through RSSI maximum. This parameter is a measure by the PHY of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured during the reception of the PLCP preamble. RSSI is intended to be used in a relative manner, and it shall be a monotonically increasing function of the received power.

#### 17.2.3.3 DATARATE

DATARATE shall represent the data rate at which the current PPDU was received. The allowed values of the DATARATE are 6, 9, 12, 18, 24, 36, 48, or 54 Mb/s for 20 MHz channel spacing; 3, 4.5, 6, 9, 12, 18, 24, or 27 Mb/s for 10 MHz channel spacing; and 1.5, 2.25, 3, 4.5, 6, 9, 12, or 13.5 Mb/s for 5 MHz channel spacing.

#### 17.2.3.4 SERVICE

The SERVICE field shall be null.

## 17.3 OFDM PLCP sublayer

### 17.3.1 Introduction

This subclause provides a convergence procedure in which PSDUs are converted to and from PPDUs. During transmission, the PSDU shall be provided with a PLCP preamble and header to create the PPDU. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the PSDU.


Copyright © 2007 IEEE. All rights reserved.

### 17.3.2 PLCP frame format

Figure 17-1 shows the format for the PPDU including the OFDM PLCP preamble, OFDM PLCP header, PSDU, tail bits, and pad bits. The PLCP header contains the following fields: LENGTH, RATE, a reserved bit, an even parity bit, and the SERVICE field. In terms of modulation, the LENGTH, RATE, reserved bit, and parity bit (with 6 zero tail bits appended) constitute a separate single OFDM symbol, denoted SIGNAL, which is transmitted with the most robust combination of BPSK modulation and a coding rate of R = 1/2. The SERVICE field of the PLCP header and the PSDU (with 6 zero tail bits and pad bits appended), denoted as DATA, are transmitted at the data rate described in the RATE field and may constitute multiple OFDM symbols. The tail bits in the SIGNAL symbol enable decoding of the RATE and LENGTH fields immediately after the reception of the tail bits. The RATE and LENGTH fields are required for decoding the DATA part of the packet. In addition, the CCA mechanism can be augmented by predicting the duration of the packet from the contents of the RATE and LENGTH fields, even if the data rate is not supported by the STA. Each of these fields is described in detail in 17.3.3, 17.3.4, and 17.3.5.

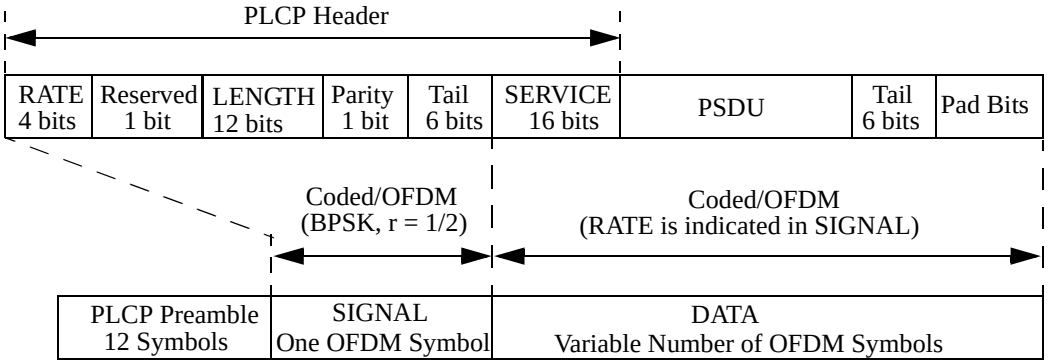



**Figure 17-1—PPDU frame format**

#### 17.3.2.1 Overview of the PPDU encoding process

The encoding process is composed of many detailed steps, which are described fully in later subclauses, as noted below. The following overview intends to facilitate understanding the details of the convergence procedure:

a) Produce the PLCP Preamble field, composed of 10 repetitions of a "short training sequence" (used for AGC convergence, diversity selection, timing acquisition, and coarse frequency acquisition in the receiver) and two repetitions of a "long training sequence" (used for channel estimation and fine frequency acquisition in the receiver), preceded by a guard interval (GI). Refer to 17.3.3 for details.

b) Produce the PLCP header field from the RATE, LENGTH, and SERVICE fields of the TXVECTOR by filling the appropriate bit fields. The RATE and LENGTH fields of the PLCP header are encoded by a convolutional code at a rate of R = 1/2, and are subsequently mapped onto a single BPSK encoded OFDM symbol, denoted as the SIGNAL symbol. In order to facilitate a reliable and timely detection of the RATE and LENGTH fields, 6 zero tail bits are inserted into the PLCP header. The encoding of the SIGNAL field into an OFDM symbol follows the same steps for convolutional encoding, interleaving, BPSK modulation, pilot insertion, Fourier transform, and prepending a GI as described subsequently for data transmission with BPSK-OFDM modulated at coding rate 1/2. The contents of the SIGNAL field are not scrambled. Refer to 17.3.4 for details.

c) Calculate from RATE field of the TXVECTOR the number of data bits per OFDM symbol ($N_{DBPS}$), the coding rate (R), the number of bits in each OFDM subcarrier ($N_{BPSC}$), and the number of coded bits per OFDM symbol ($N_{CBPS}$). Refer to 17.3.2.2 for details.

Copyright © 2007 IEEE. All rights reserved.

d) Append the PSDU to the SERVICE field of the TXVECTOR. Extend the resulting bit string with zero bits (at least 6 bits) so that the resulting length will be a multiple of $N_{DBPS}$. The resulting bit string constitutes the DATA part of the packet. Refer to 17.3.5.3 for details.

e) Initiate the scrambler with a pseudo-random nonzero seed, generate a scrambling sequence, and XOR it with the extended string of data bits. Refer to 17.3.5.4 for details.

f) Replace the six scrambled zero bits following the data with six nonscrambled zero bits. (Those bits return the convolutional encoder to the zero state and are denoted as tail bits.) Refer to 17.3.5.2 for details.

g) Encode the extended, scrambled data string with a convolutional encoder (R = 1/2). Omit (puncture) some of the encoder output string (chosen according to "puncturing pattern") to reach the desired "coding rate." Refer to 17.3.5.5 for details.

h) Divide the encoded bit string into groups of $N_{CBPS}$ bits. Within each group, perform an "interleaving" (reordering) of the bits according to a rule corresponding to the desired RATE. Refer to 17.3.5.6 for details.

i) Divide the resulting coded and interleaved data string into groups of $N_{CBPS}$ bits. For each of the bit groups, convert the bit group into a complex number according to the modulation encoding tables. Refer to 17.3.5.7 for details.

j) Divide the complex number string into groups of 48 complex numbers. Each such group will be associated with one OFDM symbol. In each group, the complex numbers will be numbered 0 to 47 and mapped hereafter into OFDM subcarriers numbered –26 to –22, –20 to –8, –6 to –1, 1 to 6, 8 to 20, and 22 to 26. The subcarriers –21, –7, 7, and 21 are skipped and, subsequently, used for inserting pilot subcarriers. The 0 subcarrier, associated with center frequency, is omitted and filled with zero value. Refer to 17.3.5.9 for details.

k) Four subcarriers are inserted as pilots into positions –21, –7, 7, and 21. The total number of the subcarriers is 52 (48 + 4). Refer to 17.3.5.8 for details.

l) For each group of subcarriers –26 to 26, convert the subcarriers to time domain using inverse Fourier transform. Prepend to the Fourier-transformed waveform a circular extension of itself thus forming a GI, and truncate the resulting periodic waveform to a single OFDM symbol length by applying time domain windowing. Refer to 17.3.5.9 for details.

m) Append the OFDM symbols one after another, starting after the SIGNAL symbol describing the RATE and LENGTH fields. Refer to 17.3.5.9 for details.

n) Up-convert the resulting "complex baseband" waveform to an RF according to the center frequency of the desired channel and transmit. Refer to 17.3.2.4 and 17.3.8.1 for details.

An illustration of the transmitted frame and its parts appears in Figure 17-4 (in 17.3.3).

#### 17.3.2.2 Modulation-dependent parameters

The modulation parameters dependent on the data rate used shall be set according to Table 17-3.


Copyright © 2007 IEEE. All rights reserved.

**Table 17-3—Modulation-dependent parameters**

| Modulation | Coding rate ($R$) | Coded bits per subcarrier ($N_{BPSC}$) | Coded bits per OFDM symbol ($N_{CBPS}$) | Data bits per OFDM symbol ($N_{DBPS}$) | Data rate (Mb/s) (20 MHz channel spacing) | Data rate (Mb/s) (10 MHz channel spacing) | Data rate (Mb/s) (5 MHz channel spacing) |
|---|---|---|---|---|---|---|---|
| BPSK | 1/2 | 1 | 48 | 24 | 6 | 3 | 1.5 |
| BPSK | 3/4 | 1 | 48 | 36 | 9 | 4.5 | 2.25 |
| QPSK | 1/2 | 2 | 96 | 48 | 12 | 6 | 3 |
| QPSK | 3/4 | 2 | 96 | 72 | 18 | 9 | 4.5 |
| 16-QAM | 1/2 | 4 | 192 | 96 | 24 | 12 | 6 |
| 16-QAM | 3/4 | 4 | 192 | 144 | 36 | 18 | 9 |
| 64-QAM | 2/3 | 6 | 288 | 192 | 48 | 24 | 12 |
| 64-QAM | 3/4 | 6 | 288 | 216 | 54 | 27 | 13.5 |

### 17.3.2.3 Timing related parameters

Table 17-4 is the list of timing parameters associated with the OFDM PLCP.

**Table 17-4—Timing-related parameters**

| Parameter | Value (20 MHz channel spacing) | Value (10 MHz channel spacing) | Value (5 MHz channel spacing) |
|---|---|---|---|
| $N_{SD}$: Number of data subcarriers | 48 | 48 | 48 |
| $N_{SP}$: Number of pilot subcarriers | 4 | 4 | 4 |
| $N_{ST}$: Number of subcarriers, total | 52 ($N_{SD} + N_{SP}$) | 52 ($N_{SD} + N_{SP}$) | 52 ($N_{SD} + N_{SP}$) |
| $\Delta_F$: Subcarrier frequency spacing | 0.3125 MHz (=20 MHz/64) | 0.15625 MHz (= 10 MHz/64) | 0.078125 MHz (= 5 MHz/64) |
| $T_{FFT}$: Inverse Fast Fourier Transform (IFFT) / Fast Fourier Transform (FFT) period | 3.2 μs ($1/\Delta_F$) | 6.4 μs ($1/\Delta_F$) | 12.8 μs ($1/\Delta_F$) |
| $T_{PREAMBLE}$: PLCP preamble duration | 16 μs ($T_{SHORT} + T_{LONG}$) | 32 μs ($T_{SHORT} + T_{LONG}$) | 64 μs ($T_{SHORT} + T_{LONG}$) |
| $T_{SIGNAL}$: Duration of the SIGNAL BPSK-OFDM symbol | 4.0 μs ($T_{GI} + T_{FFT}$) | 8.0 μs ($T_{GI} + T_{FFT}$) | 16.0 μs ($T_{GI} + T_{FFT}$) |
| $T_{GI}$: GI duration | 0.8 μs ($T_{FFT}/4$) | 1.6 μs ($T_{FFT}/4$) | 3.2 μs ($T_{FFT}/4$) |
| $T_{GI2}$: Training symbol GI duration | 1.6 μs ($T_{FFT}/2$) | 3.2 μs ($T_{FFT}/2$) | 6.4 μs ($T_{FFT}/2$) |
| $T_{SYM}$: Symbol interval | 4 μs ($T_{GI} + T_{FFT}$) | 8 μs ($T_{GI} + T_{FFT}$) | 16 μs ($T_{GI} + T_{FFT}$) |

Copyright © 2007 IEEE. All rights reserved.

**Table 17-4—Timing-related parameters *(continued)***

| Parameter | Value (20 MHz channel spacing) | Value (10 MHz channel spacing) | Value (5 MHz channel spacing) |
|---|---|---|---|
| $T_{SHORT}$: Short training sequence duration | 8 μs ($10 \times T_{FFT}/4$) | 16 μs ($10 \times T_{FFT}/4$) | 32 μs ($10 \times T_{FFT}/4$) |
| $T_{LONG}$: Long training sequence duration | 8 μs ($T_{GI2} + 2 \times T_{FFT}$) | 16 μs ($T_{GI2} + 2 \times T_{FFT}$) | 32 μs ($T_{GI2} + 2 \times T_{FFT}$) |

### 17.3.2.4 Mathematical conventions in the signal descriptions

The transmitted signals will be described in a complex baseband signal notation. The actual transmitted signal is related to the complex baseband signal by the following relation:

$$r_{(RF)\langle t\rangle} = Re\{r\langle t\rangle \exp\langle j2\pi f_c t\rangle\} \quad (17\text{-}1)$$

where

$Re(.)$ represents the real part of a complex variable

$f_c$ denotes the carrier center frequency

The transmitted baseband signal is composed of contributions from several OFDM symbols.

$$r_{PACKET}(t) = r_{PREAMBLE}(t) + r_{SIGNAL}(t - t_{SIGNAL}) + r_{DATA}(t - t_{DATA}) \quad (17\text{-}2)$$

The subframes of which Equation (17-2) are composed are described in 17.3.3, 17.3.4, and 17.3.5.9. The time offsets $t_{SUBFRAME}$ determine the starting time of the corresponding subframe; $t_{SIGNAL}$ is equal to 16 μs for 20 MHz channel spacing, 32 μs for 10 MHz channel spacing, and 64 μs for 5 MHz channel spacing, and $t_{DATA}$ is equal to 20 μs for 20 MHz channel spacing, 40 μs for 10 MHz channel spacing, and 80 μs for 5 MHz channel spacing.

All the subframes of the signal are constructed as an inverse Fourier transform of a set of coefficients, $C_k$, with $C_k$ defined later as data, pilots, or training symbols in 17.3.3 through 17.3.5.

$$r_{SUBFRAME}(t) = w_{TSUBFRAME}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} C_k \exp(j2\pi k\Delta_f)(t - T_{GUARD}) \quad (17\text{-}3)$$

The parameters $\Delta_F$ and $N_{ST}$ are described in Table 17-4. The resulting waveform is periodic with a period of $T_{FFT} = 1/\Delta_F$. Shifting the time by $T_{GUARD}$ creates the "circular prefix" used in OFDM to avoid ISI from the previous frame. Three kinds of $T_{GUARD}$ are defined: for the short training sequence (= 0 μs), for the long training sequence (= $T_{GI2}$), and for data OFDM symbols (= $T_{GI}$). (Refer to Table 17-4.) The boundaries of the subframe are set by a multiplication by a time-windowing function, $w_{TSUBFRAME}(t)$, which is defined as a rectangular pulse, $w_T(t)$, of duration $T$, accepting the value $T_{SUBFRAME}$. The time-windowing function, $w_T(t)$, depending on the value of the duration parameter, $T$, may extend over more than one period, $T_{FFT}$. In particular, window functions that extend over multiple periods of the FFT are utilized in the definition of the preamble. Figure 17-2 illustrates the possibility of extending the windowing function over more than one period, $T_{FFT}$, and additionally shows smoothed transitions by application of a windowing function, as exemplified in Equation (17-4). In particular, window functions that extend over multiple periods of the FFT are utilized in the definition of the preamble.


Copyright © 2007 IEEE. All rights reserved.

$$w_T(t) = \begin{cases} \sin^2\left(\frac{\pi}{2}(0.5 + t/T_{TR})\right) & (-T_{TR}/2 < t < T_{TR}/2) \\ 1 & (T_{TR}/2 \le t < T - T_{TR}/2) \\ \sin^2\left(\frac{\pi}{2}(0.5 - (t-T)/T_{TR})\right) & (T - T_{TR}/2 \le t < T + T_{TR}/2) \end{cases} \quad (17\text{-}4)$$

In the case of vanishing $T_{TR}$, the windowing function degenerates into a rectangular pulse of duration $T$. The normative specifications of generating the transmitted waveforms shall utilize the rectangular pulse shape. In implementation, higher $T_{TR}$ is typically implemented in order to smooth the transitions between the consecutive subsections. This creates a small overlap between them, of duration $T_{TR}$, as shown in Figure 17-2. The transition time, $T_{TR}$, is about 100 ns. Smoothing the transition is required in order to reduce the spectral sidelobes of the transmitted waveform. However, the binding requirements are the spectral mask and modulation accuracy requirements, as detailed in 17.3.9.2 and 17.3.9.6. Time domain windowing, as described here, is just one way to achieve those objectives. The implementer may use other methods to achieve the same goal, such as frequency domain filtering. Therefore, the transition shape and duration of the transition are informative parameters.

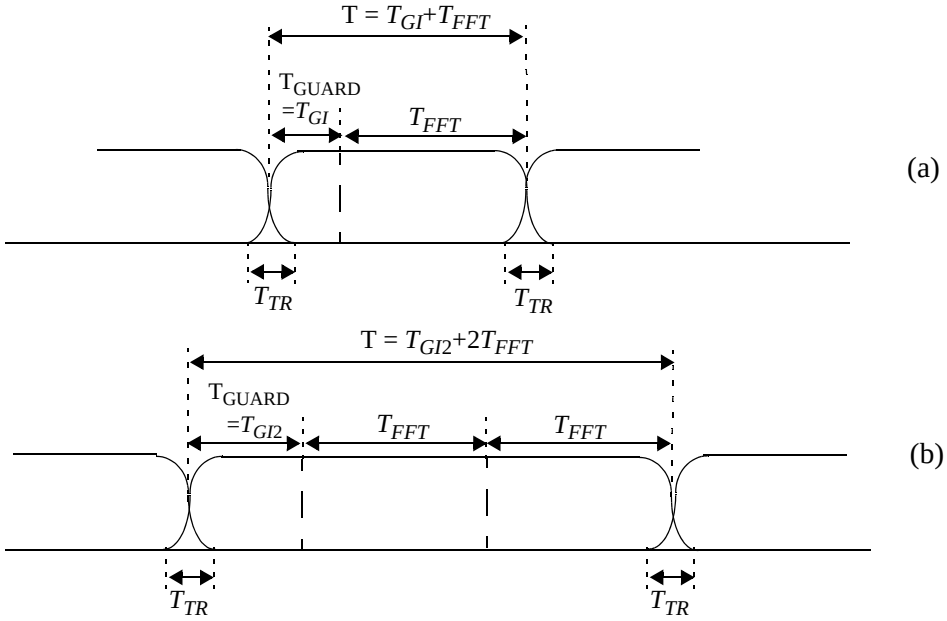



**Figure 17-2—Illustration of OFDM frame with cyclic extension and windowing for (a) single reception or (b) two receptions of the FFT period**

### 17.3.2.5 Discrete time implementation considerations

The following descriptions of the discrete time implementation are informational.

In a typical implementation, the windowing function will be represented in discrete time. As an example, when a windowing function with parameters T = 4.0 μs and a $T_{TR}$ = 100 ns is applied, and the signal is sampled at 20 Msample/s, it becomes

Copyright © 2007 IEEE. All rights reserved.

$$w_T[n] = w_T(nT_S) = \begin{Bmatrix} 1 & 1 \le n \le 79 \\ 0.5 & 0, 80 \\ 0 & otherwise \end{Bmatrix} \quad (17\text{-}5)$$

The common way to implement the inverse Fourier transform, as shown in Equation (17-3), is by an IFFT algorithm. If, for example, a 64-point IFFT is used, the coefficients 1 to 26 are mapped to the same numbered IFFT inputs, while the coefficients –26 to –1 are copied into IFFT inputs 38 to 63. The rest of the inputs, 27 to 37 and the 0 (dc) input, are set to 0. This mapping is illustrated in Figure 17-3. After performing an IFFT, the output is cyclically extended to the desired length.

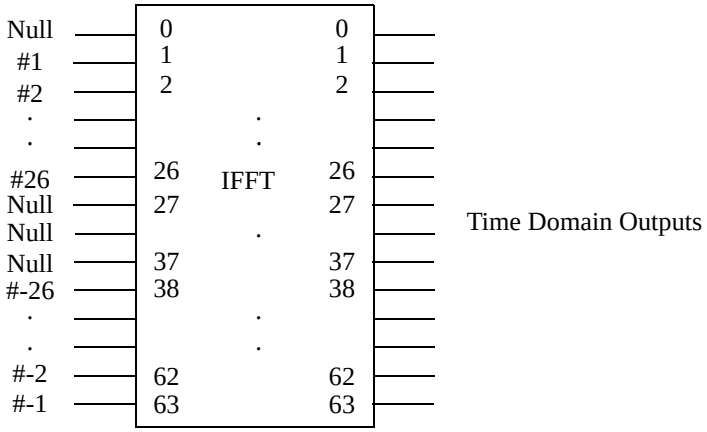



**Figure 17-3—Inputs and outputs of inverse Fourier transform**

### 17.3.3 PLCP preamble (SYNC)

The PLCP Preamble field is used for synchronization. It consists of 10 short symbols and two long symbols that are shown in Figure 17-4 and described in this subclause. The timings described in this subclause and shown in Figure 17-4 are for 20 MHz channel spacing. They are doubled for half-clocked (i.e., 10 MHz) channel spacing and are quadrupled for quarter-clocked (i.e., 5 MHz) channel spacing.

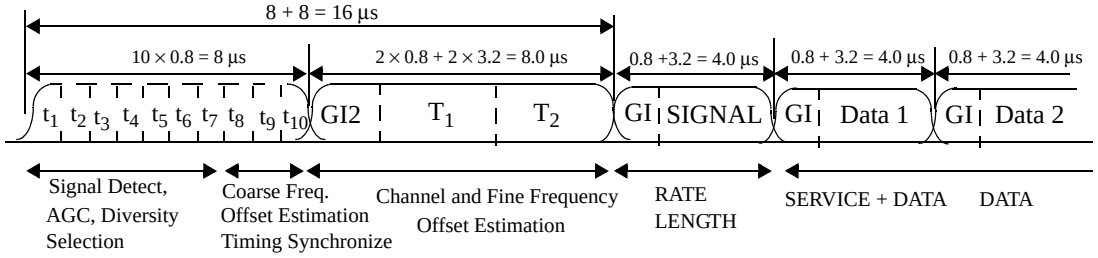



**Figure 17-4—OFDM training structure**

Figure 17-4 shows the OFDM training structure (PLCP preamble), where $t_1$ to $t_{10}$ denote short training symbols and $T_1$ and $T_2$ denote long training symbols. The PLCP preamble is followed by the SIGNAL field and DATA. The total training length is 16 µs. The dashed boundaries in the figure denote repetitions due to the periodicity of the inverse Fourier transform.


Copyright © 2007 IEEE. All rights reserved.

A short OFDM training symbol consists of 12 subcarriers, which are modulated by the elements of the sequence S, given by

$$S_{-26,26} = \sqrt{(13/6)} \times \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0,$$

$$0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0,0\} \quad (17\text{-}6)$$

The multiplication by a factor of $\sqrt{(13/6)}$ is in order to normalize the average power of the resulting OFDM symbol, which utilizes 12 out of 52 subcarriers.

The signal shall be generated according to the following equation:

$$r_{SHORT}(t) = w_{TSHORT}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_k \exp(j2\pi k\Delta_F t) \quad (17\text{-}7)$$

The fact that only spectral lines of $S_{-26:26}$ with indices that are a multiple of 4 have nonzero amplitude results in a periodicity of $T_{FFT}/4 = 0.8$ μs. The interval $T_{SHORT}$ is equal to ten 0.8 μs periods (i.e., 8 μs).

Generation of the short training sequence is illustrated in Table G.2.

A long OFDM training symbol consists of 53 subcarriers (including a zero value at dc), which are modulated by the elements of the sequence L, given by

$$L_{-26,26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 0,$$

$$1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1\} \quad (17\text{-}8)$$

A long OFDM training symbol shall be generated according to the following equation:

$$r_{LONG}(t) = w_{TLONG}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} L_k \exp(j2\pi k\Delta_F (t - T_{G12})) \quad (17\text{-}9)$$

where

$T_{G12} = 1.6$ μs

Two periods of the long sequence are transmitted for improved channel estimation accuracy, yielding $T_{LONG} = 1.6 + 2 \times 3.2 = 8$ μs.

An illustration of the long training sequence generation is given in Table G.5.

The sections of short repetitions and long repetitions shall be concatenated to form the preamble

$$r_{PREAMBLE}(t) = r_{SHORT}(t) + r_{LONG}(t - T_{SHORT}) \quad (17\text{-}10)$$

### 17.3.4 SIGNAL field

The OFDM training symbols shall be followed by the SIGNAL field, which contains the RATE and the LENGTH fields of the TXVECTOR. The RATE field conveys information about the type of modulation and the coding rate as used in the rest of the packet. The encoding of the SIGNAL single OFDM symbol

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 649 of 1232

shall be performed with BPSK modulation of the subcarriers and using convolutional coding at R = 1/2. The encoding procedure, which includes convolutional encoding, interleaving, modulation mapping processes, pilot insertion, and OFDM modulation, follows the steps described in 17.3.5.5, 17.3.5.6, and 17.3.5.8, as used for transmission of data with BPSK-OFDM modulated at coding rate 1/2. The contents of the SIGNAL field are not scrambled.

The SIGNAL field shall be composed of 24 bits, as illustrated in Figure 17-5. The four bits 0 to 3 shall encode the RATE. Bit 4 shall be reserved for future use. Bits 5–16 shall encode the LENGTH field of the TXVECTOR, with the LSB being transmitted first.

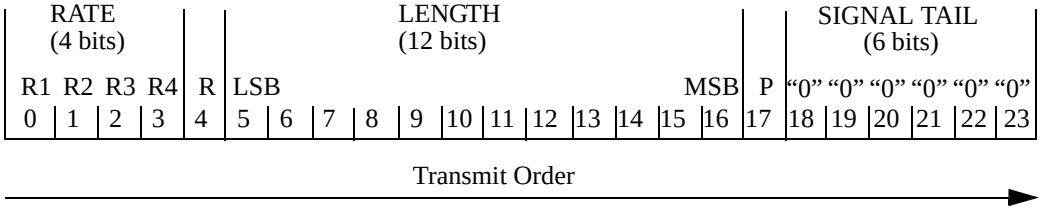



**Figure 17-5—SIGNAL field bit assignment**

The process of generating the SIGNAL OFDM symbol is illustrated in G.4.

### 17.3.4.1 RATE field

The bits R1–R4 shall be set, dependent on RATE, according to the values in Table 17-5.

**Table 17-5—Contents of the SIGNAL field**

| R1–R4 | Rate (Mb/s) (20 MHz channel spacing) | Rate (Mb/s) (10 MHz channel spacing) | Rate (Mb/s) (5 MHz channel spacing) |
|---|---|---|---|
| 1101 | 6 | 3 | 1.5 |
| 1111 | 9 | 4.5 | 2.25 |
| 0101 | 12 | 6 | 3 |
| 0111 | 18 | 9 | 4.5 |
| 1001 | 24 | 12 | 6 |
| 1011 | 36 | 18 | 9 |
| 0001 | 48 | 24 | 12 |
| 0011 | 54 | 27 | 13.5 |

### 17.3.4.2 PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 12-bit integer that indicates the number of octets in the PSDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start transmission. The transmitted value shall be determined from the LENGTH parameter in the


Copyright © 2007 IEEE. All rights reserved.

TXVECTOR issued with the PHY-TXSTART.request primitive described in 12.3.5.4. The LSB shall be transmitted first in time. This field shall be encoded by the convolutional encoder described in 17.3.5.5.

#### 17.3.4.3 Parity (P), Reserved (R), and SIGNAL TAIL fields

Bit 4 shall be reserved for future use. Bit 17 shall be a positive parity (even parity) bit for bits 0–16. The bits 18–23 constitute the SIGNAL TAIL field, and all 6 bits shall be set to 0.

### 17.3.5 DATA field

The DATA field contains the SERVICE field, the PSDU, the TAIL bits, and the PAD bits, if needed, as described in 17.3.5.2 and 17.3.5.3. All bits in the DATA field are scrambled, as described in 17.3.5.4.

#### 17.3.5.1 SERVICE field

The IEEE 802.11 SERVICE field has 16 bits, which shall be denoted as bits 0–15. The bit 0 shall be transmitted first in time. The bits from 0–6 of the SERVICE field, which are transmitted first, are set to zeros and are used to synchronize the descrambler in the receiver. The remaining 9 bits (7–15) of the SERVICE field shall be reserved for future use. All reserved bits shall be set to 0. Refer to Figure 17-6.

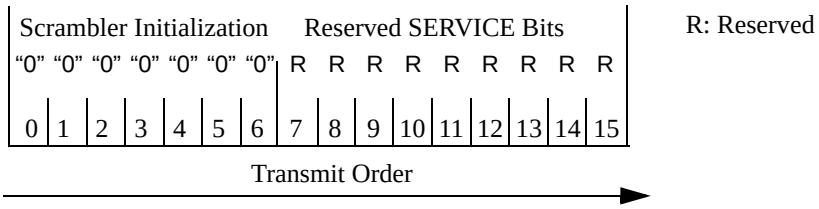



**Figure 17-6—SERVICE field bit assignment**

#### 17.3.5.2 PPDU TAIL field

The PPDU TAIL field shall be six bits of zero, which are required to return the convolutional encoder to the zero state. This procedure improves the error probability of the convolutional decoder, which relies on future bits when decoding and which may be not be available past the end of the message. The PLCP tail bit field shall be produced by replacing six scrambled zero bits following the message end with six nonscrambled zero bits.

#### 17.3.5.3 Pad bits (PAD)

The number of bits in the DATA field shall be a multiple of $N_{CBPS}$, the number of coded bits in an OFDM symbol (48, 96, 192, or 288 bits). To achieve that, the length of the message is extended so that it becomes a multiple of $N_{DBPS}$, the number of data bits per OFDM symbol. At least 6 bits are appended to the message, in order to accommodate the TAIL bits, as described in 17.3.5.2. The number of OFDM symbols, $N_{SYM}$; the number of bits in the DATA field, $N_{DATA}$; and the number of pad bits, $N_{PAD}$, are computed from the length of the PSDU (LENGTH) as follows:

$$N_{SYM} = \text{Ceiling}\left((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}\right) \quad (17\text{-}11)$$

$$N_{DATA} = N_{SYM} \times N_{DBPS} \quad (17\text{-}12)$$

$$N_{PAD} = N_{DATA} - (16 + 8 \times \text{LENGTH} + 6) \quad (17\text{-}13)$$

Copyright © 2007 IEEE. All rights reserved.

The function ceiling (.) is a function that returns the smallest integer value greater than or equal to its argument value. The appended bits ("pad bits") are set to zeros and are subsequently scrambled with the rest of the bits in the DATA field.

An example of a DATA field that contains the SERVICE field, DATA, tail, and pad bits is given in G.5.1.

#### 17.3.5.4 PLCP DATA scrambler and descrambler

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be scrambled with a length-127 frame-synchronous scrambler. The octets of the PSDU are placed in the transmit serial bit stream, bit 0 first and bit 7 last. The frame synchronous scrambler uses the generator polynomial *S(x)* as follows, and is illustrated in Figure 17-7:

$$S(x) = x^7 + x^4 + 1 \quad (17\text{-}14)$$

The 127-bit sequence generated repeatedly by the scrambler shall be (leftmost used first), 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111, when the all ones initial state is used. The same scrambler is used to scramble transmit data and to descramble receive data. When transmitting, the initial state of the scrambler will be set to a pseudo-random nonzero state. The seven LSBs of the SERVICE field will be set to all zeros prior to scrambling to enable estimation of the initial state of the scrambler in the receiver.

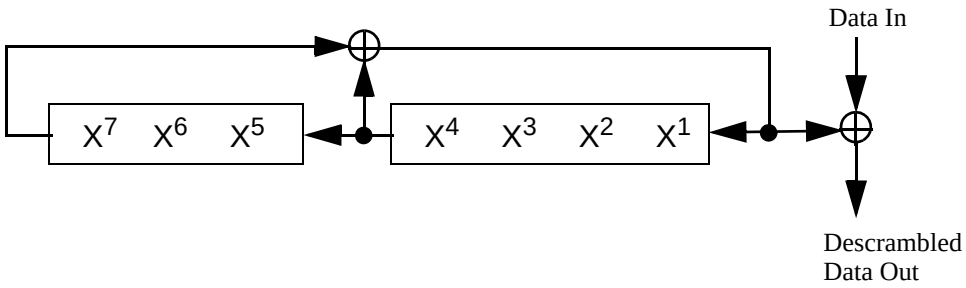



**Figure 17-7—Data scrambler**

An example of the scrambler output is illustrated in G.5.2.

#### 17.3.5.5 Convolutional encoder

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be coded with a convolutional encoder of coding rate R = 1/2, 2/3, or 3/4, corresponding to the desired data rate. The convolutional encoder shall use the industry-standard generator polynomials, $g_0 = 133_8$ and $g_1 = 171_8$, of rate R = 1/2, as shown in Figure 17-8. The bit denoted as "A" shall be output from the encoder before the bit denoted as "B." Higher rates are derived from it by employing "puncturing." Puncturing is a procedure for omitting some of the encoded bits in the transmitter (thus reducing the number of transmitted bits and increasing the coding rate) and inserting a dummy "zero" metric into the convolutional decoder on the receive side in place of the omitted bits. The puncturing patterns are illustrated in Figure 17-9. Decoding by the Viterbi algorithm is recommended.

An example of encoding operation is shown in G.6.1.


Copyright © 2007 IEEE. All rights reserved.

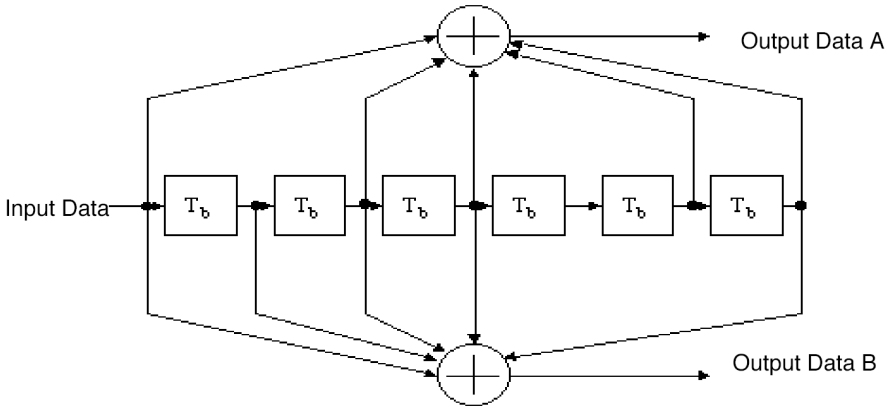



**Figure 17-8—Convolutional encoder (k = 7)**

### 17.3.5.6 Data interleaving

All encoded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of bits in a single OFDM symbol, $N_{CBPS}$. The interleaver is defined by a two-step permutation. The first permutation ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second ensures that adjacent coded bits are mapped alternately onto less and more significant bits of the constellation and, thereby, long runs of low reliability (LSB) bits are avoided.

The index of the coded bit before the first permutation shall be denoted by $k$; $i$ shall be the index after the first and before the second permutation; and $j$ shall be the index after the second permutation, just prior to modulation mapping.

The first permutation is defined by the rule

$$i = (N_{CBPS}/16)\,(k \bmod 16) + \text{floor}(k/16) \qquad k = 0,1,\ldots,N_{CBPS} - 1 \tag{17-15}$$

The function floor (.) denotes the largest integer not exceeding the parameter.

The second permutation is defined by the rule

$$j = s \times \text{floor}(i/s) + (i + N_{CBPS} - \text{floor}(16 \times i/N_{CBPS})) \bmod s \qquad i = 0,1,\ldots N_{CBPS} - 1 \tag{17-16}$$

The value of s is determined by the number of coded bits per subcarrier, $N_{BPSC}$, according to

$$s = \max(N_{BPSC}/2,1) \tag{17-17}$$

The deinterleaver, which performs the inverse relation, is also defined by two permutations.

Copyright © 2007 IEEE. All rights reserved.

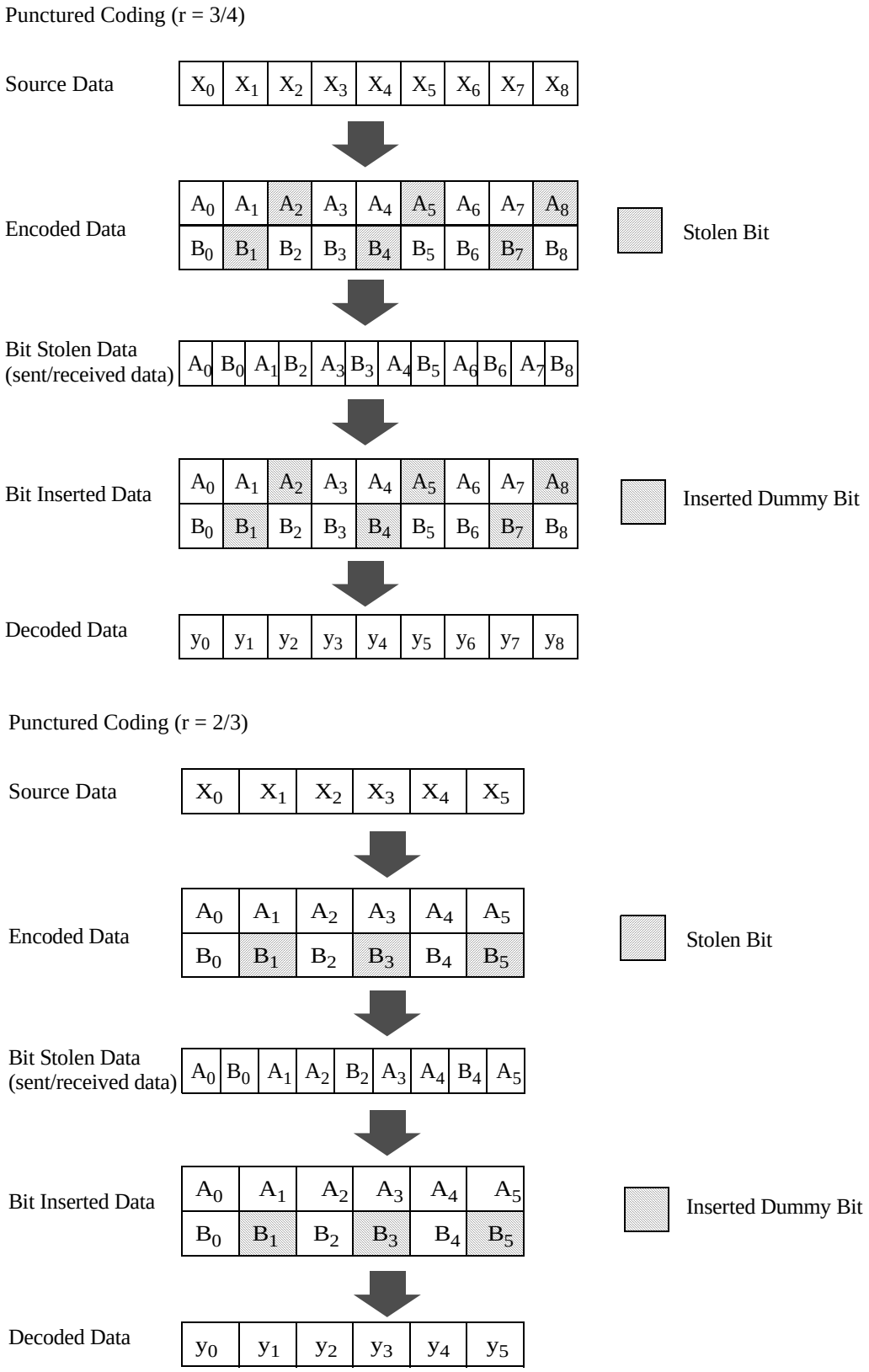



**Figure 17-9—Example of the bit-stealing and bit-insertion procedure (r = 3/4, 2/3)**


Copyright © 2007 IEEE. All rights reserved.

Here the index of the original received bit before the first permutation shall be denoted by *j*; *i* shall be the index after the first and before the second permutation; and *k* shall be the index after the second permutation, just prior to delivering the coded bits to the convolutional (Viterbi) decoder.

The first permutation is defined by the rule

$$i = s \times \text{floor}(j/s) + (j + \text{floor}(16 \times j/N_{CBPS})) \bmod s \qquad j = 0,1,\ldots N_{CBPS} - 1 \tag{17-18}$$

where

*s* is defined in Equation (17-17).

This permutation is the inverse of the permutation described in Equation (17-16).

The second permutation is defined by the rule

$$k = 16 \times i - (N_{CBPS} - 1)\text{floor}(16 \times i/N_{CBPS}) \qquad i = 0,1,\ldots N_{CBPS} - 1 \tag{17-19}$$

This permutation is the inverse of the permutation described in Equation (17-15).

An example of interleaving operation is illustrated in G.6.2.

### 17.3.5.7 Subcarrier modulation mapping

The OFDM subcarriers shall be modulated by using BPSK, QPSK, 16-QAM, or 64-QAM, depending on the RATE requested. The encoded and interleaved binary serial input data shall be divided into groups of $N_{BPSC}$ (1, 2, 4, or 6) bits and converted into complex numbers representing BPSK, QPSK, 16-QAM, or 64-QAM constellation points. The conversion shall be performed according to Gray-coded constellation mappings, illustrated in Figure 17-10, with the input bit, $b_0$, being the earliest in the stream. The output values, d, are formed by multiplying the resulting (I+jQ) value by a normalization factor $K_{MOD}$, as described in Equation (17-20).

$$d = (I + jQ) \times K_{MOD} \tag{17-20}$$

The normalization factor, $K_{MOD}$, depends on the base modulation mode, as prescribed in Table 17-6. Note that the modulation type can be different from the start to the end of the transmission, as the signal changes from SIGNAL to DATA, as shown in Figure 17-1. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor can be used, as long as the device conforms with the modulation accuracy requirements described in 17.3.9.6.

**Table 17-6—Modulation-dependent normalization factor $K_{MOD}$**

| Modulation | $K_{MOD}$ |
|---|---|
| BPSK | 1 |
| QPSK | $1/\sqrt{2}$ |
| 16-QAM | $1/\sqrt{10}$ |
| 64-QAM | $1/\sqrt{42}$ |

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 655 of 1232

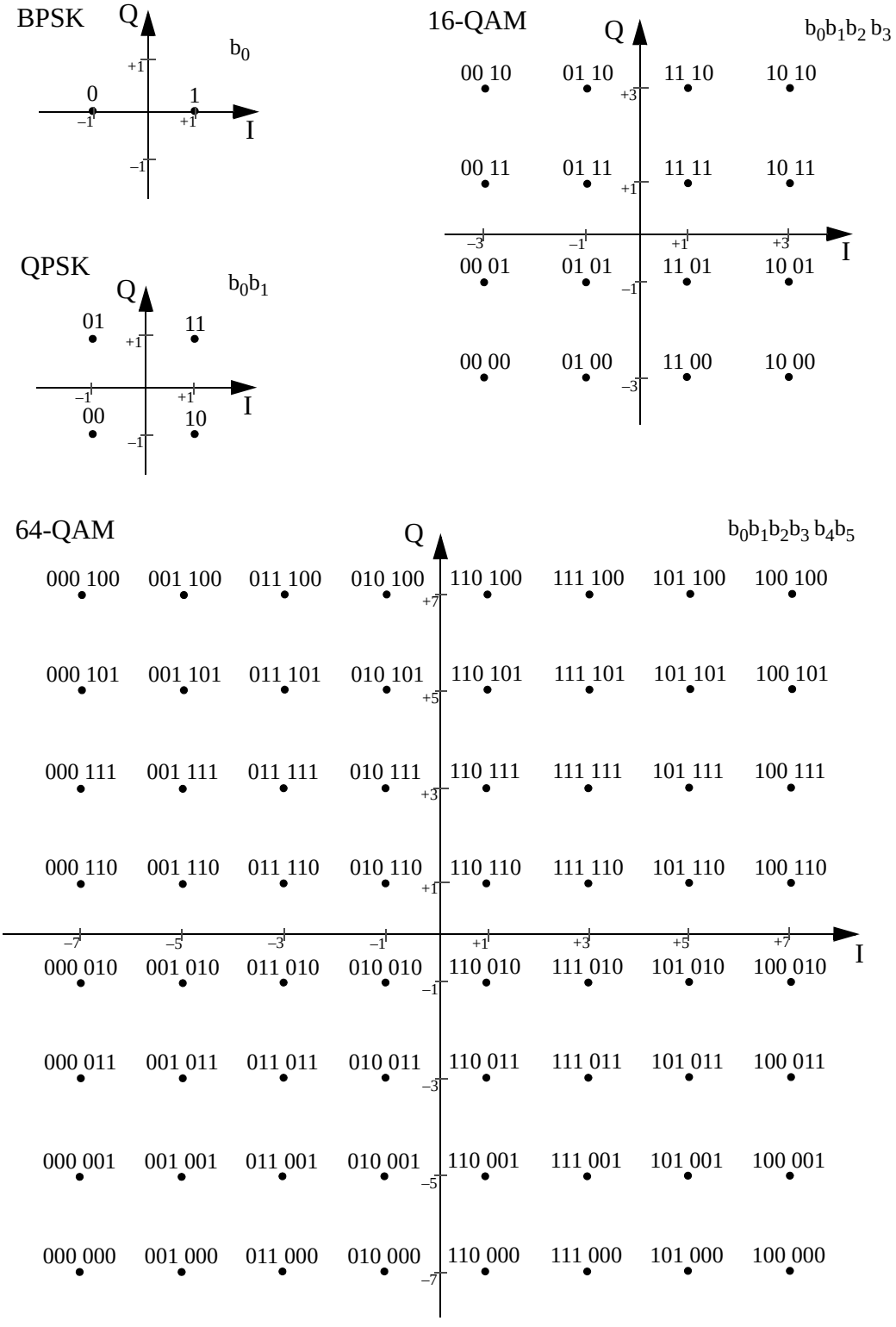



**Figure 17-10—BPSK, QPSK, 16-QAM, and 64-QAM constellation bit encoding**


Copyright © 2007 IEEE. All rights reserved.

For BPSK, $b_0$ determines the I value, as illustrated in Table 17-7. For QPSK, $b_0$ determines the I value and $b_1$ determines the Q value, as illustrated in Table 17-8. For 16-QAM, $b_0b_1$ determines the I value and $b_2b_3$ determines the Q value, as illustrated in Table 17-9. For 64-QAM, $b_0b_1b_2$ determines the I value and $b_3b_4b_5$ determines the Q value, as illustrated in Table 17-10.

**Table 17-7—BPSK encoding table**

| Input bit ($b_0$) | I-out | Q-out |
|---|---|---|
| 0 | –1 | 0 |
| 1 | 1 | 0 |

**Table 17-8—QPSK encoding table**

| Input bit ($b_0$) | I-out |
|---|---|
| 0 | –1 |
| 1 | 1 |

| Input bit ($b_1$) | Q-out |
|---|---|
| 0 | –1 |
| 1 | 1 |

**Table 17-9—16-QAM encoding table**

| Input bits ($b_0$ $b_1$) | I-out |
|---|---|
| 00 | –3 |
| 01 | –1 |
| 11 | 1 |
| 10 | 3 |

| Input bits ($b_2$ $b_3$) | Q-out |
|---|---|
| 00 | –3 |
| 01 | –1 |
| 11 | 1 |
| 10 | 3 |

**Table 17-10—64-QAM encoding table**

| Input bits ($b_0$ $b_1$ $b_2$) | I-out |
|---|---|
| 000 | –7 |
| 001 | –5 |
| 011 | –3 |
| 010 | –1 |
| 110 | 1 |
| 111 | 3 |
| 101 | 5 |
| 100 | 7 |

| Input bits ($b_3$ $b_4$ $b_5$) | Q-out |
|---|---|
| 000 | –7 |
| 001 | –5 |
| 011 | –3 |
| 010 | –1 |
| 110 | 1 |
| 111 | 3 |
| 101 | 5 |
| 100 | 7 |

Copyright © 2007 IEEE. All rights reserved.

### 17.3.5.8 Pilot subcarriers

In each OFDM symbol, four of the subcarriers are dedicated to pilot signals in order to make the coherent detection robust against frequency offsets and phase noise. These pilot signals shall be put in subcarriers –21, –7, 7, and 21. The pilots shall be BPSK modulated by a pseudo-binary sequence to prevent the generation of spectral lines. The contribution of the pilot subcarriers to each OFDM symbol is described in 17.3.5.9.

### 17.3.5.9 OFDM modulation

The stream of complex numbers is divided into groups of $N_{SD} = 48$ complex numbers. This shall be denoted by writing the complex number $d_{k,n}$, which corresponds to subcarrier k of OFDM symbol n, as follows:

$$d_{k,n} \equiv d_{k+N_{SD}\times n}, \qquad k = 0, \ldots N_{SD} - 1,\ n = 0, \ldots N_{SYM} - 1 \tag{17-21}$$

The number of OFDM symbols, $N_{SYM}$, was introduced in 17.3.5.3.

An OFDM symbol, $r_{DATA,n}(t)$, is defined as

$$r_{DATA,n}(t) = w_{TSYM}(t)\left(\sum_{k=0}^{N_{SD}-1} d_{k,n}\exp((j2\pi M(k)\Delta_F(t-T_{GI})) + p_{n+1}\sum_{k=-N_{ST}/2}^{N_{ST}/2} P_k\exp(j2\pi k\Delta_F(t-T_{GI}))\right) \tag{17-22}$$

where the function, $M(k)$, defines a mapping from the logical subcarrier number 0 to 47 into frequency offset index -26 to 26, while skipping the pilot subcarrier locations and the $0^{th}$ (dc) subcarrier.

$$M(k) = \begin{cases} k-26 & 0 \le k \le 4 \\ k-25 & 5 \le k \le 17 \\ k-24 & 18 \le k \le 23 \\ k-23 & 24 \le k \le 29 \\ k-22 & 30 \le k \le 42 \\ k-21 & 43 \le k \le 47 \end{cases} \tag{17-23}$$

The contribution of the pilot subcarriers for the $n^{th}$ OFDM symbol is produced by inverse Fourier transform of sequence P, given by

$$\begin{aligned} P_{-26,26} = \{&0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, \\ &0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0\} \end{aligned} \tag{17-24}$$


Copyright © 2007 IEEE. All rights reserved.

The polarity of the pilot subcarriers is controlled by the sequence, $p_n$, which is a cyclic extension of the 127 elements sequence and is given by

$$p_{0..126v} = \{1,1,1,1,\ -1,-1,-1,1,\ -1,-1,-1,-1,\ 1,1,-1,1,\ -1,-1,1,1,\ -1,1,1,-1,\ 1,1,1,1,\ 1,1,-1,1,$$
$$1,1,-1,1,\ 1,-1,-1,1,\ 1,1,-1,1,\ -1,-1,-1,1,\ -1,1,-1,-1,\ 1,-1,-1,1,\ 1,1,1,1,\ -1,-1,1,1,$$
$$-1,-1,1,-1,\ 1,-1,1,1,\ -1,-1,-1,1,\ 1,-1,-1,-1,\ -1,1,-1,-1,\ 1,-1,1,1,\ 1,1,-1,1,\ -1,1,-1,1,$$
$$-1,-1,-1,-1,\ -1,1,-1,1,\ 1,-1,1,-1,\ 1,1,1,-1,\ -1,1,-1,-1,\ -1,1,1,1,\ -1,-1,-1,-1,\ -1,-1,-1\} \quad (17\text{-}25)$$

The sequence $p_n$ can be generated by the scrambler defined by Figure 17-7 when the all ones initial state is used, and by replacing all 1's with –1 and all 0's with 1. Each sequence element is used for one OFDM symbol. The first element, $p_0$, multiplies the pilot subcarriers of the SIGNAL symbol, while the elements from $p_1$ on are used for the DATA symbols.

The subcarrier frequency allocation is shown in Figure 17-11. To avoid difficulties in D/A and A/D converter offsets and carrier feedthrough in the RF system, the subcarrier falling at DC ($0^{th}$ subcarrier) is not used.

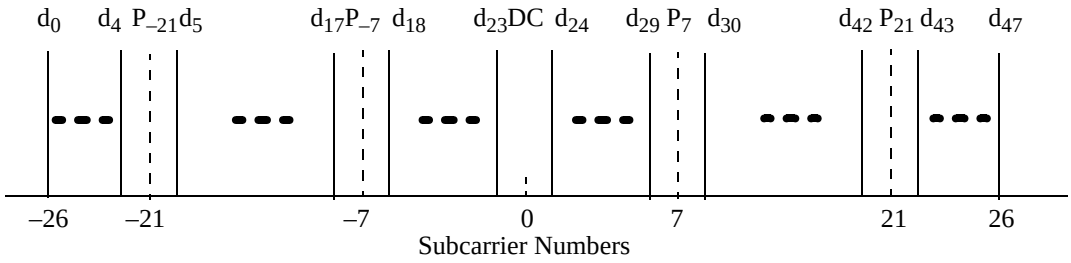



**Figure 17-11—Subcarrier frequency allocation**

The concatenation of $N_{SYM}$ OFDM symbols can now be written as

$$r_{DATA}(t) = \sum_{n=0}^{N_{SYM}-1} r_{DATA,n}(t - nT_{SYM}) \quad (17\text{-}26)$$

An example of mapping into symbols is shown in G.6.3, as well as the scrambling of the pilot signals (see G.7). The final output of these operations is also shown in G.8.

### 17.3.6 CCA

PLCP shall provide the capability to perform CCA and report the result to the MAC. The CCA mechanism shall detect a "medium busy" condition with a performance specified in 17.3.10.5. This medium status report is indicated by the primitive PHY_CCA.indicate.

### 17.3.7 PLCP data modulation and modulation rate change

The PLCP preamble shall be transmitted using an OFDM modulated fixed waveform. The IEEE 802.11 SIGNAL field, BPSK-OFDM modulated with coding rate 1/2, shall indicate the modulation and coding rate that shall be used to transmit the MPDU. The transmitter (receiver) shall initiate the modulation (demodulation) constellation and the coding rate according to the RATE indicated in the SIGNAL field. The MPDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 17.2.2.

Copyright © 2007 IEEE. All rights reserved.

### 17.3.8 PMD operating specifications (general)

General specifications for the BPSK OFDM, QPSK OFDM, 16-QAM OFDM, and 64-QAM OFDM PMD sublayers are provided in 17.3.8.1 through 17.3.8.8. These specifications apply to both the receive and transmit functions and general operation of the OFDM PHY.

#### 17.3.8.1 Outline description

The general block diagram of the transmitter and receiver for the OFDM PHY is shown in Figure 17-12. Major specifications for the OFDM PHY are listed in Table 17-11.

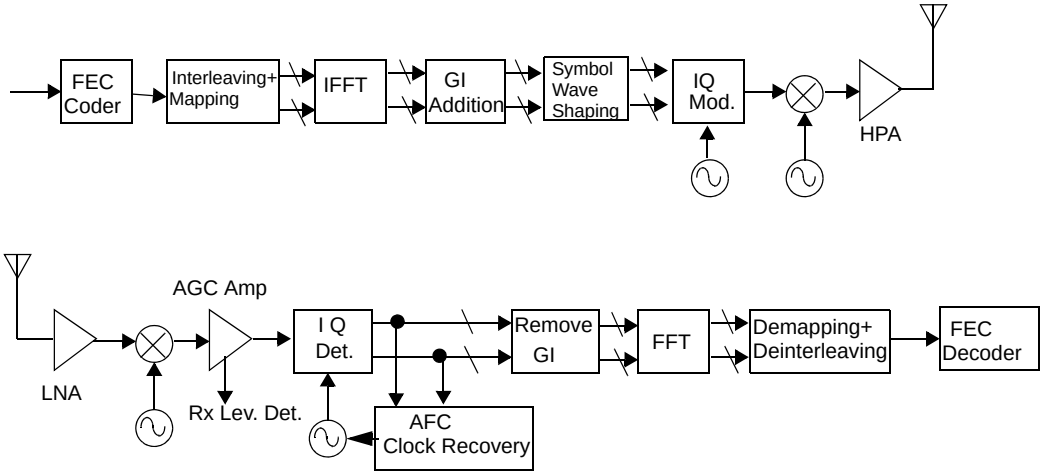



**Figure 17-12—Transmitter and receiver block diagram for the OFDM PHY**

**Table 17-11—Major parameters of the OFDM PHY**

| Information data rate | 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s (6, 12, and 24 Mb/s are mandatory) (20 MHz channel spacing) | 3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s (3, 6, and 12 Mb/s are mandatory) (10 MHz channel spacing) | 1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s (1.5, 3, and 6 Mb/s are mandatory) (5 MHz channel spacing) |
|---|---|---|---|
| Modulation | BPSK OFDM<br>QPSK OFDM<br>16-QAM OFDM<br>64-QAM OFDM | BPSK OFDM<br>QPSK OFDM<br>16-QAM OFDM<br>64-QAM OFDM | BPSK OFDM<br>QPSK OFDM<br>16-QAM OFDM<br>64-QAM OFDM |
| Error correcting code | K = 7 (64 states) convolutional code | K = 7 (64 states) convolutional code | K = 7 (64 states) convolutional code |
| Coding rate | 1/2, 2/3, 3/4 | 1/2, 2/3, 3/4 | 1/2, 2/3, 3/4 |
| Number of subcarriers | 52 | 52 | 52 |
| OFDM symbol duration | 4.0 μs | 8.0 μs | 16.0 μs |
| GI | 0.8 μs* ($T_{GI}$) | 1.6 μs ($T_{GI}$) | 3.2 μs ($T_{GI}$) |
| Occupied bandwidth | 16.6 MHz | 8.3 MHz | 4.15 MHz |

*Refer to 17.3.2.4.


Copyright © 2007 IEEE. All rights reserved.

#### 17.3.8.2 Regulatory requirements

WLANs implemented in accordance with this standard are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the regulatory sources in Annex I for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

#### 17.3.8.3 Operating channel frequencies

##### 17.3.8.3.1 Operating frequency range

The OFDM PHY shall operate in the 5 GHz band, as allocated by a regulatory body in its operational region. Spectrum allocation in the 5 GHz band is subject to authorities responsible for geographic-specific regulatory domains (e.g., global, regional, and national). The particular channelization to be used for this standard is dependent on such allocation, as well as the associated regulations for use of the allocations. These regulations are subject to revision, or may be superseded.

In some regulatory domains, several frequency bands may be available for OFDM PHY-based WLANs. These bands may be contiguous or not, and different regulatory limits may be applicable. A compliant OFDM PHY shall support at least one frequency band in at least one regulatory domain. The support of specific regulatory domains, and bands within the domains, shall be indicated by PLME attributes dot11RegDomainsSupported and dot11FrequencyBandsSupported.

##### 17.3.8.3.2 Channel numbering

Channel center frequencies are defined at every integral multiple of 5 MHz above Channel starting frequency. The relationship between center frequency and channel number is given by Equation (17-27):

$$\text{Channel center frequency} = \text{Channel starting frequency} + 5 \times n_{ch} \text{ (MHz)} \quad (17\text{-}27)$$

where

$n_{ch} = 0,1,\ldots 200$

Channel starting frequency is defined as dot11ChannelStartingFactor × 500 kHz or
is defined as 5 GHz for systems where dot11RegulatoryClassesRequired is false or not defined.

For example, dot11ChannelStartingFactor = 10000 indicates that Channel zero center frequency is 5.000 GHz. The value NULL for $n_{ch}$ shall be reserved, and a channel center frequency of 5.000 GHz shall be indicated by dot11ChannelStartingFactor = 8000 and $n_{ch}$ = 200. An SME managing multiple channel sets can change the channel set being managed by changing the value of dot11ChannelStartingFactor.

This definition provides a unique numbering system for all channels with 5 MHz between center frequencies, as well as the flexibility to define channelization sets for all current and future regulatory domains.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 661 of 1232

#### 17.3.8.3.3 Channelization

The set of valid operating channel numbers by regulatory domain is defined in Annex J. As shown in Figure 17-11, no subcarrier is allocated on the channel center frequency.

### 17.3.8.4 Transmit and receive in-band and out-of-band spurious emissions

The OFDM PHY shall conform to in-band and out-of-band spurious emissions as set by regulatory bodies.

### 17.3.8.5 TX RF delay

The TX RF delay time shall be defined as the time between the issuance of a PMD.DATA.request to the PMD and the start of the corresponding symbol at the air interface.

### 17.3.8.6 Slot time

The slot time for the OFDM PHY shall be 9 μs for 20 MHz channel spacing, shall be 13 μs for 10 MHz channel spacing, and shall be 21 μs for 5 MHz channel spacing.

Where dot11RegulatoryClassesRequired is true, the value of the slot time shall be increased by the value of 3 μs × coverage class. The default value of coverage class shall be zero.

NOTE—Distributed coordination function (DCF) operation over larger BSS diameters is facilitated by relaxing some PHY timing parameters, while maintaining compatibility with existing implementations in small BSS diameters.

### 17.3.8.7 Transmit and receive antenna port impedance

The transmit and receive antenna port(s) impedance shall be 50 Ω if the port is exposed.

### 17.3.8.8 Transmit and receive operating temperature range

Three temperature ranges for full operation compliance to the OFDM PHY are specified in Clause 13. Type 1, defined as 0 °C to 40 °C, is designated for office environments. Type 2, defined as –20 °C to 50 °C, and Type 3, defined as –30 °C to 70 °C, are designated for industrial environments.

## 17.3.9 PMD transmit specifications

The transmit specifications associated with the PMD sublayer are described in 17.3.9.1 through 17.3.9.7. In general, these are specified by primitives from the PLCP, and the transmit PMD entity provides the actual means by which the signals required by the PLCP primitives are imposed onto the medium.

### 17.3.9.1 Transmit power levels

The maximum allowable transmit power by regulatory domain is defined in Annex I.

### 17.3.9.2 Transmit spectrum mask

The transmit spectrum mask by regulatory domain is defined in Annex I and Annex J.

### 17.3.9.3 Transmission spurious

Spurious transmissions from compliant devices shall conform to national regulations.


Copyright © 2007 IEEE. All rights reserved.

#### 17.3.9.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±20 ppm maximum for 20 MHz and 10 MHz channels and shall be ±10 ppm maximum for 5 MHz channels. The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

#### 17.3.9.5 Symbol clock frequency tolerance

The symbol clock frequency tolerance shall be ±20 ppm maximum for 20 MHz and 10 MHz channels, and shall be ±10 ppm maximum for 5 MHz channels. The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

#### 17.3.9.6 Modulation accuracy

Transmit modulation accuracy specifications are described in this subclause. The test method is described in 17.3.9.7.

#### 17.3.9.6.1 Transmitter center frequency leakage

Certain transmitter implementations may cause leakage of the center frequency component. Such leakage (which manifests itself in a receiver as energy in the center frequency component) shall not exceed –15 dB relative to overall transmitted power or, equivalently, +2 dB relative to the average energy of the rest of the subcarriers. The data for this test shall be derived from the channel estimation phase.

#### 17.3.9.6.2 Transmitter spectral flatness

The average energy of the constellations in each of the spectral lines –16.. –1 and +1.. +16 will deviate no more than ± 2 dB from their average energy. The average energy of the constellations in each of the spectral lines –26.. –17 and +17.. +26 will deviate no more than +2/–4 dB from the average energy of spectral lines –16.. –1 and +1.. +16. The data for this test shall be derived from the channel estimation step.

#### 17.3.9.6.3 Transmitter constellation error

The relative constellation RMS error, averaged over subcarriers, OFDM frames, and packets, shall not exceed a data-rate dependent value according to Table 17-12.

**Table 17-12—Allowed relative constellation error versus data rate**

| Relative constellation error (dB) | Modulation | Coding rate (R) |
|---|---|---|
| –5 | BPSK | 1/2 |
| –8 | BPSK | 3/4 |
| –10 | QPSK | 1/2 |
| –13 | QPSK | 3/4 |
| –16 | 16-QAM | 1/2 |
| –19 | 16-QAM | 3/4 |
| –22 | 64-QAM | 2/3 |
| –25 | 64-QAM | 3/4 |

Copyright © 2007 IEEE. All rights reserved.

### 17.3.9.7 Transmit modulation accuracy test

The transmit modulation accuracy test shall be performed by instrumentation capable of converting the transmitted signal into a stream of complex samples at 20 Msample/s or more, with sufficient accuracy in terms of I/Q arm amplitude and phase balance, dc offsets, phase noise, etc. A possible embodiment of such a setup is converting the signal to a low IF frequency with a microwave synthesizer, sampling the signal with a digital oscilloscope and decomposing it digitally into quadrature components.

The sampled signal shall be processed in a manner similar to an actual receiver, according to the following steps, or an equivalent procedure:

a) Start of frame shall be detected.

b) Transition from short sequences to channel estimation sequences shall be detected, and fine timing (with one sample resolution) shall be established.

c) Coarse and fine frequency offsets shall be estimated.

d) The packet shall be derotated according to estimated frequency offset.

e) The complex channel response coefficients shall be estimated for each of the subcarriers.

f) For each of the data OFDM symbols: transform the symbol into subcarrier received values, estimate the phase from the pilot subcarriers, derotate the subcarrier values according to estimated phase, and divide each subcarrier value with a complex estimated channel response coefficient.

g) For each data-carrying subcarrier, find the closest constellation point and compute the Euclidean distance from it.

h) Compute the RMS average of all errors in a packet. It is given by

$$Error_{RMS} = \frac{\sum_{i=1}^{N_f} \sqrt{\frac{\sum_{j=1}^{L_P} \left[ \sum_{k=1}^{52} \{ (I(i,j,k) - I_0(i,j,k))^2 + (Q(i,j,k) - Q_0(i,j,k))^2 \} \right]}{52 L_P \times P_0}}}{N_f} \quad (17\text{-}28)$$

where

$L_P$ is the length of the packet;

$N_f$ is the number of frames for the measurement;

($I_0(i,j,k)$, $Q_0(i,j,k)$)denotes the ideal symbol point of the $i^{th}$ frame, $j^{th}$ OFDM symbol of the frame, $k^{th}$ sub-carrier of the OFDM symbol in the complex plane;

($I(i,j,k)$, $Q(i,j,k)$)denotes the observed point of the $i^{th}$ frame, $j^{th}$ OFDM symbol of the frame, $k^{th}$ subcarrier of the OFDM symbol in the complex plane (see Figure 17-13);

$P_0$ is the average power of the constellation.

The vector error on a phase plane is shown in Figure 17-13.

The test shall be performed over at least 20 frames ($N_f$), and the RMS average shall be taken. The packets under test shall be at least 16 OFDM symbols long. Random data shall be used for the symbols.


Copyright © 2007 IEEE. All rights reserved.

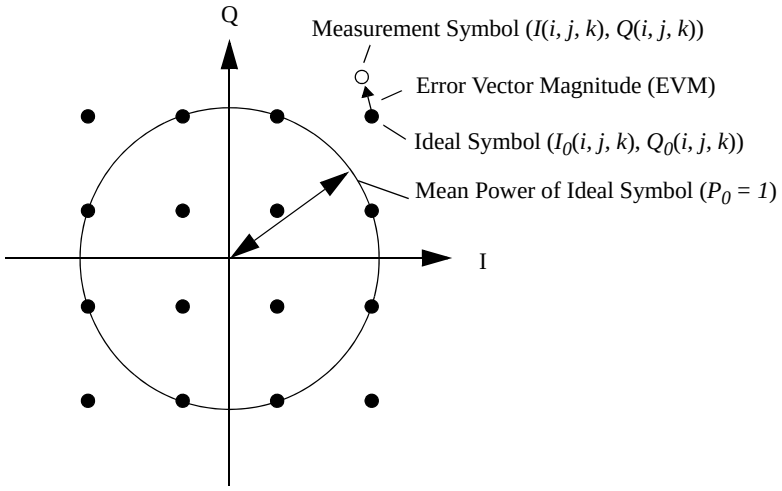



**Figure 17-13—Constellation error**

### 17.3.10 PMD receiver specifications

The receive specifications associated with the PMD sublayer are described in 17.3.10.1 through 17.3.10.5.

#### 17.3.10.1 Receiver minimum input sensitivity

The packet error rate (PER) shall be less than 10% at a PSDU length of 1000 octets for rate-dependent input levels shall be the numbers listed in Table 17-13 or less. The minimum input levels are measured at the antenna connector (noise factor of 10 dB and 5 dB implementation margins are assumed).

**Table 17-13—Receiver performance requirements**

| Modulation | Coding rate (R) | Adjacent channel rejection (dB) | Alternate adjacent channel rejection (dB) | Minimum sensitivity (dBm) (20 MHz channel spacing) | Minimum sensitivity (dBm) (10 MHz channel spacing) | Minimum sensitivity (dBm) (5 MHz channel spacing) |
|---|---|---|---|---|---|---|
| BPSK | 1/2 | 16 | 32 | –82 | –85 | –88 |
| BPSK | 3/4 | 15 | 31 | –81 | –84 | –87 |
| QPSK | 1/2 | 13 | 29 | –79 | –82 | –85 |
| QPSK | 3/4 | 11 | 27 | –77 | –80 | –83 |
| 16-QAM | 1/2 | 8 | 24 | –74 | –77 | –80 |
| 16-QAM | 3/4 | 4 | 20 | –70 | –73 | –76 |
| 64-QAM | 2/3 | 0 | 16 | –66 | –69 | –72 |
| 64-QAM | 3/4 | –1 | 15 | –65 | –68 | –71 |

Copyright © 2007 IEEE. All rights reserved.

#### 17.3.10.2 Adjacent channel rejection

The adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 17-13 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding adjacent channel rejection. The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conformant OFDM PHY the corresponding rejection shall be no less than specified in Table 17-13.

#### 17.3.10.3 Nonadjacent channel rejection

The nonadjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 17-13, and raising the power of the interfering signal until a 10% PER occurs for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding nonadjacent channel rejection. The interfering signal in the nonadjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For a conformed OFDM PHY, the corresponding rejection shall be no less than specified in Table 17-13.

#### 17.3.10.4 Receiver maximum input level

The receiver shall provide a maximum PER of 10% at a PSDU length of 1000 octets, for a maximum input level of –30 dBm measured at the antenna for any baseband modulation.

#### 17.3.10.5 CCA sensitivity

The start of a valid OFDM transmission at a receive level equal to or greater than the minimum modulation and coding rate sensitivity (–82 dBm for 20 MHz channel spacing, –85 dBm for 10 MHz channel spacing, and –88 dBm for 5 MHz channel spacing) shall cause CCA to indicate busy with a probability > 90% within 4 μs for 20 MHz channel spacing, 8 μs for 10 MHz channel spacing, and 16 μs for 5 MHz channel spacing. If the preamble portion was missed, the receiver shall hold the CS signal busy for any signal 20 dB above the minimum modulation and coding rate sensitivity (–62 dBm for 20 MHz channel spacing, –65 dBm for 10 MHz channel spacing, and –68 dBm for 5 MHz channel spacing).

NOTE—CCA detect time is based on finding the short sequences in the preamble, so when $T_{SYM}$ doubles, so does CCA detect time.

### 17.3.11 Transmit PLCP

The transmit PLCP is shown in Figure 17-14. In order to transmit data, PHY-TXSTART.request shall be enabled so that the PHY entity shall be in the transmit state. Further, the PHY shall be set to operate at the appropriate frequency through STA management via the PLME. Other transmit parameters, such as DATARATE and TX power, are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR), as described in 17.2.2.

A clear channel shall be indicated by PHY-CCA.indicate(IDLE). The MAC considers this indication before issuing the PHY-TXSTART.request. Transmission of the PPDU shall be initiated after receiving the PHY-TXSTART.request(TXVECTOR) primitive. The TXVECTOR elements for the PHY-TXSTART.request are the PLCP header parameters DATARATE, SERVICE, and LENGTH, and the PMD parameter TXPWR_LEVEL.

The PLCP shall issue PMD_TXPWRLVL and PMD_RATE primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request, and transmission of the PLCP preamble and PLCP header, based on the parameters passed in the PHY-TXSTART.request primitive. Once PLCP preamble transmission is started, the PHY entity shall immediately initiate data scrambling and data encoding. The scrambled and encoded data shall then be exchanged between the MAC and the PHY through a series of


Copyright © 2007 IEEE. All rights reserved.

PHY-DATA.request(DATA) primitives issued by the MAC, and PHY-DATA.confirm primitives issued by the PHY. The modulation rate change, if any, shall be initiated from the SERVICE field data of the PLCP header, as described in 17.3.2.

The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. The PLCP header parameter, SERVICE, and PSDU are encoded by the convolutional encoder with the bit-stealing function described in 17.3.5.5. At the PMD layer, the data octets are sent in bit 0–7 order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request. Normal termination occurs after the transmission of the final bit of the last PSDU octet, according to the number supplied in the OFDM PHY preamble LENGTH field.

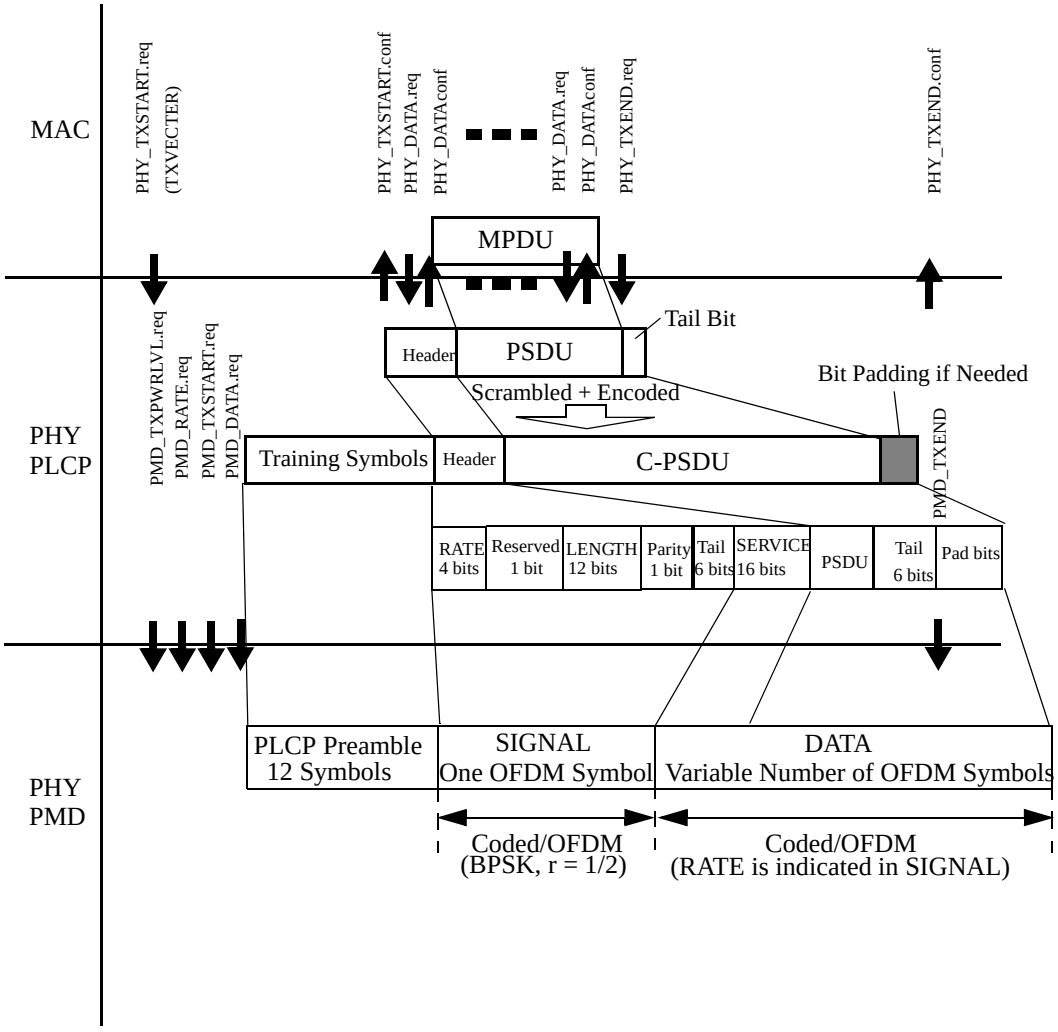



**Figure 17-14—Transmit PLCP**

The packet transmission shall be completed and the PHY entity shall enter the receive state (i.e., PHY-TXSTART shall be disabled). Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY. If the coded PSDU (C-PSDU) is not multiples of the OFDM symbol, bits shall be stuffed to make the C-PSDU length multiples of the OFDM symbol.

Copyright © 2007 IEEE. All rights reserved.

In the PMD, the GI shall be inserted in every OFDM symbol as a countermeasure against severe delay spread.

A typical state machine implementation of the transmit PLCP is provided in Figure 17-15. Requests (.req) and confirmations(.confirm) are issued once with designated states.

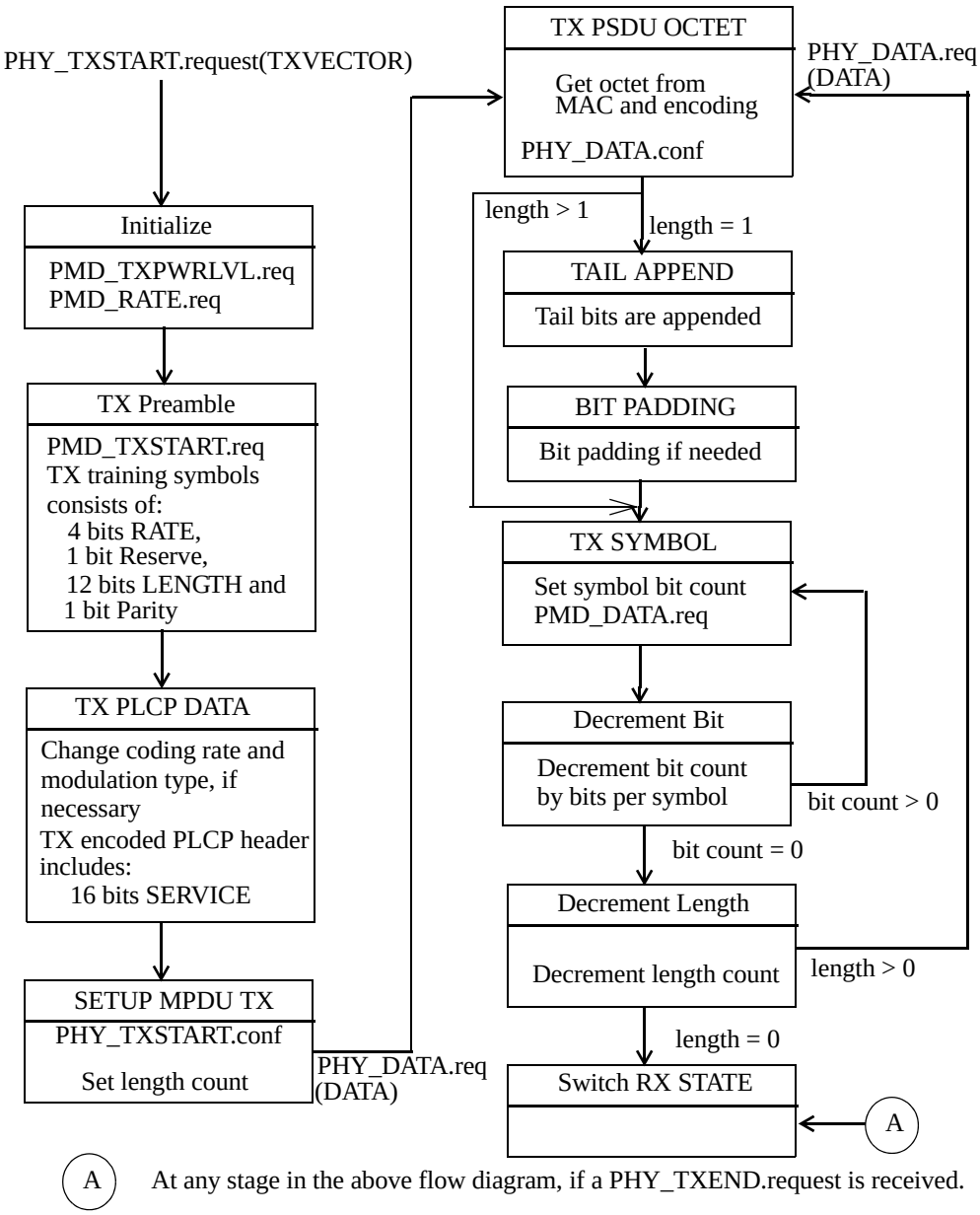



**Figure 17-15—PLCP transmit state machine**


Copyright © 2007 IEEE. All rights reserved.

### 17.3.12 Receive PLCP

The receive PLCP is shown in Figure 17-16. In order to receive data, PHY-TXSTART.request shall be disabled so that the PHY entity is in the receive state. Further, through STA management (via the PLME) the PHY is set to the appropriate frequency. Other receive parameters, such as RSSI and indicated DATARATE, may be accessed via the PHY-SAP.

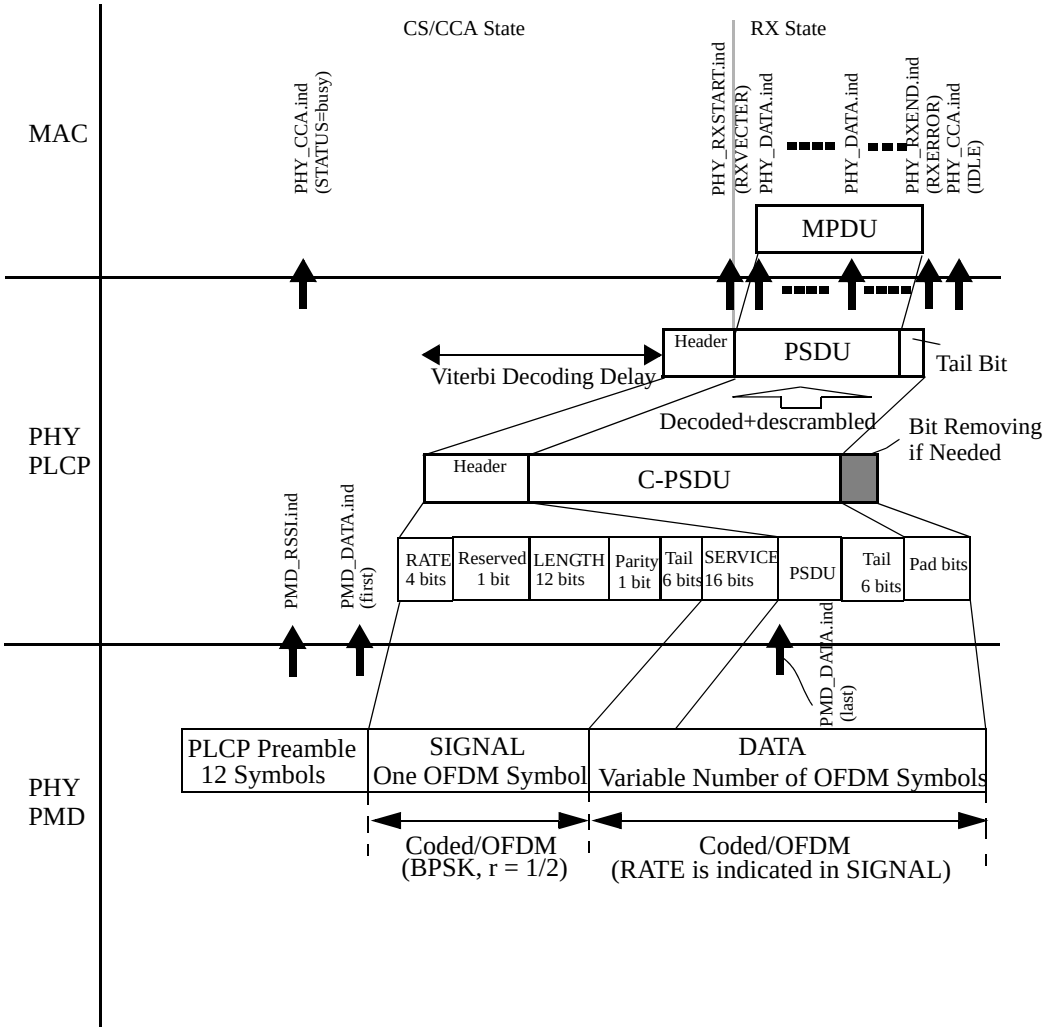



**Figure 17-16—Receive PLCP**

Upon receiving the transmitted PLCP preamble, PMD_RSSI.indicate shall report a significant received signal strength level to the PLCP. This indicates activity to the MAC via PHY_CCA.indicate. PHY_CCA.indicate(BUSY) shall be issued for reception of a signal prior to correct reception of the PLCP frame. The PMD primitive PMD_RSSI is issued to update the RSSI and parameter reported to the MAC.

After PHY-CCA.indicate is issued, the PHY entity shall begin receiving the training symbols and searching for the SIGNAL in order to set the length of the data stream, the demodulation type, and the decoding rate. Once the SIGNAL is detected, without any errors detected by a single parity (even), FEC decode shall be

Copyright © 2007 IEEE. All rights reserved.

initiated and the PLCP IEEE 802.11 SERVICE fields and data shall be received, decoded (a Viterbi decoder is recommended), and checked by ITU-T CRC-32. If the FCS by the ITU-T CRC-32 check fails, the PHY receiver shall return to the RX IDLE state, as depicted in Figure 17-16. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indicate(RXVECTOR) shall be issued. The RXVECTOR associated with this primitive includes the SIGNAL field, the SERVICE field, the PSDU length in octets, and the RSSI. Also, in this case, the OFDM PHY will ensure that the CCA shall indicate a busy medium for the intended duration of the transmitted frame, as indicated by the LENGTH field.

The received PSDU bits are assembled into octets, decoded, and presented to the MAC using a series of PHY-DATA.indicate(DATA) primitive exchanges. The rate change indicated in the IEEE 802.11 SIGNAL field shall be initiated from the SERVICE field data of the PLCP header, as described in 17.3.2. The PHY shall proceed with PSDU reception. After the reception of the final bit of the last PSDU octet indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state, as shown in Figure 17-16. A PHY-RXEND.indicate(NoError) primitive shall be issued.

In the event that a change in the RSSI causes the status of the CCA to return to the IDLE state before the complete reception of the PSDU, as indicated by the PLCP LENGTH field, the error condition PHY-RXEND.indicate(CarrierLost) shall be reported to the MAC. The OFDM PHY will ensure that the CCA indicates a busy medium for the intended duration of the transmitted packet.

If the indicated rate in the SIGNAL field is not receivable, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(UnsupportedRate). If the PLCP header is receivable, but the parity check of the PLCP header is not valid, a PHY-RXSTART.indicate will not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(FormatViolation).

Any data received after the indicated data length are considered pad bits (to fill out an OFDM symbol) and should be discarded.

A typical state machine implementation of the receive PLCP is given in Figure 17-17.

## 17.4 OFDM PLME

### 17.4.1 PLME_SAP sublayer management primitives

Table 17-14 lists the MIB attributes that may be accessed by the PHY entities and the intralayer of higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, PLME-RESET, and PLME-CHARACTERISTICS primitives defined in 10.4.

### 17.4.2 OFDM PHY MIB

All OFDM PHY MIB attributes are defined in Clause 13, with specific values defined in Table 17-14. The column titled “Operational semantics” in Table 17-14 contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. Dynamic MIB attributes can be modified by some management entity.


Copyright © 2007 IEEE. All rights reserved.

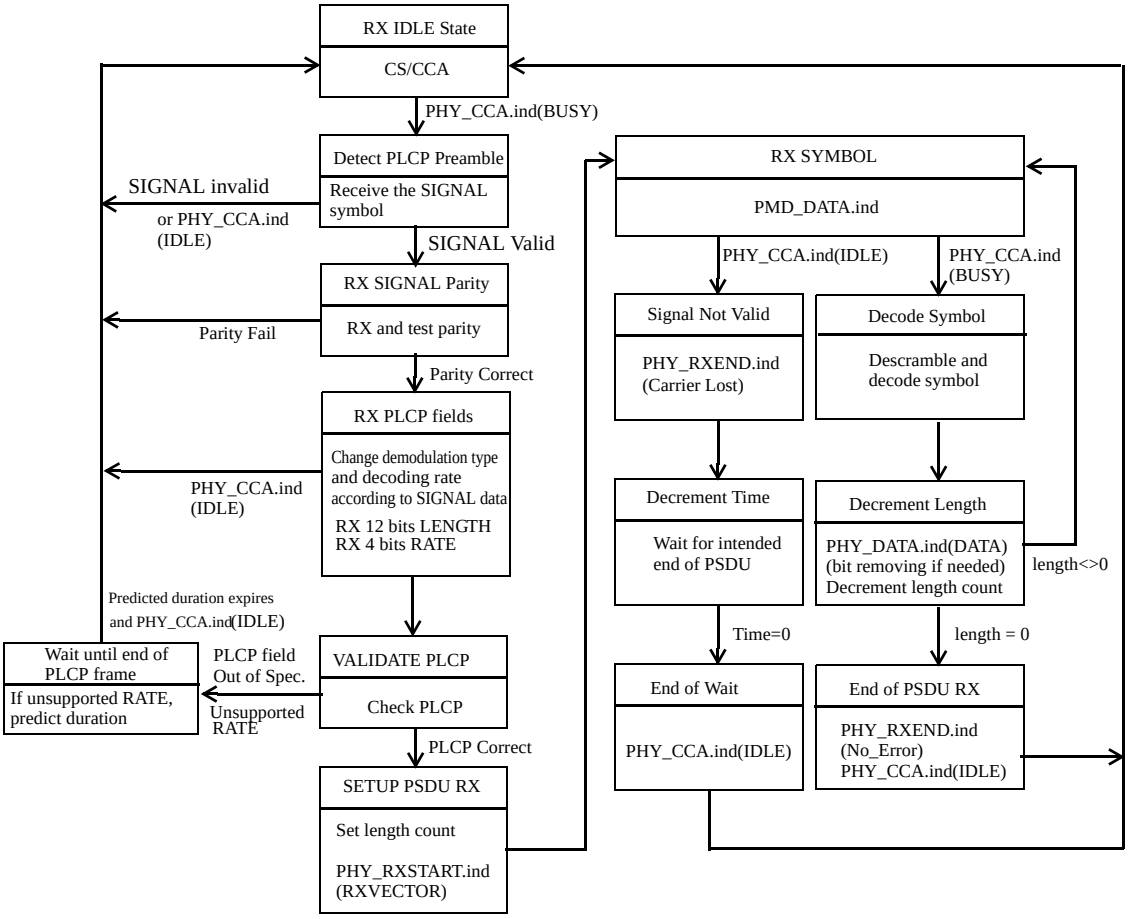



**Figure 17-17—PLCP receive state machine**

**Table 17-14—MIB attribute default values/ranges**

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11 PHY Operation Table** | | |
| dot11 PHY type | OFDM-5. (04) | Static |
| dot11 Current reg domain | Implementation dependent | Dynamic |
| dot11 Current frequency band | Implementation dependent | Dynamic |
| dot11 Temp type | Implementation dependent | Static |
| **dot11 PHY Antenna Table** | | |
| dot11 Current Tx antenna | Implementation dependent | Dynamic |
| dot11 Diversity support | Implementation dependent | Static |
| dot11 Current Rx antenna | Implementation dependent | Dynamic |

Copyright © 2007 IEEE. All rights reserved.

**Table 17-14—MIB attribute default values/ranges *(continued)***

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11 PHY Tx Power Table** | | |
| dot11 Number supported power levels | Implementation dependent | Static |
| dot11 Tx power level 1 | Implementation dependent | Static |
| dot11 Tx power level 2 | Implementation dependent | Static |
| dot11 Tx power level 3 | Implementation dependent | Static |
| dot11 Tx power level 4 | Implementation dependent | Static |
| dot11 Tx power level 5 | Implementation dependent | Static |
| dot11 Tx power level 6 | Implementation dependent | Static |
| dot11 Tx power level 7 | Implementation dependent | Static |
| dot11 Tx power level 8 | Implementation dependent | Static |
| dot11 current Tx Power Level | Implementation dependent | Dynamic |
| **dot11 Reg Domains Supported Table** | | |
| dot11 Reg domains supported | Implementation dependent | Static |
| dot11 Frequency bands supported | Implementation dependent | Static |
| **dot11 PHY Antennas List Table** | | |
| dot 11 Supported Tx antenna | Implementation dependent | Static |
| dot11 Supported Rx antenna | Implementation dependent | Static |
| dot 11 Diversity selection Rx | Implementation dependent | Dynamic |
| **dot11 Supported Data Rates Tx Table** | | |
| dot11 Supported data rates Tx value | 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Mandatory rates: 6, 12, and 24)<br><br>3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Mandatory rates: 3, 6, and 12)<br><br>1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Mandatory rates: 1.5, 3, and 6) | Static |


Copyright © 2007 IEEE. All rights reserved.

**Table 17-14—MIB attribute default values/ranges *(continued)***

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11SupportedDataRatesRxTable** | | |
| dot11 Supported data rates Rx value | 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s for 20 MHz channel spacing (Mandatory rates: 6, 12, and 24)<br><br>3, 4.5, 6, 9, 12, 18, 24, and 27 Mb/s for 10 MHz channel spacing (Mandatory rates: 3, 6, and 12)<br><br>1.5, 2.25, 3, 4.5, 6, 9, 12, and 13.5 Mb/s for 5 MHz channel spacing (Mandatory rates: 1.5, 3, and 6) | Static |
| **dot11 PHY OFDM Table** | | |
| dot11 Current frequency | Implementation dependent | Dynamic |
| dot11 TI threshold | Implementation dependent | Dynamic |
| dot11 Channel starting factor | Implementation dependent | Dynamic |

### 17.4.3 OFDM TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive shall be calculated according to the following equation:

$$\text{TXTIME} = T_{PREAMBLE} + T_{SIGNAL} + T_{SYM} \times \text{Ceiling}((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}) \quad (17\text{-}29)$$

where

$N_{DBPS}$ is derived from the DATARATE parameter. (Ceiling is a function that returns the smallest integer value greater than or equal to its argument value.)

$N_{SYM}$ is given by Equation (17-11).

A simplified equation may be used.

$$\text{TXTIME} = T_{PREAMBLE} + T_{SIGNAL} + (16 + 8 \times \text{LENGTH} + 6)/\text{DATARATE} + T_{SYM}/2 \quad (17\text{-}30)$$

Equation (17-30) does not include the effect of rounding to the next OFDM symbol and may be in error by $T_{SYM}/2$.

### 17.4.4 OFDM PHY characteristics

The static OFDM PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 17-15. The definitions for these characteristics are given in 10.4.

Copyright © 2007 IEEE. All rights reserved.

**Table 17-15—OFDM PHY characteristics**

| Characteristics | Value (20 MHz channel spacing) | Value (10 MHz channel spacing) | Value (5 MHz channel spacing) |
|---|---|---|---|
| aSlotTime | 9 μs | 13 μs | 21 μs |
| aSIFSTime | 16 μs | 32 μs | 64 μs |
| aCCATime | < 4 μs | < 8 μs | < 16 μs |
| aPHY-RX-START-Delay | 25 μs | 49 μs | 97 μs |
| aRxTxTurnaroundTime | < 2 μs | < 2 μs | < 2 μs |
| aTxPLCPDelay | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxPLCPDelay | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. |
| aRxTxSwitchTime | << 1 μs | << 1 μs | << 1 μs |
| aTxRampOnTime | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. |
| aTxRampOffTime | Implementation dependent as long as the requirements of aSIFSTime are met. | Implementation dependent as long as the requirements of aSIFSTime are met. | Implementation dependent as long as the requirements of aSIFSTime are met. |
| aTxRFDelay | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. | Implementation dependent as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxRFDelay | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. | Implementation dependent as long as the requirements of aSIFSTime and aCCA-Time are met. |
| aAirPropagationTime | << 1 μs | << 1 μs | << 1 μs |
| aMACProcessingDelay | < 2 μs | < 2 μs | < 2 μs |
| aPreambleLength | 16 μs | 32 μs | 64 μs |
| aPLCPHeaderLength | 4 μs | 8 μs | 16 μs |
| aMPDUMaxLength | 4095 | 4095 | 4095 |
| aCWmin | 15 | 15 | 15 |
| aCWmax | 1023 | 1023 | 1023 |


Copyright © 2007 IEEE. All rights reserved.

## 17.5 OFDM PMD sublayer

### 17.5.1 Scope and field of application

This subclause describes the PMD services provided to the PLCP for the OFDM PHY. Also defined in this subclause are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire OFDM PHY is shown in Figure 17-18.

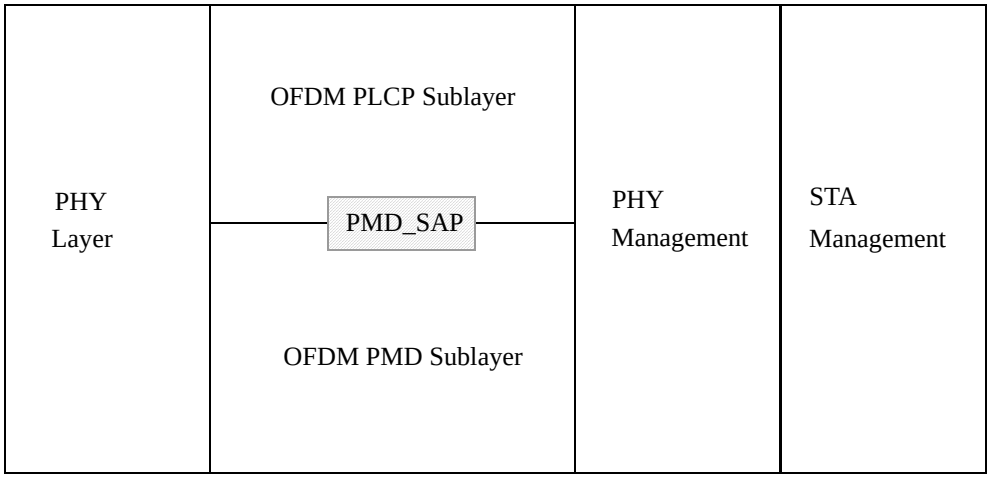



**Figure 17-18—PMD layer reference model**

### 17.5.2 Overview of service

The OFDM PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined function of the OFDM PMD sublayer primitives and parameters for the receive function results in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality shall be provided for data transmission.

### 17.5.3 Overview of interactions

The primitives associated with the IEEE 802.11 PLCP sublayer to the OFDM PMD fall into two basic categories

a) Service primitives that support PLCP peer-to-peer interactions;

b) Service primitives that have local significance and support sublayer-to-sublayer interactions.

### 17.5.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory, unless otherwise specified.

#### 17.5.4.1 PMD_SAP peer-to-peer service primitives

Table 17-16 indicates the primitives for peer-to-peer interactions.

Copyright © 2007 IEEE. All rights reserved.

**Table 17-16—PMD_SAP peer-to-peer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_DATA | X | X | — | — |

#### 17.5.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 17-17 indicates the primitives for sublayer-to-sublayer interactions.

**Table 17-17—PMD_SAP sublayer-to-sublayer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_TXSTART | X | — | — | — |
| PMD_TXEND | X | — | — | — |
| PMD_TXPWRLVL | X | — | — | — |
| PMD_RATE | X | — | — | — |
| PMD_RSSI | — | X | — | — |

#### 17.5.4.3 PMD_SAP service primitive parameters

Table 17-18 shows the parameters used by one or more of the PMD_SAP service primitives.

**Table 17-18—List of parameters for the PMD primitives**

| Parameter | Associate primitive | Value (20 MHz channel spacing) | Value (10 MHz channel spacing) | Value (5 MHz channel spacing) |
|---|---|---|---|---|
| TXD_UNIT | PMD_DATA.request | One(1), Zero(0): one OFDM symbol value | One(1),Zero(0): one OFDM symbol value | One(1),Zero(0): one OFDM symbol value |
| RXD_UNIT | PMD_DATA.indicate | One(1), Zero(0): one OFDM symbol value | One(1),Zero(0): one OFDM symbol value | One(1),Zero(0): one OFDM symbol value |
| TXPWR_LEVEL | PMD_TXPWRLVL .request | 1–8 (max of 8 levels) | 1–8 (max of 8 levels) | 1–8 (max of 8 levels) |
| RATE | PMD_RATE.request | 12 Mb/s (for BPSK)<br>24 Mb/s (for QPSK)<br>48 Mb/s (for 16-QAM)<br>72 Mb/s (for 64-QAM) | 6 Mb/s (for BPSK)<br>12 Mb/s (for QPSK)<br>24 Mb/s (for 16-QAM)<br>36 Mb/s (for 64-QAM) | 3 Mb/s (for BPSK)<br>6 Mb/s (for QPSK)<br>12Mb/s (for 16-QAM)<br>18 Mb/s (for 64-QAM) |
| RSSI | PMD_RSSI.indicate | 0–8 bits of RSSI | 0–8 bits of RSSI | 0–8 bits of RSSI |


Copyright © 2007 IEEE. All rights reserved.

### 17.5.5 PMD_SAP detailed service specification

This subclause describes the services provided by each PMD primitive.

#### 17.5.5.1 PMD_DATA.request

##### 17.5.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

##### 17.5.5.1.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_DATA.request(TXD_UNIT)

The TXD_UNIT parameter shall be the n-bit combination of 0 and 1 for one symbol of OFDM modulation. If the length of a coded MPDU (C-MPDU) is shorter than n bits, 0 bits are added to form an OFDM symbol. This parameter represents a single block of data which, in turn, shall be used by the PHY to be encoded into an OFDM transmitted symbol.

##### 17.5.5.1.3 When generated

This primitive shall be generated by the PLCP sublayer to request transmission of one OFDM symbol. The data clock for this primitive shall be supplied by the PMD layer based on the OFDM symbol clock.

##### 17.5.5.1.4 Effect of receipt

The PMD performs transmission of the data.

Copyright © 2007 IEEE. All rights reserved.

#### 17.5.5.2 PMD_DATA.indicate

##### 17.5.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

##### 17.5.5.2.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_DATA.indicate(RXD_UNIT)

The RXD_UNIT parameter shall be 0 or 1, and shall represent either a signal field bit or a data field bit after the decoding of the convolutional code by the PMD entity.

##### 17.5.5.2.3 When generated

This primitive, generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive shall be supplied by the PMD layer based on the OFDM symbol clock.

##### 17.5.5.2.4 Effect of receipt

The PLCP sublayer interprets the bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the MPDU.


Copyright © 2007 IEEE. All rights reserved.

#### 17.5.5.3 PMD_TXSTART.request

##### 17.5.5.3.1 Function

This primitive, generated by the PHY PLCP sublayer, initiates PPDU transmission by the PMD layer.

##### 17.5.5.3.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_TXSTART.request

##### 17.5.5.3.3 When generated

This primitive shall be generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-TXSTART.request primitive shall be provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

##### 17.5.5.3.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

Copyright © 2007 IEEE. All rights reserved.

#### 17.5.5.4 PMD_TXEND.request

##### 17.5.5.4.1 Function

This primitive, generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

##### 17.5.5.4.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_TXEND.request

##### 17.5.5.4.3 When generated

This primitive shall be generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

##### 17.5.5.4.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.


Copyright © 2007 IEEE. All rights reserved.

#### 17.5.5.5 PMD_TXPWRLVL.request

##### 17.5.5.5.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

##### 17.5.5.5.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_TXPWRLVL.request(TXPWR_LEVEL)

TXPWR_LEVEL selects which of the transmit power levels should be used for the current packet transmission. The number of available power levels shall be determined by the MIB parameter aNumberSupportedPowerLevels. See 17.3.9.1 for further information on the OFDM PHY power level control capabilities.

##### 17.5.5.5.3 When generated

This primitive shall be generated by the PLCP sublayer to select a specific transmit power. This primitive shall be applied prior to setting PMD_TXSTART into the transmit state.

##### 17.5.5.5.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level to that given by TXPWR_LEVEL.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 681 of 1232

#### 17.5.5.6 PMD_RATE.request

##### 17.5.5.6.1 Function

This primitive, generated by the PHY PLCP sublayer, selects the modulation rate that shall be used by the OFDM PHY for transmission.

##### 17.5.5.6.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_RATE.request(RATE)

RATE selects which of the OFDM PHY data rates shall be used for MPDU transmission. See 17.3.8.6 for further information on the OFDM PHY modulation rates. The OFDM PHY rate change capability is described in detail in 17.3.7.

##### 17.5.5.6.3 When generated

This primitive shall be generated by the PLCP sublayer to change or set the current OFDM PHY modulation rate used for the MPDU portion of a PPDU.

##### 17.5.5.6.4 Effect of receipt

The receipt of PMD_RATE selects the rate that shall be used for all subsequent MPDU transmissions. This rate shall be used for transmission only. The OFDM PHY shall still be capable of receiving all the required OFDM PHY modulation rates.


Copyright © 2007 IEEE. All rights reserved.

#### 17.5.5.7 PMD_RSSI.indicate

##### 17.5.5.7.1 Function

This primitive, generated by the PMD sublayer, provides the receive signal strength to the PLCP and MAC entity.

##### 17.5.5.7.2 Semantic of the service primitive

This primitive shall provide the following parameters:
PMD_RSSI.indicate(RSSI)

The RSSI shall be a measure of the RF energy received by the OFDM PHY. RSSIs of up to 8 bits (256 levels) are supported.

##### 17.5.5.7.3 When generated

This primitive shall be generated by the PMD when the OFDM PHY is in the receive state. It shall be available continuously to the PLCP which, in turn, shall provide the parameter to the MAC entity.

##### 17.5.5.7.4 Effect of receipt

This parameter shall be provided to the PLCP layer for information only. The RSSI may be used as part of a CCA scheme.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.
**Samsung Ex. 1044, Page 684 of 1232**

# 18. High Rate direct sequence spread spectrum (HR/DSSS) PHY specification

## 18.1 Overview

This clause specifies the High Rate extension of the PHY for the DSSS system (see Clause 15), hereinafter known as the High Rate PHY for the 2.4 GHz band designated for ISM applications.

This extension of the DSSS system builds on the data rate capabilities, as described in Clause 15, to provide 5.5 Mb/s and 11 Mb/s payload data rates in addition to the 1 Mb/s and 2 Mb/s rates. To provide the higher rates, 8-chip complementary code keying (CCK) is employed as the modulation scheme. The chipping rate is 11 MHz, which is the same as the DSSS system described in Clause 15, thus providing the same occupied channel bandwidth. The basic new capability described in this clause is called HR/DSSS. The basic High Rate PHY uses the same PLCP preamble and header as the DSSS PHY, so both PHYs can co-exist in the same BSS and can use the rate switching mechanism as provided.

In addition to providing higher speed extensions to the DSSS system, a number of optional features allow the performance of the RF LAN system to be improved as technology allows the implementation of these options to become cost effective.

An optional mode replacing the CCK modulation with HR/DSSS/PBCC is provided.

Another optional mode is provided that allows data throughput at the higher rates (2, 5.5, and 11 Mb/s) to be significantly increased by using a shorter PLCP preamble. This mode is called HR/DSSS/short, or HR/DSSS/PBCC/short. This short preamble mode can coexist with DSSS, HR/DSSS, or HR/DSSS/PBCC under limited circumstances, such as on different channels or with appropriate CCA mechanisms.

An optional capability for Channel Agility is also provided. This option allows an implementation to overcome some inherent difficulty with static channel assignments (a tone jammer), without burdening all implementations with the added cost of this capability. This option can also be used to implement IEEE 802.11-compliant systems that are interoperable with both FH and DS modulations. See Annex F for more details.

### 18.1.1 Scope

This clause specifies the PHY entity for the HR/DSSS extension and explains how this standard accommodates the High Rate PHY.

The High Rate PHY consists of the following two protocol functions:

a) A PHY convergence function, which adapts the capabilities of the PMD system to the PHY service. This function is supported by the PLCP, which defines a method for mapping the MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system. The PHY exchanges PPDUs that contain PSDUs. The MAC uses the PHY service, so each MPDU corresponds to a PSDU that is carried in a PPDU.

b) A PMD system, whose function defines the characteristics of, and method of transmitting and receiving data through, a WM between two or more STAs, each using the High Rate PHY system.

### 18.1.2 High Rate PHY functions

The 2.4 GHz High Rate PHY architecture is depicted in the ISO/IEC basic reference model shown in Figure 18-11 (in 18.4.1). The High Rate PHY contains three functional entities: the PMD function, the PHY

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 685 of 1232

convergence function, and the layer management function. Each of these functions is described in detail in 18.1.2.1, 18.1.2.2, and 18.1.2.3. For the purposes of MAC and MAC management, when Channel Agility is both present and enabled (see 18.3.2 and Annex C), the High Rate PHY shall be interpreted to be both a High Rate and an FH PHY.

The High Rate PHY service shall be provided to the MAC through the PHY service primitives described in Clause 12.

#### 18.1.2.1 PLCP sublayer

To allow the MAC to operate with minimum dependence on the PMD sublayer, a PLCP sublayer is defined. This function simplifies the PHY service interface to the MAC services.

#### 18.1.2.2 PMD sublayer

The PMD sublayer provides a means and method of transmitting and receiving data through a WM between two or more STAs, each using the High Rate system.

#### 18.1.2.3 PLME

The PLME performs management of the local PHY functions in conjunction with the MLME.

### 18.1.3 Service specification method and notation

The models represented by figures and state diagrams are intended to be illustrations of functions provided. It is important to distinguish between a model and a real implementation. The models are optimized for simplicity and clarity of presentation; the actual method of implementation is left to the discretion of the High-Rate-PHY-compliant developer.

The service of a layer or sublayer is a set of capabilities that it offers to a user in the next-higher layer (or sublayer). Abstract services are specified here by describing the service primitives and parameters that characterize each service. This definition is independent of any particular implementation.

## 18.2 High Rate PLCP sublayer

### 18.2.1 Overview

This subclause provides a convergence procedure for the 2 Mb/s, 5.5 Mb/s, and 11 Mb/s specification, in which PSDUs are converted to and from PPDUs. During transmission, the PSDU shall be appended to a PLCP preamble and header to create the PPDU. Two different preambles and headers are defined: the mandatory supported long preamble and header, which interoperates with the current 1 Mb/s and 2 Mb/s DSSS specification, and an optional short preamble and header. At the receiver, the PLCP preamble and header are processed to aid in demodulation and delivery of the PSDU.

The optional short preamble and header is intended for applications where maximum throughput is desired and interoperability with legacy and nonshort-preamble-capable equipment is not a consideration. That is, it is expected to be used only in networks of like equipment, which can all handle the optional mode.

### 18.2.2 PPDU format

Two different preambles and headers are defined: the mandatory supported long preamble and header, which is interoperable with the current 1 Mb/s and 2 Mb/s DSSS specification, and an optional short preamble and header.


Copyright © 2007 IEEE. All rights reserved.

### 18.2.2.1 Long PPDU format

Figure 18-1 shows the format for the interoperable (long) PPDU, including the High Rate PLCP preamble, the High Rate PLCP header, and the PSDU. The PLCP preamble contains the following fields: SYNC and SFD. The PLCP header contains the following fields: signaling (SIGNAL), service (SERVICE), length (LENGTH), and CRC-16. Each of these fields is described in detail in 18.2.3. The format for the PPDU, including the long High Rate PLCP preamble, the long High Rate PLCP header, and the PSDU, does not differ from the format for 1 Mb/s and 2 Mb/s. The only exceptions are

a) The encoding of the rate in the SIGNAL field;

b) The use of a bit in the SERVICE field to resolve an ambiguity in PSDU length in octets, when the length is expressed in whole microseconds;

c) The use of a bit in the SERVICE field to indicate if the optional PBCC mode is being used;

d) The use of a bit in the SERVICE field to indicate that the transit frequency and bit clocks are locked.

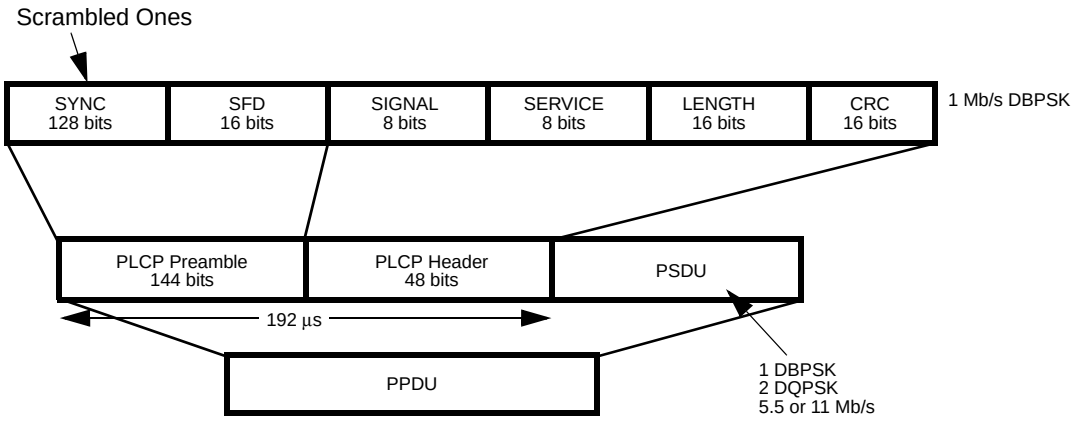



**Figure 18-1—Long PPDU format**

### 18.2.2.2 Short PPDU format

The short PLCP preamble and header (HR/DSSS/short) is defined as optional for HR/DSSS. The short preamble and header may be used to minimize overhead and, thus, maximize the network data throughput. The format of the PPDU, with HR/DSSS/short, is depicted in Figure 18-2. For Clause 19 STAs support of this preamble type is mandatory.

A transmitter using the short PLCP will only be interoperable with another receiver that is also capable of receiving this short PLCP. To interoperate with a receiver that is not capable of receiving a short preamble and header, the transmitter shall use the long PLCP preamble and header. The short PLCP preamble uses the 1 Mb/s Barker code spreading with DBPSK modulation. The short PLCP header uses the 2 Mb/s Barker code spreading with DQPSK modulation, and the PSDU is transmitted at 2 Mb/s, 5.5 Mb/s, or 11 Mb/s.

STAs not implementing this option that do active scanning will get a response even when the network is using Short Preambles, because all management traffic is returned with the same type preamble as received.

### 18.2.3 PPDU field definitions

In the PLCP field definition subclauses (18.2.3.1 through 18.2.3.14), the definitions of the long (Clause 15) PLCP fields are given first, followed by the definitions of the short PLCP. The names for the short PLCP fields are preceded by the term *short*.

Copyright © 2007 IEEE. All rights reserved.

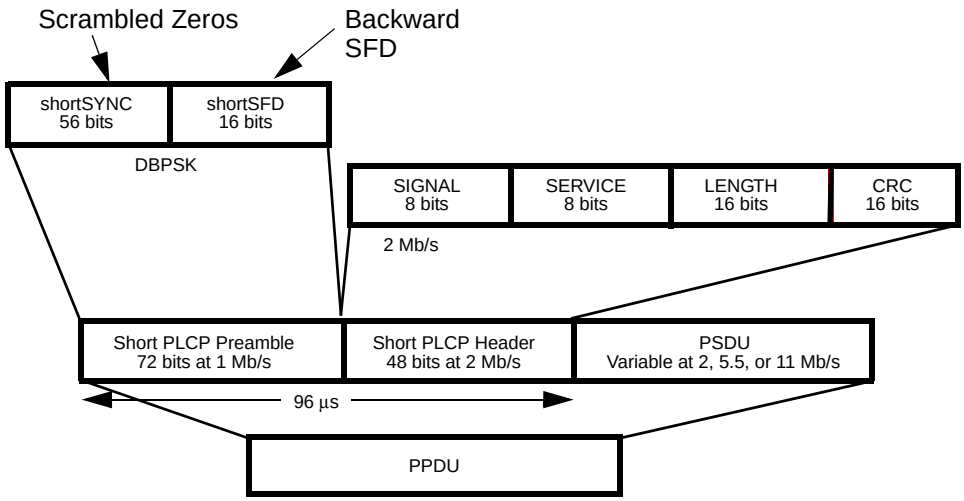



**Figure 18-2—Short PPDU format**

### 18.2.3.1 Long PLCP SYNC field

The SYNC field shall consist of 128 bits of scrambled 1 bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [1101100], where the leftmost bit specifies the value to put in the first delay element ($Z^1$) in Figure 18-5 (in 18.2.4), and the rightmost bit specifies the value to put in the last delay element in the scrambler.

To support the reception of DSSS signals generated with implementations based on Clause 15, the receiver shall also be capable of synchronization on a SYNC field derived from any nonzero scrambler initial state.

### 18.2.3.2 Long PLCP SFD

The SFD shall be provided to indicate the start of PHY-dependent parameters within the PLCP preamble. The SFD shall be a 16-bit field, [1111 0011 1010 0000], where the rightmost bit shall be transmitted first in time.

### 18.2.3.3 Long PLCP SIGNAL field

The 8-bit SIGNAL field indicates to the PHY the modulation that shall be used for transmission (and reception) of the PSDU. The data rate shall be equal to the SIGNAL field value multiplied by 100 kbit/s. The High Rate PHY supports four mandatory rates given by the following 8-bit words, which represent the rate in units of 100 kbit/s, where the LSB shall be transmitted first in time:

a) X’0A’ (MSB to LSB) for 1 Mb/s;

b) X’14’ (MSB to LSB) for 2 Mb/s;

c) X’37’ (MSB to LSB) for 5.5 Mb/s;

d) X’6E’ (MSB to LSB) for 11 Mb/s.

The High Rate PHY rate change capability is described in 18.2.3.14. This field shall be protected by the CRC-16 FCS described in 18.2.3.6.


Copyright © 2007 IEEE. All rights reserved.

### 18.2.3.4 Long PLCP SERVICE field

Three bits have been defined in the SERVICE field to support the High Rate extension. The rightmost bit (bit 7) shall be used to supplement the LENGTH field described in 18.2.3.5. Bit 3 shall be used to indicate whether the modulation method is CCK <0> or PBCC <1>, as shown in Table 18-1. Bit 2 shall be used to indicate that the transmit frequency and symbol clocks are derived from the same oscillator. This locked clocks bit shall be set by the PHY based on its implementation configuration. The SERVICE field shall be transmitted b0 first in time, and shall be protected by the CRC-16 FCS described in 18.2.3.6. An IEEE 802.11-compliant device shall set the values of the bits b0, b1, b4, b5, and b6 to 0.

**Table 18-1—SERVICE field definitions**

| b0 | b1 | b2 | b3 | b4 | b5 | b6 | b7 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Locked clocks bit<br>0 = not<br>1 = locked | Mod. selection bit<br>0 = XXK<br>1 = PBCC | Reserved | Reserved | Reserved | Length extension bit |

### 18.2.3.5 Long PLCP LENGTH field

The PLCP LENGTH field shall be an unsigned 16-bit integer that indicates the number of microseconds required to transmit the PSDU. The transmitted value shall be determined from the LENGTH and DATARATE parameters in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 18.4.4.2.

The LENGTH field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The LENGTH field is calculated as follows. Because there is an ambiguity in the number of octets that is described by a length in integer microseconds for any data rate over 8 Mb/s, a length extension bit shall be placed at bit position b7 in the SERVICE field to indicate when the smaller potential number of octets is correct.

a) 5.5 Mb/s CCK — Length = number of octets × 8/5.5, rounded up to the next integer.

b) 11 Mb/s CCK — Length = number of octets × 8/11, rounded up to the next integer; the service field (b7) bit shall indicate a 0 if the rounding took less than 8/11 or a 1 if the rounding took more than or equal to 8/11.

c) 5.5 Mb/s PBCC — Length = (number of octets + 1) × 8/5.5, rounded up to the next integer.

d) 11 Mb/s PBCC — Length = (number of octets + 1) × 8/11, rounded up to the next integer; the service field (b7) bit shall indicate a 0 if the rounding took less than 8/11 or a 1 if the rounding took more than or equal to 8/11.

At the receiver, the number of octets in the MPDU is calculated as follows:

a) 5.5 Mb/s CCK — Number of octets = Length × 5.5/8, rounded down to the next integer.

b) 11 Mb/s CCK — Number of octets = Length × 11/8, rounded down to the next integer, minus 1 if the service field (b7) bit is a 1.

c) 5.5 Mb/s PBCC — Number of octets = (Length × 5.5/8) –1, rounded down to the next integer.

d) 11 Mb/s PBCC — Number of octets = (Length × 11/8) –1, rounded down to the next integer, minus 1 if the service field (b7) bit is a 1.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 689 of 1232

An example for an 11 Mb/s calculation described in psuedocode form is shown below. At the transmitter, the values of the LENGTH field and length extension bit are calculated as follows:

LENGTH' = ((number of octets + P) × 8) / R

LENGTH = Ceiling (LENGTH')

If

(R = 11) and ((LENGTH–LENGTH') ≥ 8/11)

then

Length Extension = 1

else

Length Extension = 0

where

| | |
|---|---|
| R | is the data rate (Mb/s) |
| P | is 0 for CCK |
| P | is 1 for PBCC |
| Ceiling (X) | returns the smallest integer value greater than or equal to X |

At the receiver, the number of octets in the MPDU is calculated as follows:

Number of octets = Floor(((Length × R) / 8) – P) – Length Extension

where

| | |
|---|---|
| R | is the data rate (Mb/s) |
| P | is 0 for CCK |
| P | is 1 for PBCC |
| Floor (X) | returns the largest integer value less than or equal to X |

Table 18-2 shows an example calculation for several packet lengths of CCK at 11 Mb/s.

**Table 18-2—Example of LENGTH calculations for CCK**

| TX octets | Octets (× 8/11) | LENGTH | Length extension bit | LENGTH (× 11/8) | Floor (X) | RX octets |
|---|---|---|---|---|---|---|
| 1023 | 744 | 744 | 0 | 1023 | 1023 | 1023 |
| 1024 | 744.7273 | 745 | 0 | 1024.375 | 1024 | 1024 |
| 1025 | 745.4545 | 746 | 0 | 1025.75 | 1025 | 1025 |
| 1026 | 746.1818 | 747 | 1 | 1027.125 | 1027 | 1026 |


Copyright © 2007 IEEE. All rights reserved.

Table 18-3 shows an example calculation for several packet lengths of PBCC at 11 Mb/s.

**Table 18-3—Example of LENGTH calculations for PBCC**

| TX octets | (Octets × 8/11) + 1 | LENGTH | Length extension bit | (LENGTH × 11/8) – 1 | Floor (X) | RX octets |
|---|---|---|---|---|---|---|
| 1023 | 744.7273 | 745 | 0 | 1023.375 | 1023 | 1023 |
| 1024 | 745.4545 | 746 | 0 | 1024.750 | 1024 | 1024 |
| 1025 | 746.1818 | 747 | 1 | 1026.125 | 1026 | 1025 |
| 1026 | 746.9091 | 747 | 0 | 1026.125 | 1026 | 1026 |

This example illustrates why normal rounding or truncation of the number will not produce the right result. The LENGTH field is defined in units of microseconds and must correspond to the actual length, and the number of octets must be exact.

The LSB shall be transmitted first in time. This field shall be protected by the CRC-16 FCS described in 18.2.3.6.

### 18.2.3.6 PLCP CRC (CRC-16) field

The SIGNAL, SERVICE, and LENGTH fields shall be protected with a CRC-16 FCS. The CRC-16 FCS shall be the ones complement of the remainder generated by the modulo 2 division of the protected PLCP fields by the polynomial

$$x^{16} + x^{12} + x^5 + 1$$

The protected bits shall be processed in transmit order. All FCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 18-3.

As an example, the SIGNAL, SERVICE, and LENGTH fields for a DBPSK signal with a PPDU length of 192 μs (24 octets) would be given by the following:

0101 0000 0000 0000 0000 0011 0000 0000 [leftmost bit (b0) transmitted first in time]

b0.....b48

The ones complement FCS for these protected PLCP preamble bits would be the following:

0101 1011 0101 0111 [leftmost bit (b0) transmitted first in time]

b0.....b16

Figure 18-3 depicts this example.

An illustrative example of the CRC-16 FCS using the information from Figure 18-3 is shown in Figure 18-4.

Copyright © 2007 IEEE. All rights reserved.

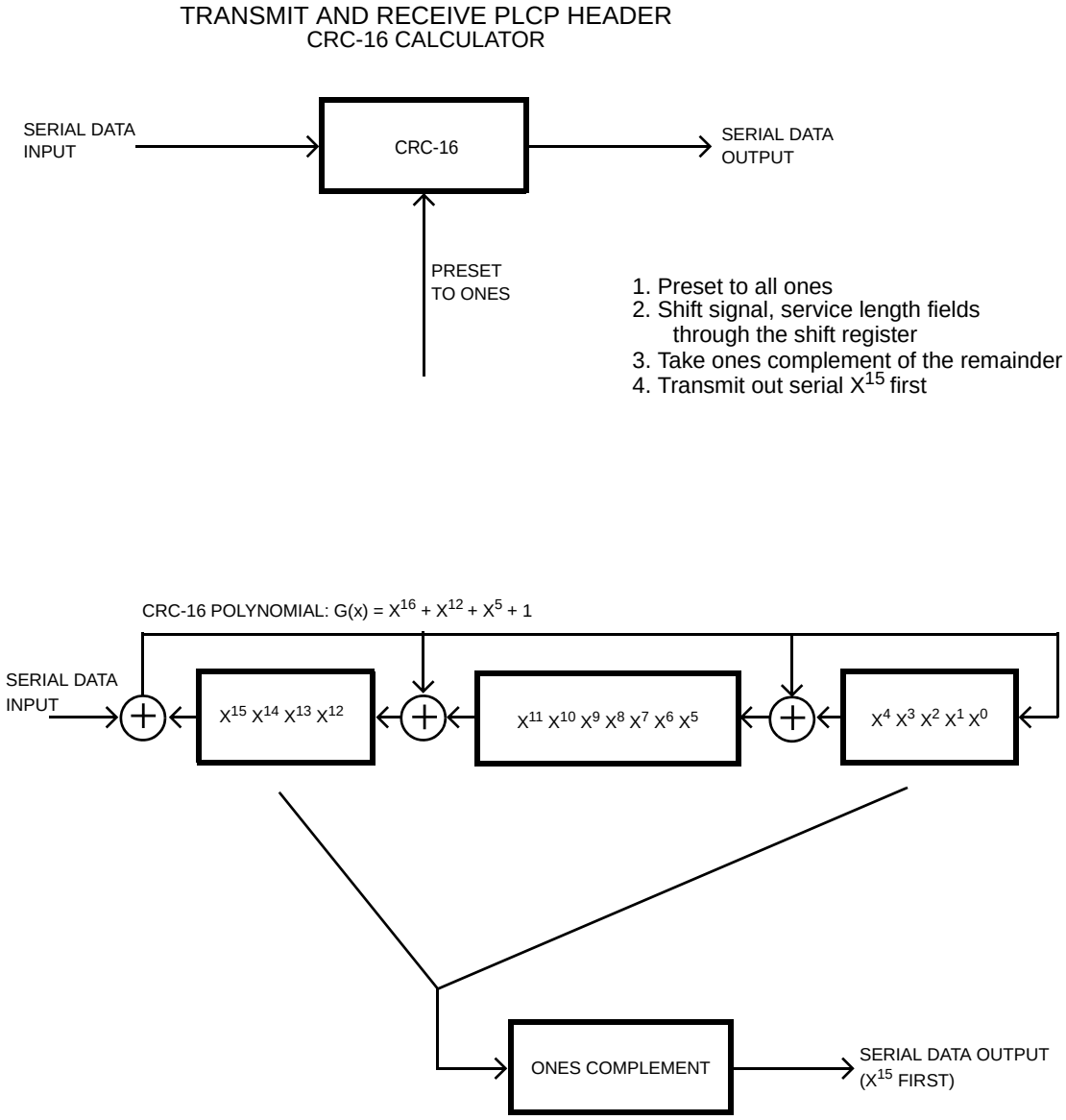



**Figure 18-3—CRC-16 implementation**


Copyright © 2007 IEEE. All rights reserved.

```
Data        CRC Registers
            MSB             LSB

            1111111111111111        ; Initialize preset to ones
  0         1110111111011111
  1         1101111110111110
  0         1010111101011101
  1         0101111010111010
  0         1011110101110100
  0         0110101011001001
  0         1101010110010010
  0         1011101100000101
  0         0110011000101011
  0         1100110001010110
  0         1000100010001101
  0         0000000100111011
  0         0000001001110110
  0         0000010011101100
  0         0000100111011000
  0         0001001110110000
  0         0010011101100000
  0         0100111011000000
  0         1001110110000000
  0         0010101100100001
  0         0101011001000010
  0         1010110010000100
  1         0101100100001000
  1         1010001000110001
  0         0101010001000011
  0         1010100010000110
  0         0100000100101101
  0         1000001001011010
  0         0001010010010101
  0         0010100100101010
  0         0101001001010100
  0         1010010010101000
            0101101101010111        ; ones complement, result = CRC FCS parity
```

**Figure 18-4—Example of CRC calculation**

### 18.2.3.7 Long PLCP data modulation and modulation rate change

The long PLCP preamble and header shall be transmitted using the 1 Mb/s DBPSK modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation that shall be used to transmit the PSDU. The SIGNAL field indicates the rate, and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 18.4.4.1.

### 18.2.3.8 Short PLCP synchronization (shortSYNC)

The shortSYNC field shall consist of 56 bits of scrambled 0 bits. This field is provided so the receiver can perform the necessary synchronization operations. The initial state of the scrambler (seed) shall be [001 1011], where the left end bit specifies the value to place in the first delay element ($Z^1$) in Figure 18-5 (in 18.2.4), and the right end bit specifies the value to place in the last delay element ($Z^7$).

### 18.2.3.9 Short PLCP SFD field (shortSFD)

The shortSFD shall be a 16-bit field and be the time reverse of the field of the SFD in the long PLCP preamble (18.2.3.2). The field is the bit pattern 0000 0101 1100 1111. The right end bit shall be transmitted first in time. A receiver not configured to use the short header option will not detect this SFD.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 693 of 1232

#### 18.2.3.10 Short PLCP SIGNAL field (shortSIGNAL)

The 8-bit SIGNAL field of the short header indicates to the PHY the data rate that shall be used for transmission (and reception) of the PSDU. A PHY operating with the HR/DSSS/short option supports three mandatory rates given by the following 8-bit words, where the LSB shall be transmitted first in time and the number represents the rate in units of 100 kBit/s:

a) X'14' (MSB to LSB) for 2 Mb/s;
b) X'37'(MSB to LSB) for 5.5 Mb/s;
c) X'6E' (MSB to LSB) for 11 Mb/s.

#### 18.2.3.11 Short PLCP SERVICE field (shortSERVICE)

The SERVICE field in the short header shall be the same as the SERVICE field described in 18.2.3.4.

#### 18.2.3.12 Short PLCP LENGTH field (shortLENGTH)

The LENGTH field in the short header shall be the same as the LENGTH field described in 18.2.3.5.

#### 18.2.3.13 Short CRC-16 field (shortCRC)

The CRC in the short header shall be the same as the CRC field defined in 18.2.3.6. The CRC-16 is calculated over the shortSIGNAL, shortSERVICE, and shortLENGTH fields.

#### 18.2.3.14 Short PLCP data modulation and modulation rate change

The short PLCP preamble shall be transmitted using the 1 Mb/s DBPSK modulation. The short PLCP header shall be transmitted using the 2 Mb/s modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation that shall be used to transmit the PSDU. The SIGNAL field indicates the rate, and the SERVICE field indicates the modulation. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 18.4.4.1.

### 18.2.4 PLCP/High Rate PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 18-5 and Figure 18-6 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

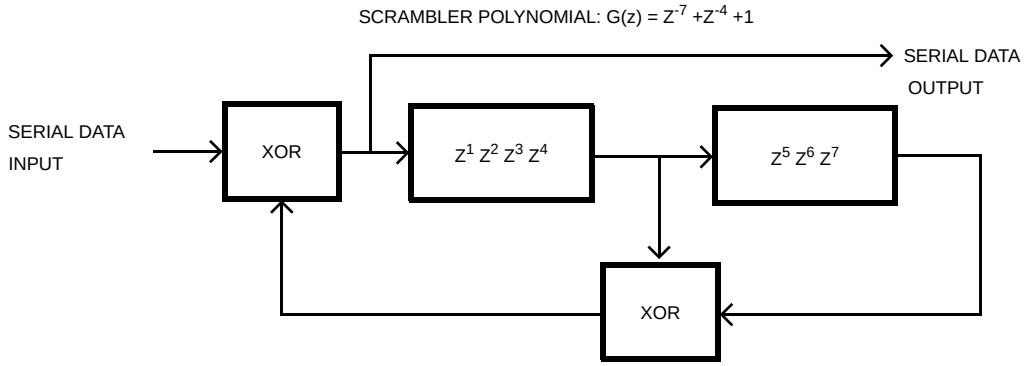



**Figure 18-5—Data scrambler**


Copyright © 2007 IEEE. All rights reserved.

The scrambler shall be initialized as specified in 18.2.3.8 for the short PLCP and 18.2.3.1 for the long PLCP. For a long preamble, this shall result in the scrambler registers $Z^1$ through $Z^7$ in Figure 18-5 having the data pattern [1101100] (i.e., $Z^1$= 1... $Z^7$= 0) when the scrambler is first started. The scrambler shall be initialized with the reverse pattern [0011011] when transmitting the optional short preamble.

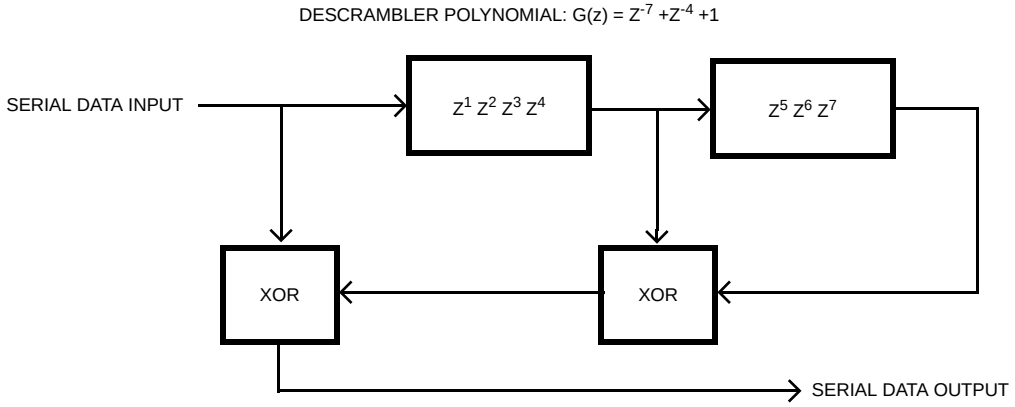



**Figure 18-6—Data descrambler**

### 18.2.5 Transmit PLCP

The transmit procedures for a High Rate PHY using the long PLCP preamble and header are the same as the transmit procedures described in 15.2.6 and 15.2.7 and do not change apart from the ability to transmit 5.5 Mb/s and 11 Mb/s.

The procedures for a transmitter employing HR/DSSS/short and HR/DSSS/PBCC/short are the same except for length and rate changes. The decision to use a long or short PLCP is beyond the scope of this standard.

The transmit PLCP is shown in Figure 18-7.

A PHY-TXSTART.request(TXVECTOR) primitive will be issued by the MAC to start the transmission of a PPDU. In addition to parameters DATARATE and LENGTH, other transmit parameters such as PREAMBLE_TYPE and MODULATION are set via the PHY-SAP with the PHY-TXSTART.request(TXVECTOR), as described in 18.3.5. The SIGNAL, SERVICE, and LENGTH fields of the PLCP header are calculated as described in 18.2.3.

The PLCP shall issue PMD_ANTSEL, PMD_RATE, and PMD_TXPWRLVL primitives to configure the PHY. The PLCP shall then issue a PMD_TXSTART.request, and the PHY entity shall immediately initiate data scrambling and transmission of the PLCP preamble based on the parameters passed in the PHY-TXSTART.request primitive. The time required for transmit power-on ramp, described in 18.4.7.6, shall be included in the PLCP SYNC field. Once the PLCP preamble transmission is complete, data shall be exchanged between the MAC and the PHY by a series of PHY-DATA.request(DATA) primitives issued by the MAC and PHY-DATA.confirm primitives issued by the PHY. The modulation and rate change, if any, shall be initiated with the first data symbol of the PSDU, as described in 18.2.3.7 and 18.2.3.14. The PHY proceeds with PSDU transmission through a series of data octet transfers from the MAC. At the PMD layer, the data octets are sent in LSB-to-MSB order and presented to the PHY through PMD_DATA.request primitives. Transmission can be prematurely terminated by the MAC through the primitive PHY-TXEND.request. PHY-TXSTART shall be disabled by the issuance of the PHY-TXEND.request. Normal termination occurs after the transmission of the final bit of the last PSDU octet, calculated from the number supplied in the PHY preamble LENGTH and SERVICE fields using the equations specified in 18.2.3.5. The PPDU transmission shall be completed and the PHY entity shall enter the receive state (i.e.,

Copyright © 2007 IEEE. All rights reserved.

PHY-TXSTART shall be disabled). It is recommended that modulation continue during power-down to prevent radiating a continuous wave carrier. Each PHY-TXEND.request is acknowledged with a PHY-TXEND.confirm primitive from the PHY.

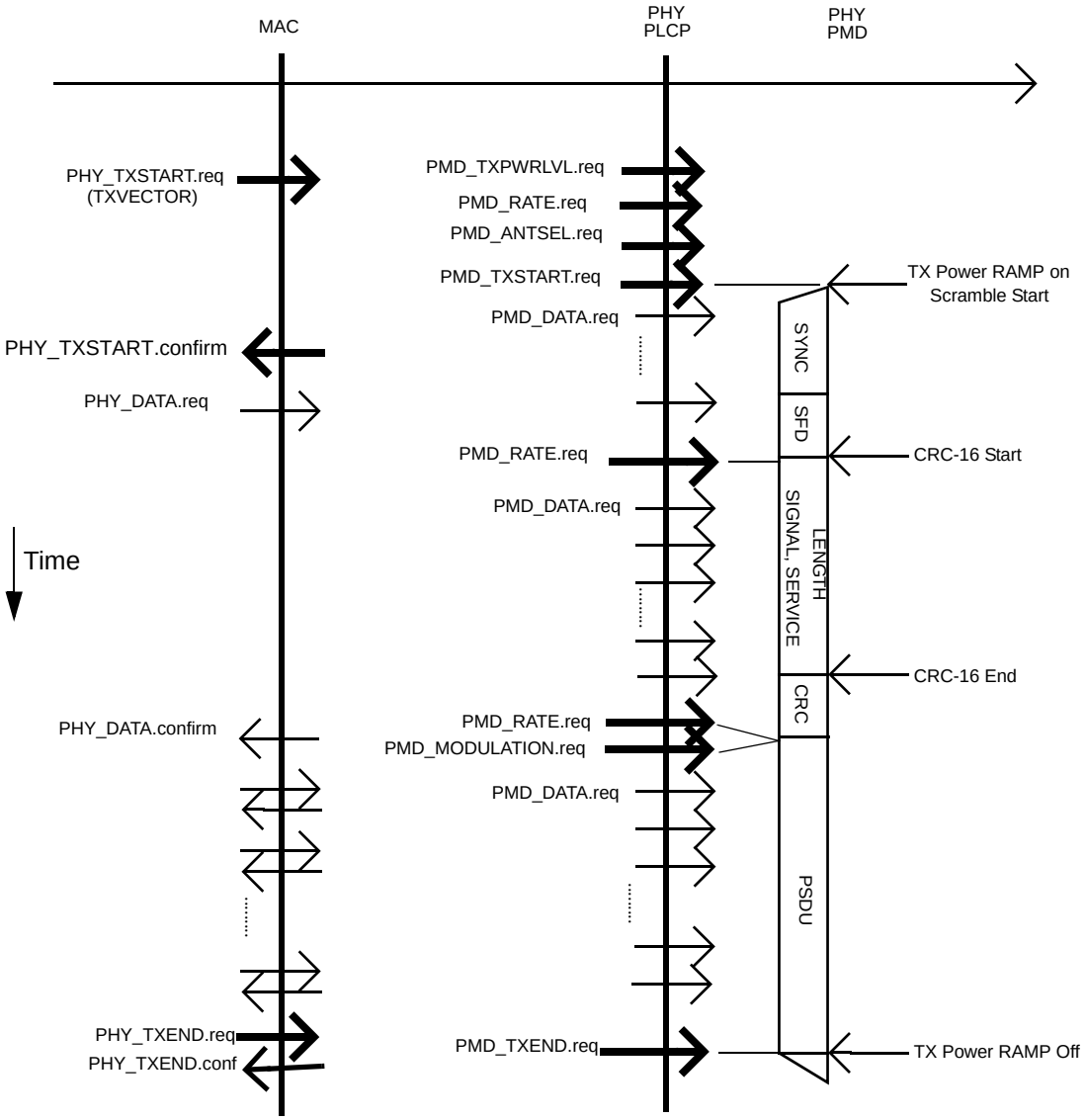



**Figure 18-7—Transmit PLCP**


Copyright © 2007 IEEE. All rights reserved.

A typical state machine implementation of the transmit PLCP is provided in Figure 18-8.

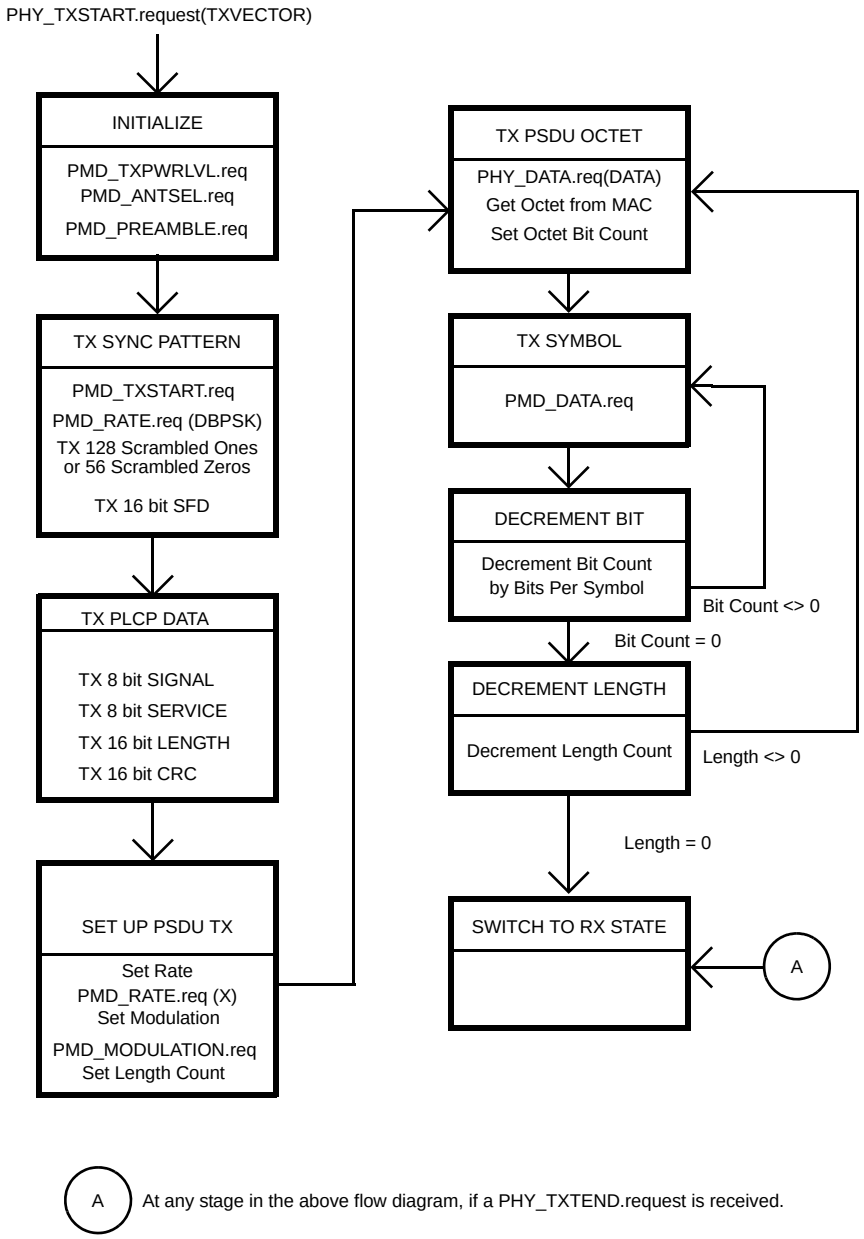



**Figure 18-8—PLCP transmit state machine**

### 18.2.6 Receive PLCP

The receive procedures for receivers configured to receive the mandatory and optional PLCPs, rates, and modulations are described in this subclause. A receiver that supports this High Rate extension of the standard is capable of receiving 5.5 Mb/s and 11 Mb/s, in addition to 1 Mb/s and 2 Mb/s. If the PHY implements the short preamble option, it shall detect both short and long preamble formats and indicate which type of preamble was received in the RXVECTOR. If the PHY implements the PBCC modulation option, it shall detect either CCK or PBCC modulations, as indicated in the SIGNAL field, and shall report the type of modulation used in the RXVECTOR.

Copyright © 2007 IEEE. All rights reserved.

The receiver shall implement the CCA procedure as defined in 18.4.8.4. Upon receiving a PPDU, the receiver shall distinguish between a long and short header format by the value of the SFD, as specified in 18.2.2. The receiver shall demodulate a long PLCP header using BPSK at 1 Mb/s. The receiver shall demodulate a short PLCP header using QPSK at 2 Mb/s. The receiver shall use the SIGNAL and SERVICE fields of the PLCP header to determine the data rate and modulation of the PSDU.

The receive PLCP is shown in Figure 18-9 In order to receive data, the PHY-TXSTART.request shall be disabled so that the PHY entity is in the receive state. Further, through STA management via the PLME, the PHY shall be set to the appropriate channel and the CCA method chosen. Other receive parameters, such as RSSI, SQ, and indicated DATARATE, may be accessed via the PHY-SAP.

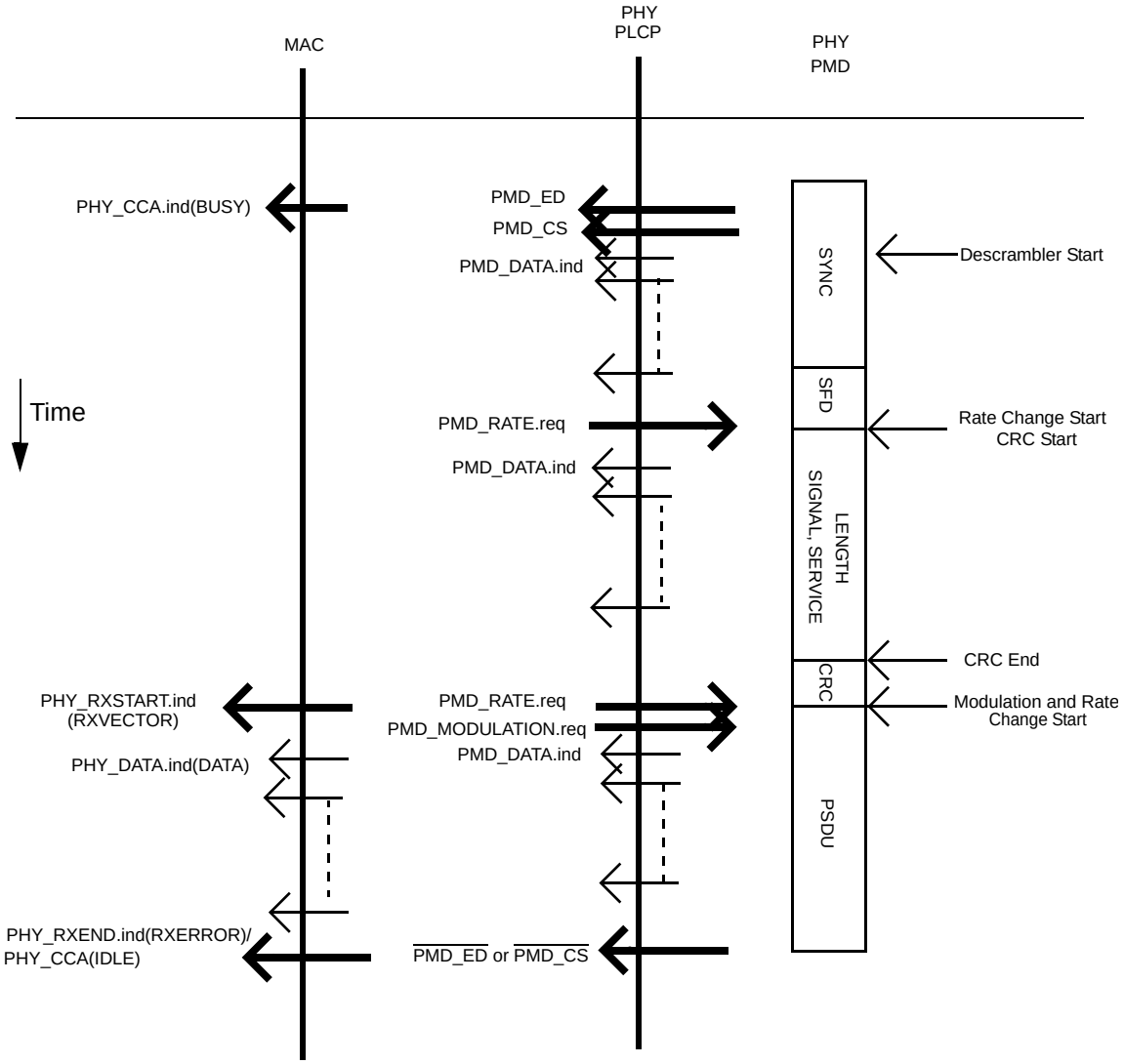



**Figure 18-9—Receive PLCP**

Upon receiving the transmitted energy, according to the selected CCA mode, the PMD_ED shall be enabled (according to 18.4.8.4) as the RSSI reaches the ED_THRESHOLD, and/or PMD_CS shall be enabled after code lock is established. These conditions are used to indicate activity to the MAC via PHY-CCA.indicate, according to 18.4.8.4. PHY-CCA.indicate(BUSY) shall be issued for ED and/or code lock prior to correct reception of the PLCP header. The PMD primitives, PMD_SQ and PMD_RSSI, are issued to update the RSSI and SQ parameters reported to the MAC.


Copyright © 2007 IEEE. All rights reserved.

After PHY-CCA.indicate is issued, the PHY entity shall begin searching for the SFD field. Once the SFD field is detected, CRC-16 processing shall be initiated and the PLCP SIGNAL, SERVICE, and LENGTH fields shall be received. The CRC-16 FCS shall be processed. If the CRC-16 FCS check fails, the PHY receiver shall return to the RX IDLE state, as depicted in Figure 18-10. Should the status of CCA return to the IDLE state during reception prior to completion of the full PLCP processing, the PHY receiver shall return to the RX IDLE state.

If the PLCP header reception is successful (and the SIGNAL field is completely recognizable and supported), a PHY-RXSTART.indicate(RXVECTOR) shall be issued. The RXVECTOR associated with this primitive includes

a) The SIGNAL field
b) The SERVICE field
c) The PSDU length in octets (calculated from the LENGTH field in microseconds and the DATARATE in Mb/s, in accordance with the formula in 18.2.3.5)
d) RXPREAMBLE_TYPE (which is an enumerated type taking on values SHORTPREAMBLE or LONGPREAMBLE)
e) The antenna used for receive (RX_ANTENNA), RSSI, and SQ

The received PSDU bits are assembled into octets and presented to the MAC using a series of PHY-DATA.indicate(DATA) primitive exchanges. The rate and modulation change indicated in the SIGNAL field shall be initiated with the first symbol of the PSDU, as described in 18.2.5. The PHY proceeds with PSDU reception. After reception of the final bit of the last PSDU octet, indicated by the PLCP preamble LENGTH field, the receiver shall be returned to the RX IDLE state shown in Figure 18-10.

A PHY-RXEND.indicate(NoError) primitive shall be issued. A PHY-CCA.indicate(IDLE) primitive shall be issued following a change in PHYCS and/or PHYED according to the selected CCA method.

In the event that a change in PHYCS or PHYED would cause the status of CCA to return to the IDLE state before the complete reception of the PSDU, as indicated by the PLCP LENGTH field, the error condition PHY-RXEND.indicate(CarrierLost) shall be reported to the MAC. The High Rate PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted PPDU.

If the PLCP header is successful, but the indicated rate or modulation in the SIGNAL and SERVICE fields is not within the capabilities of the receiver, a PHY-RXSTART.indicate shall not be issued. The PHY shall issue the error condition PHY-RXEND.indicate(UnsupportedRate). If the PLCP header is invalid, a PHY-RXSTART.indicate shall not be issued, and the PHY shall issue the error condition PHY-RXEND.indicate(FormatViolation). Also, in both cases, the High Rate PHY shall ensure that the CCA indicates a busy medium for the intended duration of the transmitted PSDU, as indicated by the LENGTH field. The intended duration is indicated by the LENGTH field (LENGTH $\times$ 1 μs).

A typical state machine implementation of the receive PLCP is shown in Figure 18-10.

Copyright © 2007 IEEE. All rights reserved.

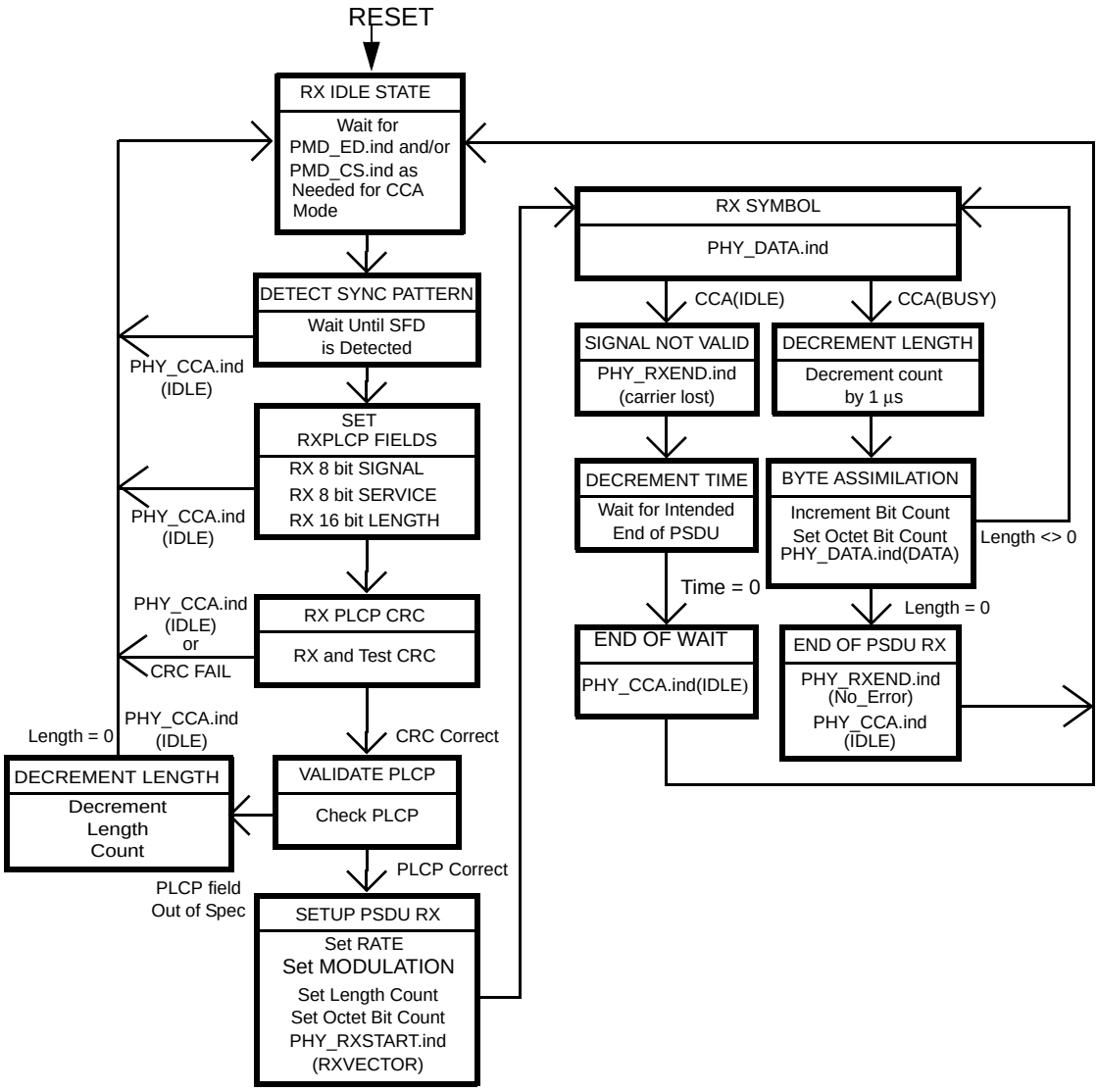



**Figure 18-10—PLCP receive state machine**

## 18.3 High Rate PLME

### 18.3.1 PLME_SAP sublayer management primitives

Table 18-4 lists the MIB attributes that may be accessed by the PHY entities and intralayer or higher level LMEs. These attributes are accessed via the PLME-GET, PLME-SET, and PLME-RESET primitives defined in Clause 10.


Copyright © 2007 IEEE. All rights reserved.

### 18.3.2 High Rate PHY MIB

All High Rate PHY MIB attributes are defined in Annex D, with specific values defined in Table 18-4.

**Table 18-4—MIB attribute default values/ranges**

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11PhyOperationTable** | | |
| dot11PHYType | High Rate–2.4 (X‘05’) | Static |
| dot11TempType | Implementation dependent | Static |
| dot11CurrentRegDomain | Implementation dependent | Static |
| dot11ShortPreambleOptionImplemented | Implementation dependent | Static |
| dot11PBCCOptionImplemented | Implementation dependent | Static |
| dot11ChannelAgility Present | Implementation dependent | Static |
| dot11ChannelAgilityEnabled | False/Boolean | Dynamic |
| **dot11PhyAntennaTable** | | |
| dot11CurrentTxAntenna | Implementation dependent | Dynamic |
| dot11DiversitySupport | Implementation dependent | Static |
| dot11CurrentRxAntenna | Implementation dependent | Dynamic |
| **dot11PhyTxPowerTable** | | |
| dot11NumberSupportedPowerLevels | Implementation dependent | Static |
| dot11TxPowerLevel1 | Implementation dependent | Static |
| dot11TxPowerLevel2 | Implementation dependent | Static |
| dot11TxPowerLevel3 | Implementation dependent | Static |
| dot11TxPowerLevel4 | Implementation dependent | Static |
| dot11TxPowerLevel5 | Implementation dependent | Static |
| dot11TxPowerLevel6 | Implementation dependent | Static |
| dot11TxPowerLevel7 | Implementation dependent | Static |
| dot11TxPowerLevel8 | Implementation dependent | Static |
| dot11CurrentTxPowerLevel | Implementation dependent | Dynamic |
| **dot11PhyDSSSTable** | | |
| dot11CurrentChannel | Implementation dependent | Dynamic |
| dot11CCAModeSupported | Implementation dependent | Static |
| dot11CurrentCCAMode | Implementation dependent | Dynamic |
| dot11EDThreshold | Implementation dependent | Dynamic |
| **dot11AntennasListTable** | | |
| dot11SupportTxAntenna | Implementation dependent | Static |
| dot11SupportRxAntenna | Implementation dependent | Static |
| dot11DiversitySelectionRx | Implementation dependent | Dynamic |

Copyright © 2007 IEEE. All rights reserved.

**Table 18-4—MIB attribute default values/ranges *(continued)***

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11RegDomainsSupportedTable** | | |
| dot11RegDomainsSupported | Implementation dependent | Static |
| dot11SupportedDataRatesTx | Table Tx X'02', X'04', X'0B', X'16' | Static |
| dot11SupportedDataRatesRx | Table Rx X'02', X'04', 'X'0B', X'16' | Static |
| NOTE—The column titled "Operational semantics" contains two types: static and dynamic. Static MIB attributes are fixed and cannot be modified for a given PHY implementation. Dynamic MIB attributes can be modified by some management entities. | | |

### 18.3.3 DS PHY characteristics

The static DS PHY characteristics, provided through the PLME-CHARACTERISTICS service primitive, are shown in Table 18-5. The definitions of these characteristics are in 10.4.3.

**Table 18-5—High Rate PHY characteristics**

| Characteristic | Value |
|---|---|
| aSlotTime | 20 μs |
| aSIFSTime | 10 μs |
| aCCATime | ≤ 15 μs |
| aPHY-RX-START-Delay | 192 μs for long preamble and<br>96 μs for short preamble |
| aRxTxTurnaroundTime | ≤ 5 μs |
| aTxPLCPDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxPLCPDelay | Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aRxTxSwitchTime | ≤ 5 μs |
| aTxRampOnTime | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aTxRampOffTime | Implementers may choose any value for this delay as long as the requirements of aSIFSTime are met. |
| aTxRFDelay | Implementers may choose any value for this delay as long as the requirements of aRxTxTurnaroundTime are met. |
| aRxRFDelay | Implementers may choose any value for this delay as long as the requirements of aSIFSTime and aCCATime are met. |
| aAirPropagationTime | 1 μs |
| aMACProcessingDelay | ≤ 2 μs |
| aPreambleLength | 144 μs |
| aPLCPHeaderLength | 48 μs |
| aMPUMaxLength | $14 \le x \le (2^{12} - 1)$ |
| aCWmin | 31 |
| aCWmax | 1023 |


Copyright © 2007 IEEE. All rights reserved.

### 18.3.4 High Rate TXTIME calculation

The value of the TXTIME parameter returned by the PLME-TXTIME.confirm primitive shall be calculated according to the following equation:

$$TXTIME = PreambleLength + PLCPHeaderTime + Ceiling(((LENGTH+PBCC) \times 8) / DATARATE)$$

where

| | |
|---|---|
| LENGTH and DATARATE | are values from the TXVECTOR parameter of the corresponding PLME-TXTIME.request primitive |
| LENGTH | is in units of octets |
| DATARATE | is in units of Mb/s |
| Ceiling | is a function that returns the smallest integer value greater than or equal to its argument value |
| PBCC | has a value of 1 if the SIGNAL value from the TXVECTOR parameter specifies PBCC and has a value of 0 otherwise |
| The value of PreambleLength | is 144 μs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "LONGPREAMBLE," or 72 μs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "SHORTPREAMBLE" |
| The value of PLCPHeaderTime | is 48 μs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "LONGPREAMBLE," or 24 μs if the TXPREAMBLE_TYPE value from the TXVECTOR parameter indicates "SHORTPREAMBLE" |

### 18.3.5 Vector descriptions

Several service primitives include a parameter vector. These vectors are a list of parameters as described in Table 18-6. DATARATE and LENGTH are described in 12.3.4.4. The remaining parameters are considered to be management parameters and are specific to this PHY.

**Table 18-6—Parameter vectors**

| Parameter | Associated vector | Value |
|---|---|---|
| DATARATE | RXVECTOR, TXVECTOR | The rate used to transmit the PSDU in Mb/s. |
| LENGTH | RXVECTOR, TXVECTOR | The length of the PSDU in octets. |
| PREAMBLE_TYPE | RXVECTOR, TXVECTOR | The preamble used for the transmission of this PPDU. This is an enumerated type that can take the value SHORTPREAMBLE or LONGPREAMBLE. |
| MODULATION | RXVECTOR, TXVECTOR | The modulation used for the transmission of this PSDU. This is an integer where 0 means CCK and 1 means PBCC. |

Copyright © 2007 IEEE. All rights reserved.

## 18.4 High Rate PMD sublayer

### 18.4.1 Scope and field of application

The PMD services provided to the PLCP for the High Rate PHY are described in 18.4. Also defined in 18.4 are the functional, electrical, and RF characteristics required for interoperability of implementations conforming to this specification. The relationship of this specification to the entire High Rate PHY is shown in Figure 18-11.

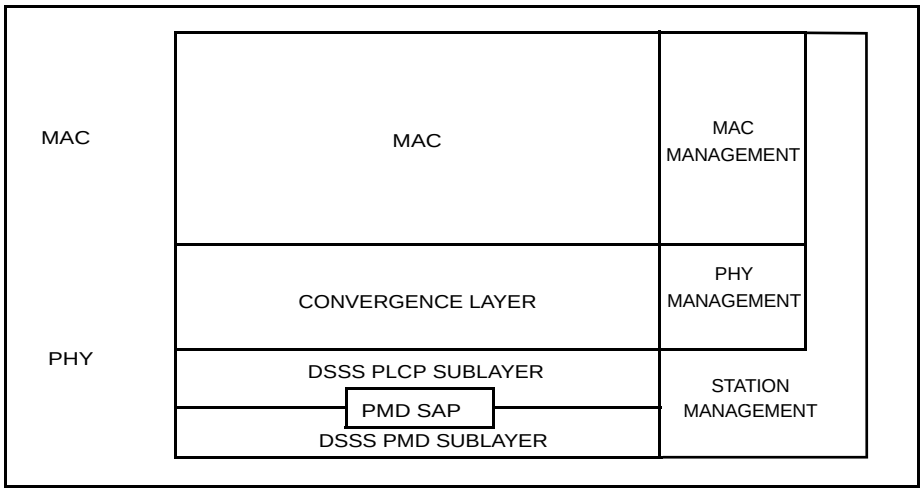



**Figure 18-11—Layer reference model**

### 18.4.2 Overview of service

The High Rate PMD sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined functions of the High Rate PMD sublayer primitives and parameters for the receive function result in a data stream, timing information, and associated receive signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

### 18.4.3 Overview of interactions

The primitives associated with the PLCP sublayer to the High Rate PMD fall into two basic categories

a) Service primitives that support PLCP peer-to-peer interactions;
b) Service primitives that have local significance and that support sublayer-to-sublayer interactions.

### 18.4.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory, unless otherwise specified.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.4.1 PMD_SAP peer-to-peer service primitives

Table 18-7 indicates the primitives for peer-to-peer interactions.

**Table 18-7—PMD_SAP peer-to-peer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_DATA | X | X | — | — |

#### 18.4.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 18-8 indicates the primitives for sublayer-to-sublayer interactions.

**Table 18-8—PMD_SAP sublayer-to-sublayer service primitives**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_TXSTART | X | — | — | — |
| PMD_TXEND | X | — | — | — |
| PMD_ANTSEL | X | X | — | — |
| PMD_TXPWRLVL | X | — | — | — |
| PMD_MODULATION | X | X | — | — |
| PMD_PREAMBLE | X | X | — | — |
| PMD_RATE | X | X | — | — |
| PMD_RSSI | — | X | — | — |
| PMD_SQ | — | X | — | — |
| PMD_CS | — | X | — | — |
| PMD_ED | X | X | — | — |

Copyright © 2007 IEEE. All rights reserved.

### 18.4.5 PMD_SAP detailed service specification

The services provided by each PMD primitive are described in 18.4.5.1 through 18.4.5.15.

#### 18.4.5.1 PMD_DATA.request

##### 18.4.5.1.1 Function

This primitive defines the transfer of data from the PLCP sublayer to the PMD entity.

##### 18.4.5.1.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value (Mb/s) | Description |
|---|---|---|---|
| TXD_UNIT | PMD_DATA.request | 0,1: 1<br>00,01,11,10:2<br>X'0'–X'F': 5.5<br>X'00'–X'FF': 11 | This parameter represents a single block of data, which, in turn, is used by the PMD to be differentially encoded into a transmitted symbol. The symbol itself is spread by the PN code prior to transmission. |

##### 18.4.5.1.3 When generated

This primitive is generated by the PLCP sublayer to request transmission of a symbol. The data clock for this primitive is supplied by the PMD layer based on the PN code repetition.

##### 18.4.5.1.4 Effect of receipt

The PMD performs the differential encoding, PN code modulation, and transmission of data.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.2 PMD_DATA.indicate

##### 18.4.5.2.1 Function

This primitive defines the transfer of data from the PMD entity to the PLCP sublayer.

##### 18.4.5.2.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value (Mb/s) | Description |
|---|---|---|---|
| RXD_UNIT | PMD_DATA.indicate | 0,1: 1<br>00,01,11,10:2<br>X'0' - X'F': 5.5<br>X'00' - X'FF': 11 | This parameter represents a single symbol that has been demodulated by the PMD entity. |

##### 18.4.5.2.3 When generated

This primitive, which is generated by the PMD entity, forwards received data to the PLCP sublayer. The data clock for this primitive is supplied by the PMD layer based on the PN code repetition.

##### 18.4.5.2.4 Effect of receipt

The PLCP sublayer either interprets the bit or bits that are recovered as part of the PLCP or passes the data to the MAC sublayer as part of the PSDU.

Copyright © 2007 IEEE. All rights reserved.

### 18.4.5.3 PMD_MODULATION.request

#### 18.4.5.3.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the modulation code that is used by the High Rate PHY for transmission.

#### 18.4.5.3.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| MODULATION | PMD_MODULATION.request<br>PMD_MODULATION.indicate | 1 MbBarker,<br>2 MbBarker, 5.5 CCK,<br>11 CCK,<br>5.5 PBCC, or 11 PBCC | In Receive mode, the MODULATION parameter informs the PLCP layer which PHY data modulation was used to process the PSDU portion of the PPDU. See 18.4.6.3 for further information on the High Rate PHY modulation codes. |

#### 18.4.5.3.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY modulation code used for the PSDU portion of a PPDU. The PMD_MODULATION.request primitive is normally issued prior to issuing the PMD_TXSTART command.

#### 18.4.5.3.4 Effect of receipt

The receipt of PMD_MODULATION selects the modulation that is used for all subsequent PSDU transmissions. This code is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY modulations. This primitive, which is generated by the PMD entity, sets the state of the PHY for demodulation of the appropriate modulation.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.4 PMD_PREAMBLE.request

##### 18.4.5.4.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the preamble mode that is used by the High Rate PHY for transmission.

##### 18.4.5.4.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PREAMBLE | PMD_PREAMBLE.request | 0 for long<br>1 for short | PREAMBLE selects which of the High Rate PHY preamble types is used for PLCP transmission. See 18.2.2 for further information on the High Rate PHY preamble modes. |

##### 18.4.5.4.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY preamble mode used for the PLCP portion of a PPDU. The PMD_PREAMBLE.request primitive is normally issued prior to issuing the PMD_TXSTART command.

##### 18.4.5.4.4 Effect of receipt

The receipt of PMD_PREAMBLE selects the preamble mode that is used for all subsequent PSDU transmissions. This mode is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY preambles. This primitive sets the state of the PHY for modulation of the appropriate mode.

Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.5 PMD_PREAMBLE.indicate

##### 18.4.5.5.1 Function

This primitive, which is generated by the PMD sublayer, indicates which preamble mode was used to receive the PLCP portion of the PPDU.

##### 18.4.5.5.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PREAMBLE | PMD_PREAMBLE. indicate | 0 for long<br>1 for short | In RECEIVE mode, the PREAMBLE parameter informs the PLCP layer which of the High Rate PHY preamble modes was used to send the PLCP portion of the PPDU. |

##### 18.4.5.5.3 When generated

This primitive is generated by the PMD sublayer when the PLCP preamble has been properly detected.

##### 18.4.5.5.4 Effect of receipt

This parameter is provided to the PLCP layer for information only.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.6 PMD_TXSTART.request

##### 18.4.5.6.1 Function

As a result of receiving a PHY_DATA.request from the MAC, the PLCP issues this primitive, which initiates PPDU transmission by the PMD layer.

##### 18.4.5.6.2 Semantics of the service primitive

This primitive has no parameters.

##### 18.4.5.6.3 When generated

This primitive is generated by the PLCP sublayer to initiate the PMD layer transmission of the PPDU. The PHY-DATA.request primitive is provided to the PLCP sublayer prior to issuing the PMD_TXSTART command.

##### 18.4.5.6.4 Effect of receipt

PMD_TXSTART initiates transmission of a PPDU by the PMD sublayer.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 711 of 1232

#### 18.4.5.7 PMD_TXEND.request

##### 18.4.5.7.1 Function

This primitive, which is generated by the PHY PLCP sublayer, ends PPDU transmission by the PMD layer.

##### 18.4.5.7.2 Semantics of the service primitive

This primitive has no parameters.

##### 18.4.5.7.3 When generated

This primitive is generated by the PLCP sublayer to terminate the PMD layer transmission of the PPDU.

##### 18.4.5.7.4 Effect of receipt

PMD_TXEND terminates transmission of a PPDU by the PMD sublayer.


Copyright © 2007 IEEE. All rights reserved.

### 18.4.5.8 PMD_ANTSEL.request

#### 18.4.5.8.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the antenna used by the PHY for transmission or reception (when diversity is disabled).

#### 18.4.5.8.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| ANT_STATE | PMD_ANTSEL.request<br>PMD_ANTSEL.indicate | 1 to 256 | ANT_STATE selects which of the available antennas should be used for transmit. The number of available antennas is determined from the MIB table parameters, aSuprtRxAntennas and aSuprtTxAntennas. |

#### 18.4.5.8.3 When generated

This primitive is generated by the PLCP sublayer to select a specific antenna for transmission (or reception when diversity is disabled).

#### 18.4.5.8.4 Effect of receipt

PMD_ANTSEL immediately selects the antenna specified by ANT_STATE.

Copyright © 2007 IEEE. All rights reserved.

### 18.4.5.9 PMD_TXPWRLVL.request

#### 18.4.5.9.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the power level used by the PHY for transmission.

#### 18.4.5.9.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| TXPWR_LEVEL | PMD_TXPWRLVL.request | 0, 1, 2, 3 (maximum of 4 levels) | TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current PPDU transmission. The number of available power levels is determined by the MIB parameter dot11Number-SupportedPowerLevels. See 18.4.7.2 for further information on the optional High Rate PHY power-level control capabilities. |

#### 18.4.5.9.3 When generated

This primitive is generated by the PLCP sublayer to select a specific transmit power. This primitive is applied prior to setting PMD_TXSTART to the transmit state.

#### 18.4.5.9.4 Effect of receipt

PMD_TXPWRLVL immediately sets the transmit power level given by TXPWR_LEVEL.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.10 PMD_RATE.request

##### 18.4.5.10.1 Function

This primitive, which is generated by the PHY PLCP sublayer, selects the data rate that shall be used by the High Rate PHY for transmission.

##### 18.4.5.10.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value (Mb/s) | Description |
|---|---|---|---|
| RATE | PMD_RATE.indicate<br>PMD_RATE.request | X'0A' or 1<br>X'14' for 2<br>X'37' for 5.5<br>X'6E' for 11 | RATE selects which of the High Rate PHY data rates is used for PSDU transmission. See 18.4.6.3 for further information on the High Rate PHY data rates. The High Rate PHY rate change capability is described in 18.2. |

##### 18.4.5.10.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY data rate used for the PSDU portion of a PPDU.

##### 18.4.5.10.4 Effect of receipt

The receipt of PMD_RATE selects the rate that is used for all subsequent PSDU transmissions. This rate is used for transmission only. The High Rate PHY shall still be capable of receiving all the required High Rate PHY data rates.

Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.11 PMD_RSSI.indicate

##### 18.4.5.11.1 Function

This optional primitive may be generated by the PMD to provide the receive signal strength to the PLCP.

##### 18.4.5.11.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| RSSI | PMD_RSSI.indicate | 0–8 bits of RSSI | The RSSI is a measure of the RF energy received by the High Rate PHY. |

##### 18.4.5.11.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

##### 18.4.5.11.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The RSSI may be used in conjunction with SQ as part of a CCA scheme.


Copyright © 2007 IEEE. All rights reserved.

### 18.4.5.12 PMD_SQ.indicate

#### 18.4.5.12.1 Function

This optional primitive may be generated by the PMD to provide an indication of the SQ of the High Rate PHY PN code correlation to the PLCP. SQ is a measure of the quality of BARKER code lock, providing an effective measure during the full reception of a PLCP preamble and header.

#### 18.4.5.12.2 Semantics of the service primitive

This primitive provides the following parameters

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| SQ | PMD_SQ.indicate | 0–8 bits of SQ | This primitive is a measure of the SQ received by the HR/ DSSS PHY. |

#### 18.4.5.12.3 When generated

This primitive is generated by the PMD when the High Rate PHY is in the receive state and Barker code lock is achieved. It is continuously available to the PLCP, which, in turn, provides the parameter to the MAC entity.

#### 18.4.5.12.4 Effect of receipt

This parameter is provided to the PLCP layer for information only. The SQ may be used in conjunction with RSSI as part of a CCA scheme.

Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.13 PMD_CS.indicate

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data are being demodulated.

#### 18.4.5.13.1 Function

This primitive, which is generated by the PMD, shall indicate to the PLCP layer that the receiver has acquired (locked) the Barker code and data are being demodulated.

#### 18.4.5.13.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PMD_CS | PMD_CS.indicate | 0 for DISABLED<br>1 for ENABLED | The PMD_CS primitive, in conjunction with PMD_ED, provides CCA status through the PLCP layer PHYCCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS is ENABLED when the correlator SQ indicated in PMD_SQ is greater than the correlation threshold. PMD_CS is DISABLED when the PMD_SQ falls below the correlation threshold. |

#### 18.4.5.13.3 When generated

This primitive is generated by the PMD sublayer when the High Rate PHY is receiving a PPDU and the PN code has been acquired.

#### 18.4.5.13.4 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PHY-CCA indicator. This parameter shall indicate that the RF medium is busy and occupied by a High Rate PHY signal. The High Rate PHY should not be placed into the transmit state when PMD_CS is ENABLED.


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.5.14 PMD_ED.indicate

##### 18.4.5.14.1 Function

This optional primitive may be generated by the PMD to provide an indication that the receiver has detected RF energy indicated by the PMD_RSSI primitive that is above a predefined threshold.

##### 18.4.5.14.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PMD_ED | PMD_ED.indicate | 0 for DISABLED<br>1 for ENABLED | The PMD_ED primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED is ENABLED when the RSSI in PMD_RSSI is greater than the ED_THRESHOLD parameter. PMD_ED is DISABLED when the PMD_RSSI falls below the energy detect threshold. |

##### 18.4.5.14.3 When generated

This primitive is generated by the PHY when the PHY is receiving RF energy from any source that exceeds the ED_THRESHOLD parameter.

##### 18.4.5.14.4 Effect of receipt

This indicator is provided to the PLCP for forwarding to the MAC entity for information purposes through the PMD_ED indicator. This parameter shall indicate that the RF medium may be busy with an RF energy source that is not High Rate PHY compliant. If a High Rate PHY source is being received, the PMD_CS function is enabled shortly after the PMD_ED function is enabled.

Copyright © 2007 IEEE. All rights reserved.

### 18.4.5.15 PMD_ED.request

#### 18.4.5.15.1 Function

This optional primitive may be generated by the PLCP to set a value for the energy detect ED_THRESHOLD.

#### 18.4.5.15.2 Semantics of the service primitive

This primitive provides the following parameters.

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PMD_ED | PMD_ED.request | ED_THRESHOLD | ED_THRESHOLD is the threshold that the RSSI should be greater than in order for PMD_ED to be enabled. PMD_ED is DISABLED when the PMD_RSSI falls below the energy detect threshold. |

#### 18.4.5.15.3 When generated

This primitive is generated by the PLCP sublayer to change or set the current High Rate PHY energy detect threshold.

#### 18.4.5.15.4 Effect of receipt

The receipt of PMD_ED immediately changes the energy detect threshold as set by the ED_THRESHOLD parameter.


Copyright © 2007 IEEE. All rights reserved.

### 18.4.6 PMD operating specifications, general

General specifications for the High Rate PMD sublayer are provided in 18.4.6.1 through 18.4.6.14. These specifications apply to both the receive and transmit functions and general operation of a High Rate PHY.

WLANs implemented in accordance with this standard are subject to equipment certification and operation requirements established by regional and national regulatory administrations. The PMD specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regulations are subject to revision, or may be superseded. Requirements that are subject to local geographic regulations are annotated within the PMD specification. Regulatory requirements that do not affect interoperability are not addressed in this standard. Implementers are referred to the following regulatory sources for further information. Operation in countries within defined regulatory domains may be subject to additional regulations.

#### 18.4.6.1 Operating frequency range

The High Rate PHY shall operate in the 2.4–2.4835 GHz frequency range, as allocated by regulatory bodies in the China, United States, Europe, and Japan, or in the 2.471–2.497 GHz frequency range, as allocated by regulatory authority in Japan.

#### 18.4.6.2 Number of operating channels

The channel center frequencies and CHNL_ID numbers shall be as shown in Table 18-9. See the applicable regulations for the countries in which the implementation will operate. For each supported regulatory domain, all channels in Table 18-9 marked with "X" shall be supported.

**Table 18-9—High Rate PHY frequency channel plan**

| CHNL_ID | Frequency (MHz) | Regulatory domains | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | X'10' FCC | X'20' IC | X'30' ETSI | X'31' Spain | X'32 France | X'40' Japan | X'41' Japan | X'51' China |
| 1 | 2412 | X | X | X | — | — | — | X | X |
| 2 | 2417 | X | X | X | — | — | — | X | X |
| 3 | 2422 | X | X | X | — | — | — | X | X |
| 4 | 2427 | X | X | X | — | — | — | X | X |
| 5 | 2432 | X | X | X | — | — | — | X | X |
| 6 | 2437 | X | X | X | — | — | — | X | X |
| 7 | 2442 | X | X | X | — | — | — | X | X |
| 8 | 2447 | X | X | X | — | — | — | X | X |
| 9 | 2452 | X | X | X | — | — | — | X | X |
| 10 | 2457 | X | X | X | X | X | — | X | X |
| 11 | 2462 | X | X | X | X | X | — | X | X |
| 12 | 2467 | — | — | X | — | X | — | X | X |
| 13 | 2472 | — | — | X | — | X | — | X | X |
| 14 | 2484 | — | — | — | — | — | X | — | — |

Copyright © 2007 IEEE. All rights reserved.

In a multiple cell network topology, overlapping and/or adjacent cells using different channels can operate simultaneously without interference if the distance between the center frequencies is at least 25 MHz. Channel 14 shall be designated specifically for operation in Japan.

#### 18.4.6.3 Modulation and channel data rates

Four modulation formats and data rates are specified for the High Rate PHY. The basic access rate shall be based on 1 Mb/s DBPSK modulation. The enhanced access rate shall be based on 2 Mb/s DQPSK. The extended direct sequence specification defines two additional data rates. The High Rate access rates shall be based on the CCK modulation scheme for 5.5 Mb/s and 11 Mb/s. An optional PBCC mode is also provided for potentially enhanced performance.

#### 18.4.6.4 Spreading sequence and modulation for 1 Mb/s and 2 Mb/s

The following 11-chip Barker sequence shall be used as the PN code sequence for the 1 Mb/s and 2 Mb/s modulation:

+1, –1, +1, +1, –1, +1, +1, +1, –1, –1, –1

The leftmost chip shall be output first in time. The first chip shall be aligned at the start of a transmitted symbol. The symbol duration shall be exactly 11 chips long.

The DBPSK encoder for the basic access rate is specified in Table 18-10. The DQPSK encoder is specified in Table 18-11. (In these tables, $+j\omega$ shall be defined as counterclockwise rotation.)

**Table 18-10—1 Mb/s DBPSK encoding table**

| Bit input | Phase change ($+j\omega$) |
|---|---|
| 0 | 0 |
| 1 | $\pi$ |

**Table 18-11—2 Mb/s DQPSK encoding table**

| Dibit pattern (d0,d1) (d0 is first in time) | Phase change ($+j\omega$) |
|---|---|
| 00 | 0 |
| 01 | $\pi/2$ |
| 11 | $\pi$ |
| 10 | $3\pi/2$ ($-\pi/2$) |

#### 18.4.6.5 Spreading sequences and modulation for CCK modulation at 5.5 Mb/s and 11 Mb/s

For the CCK modulation modes, the spreading code length is 8 and is based on complementary codes. The chipping rate is 11 Mchip/s. The symbol duration shall be exactly 8 complex chips long.


Copyright © 2007 IEEE. All rights reserved.

The following formula shall be used to derive the CCK code words that shall be used for spreading both 5.5 Mb/s and 11 Mb/s

$$c = \{e^{j(\varphi_1+\varphi_2+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_3+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_4)}, -e^{j(\varphi_1+\varphi_4)}, e^{j(\varphi_1+\varphi_2+\varphi_3)}, e^{j(\varphi_1+\varphi_3)}, -e^{j(\varphi_1+\varphi_2)}, e^{j\varphi_1}\} \quad (18\text{-}1)$$

where C is the code word

C = {c0 to c7}

The terms φ1, φ2, φ3, and φ4 are defined in 18.4.6.5.2 for 5.5 Mb/s and 18.4.6.5.3 for 11 Mb/s.

This formula creates 8 complex chips (c0 to c7), where c0 is transmitted first in time.

This is a form of the generalized Hadamard transform encoding, where φ1 is added to all code chips, φ2 is added to all odd code chips, φ3 is added to all odd pairs of code chips, and φ4 is added to all odd quads of code chips.

The term φ1 modifies the phase of all code chips of the sequence and shall be DQPSK encoded for 5.5 Mb/s and 11 Mb/s. This shall take the form of rotating the whole symbol by the appropriate amount relative to the phase of the preceding symbol. Note that the chip c7 of the symbol defined above is the chip that indicates the symbol's phase and is transmitted last.

#### 18.4.6.5.1 Cover code for CCK

The fourth and seventh chips are rotated 180° by a cover sequence to optimize the sequence correlation properties and minimize dc offsets in the codes. This can be seen by the minus sign on the fourth and seventh terms in Equation (18-1).

#### 18.4.6.5.2 CCK 5.5 Mb/s modulation

At 5.5 Mb/s, 4 bits (d0 to d3; d0 first in time) are transmitted per symbol.

The data bits d0 and d1 encode φ1 based on DQPSK. The DQPSK encoder is specified in Table 18-11. (In the table, +jω shall be defined as counterclockwise rotation.) The phase change for φ1 is relative to the phase φ1 of the preceding symbol. For the header to PSDU transition, the phase change for φ1 is relative to the phase of the preceding DQPSK (2 Mb/s) symbol. That is, the phase of the last symbol of the CRC-16 is the reference phase for the first symbol generated from the PSDU octets. (See the definition in 18.4.6.4 for the reference phase of this Barker coded symbol.) A "+1" chip in the Barker code shall represent the same carrier phase as a "+1" chip in the CCK code.

All odd-numbered symbols generated from the PSDU octets shall be given an extra 180 degree (π) rotation, in addition to the standard DQPSK modulation as shown in Table 18-12. The symbols of the PSDU shall be numbered starting with 0 for the first symbol, for the purposes of determining odd and even symbols. That is, the PSDU transmission starts on an even-numbered symbol.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 723 of 1232

**Table 18-12—DQPSK encoding table**

| Dibit pattern (d0, d1) (d0 is first in time) | Even symbols phase change (+jω) | Odd symbols phase change (+jω) |
|---|---|---|
| 00 | 0 | π |
| 01 | π/2 | 3π/2 (–π/2) |
| 11 | π | 0 |
| 10 | 3π/2 (–π/2) | π/2 |

The data dibits d2 and d3 CCK encode the basic symbol, as specified in Table 18-13. This table is derived from the formula above by setting $\varphi 2 = (d2 \times \pi) + \pi/2$, $\varphi 3 = 0$, and $\varphi 4 = d3 \times \pi$. In this table, d2 and d3 are in the order shown, and the complex chips are shown c0 to c7 (left to right), with c0 transmitted first in time.

**Table 18-13—5.5 Mb/s CCK encoding table**

| d2, d3 | c1 | c2 | c3 | c4 | c5 | c6 | c7 | c8 |
|---|---|---|---|---|---|---|---|---|
| 00 | 1j | 1 | 1j | –1 | 1j | 1 | –1j | 1 |
| 01 | –1j | –1 | –1j | 1 | 1j | 1 | –1j | 1 |
| 10 | –1j | 1 | –1j | –1 | –1j | 1 | 1j | 1 |
| 11 | 1j | –1 | 1j | 1 | –1j | 1 | 1j | 1 |

#### 18.4.6.5.3 CCK 11 Mb/s modulation

At 11 Mb/s, 8 bits (d0 to d7; d0 first in time) are transmitted per symbol.

The first dibit (d0, d1) encodes φ1 based on DQPSK. The DQPSK encoder is specified in Table 18-11. The phase change for φ1 is relative to the phase φ1 of the preceding symbol. In the case of header to PSDU transition, the phase change for φ1 is relative to the phase of the preceding DQPSK symbol. All odd-numbered symbols of the PSDU are given an extra 180 degree (π) rotation, in accordance with the DQPSK modulation shown in Table 18-11. Symbol numbering starts with 0 for the first symbol of the PSDU.

The data dibits (d2, d3), (d4, d5), and (d6, d7) encode φ2, φ3, and φ4, respectively, based on QPSK as specified in Table 18-14. Note that this table is binary (not Gray) coded.

**Table 18-14—QPSK encoding table**

| Dibit pattern [di, d(i+1)] (di is first in time) | Phase |
|---|---|
| 00 | 0 |
| 01 | π/2 |
| 10 | π |
| 11 | 3π/2 (–π/2) |


Copyright © 2007 IEEE. All rights reserved.

### 18.4.6.6 DSSS/PBCC data modulation and modulation rate (optional)

This optional coding scheme uses a binary convolutional coding with a 64-state binary convolutional code (BCC) and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as described in this subclause.

The encoder for this scheme is shown in Figure 18-12. Incoming data are first encoded with a binary convolutional code. A cover code is applied to the encoded data prior to transmission through the channel.

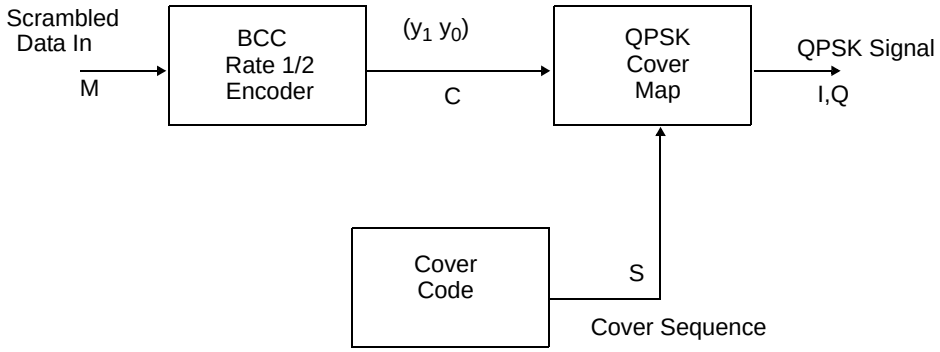



**Figure 18-12—PBCC modulator scheme**

The BCC that is used is a 64-state, rate ½ code. The generator matrix for the code is given as

$$G = [D^6 + D^4 + D^3 + D + 1, \quad D^6 + D^5 + D^4 + D^3 + D^2 + 1]$$

or in octal notation, it is given by

$$G = [133, \quad 175]$$

Because the system is frame (PPDU) based, the encoder shall be in state zero (i.e., all memory elements contain zero at the beginning of each PPDU). The encoder must also be placed in a known state at the end of each PPDU to prevent the data bits near the end of the PPDU from being substantially less reliable than those early on in the PPDU. To place the encoder in a known state at the end of a PPDU, at least six deterministic bits must be input immediately following the last data bit input to the convolutional encoder. This is achieved by appending one octet containing all zeros to the end of the PPDU prior to transmission, and discarding the final octet of each received PPDU. In this manner, the decoding process can be completed reliably on the last data bits.

An encoder block diagram is shown in Figure 18-13. It consists of six memory elements. For every data bit input, two output bits are generated.

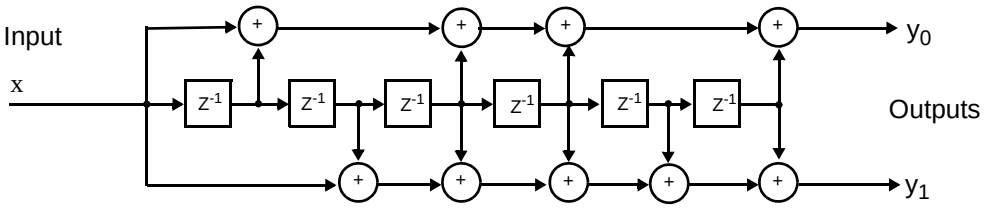



**Figure 18-13—PBCC convolutional encoder**

Copyright © 2007 IEEE. All rights reserved.

The output of the binary convolutional code described above is mapped to a constellation using one of two possible rates. The 5.5 Mb/s rate uses BPSK, and the 11 Mb/s rate uses QPSK. In QPSK mode, each pair of output bits from the binary convolutional code is used to produce one symbol; in BPSK mode, each pair of bits from the BCC is taken serially ($y_0$ first) and used to produce two BPSK symbols. This yields a throughput of one bit per symbol in QPSK mode and one-half a bit per symbol in BPSK mode.

The phase of the first complex chip of the PSDU shall be defined with respect to the phase of the last chip of the PCLP header (i.e., the last chip of the CRC check). The bits ($y_1$ $y_0$) = (0, 0) shall indicate the same phase as the last chip of the CRC check. The other three combinations of ($y_1$ $y_0$) shall be defined with respect to this reference phase, as shown in Figure 18-14.

The mapping from BCC outputs to PSK constellation points in BPSK and QPSK modes is determined by a pseudo-random cover sequence. This is shown for both modes in Figure 18-14. Note that this is an absolute phase table, not differential as in CCK.

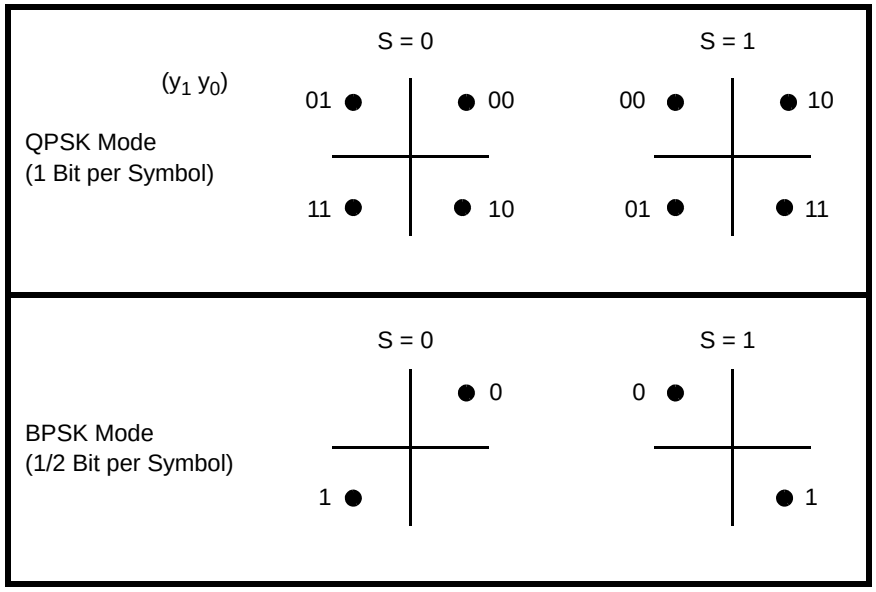



**Figure 18-14—Cover code mapping**

The pseudo-random cover sequence is generated from a seed sequence. The 16-bit seed sequence is 0011001110001011, where the first bit of the sequence in time is the leftmost bit. This sequence in octal notation is given as 150714, where the LSB is the first in time. This seed sequence is used to generate the 256-bit pseudo-random cover sequence, which is used in the mapping of the current PSK symbol. It is the current binary value of this sequence at every given point in time that is taken as S in Figure 18-14.

This sequence of 256 bits is produced by taking the first sixteen bits of the sequence as the seed sequence, the second sixteen bits as the seed sequence cyclically left rotated by three, the third sixteen bits as the seed sequence cyclically left rotated by six, etc. If c*i* is the $i^{th}$ bit of the seed sequence, where $0 \le I \le 15$, then the sequence that is used to cover the data is given row-wise as follows:

c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15
c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2
c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5
c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8
c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11
c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14


Copyright © 2007 IEEE. All rights reserved.

c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1
c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4
c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7
c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10
c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0
c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3
c7 c8 c9 c10 c11 c12 c13 c14 c15 c0 c1 c2 c3 c4 c5 c6
c10 c11 c12c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9
c13 c14 c15 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12

For PPDUs with more than 256 data bits, this sequence of 256 bits is simply repeated.

### 18.4.6.7 Channel Agility (optional)

This Channel Agility option allows an implementation to overcome some inherent difficulty with static channel assignments (a tone jammer), without burdening all implementations with the added cost of this capability. When the Channel Agility option is enabled, the PHY shall meet the requirements on channel switching and settling time, as described in 18.4.6.12, and the hop sequences described below. This option can also be used to implement IEEE 802.11-compliant systems that are interoperable between both FH and DS modulations. Annex F contains a description of the expected behavior when such networks are employed.

#### 18.4.6.7.1 Hop sequences

The hop sequences for each of the specified geographical areas are defined with two sets. High Rate frequency channels referred to in this subclause are defined in Table 18-9.

The first set (Figure 18-15 and Figure 18-17) uses nonoverlapping frequency channels to allow the High Rate systems to minimize interference degradation. The synchronization of FH is performed by the MLME, as defined in 11.1.5 for the FH PHY. The PLME SAP service primitives used to command a new frequency channel are defined in 10.4.

The second set (Figure 18-16 and Figure 18-18) uses half overlapping frequency channels, with 10 MHz center frequency spacing, to enable interoperability with 1 Mb/s and 2 Mb/s FH systems hopping with the approved IEEE 802.11 hop sequences. The High Rate hop frequency is calculated from the specific 1 MHz channel chosen for a given hop by picking the closest High Rate channel within the set. Where there is a choice of two DSSS channels, the lower one shall be the one chosen. Therefore, the chosen channel shall be no more than ±5 MHz of the channel center of the FH channel. When operating on the FH channels beyond ±5 MHz of the closest High Rate channel specified in the set, the High Rate mode shall not be used and all FH transmissions shall occur at the 1 Mb/s or 2 Mb/s rate.

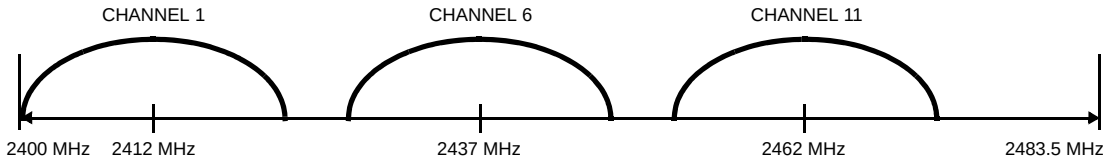



**Figure 18-15—China and North American channel selection—nonoverlapping**

Copyright © 2007 IEEE. All rights reserved.

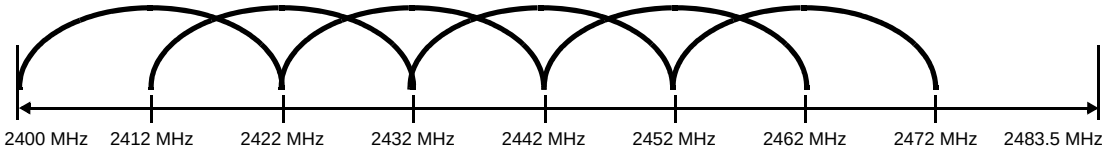



**Figure 18-16—China and North American channel selection—overlapping**

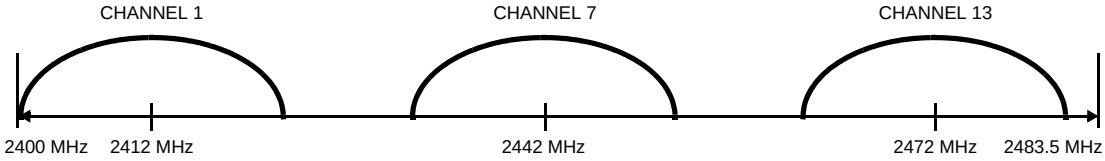



**Figure 18-17—European channel selection—nonoverlapping**

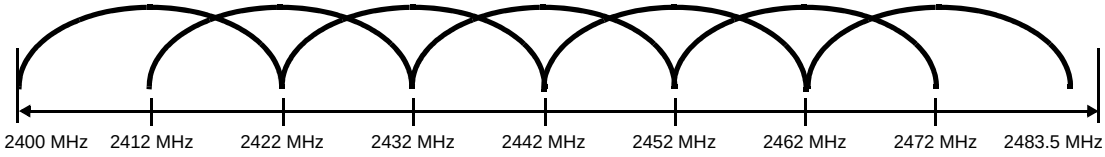



**Figure 18-18—European channel selection—overlapping**

### 18.4.6.7.2 Operating channels

The operating channels for specified geographical areas are defined in Table 18-15 and Table 18-16.

**Table 18-15—China and North American operating channels**

| Set | Number of channels | HR/DSSS channel numbers |
|---|---|---|
| 1 | 3 | 1, 6, 11 |
| 2 | 6 | 1, 3, 5, 7, 9, 11 |

**Table 18-16—European operating channels (except France and Spain)**

| Set | Number of channels | HR/DSS channel numbers |
|---|---|---|
| 1 | 3 | 1, 7, 13 |
| 2 | 7 | 1, 3, 5, 7, 9, 11, 13 |


Copyright © 2007 IEEE. All rights reserved.

#### 18.4.6.7.3 Hop patterns

An FH pattern, Fx, consists of a permutation of all frequency channels defined in Table 18-15 and Table 18-16. For a given pattern number *x*, the hopping sequence can be written as

$Fx = \{fx(1), fx(2), ..., fx(p)\}$

where

$fx(i)$ is the channel number (as defined in 14.6.4) for $i^{th}$ frequency in the $x^{th}$ hopping pattern

$p$ is the number of hops in pseudo-random hopping pattern before repeating sequence (79 for North America and most of Europe)

The FH patterns for Set 1 of each geographic area are based on the hop patterns in Table 18-17 and Table 18-18.

**Table 18-17—China and North American Set 1 hop patterns**

| Index | Pattern 1 | Pattern 2 |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 6 | 11 |
| 3 | 11 | 6 |

**Table 18-18—European Set 1 hop patterns (except France and Spain)**

| Index | Pattern 1 | Pattern 2 |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 7 | 13 |
| 3 | 13 | 7 |

The FH patterns for Set 2 of each geographic area are defined by the 1/2 Mb/s FH PHY hop sequences, as described in the FH PHY (14.6.8). Given the hopping pattern number *x*, and the index for the next frequency, *i* (in the range 1 to p), the DS channel number (as defined in 18.4.6.2) shall be selected with the following algorithm:

China and North America

$f'x(i) = f'x(i)$ for $1 \le f'x(i) \le 11$;

$f'x(i)$ = null for $f'x(i) < 1$ and $f'x(i) > 11$;

$f'x(i) = 2 \times \text{Int}[(\{[b(i) + x] \bmod (79) + 2\} - 6) / 10] - 1$;

with $b(i)$ defined in Table 14-11 (in 14.6.8).

Most of Europe

$f'x(i) = f'x(i)$ for $1 \le f'x(i) \le 13$;

$f'x(i)$ = null for $f'x(i) < 1$ and $f'x(i) > 13$;

$f'x(i) = 2 \times \text{Int}[(\{[b(i) + x] \bmod (79) + 2\} - 6) / 10] - 1$;

with $b(i)$ defined in Table 14-11 (in 14.6.8).

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 729 of 1232

#### 18.4.6.8 Transmit and receive in-band and out-of-band spurious emissions

The High Rate PHY conforms with in-band and out-of-band spurious emissions as set by the appropirate regulatory bodies.

#### 18.4.6.9 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than 10 μs, including the power-down ramp specified in 18.4.7.6.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol to the valid CCA detection of the incoming signal. The CCA should occur within 25 μs (10 μs for turnaround time, plus 15 μs for energy detect), or by the next slot boundary occurring after the 25 μs has elapsed (see 18.4.8.4). A receiver input signal 3 dB above the ED threshold described in 18.4.8.4 shall be present at the receiver.

#### 18.4.6.10 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured at the MAC/PHY interface using PHYTXSTART.request, and shall be 5 μs. This includes the transmit power-on ramp described in 18.4.7.6.

#### 18.4.6.11 Slot time

The slot time for the High Rate PHY shall be the sum of the RX-to-TX turnaround time (5 μs) and the energy detect time (15 μs specified in 18.4.8.4). The propagation delay shall be regarded as being included in the energy detect time.

#### 18.4.6.12 Channel switching/settling time

When the Channel Agility option is enabled, the time to change from one operating channel frequency to another, as specified in 18.4.6.2, is 224 μs. A conformant PMD meets this switching time specification when the operating channel center frequency has settled to within ±60 kHz of the nominal channel center. STAs shall not transmit until after the channel change settling time.

#### 18.4.6.13 Transmit and receive antenna port impedance

The impedance of the transmit and receive antenna port(s) shall be 50 Ω if the port is exposed.

#### 18.4.6.14 Transmit and receive operating temperature range

Two temperature ranges are specified for full operation compliance to the High Rate PHY. Type 1 shall be defined as 0 °C to 40 °C, and is designated for office environments. Type 2 shall be defined as –30 °C to +70 °C, and is designated for industrial environments.

### 18.4.7 PMD transmit specifications

The transmit functions and parameters associated with the PMD sublayer are described in 18.4.7.1 through 18.4.7.8.

#### 18.4.7.1 Transmit power levels

The maximum allowable output power is measured in accordance with practices specified by the appropriate regulatory bodies.


Copyright © 2007 IEEE. All rights reserved.

### 18.4.7.2 Transmit power level control

Power control shall be provided for transmitted power greater than 100 mW. A maximum of four power levels may be provided. As a minimum, a radio capable of transmission greater than 100 mW shall be capable of switching power back to 100 mW or less.

### 18.4.7.3 Transmit spectrum mask

The transmitted spectral products shall be less than –30 dBr (decibel relative to the SINx/x peak) for

$f_c$ – 22 MHz < $f$ < $f_c$ –11 MHz; and

$f_c$ + 11 MHz < $f$ < $f_c$ + 22 MHz;

and shall be less than –50 dBr for

$f$ < $f_c$ – 22 MHz; and

$f$ > $f_c$ + 22 MHz.

where

$f_c$ is the channel center frequency

The transmit spectral mask is shown in Figure 18-19. The measurements shall be made using a 100 kHz resolution bandwidth and a 100 kHz video bandwidth.

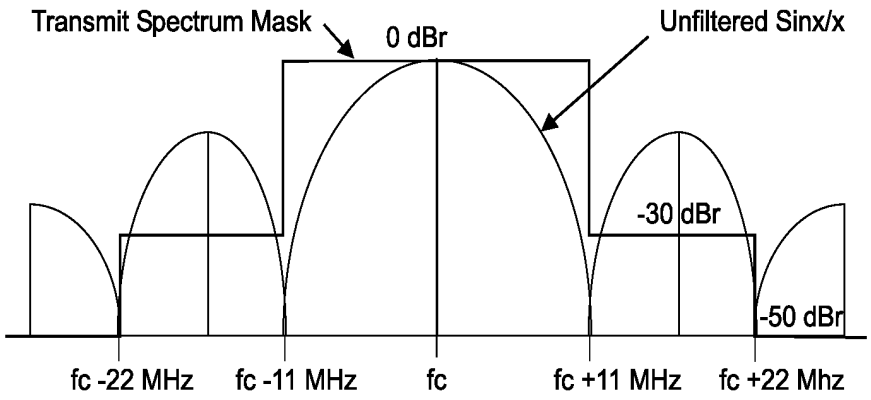



**Figure 18-19—Transmit spectrum mask**

### 18.4.7.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

### 18.4.7.5 Chip clock frequency tolerance

The PN code chip clock frequency tolerance shall be better than ±25 ppm maximum. It is highly recommended that the chip clock and the transmit frequency be locked (coupled) for optimum demodulation performance. If these clocks are locked, it is recommended that bit 2 of the SERVICE field be set to 1, as indicated in 18.2.3.4.

Copyright © 2007 IEEE. All rights reserved.

### 18.4.7.6 Transmit power-on and power-down ramp

The transmit power-on ramp for 10% to 90% of maximum power shall be no greater than 2 μs. The transmit power-on ramp is shown in Figure 18-20.

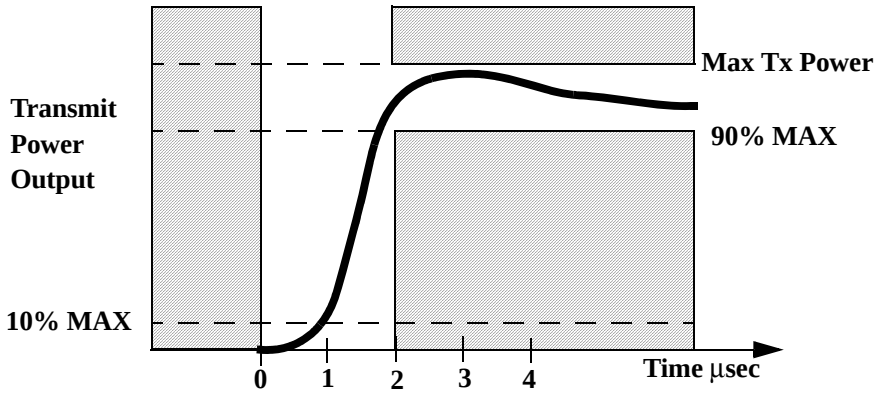



**Figure 18-20—Transmit power-on ramp**

The transmit power-down ramp for 90% to 10% maximum power shall be no greater than 2 μs. The transmit power-down ramp is shown in Figure 18-21.

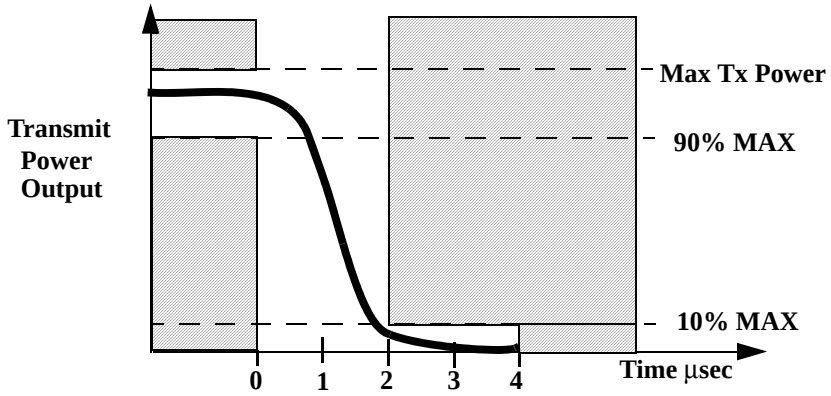



**Figure 18-21—Transmit power-down ramp**

The transmit power ramps shall be constructed such that the High Rate PHY emissions conform with spurious frequency product specification defined in 18.4.6.8.

### 18.4.7.7 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak SIN(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.


Copyright © 2007 IEEE. All rights reserved.

### 18.4.7.8 Transmit modulation accuracy

The transmit modulation accuracy requirement for the High Rate PHY shall be based on the difference between the actual transmitted waveform and the ideal signal waveform. Modulation accuracy shall be determined by measuring the peak vector error magnitude during each chip period. Worst-case vector error magnitude shall not exceeded 0.35 for the normalized sampled chip data. The ideal complex I and Q constellation points associated with DQPSK modulation, (0.707, 0.707), (0.707, –0.707), (–0.707, 0.707), (–0.707, –0.707), shall be used as the reference. These measurements shall be from baseband I and Q sampled data after recovery through a reference receiver system.

Figure 18-22 illustrates the ideal DQPSK constellation points and range of worst-case error specified for modulation accuracy.

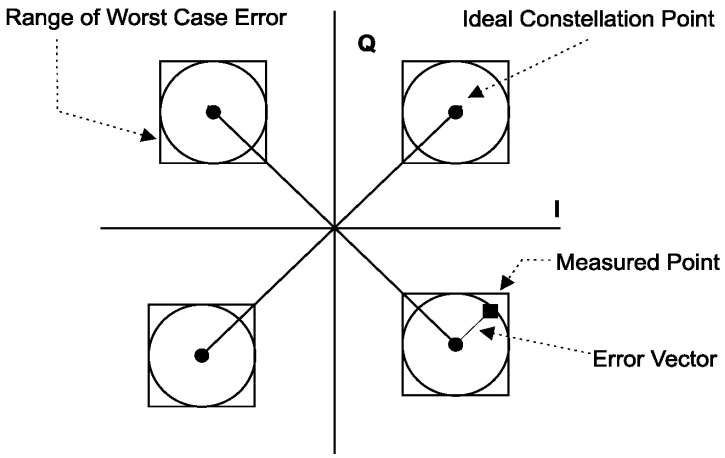



**Figure 18-22—Modulation accuracy measurement example**

Error vector measurement requires a reference receiver capable of carrier lock. All measurements shall be made under carrier lock conditions. The distortion induced in the constellation by the reference receiver shall be calibrated and measured. The test data error vectors described below shall be corrected to compensate for the reference receiver distortion.

The IEEE 802.11-compatible radio shall provide an exposed TX chip clock, which shall be used to sample the I and Q outputs of the reference receiver.

The measurement shall be made under the conditions of continuous DQPSK transmission using scrambled all ones.

The eye pattern of the I channel shall be used to determine the I and Q sampling point. The chip clock provided by the vendor radio shall be time delayed, such that the samples fall at a 1/2 chip period offset from the mean of the zero crossing positions of the eye (see Figure 18-23). This is the ideal center of the eye and may not be the point of maximum eye opening.

Copyright © 2007 IEEE. All rights reserved.

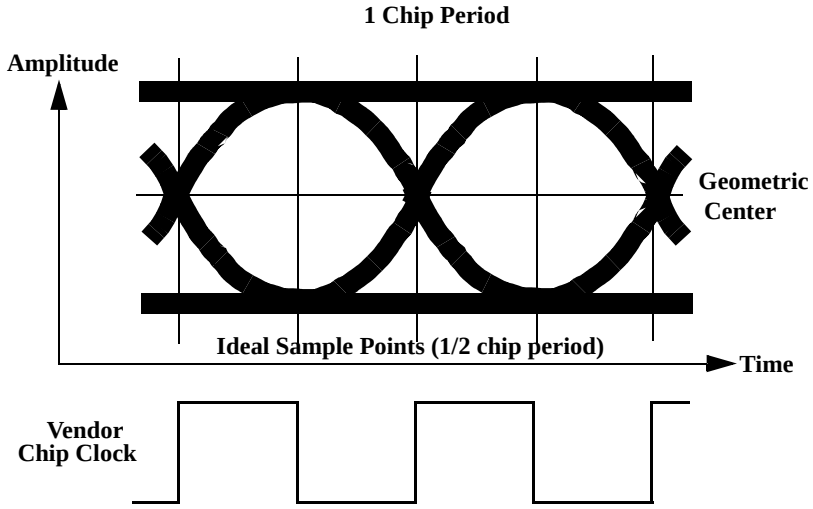



**Figure 18-23—Chip clock alignment with baseband eye pattern**

Using the aligned chip clock, 1000 samples of the I and Q baseband outputs from the reference receiver are captured. The vector error magnitudes shall be calculated as follows:

Calculate the dc offsets for I and Q samples

$$I_{\text{mean}} = \sum_{n=0}^{999} I(n) / 1000$$

$$Q_{\text{mean}} = \sum_{n=0}^{999} Q(n) / 1000$$

Calculate the dc corrected I and Q samples for all n = 1000 sample pairs

$$I_{dc}(n) = I(n) - I_{\text{mean}}$$

$$Q_{dc}(n) = Q(n) - Q_{\text{mean}}$$

Calculate the average magnitude of I and Q samples

$$I_{\text{mag}} = \sum_{n=0}^{999} \left| I_{dc}(n) \right| / 1000$$

$$Q_{\text{mag}} = \sum_{n=0}^{999} \left| Q_{dc}(n) \right| / 1000$$


Copyright © 2007 IEEE. All rights reserved.

Calculate the normalized error vector magnitude for the $I_{dc}(n)/Q_{dc}(n)$ pairs

$$V_{err}(n) = [\{|I_{dc}(n)|/I_{mag} - 1\}^2 + \{|Q_{dc}(n)|/Q_{mag} - 1\}^2]^{\frac{1}{2}} - V_{correction}$$

where

$V_{correction}$ is the error induced by the reference receiver system

A vendor High Rate PHY implementation shall be compliant if for all n = 1000 samples, the following condition is met:

$V_{err}(n) < 0.35$

### 18.4.8 PMD receiver specifications

The receive functions and parameters associated with the PMD sublayer are described in 18.4.8.1 through 18.4.8.4.

#### 18.4.8.1 Receiver minimum input level sensitivity

The FER shall be less than $8\times10^{-2}$ at a PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. This FER shall be specified for 11 Mb/s CCK modulation. The test for the minimum input level sensitivity shall be conducted with the ED threshold set less than or equal to –76 dBm.

#### 18.4.8.2 Receiver maximum input level

The receiver shall provide a maximum FER of $8\times10^{-2}$ at a PSDU length of 1024 octets for a maximum input level of –10 dBm measured at the antenna. This FER shall be specified for 11 Mb/s CCK modulation.

#### 18.4.8.3 Receiver adjacent channel rejection

Adjacent channel rejection is defined between any two channels with ≥ 25 MHz separation in each channel group, as defined in 18.4.6.2.

The adjacent channel rejection shall be equal to or better than 35 dB, with an FER of $8\times10^{-2}$ using 11 Mbit/s CCK modulation described in 18.4.6.3 and a PSDU length of 1024 octets.

The adjacent channel rejection shall be measured using the following method.

Input an 11 Mb/s CCK modulated signal at a level 6 dB greater than specified in 18.4.8.1. In an adjacent channel (≥ 25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion, which adheres to the transmit mask specified in 18.4.7.3, to a level 41 dB above the level specified in 18.4.8.1. The adjacent channel signal shall be derived from a separate signal source. It cannot be a frequency shifted version of the reference channel. Under these conditions, the FER shall be no worse than $8\times10^{-2}$.

#### 18.4.8.4 CCA

The High Rate PHY shall provide the capability to perform CCA according to at least one of the following three methods:

— CCA Mode 1: Energy above threshold. CCA shall report a busy medium upon detecting any energy above the ED threshold.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 735 of 1232

— CCA Mode 4: CS with timer. CCA shall start a timer whose duration is 3.65 ms and report a busy medium only upon the detection of a High Rate PHY signal. CCA shall report an IDLE medium after the timer expires and no High Rate PHY signal is detected. The 3.65 ms timeout is the duration of the longest possible 5.5 Mb/s PSDU.

— CCA Mode 5: A combination of CS and energy above threshold. CCA shall report busy at least while a High Rate PPDU with energy above the ED threshold is being received at the antenna.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP to indicate activity to the MAC through the PHY interface primitive, PHY-CCA.indicate.

A busy channel shall be indicated by PHY-CCA.indicate of class BUSY. A clear channel shall be indicated by PHY-CCA.indicate of class IDLE.

The PHY MIB attribute, dot11CCAModeSupported, shall indicate the appropriate operation modes. The PHY shall be configured through the PHY MIB attribute, dot11CurrentCCAMode.

The CCA shall indicate TRUE if there is no energy detect or CS. The CCA parameters are subject to the following criteria:

a) If a valid High Rate signal is detected during its preamble within the CCA window, the ED threshold shall be less than or equal to –76 dBm for TX power > 100 mW; –73 dBm for 50 mW < TX power ≤ 100 mW; and –70 dBm for TX power ≤ 50 mW.

b) With a valid signal (according to the CCA mode of operation) present at the receiver antenna within 5 μs of the start of a MAC slot boundary, the CCA indicator shall report channel busy before the end of the slot time. This implies that the CCA signal is available as an exposed test point. Refer to Figure 9-12 (in 9.2.10) for a slot time boundary definition.

c) In the event that a correct PLCP header is received, the High Rate PHY shall hold the CCA signal inactive (channel busy) for the full duration, as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted PPDU. Upon reception of a correct PLCP header, the timer of CCA Mode 2 shall be overridden by this requirement.

Conformance to the High Rate PHY CCA shall be demonstrated by applying an equivalent High-Rate-compliant signal above the appropriate ED threshold (item a) so that all conditions described in item b and item c are demonstrated.


Copyright © 2007 IEEE. All rights reserved.

# 19. ERP specification

## 19.1 Overview

This clause specifies further rate extension of the PHY for the DSSS system of Clause 15 and the extensions of Clause 18. Hereinafter the PHY defined in this clause will be known as the ERP. This PHY operates in the 2.4 GHz ISM band.

### 19.1.1 Introduction

The ERP builds on the payload data rates of 1 and 2 Mb/s, as described in Clause 15, that use DSSS modulation and builds on the payload data rates of 1, 2, 5.5, and 11 Mb/s, as described in Clause 18, that use DSSS, CCK, and optional PBCC modulations. The ERP draws from Clause 17 to provide additional payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. Of these rates, transmission and reception capability for 1, 2, 5.5, 6, 11, 12, and 24 Mb/s data rates is mandatory.

Two additional optional ERP-PBCC modulation modes with payload data rates of 22 and 33 Mb/s are defined. An ERP-PBCC STA may implement 22 Mb/s alone or 22 and 33 Mb/s. An optional modulation mode known as DSSS-OFDM is also incorporated with payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s.

### 19.1.2 Operational modes

The radio portion of all Clause 19-compliant ERP systems implements all mandatory modes of Clause 17 and Clause 18, except it uses the 2.4 GHz frequency band and channelization plan specified in 18.4.6. The ERP has the capability to decode all Clause 15 and Clause 18 PLCPs and all ERP-OFDM PLCPs. In addition, it is mandatory that all ERP-compliant equipment be capable of sending and receiving the short preamble that is (and remains) optional for Clause 18 PHYs.

The ERP has the capability to detect ERP and Clause 18 preambles whenever a CCA is requested. Because protection mechanisms are not required in all cases, the ERP CCA mechanisms for all preamble types shall be active at all times.

An ERP BSS is capable of operating in any combination of available ERP modes (Clause 19 PHYs) and NonERP modes (Clause 15 or Clause 18 PHYs). For example, a BSS could operate in an ERP-OFDM-only mode, a mixed mode of ERP-OFDM and ERP-DSSS/CCK, or a mixed mode of ERP-DSSS/CCK and NonERP. When options are enabled, combinations are also allowed.

The changes to other parts of this standard required to implement the ERP are summarized as follows:

a) ERP-DSSS/CCK
   1) The PHY uses the capabilities of Clause 18 with the following exceptions:
      i) Support of the short PLCP PPDU header format capability of 18.2.2.2 is mandatory.
      ii) CCA (see 18.4.8.4) has a mechanism that will detect all mandatory Clause 19 sync symbols.
      iii) The maximum input signal level (see 18.4.8.2) is –20 dBm.
      iv) Locking the transmit center frequency and the symbol clock frequency to the same reference oscillator is mandatory.

b) ERP-OFDM
   1) The PHY uses the capabilities of Clause 17 with the following exceptions:
      i) The frequency plan is in accordance with 18.4.6.1 and 18.4.6.2 instead of 17.3.8.3.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 737 of 1232

ii) CCA has a mechanism that will detect all mandatory Clause 19 sync symbols.

iii) The frequency accuracy (see 17.3.9.4 and 17.3.9.5) is ±25 PPM.

iv) The maximum input signal level (see 17.3.10.4) is –20 dBm.

v) The slot time is 20 μs in accordance with 18.3.3, except that an optional 9 μs slot time may be used when the BSS consists of only ERP STAs.

vi) SIFS time is 10 μs in accordance with 18.3.3. See 19.3.2.3 for more detail.

c) ERP-PBCC (Optional)

1) This is a single carrier modulation scheme that encodes the payload using a 256-state packet binary convolutional code. These are extensions to the PBCC modulation in Clause 18. ERP-PBCC modes with payload data rates of 22 Mb/s and 33 Mb/s are defined in 19.6.

d) DSSS-OFDM (Optional)

1) This is a hybrid modulation combining a DSSS preamble and header with an OFDM payload transmission. DSSS-OFDM modes with payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s are defined in 19.7.

2) If the optional DSSS-OFDM mode is used, the supported rates in that mode are the same as the ERP-OFDM supported rates.

The 2.4 GHz ISM band is a shared medium, and coexistence with other devices such as Clause 15 and Clause 18 STAs is an important issue for maintaining high performance in Clause 19 (ERP) STAs. The ERP modulations (ERP-OFDM, ERP-PBCC, and DSSS-OFDM) have been designed to coexist with existing Clause 15 and Clause 18 STAs. This coexistence is achieved by several means, including virtual CS (RTS/CTS or CTS-to-self), CSMA/CA protocols, and MSDU fragmentation.

### 19.1.3 Scope

This clause specifies the ERP entity and the deviations from earlier clauses to accommodate it. It is organized by reference to the relevant earlier clauses to avoid excessive duplication.

The ERP consists of the following two protocol functions:

a) A physical layer convergence function that adapts the capabilities of the PMD system to the PHY service available. This function is supported by the PLCP, which defines a method for mapping the MPDUs into a framing format suitable for sending and receiving user data and management information between two or more STAs using the associated PMD system. The PHY exchanges PPDUs that contain PSDUs. The MAC uses the PHY service, so each MPDU corresponds to a PSDU that is carried in a PPDU.

b) A PMD system, whose function defines the characteristics and method of transmitting and receiving data through a WM between two or more STAs; each using the ERP.

### 19.1.4 ERP functions

The architecture of the ERP is depicted in the ISO/IEC basic reference model shown in Figure 18-11 of 18.4.1. The ERP contains three functional entities: the PMD function, the PLCP, and the layer management function.

The ERP service is provided to the MAC through the PHY service primitives described in Clause 12. Interoperability is addressed by use of the CS mechanism specified in 9.2.1 and the protection mechanism in 9.13. This mechanism allows NonERP STAs to know of ERP traffic that they cannot demodulate so that they may defer the medium to that traffic.


Copyright © 2007 IEEE. All rights reserved.

## 19.2 PHY-specific service parameter list

The architecture of the IEEE 802.11 MAC is intended to be PHY independent. Some PHY implementations require PHY-dependent MAC state machines running in the MAC sublayer in order to meet certain PMD requirements. The PHY-dependent MAC state machine resides in a sublayer defined as the MLME. In certain PMD implementations, the MLME may need to interact with the PLME as part of the normal PHY SAP primitives. These interactions are defined by the PLME parameter list currently defined in the PHY service primitives as TXVECTOR and RXVECTOR. The list of these parameters and the values they may represent are defined in the specific PHY specifications for each PMD. This subclause addresses the TXVECTOR and RXVECTOR for the ERP. The service parameters for RXVECTOR and TXVECTOR shall follow 17.2.2 and 17.2.3, respectively.

Several service primitives include a parameter vector. DATARATE and LENGTH are described in 12.3.4.4. The remaining parameters are considered to be management parameters and are specific to this PHY.

The parameters in Table 19-1 are defined as part of the TXVECTOR parameter list in the PHY-TXSTART.request service and PLME_TXTIME.request primitives.

**Table 19-1—TXVECTOR parameters**

| Parameter | Value |
|---|---|
| DATARATE | The rate used to transmit the PSDU in Mb/s.<br>Allowed value depends on value of MODULATION parameter:<br>ERP-DSSS: 1 and 2<br>ERP-CCK: 5.5 and 11<br>ERP-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54<br>ERP-PBCC: 5.5, 11, 22, and 33<br>DSSS-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54 |
| LENGTH | The length of the PSDU in octets.<br>Range: 1–4095 |
| PREAMBLE_TYPE | The preamble used for the transmission of the PPDU.<br>Enumerated type for which the allowed value depends on value of MODULATION parameter:<br>ERP-OFDM: null<br>ERP-DSSS, ERP-CCK, ERP-PBCC, DSSS-OFDM: SHORTPREAMBLE, LONGPREAMBLE |
| MODULATION | The modulation used for the transmission of this PSDU.<br>Enumerated type: ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM |
| SERVICE | The scrambler initialization vector.<br>When the modulation format selected is ERP-OFDM or DSSS-OFDM, seven null bits are used for scrambler initialization as described in 17.3.5.1. The remaining bits are reserved.<br>For all other ERP modulations that all start with ERP-DSSS short or long preamble, the bits of the SERVICE field are defined in Table 19-3 and the SERVICE field is not applicable in the TXVECTOR. Therefore, the entire field is reserved. |
| TXPWR_LEVEL | The transmit power level. The definition of these levels is up to the implementer.<br>1–8 |

Copyright © 2007 IEEE. All rights reserved.

The parameters in Table 19-2 are defined as part of the RXVECTOR parameter list in the PHY-RXSTART.indicate service primitive. When implementations require the use of these vectors, some or all of these parameters may be used in the vectors.

**Table 19-2—RXVECTOR parameters**

| Parameter | Value |
|---|---|
| DATARATE | The rate at which the PSDU was received in Mb/s.<br>Allowed value depends on value of MODULATION parameter:<br>ERP-DSSS: 1 and 2<br>ERP-CCK: 5.5 and 11<br>ERP-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54<br>ERP-PBCC: 5.5, 11, 22, and 33<br>DSSS-OFDM: 6, 9, 12, 18, 24, 36, 48, and 54 |
| LENGTH | The length of the PSDU in octets.<br>Range: 1–4095 |
| PREAMBLE_TYPE | The preamble type detected during reception of the PPDU.<br>Enumerated type for which the allowed value depends on value of MODULATION parameter:<br>ERP-OFDM: null<br>ERP-DSSS, ERP-CCK, ERP-PBCC, PBCC, DSSS-OFDM: SHORTPREAMBLE, LONGPREAMBLE. |
| MODULATION | The modulation used for the reception of this PSDU.<br>Enumerated types: ERP-DSSS, ERP-CCK, ERP-OFDM, ERP-PBCC, DSSS-OFDM |
| SERVICE | Null. |
| RSSI | The RSSI is a measure of the RF energy received by the ERP. The 8-bit value is in the range of 0 to RSSI maximum as described in 17.2.3.2. |

## 19.3 Extended Rate PLCP sublayer

### 19.3.1 Introduction

This subclause provides a PLCP for the ERP. The convergence procedure specifies how PSDUs are converted to and from PPDUs at the transmitter and receiver. The PPDU is formed during data transmission by appending the PSDU to the Extended Rate PLCP preamble and header. At the receiver, the PLCP preamble and header are processed to aid in the demodulation and delivery of the PSDU.

### 19.3.2 PPDU format

An ERP STA shall support three different preamble and header formats. The first is the long preamble and header described in 19.3.2.1 (and based on 18.2.2.1 with redefinition of reserved bits defined therein). This PPDU provides interoperability with Clause 18 STAs when using the 1, 2, 5.5, and 11 Mb/s data rates; the optional DSSS-OFDM modulation at all OFDM rates; and the optional ERP-PBCC modulation at all ERP-PBCC rates. The second is the short preamble and header described in 19.3.2.2 (and based on 18.2.2.2 where it is optional). The short preamble supports the rates 2, 5.5, and 11 Mb/s as well as DSSS-OFDM and ERP-PBCC. The third is the ERP-OFDM preamble and header specified in 19.3.2.3 (and based on 17.3.2). The ERP has two optional PPDU formats, described in 19.3.2.4 and 19.3.2.5, to support the optional DSSS-OFDM modulation rates.


Copyright © 2007 IEEE. All rights reserved.

#### 19.3.2.1 Long preamble PPDU format

Figure 18-1 of 18.2.2.1 shows the basic format for the long preamble PPDU. This preamble is appropriate for use with the 1, 2, 5.5, and 11 Mb/s (Clause 18) modes and is compatible with BSSs using these modes. To support the optional modes included in the ERP, the long preamble PPDU only differs from 18.2.2.1 in the following:

a) The use of one bit in the SERVICE field to indicate when the optional ERP-PBCC mode is being used.

b) The use of two additional bits in the SERVICE field to resolve the length ambiguity when the optional ERP-PBCC-22 and ERP-PBCC-33 modes are being used.

c) Three additional optional rates given by the following SIGNAL field octets where the LSB is transmitted first in time:

   1) X'DC' (MSB to LSB) for 22 Mb/s ERP-PBCC
   2) X'21' (MSB to LSB) for 33 Mb/s ERP-PBCC
   3) X'1E' (MSB to LSB) for all DSSS-OFDM rates

Three bits of the SERVICE field have been defined to support the optional modes of the ERP standard. Table 19-3 shows graphically the assignment of the bits within the SERVICE field. The bits b0, b1, and b4 are reserved and shall be set to 0. Bit b2 is used to indicate that the transmit frequency and symbol clocks are derived from the same oscillator. For all ERP systems, the Locked Clock Bit shall be set to 1, when transmitting at an ERP-PBCC rate or at a data rate described in Clause 18. Bit b3 is used to indicate if the data are modulated using the optional ERP-PBCC modulation. Bit b3 is defined in 18.2.3.4 with the caveat that the ERP-PBCC mode now has the additional optional rates of 22 Mb/s and 33 Mb/s as defined in 19.3.3.2. Bits b5, b6, and b7 are used to resolve data field length ambiguities for the optional ERP-PBCC-11 through ERP-PBCC-33 modes. These bits are fully defined in 19.6. Bit b7 is also used to resolve data field length ambiguities for the CCK 11 Mb/s mode and is defined in 18.2.3.5. Bits b3, b5, and b6 are set to 0 for CCK.

**Table 19-3—SERVICE field bit definitions**

| b0 | b1 | b2 | b3 | b4 | b5 | b6 | b7 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Locked Clock Bit<br>0 = not locked<br>1 = locked | Modulation Selection<br>0 = Not ERP-PBCC<br>1 = ERP-PBCC | Reserved | Length Extension Bit (ERP-PBCC) | Length Extension Bit (ERP-PBCC) | Length Extension Bit |

##### 19.3.2.1.1 ERP PLCP length field calculation

For the long and short preamble modes other than PBCC, the length field shall be calculated as in 18.2.3.5.

##### 19.3.2.1.2 ERP-PBCC PLCP length (LENGTH) field calculation

For the ERP-PBCC PLCP length field, the transmitted value shall be determined from the LENGTH and DataRate parameters in the TXVECTOR issued with the PMD-TXSTART.request primitive described in 18.4.5.6.

The length field provided in the TXVECTOR is in octets and is converted to microseconds for inclusion in the PLCP LENGTH field. The Length Extension bits are provided to resolve the ambiguity in the number of octets that is described by an integer number of microseconds for any data rate over 8 Mb/s. These bits are used to indicate which of the smaller potential number of octets is correct.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 741 of 1232

— 11 Mb/s PBCC: see 18.2.3.5.

— 22 Mb/s ERP-PBCC: Length = (number of octets + 1) × 4/11, rounded up to the next integer; the SERVICE field bits b6 and b7 shall each indicate a 0 if the rounding took less than 4/11; the SERVICE field bit b6 shall indicate a 0, and b7 shall indicate a 1 if the rounding took 4/11 or more and less than 8/11; and the SERVICE field bit b6 shall indicate a 1, and b7 shall indicate a 0 if the rounding took 8/11 or more.

— 33 Mb/s ERP-PBCC: Length = (number of octets + 1) × 8/33, rounded up to the next integer; the SERVICE field bits b5, b6, and b7 shall each indicate a 0 if the rounding took less than 8/33; the SERVICE field bit b5 shall indicate a 0, b6 shall indicate a 0, and b7 shall indicate a 1 if the rounding took more than or equal to 8/33 and less than 16/33; the SERVICE field bit b5 shall indicate 0, b6 shall indicate a 1, and b7 shall indicate a 0 if the rounding took more than or equal to 16/33 and less than 24/33; the SERVICE field bit b5 shall indicate 0, b6 shall indicate a 1, and b7 shall indicate a 1 if the rounding took more than or equal to 24/33 and less than 32/33; the SERVICE field bit b5 shall indicate 1, b6 shall indicate a 0, and b7 shall indicate a 0 if the rounding took 32/33 or more.

At the receiver, the number of octets in the MPDU is calculated as follows:

— 22 Mb/s ERP-PBCC: Number of octets = (Length × 11/4) – 1, rounded down to the next integer, minus 1 if the SERVICE field bit b6 is a 0 and b7 is a 1, or minus 2 if the SERVICE field bit b6 is a 1 and b7 is a 0.

— 33 Mb/s ERP-PBCC: Number of octets = (Length × 33/8) – 1, rounded down to the next integer, minus 1 if the SERVICE field bit b5 is a 0, b6 is a 0, and b7 is a 1, or minus 2 if the SERVICE field bit b5 is a 0, b6 is a 1, and b7 is a 0, or minus 3 if the SERVICE field bit b5 is a 0, b6 is a 1, and b7 is a 1, or minus 4 if the SERVICE field bit b5 is a 1, b6 is a 0, and b7 is a 0.

Table 19-4 shows an example calculation for several packet lengths of ERP-PBCC at 22 Mb/s.

**Table 19-4—Example of LENGTH calculations for ERP-PBCC-22**

| TX Octets | (Octets+1) × 4/11 | LENGTH | Length Extension bit b6 | Length Extension bit b7 | LENGTH × 11/4 | floor(X) | RX Octets |
|---|---|---|---|---|---|---|---|
| 1023 | 372.364 | 373 | 0 | 1 | 1025.75 | 1025 | 1023 |
| 1024 | 372.727 | 373 | 0 | 0 | 1025.75 | 1025 | 1024 |
| 1025 | 373.091 | 374 | 1 | 0 | 1028.50 | 1028 | 1025 |
| 1026 | 373.455 | 374 | 0 | 1 | 1028.50 | 1028 | 1026 |

### 19.3.2.2 Short preamble PPDU format

Figure 18-2 of 18.2.2.2 shows the basic format for the short preamble PPDU. For the ERP, support for this preamble is mandatory. The short preamble is appropriate for use with 2, 5.5, and 11 Mb/s modes. The bits of the Short PLCP SERVICE field and RATE field are the same as for the Long PLCP SERVICE field and RATE field and are defined in 19.3.2.1.


Copyright © 2007 IEEE. All rights reserved.

#### 19.3.2.3 ERP-OFDM PPDU format

The format, preamble, and headers for the ERP-OFDM PLCP PPDU are described in 17.3.2 through 17.3.5. For the ERP-OFDM modes, the DATA field that contains the SERVICE field, the PSDU, the TAIL bits, and the PAD bits shall follow 17.3.5.

For ERP-OFDM modes, an ERP packet is followed by a period of no transmission with a length of 6 μs called the signal extension. The purpose of this extension is to make the TXTIME calculation in 19.8.3 result in a transmission duration interval that includes an additional 6 μs. The SIFS time for Clause 17 packets is 16 μs, and the SIFS time for Clause 18 packets is 10 μs. The longer SIFS time in Clause 17 is to allow extra time for the convolutional decode process to finish. As Clause 19 packets will use a SIFS time of 10 μs, this extra 6 μs length extension is used to ensure that the transmitter computes the Duration field in the MAC header incorporating the 6 μs of "idle time" following each ERP-OFDM transmission. This ensures that the NAV value of Clause 18 STAs is set correctly.

The "CS mechanism" described in 9.2.1 combines the NAV state and the STA's transmitter status with physical CS to determine the busy/idle state of the medium. The time interval between frames is called the IFS. An STA shall determine that the medium is idle through the use of the CCA mechanism for the interval specified. The starting reference of slot boundaries is the end of the last symbol of the previous frame on the medium. For ERP-OFDM frames, this includes the length extension. For ERP-OFDM frames, a STA shall generate the PHY RX_END indication, 6 μs after the end of the last symbol of the previous frame on the medium. This adjustment shall be performed by the STA based on local configuration information set using the PLME SAP.

#### 19.3.2.4 DSSS-OFDM long preamble PPDU format

Both long and short preambles and headers as previously described in 19.3.2.1 and 19.3.2.2 are used with DSSS-OFDM.

For all DSSS-OFDM rates and preamble modes, the PLCP SIGNAL field described in 18.2.3.3 shall be set to a 3 Mb/s value. That is, the 8 bit value is set to X'1E' (MSB to LSB). For DSSS-OFDM, this value is simply a default setting used for BSS compatibility and to ensure that NonERP STAs read the length field and defer the medium for that time even though they cannot demodulate the MPDU due to unsupported rates.

Figure 19-1 shows the PPDU format for the long preamble case. As seen, the PSDU is appended to the PLCP preamble and the PLCP header. The PLCP preamble is the same as described in 18.2.3.1 and 18.2.3.2. The PLCP header is similar to the one described in 19.3.2.1. The PSDU has a format that is nearly identical to a Clause 17 PLCP. The differences are described in 19.3.3.4.

The scrambler of 18.2.4 is used to scramble the DSSS-OFDM PLCP header, and the scrambler in 17.3.5.4 is used to scramble the data symbols in the OFDM segment.

Copyright © 2007 IEEE. All rights reserved.

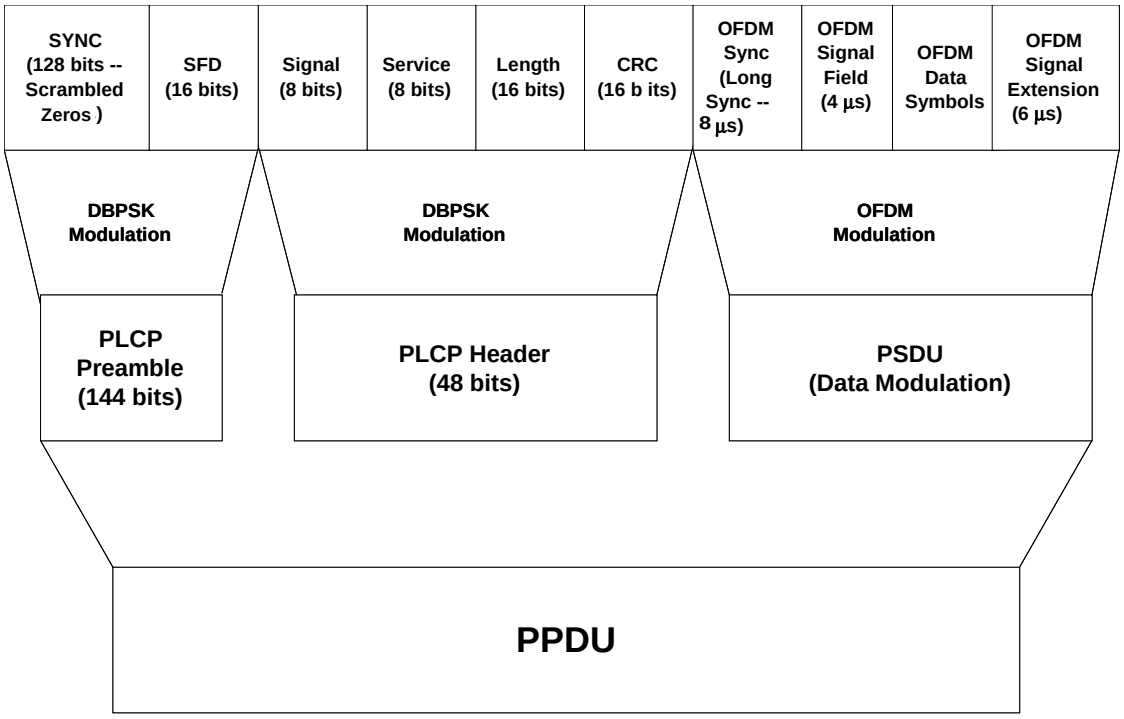



**Figure 19-1—Long preamble PPDU format for DSSS-OFDM**

#### 19.3.2.4.1 DSSS-OFDM PLCP length field calculation

For both the long and the short preamble PLCP cases, the length field calculation in terms of data packet length is as follows:

LENGTH = PSDUsyncOFDM + PSDUSignalOFDM +
4 × Ceiling((PLCPServiceBits + 8 × (NumberOfOctets) + PadBits) / $N_{DBPS}$) + SignalExtension

where

| | |
|---|---|
| PSDUsyncOFDM | is 8 µs (OFDM long training symbols) |
| PSDUSignalOFDM | is 4 µs |
| Ceiling | is a function that returns the smallest integer value greater than or equal to its argument value |
| PLCPServiceBits | is 8 bits |
| NumberOfOctets | is the number of data octets in the PSDU |
| PadBits | is 6 bits |
| $N_{DBPS}$ | is the number of data bits per OFDM symbol |
| SignalExtension | is 6 µs |

The length field is defined in units of microseconds and shall correspond to the calculated length of the PSDU. Note that the length extension bits in the Signal field are not needed or used for DSSS-OFDM.


Copyright © 2007 IEEE. All rights reserved.

#### 19.3.2.5 Short DSSS-OFDM PLCP PPDU format

The short PLCP preamble and header are used to maximize the throughput by reducing the overhead associated with the preamble and header. Figure 19-2 shows the short preamble PLCP PPDU format. As seen, the PSDU is appended to the PLCP preamble and the PLCP header. The short PLCP preamble is described in 18.2.3.8 and 18.2.3.9. The PLCP header is as described in 19.3.2.4. The PSDU has a format that is nearly identical to Clause 17 PLCP. The differences are described in 19.3.3.4.

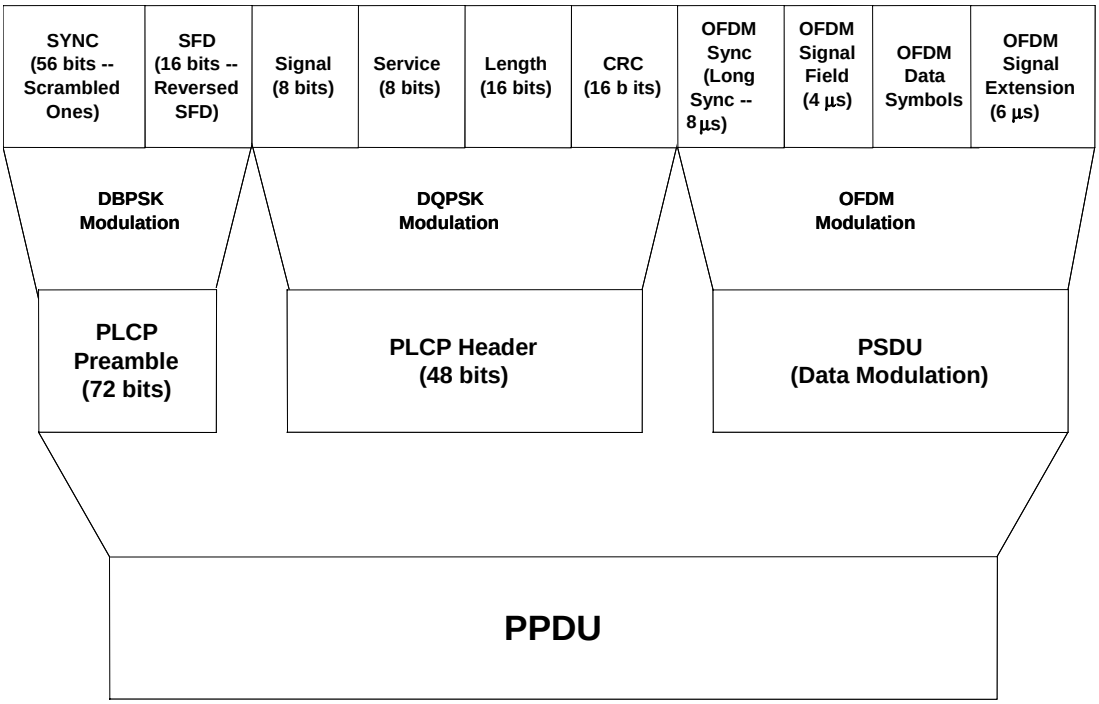



**Figure 19-2—Short preamble PPDU format for DSSS-OFDM**

### 19.3.3 PLCP data modulation and rate change

#### 19.3.3.1 Long and short preamble formats

The long and short PLCP preamble and the long PLCP header shall be transmitted using the 1 Mb/s DBPSK modulation. The short PLCP header shall be transmitted using the 2 Mb/s modulation. The SIGNAL and SERVICE fields combined shall indicate the modulation and rate that shall be used to transmit the PSDU. The transmitter and receiver shall initiate the modulation and rate indicated by the SIGNAL and SERVICE fields, starting with the first octet of the PSDU. The PSDU transmission rate shall be set by the DATARATE parameter in the TXVECTOR, issued with the PHY-TXSTART.request primitive described in 18.4.5.1.

Four modulation formats are mandatory, 1 Mb/s and 2 Mb/s ERP-DSSS and 5.5 Mb/s and 11 Mb/s ERP-CCK, and they are specified in 18.4.6.3.

Four optional ERP-PBCC modulation formats and data rates are specified for the ERP. They shall be based on PBCC 5.5, 11, 22, and 33 Mb/s modulations. The rates of 5.5 Mb/s and 11 Mb/s are described in 18.4.6.6. No change in the spectral mask of 18.4.7.3 is required for these modes.

Copyright © 2007 IEEE. All rights reserved.

### 19.3.3.2 ERP-PBCC 22 Mb/s and 33 Mb/s formats

In the PBCC encoder, incoming data are first encoded with a packet binary convolutional code. A cover code (as defined in PBCC modes in 18.4.6.6) is applied to the encoded data prior to transmission through the channel.

The packet binary convolutional code that is used is a 256-state, rate 2/3 code. The generator matrix for the code is given as

$$G = \begin{bmatrix} 1 + D^4 & D & D + D^3 \\ D^3 & 1 + D^2 + D^4 & D + D^3 \end{bmatrix} \quad (19\text{-}1)$$

In octal notation, the generator matrix is given by

$$G = \begin{bmatrix} 21 & 2 & 12 \\ 10 & 25 & 12 \end{bmatrix} \quad (19\text{-}2)$$

As the system is frame (PPDU) based, the encoder shall be in state zero; i.e., all memory elements contain zero, at the beginning of every PPDU. The encoder shall also be placed in a known state at the end of every PPDU to prevent the data bits near the end of the PPDU from being decoded incorrectly. This is achieved by appending one octet containing all zeros to the end of the PPDU prior to transmission and discarding the final octet of each received PPDU.

An encoder block diagram is shown in Figure 19-3. It consists of two paths of four memory elements each. For every pair of data bits input, three output bits are generated. The output of the convolutional code is mapped to an 8-PSK constellation; each 3-bit output sequence from the packet binary convolutional encoder is used to produce one symbol. This yields a throughput of two information bits per symbol. In ERP-PBCC-22 and ERP-PBCC-33, the input data stream is divided into pairs of adjacent bits. In each pair, the first bit is fed to the upper input of the convolutional encoder, and the second is fed to the lower input of the convolutional encoder. An illustration of the mapping for the jth (j≥0) pair of input bits (b2j, b2j+1) is given in Figure 19-3.

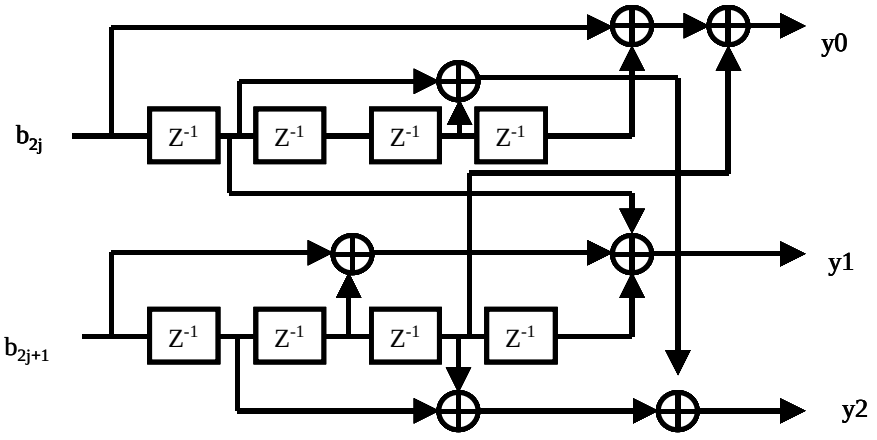



**Figure 19-3—22/33 Mb/s ERP-PBCC convolutional encoder**


Copyright © 2007 IEEE. All rights reserved.

The phase of the first complex chip of the 22 Mb/s PSDU shall be defined with respect to the phase of the last chip of the PCLP header, i.e., the last chip of the CRC check. The phase of the first complex chip of the 33 Mb/s PSDU shall be defined with respect to the phase of the last chip of the clock switch section, i.e., the last chip of the ReSync field. The bits (y2 y1 y0) = (0,0,0) shall indicate the same phase as the last chip of the CRC check. The other seven combinations of (y2 y1 y0) shall be defined with respect to this reference phase as shown in Figure 19-4.

The mapping from BCC outputs to 8-PSK constellation points is determined by a pseudo-random cover sequence. The cover sequence is the same one as described in 18.4.6.6. The current binary value of this sequence at every given point in time is taken as shown in Figure 19-4. The mapping is shown in Figure 19-4.

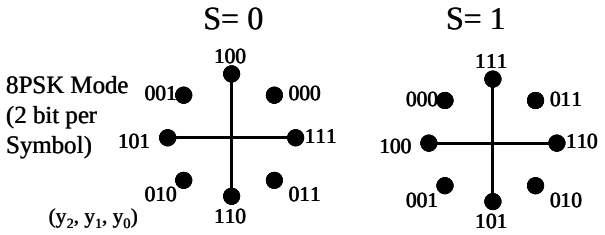



**Figure 19-4—ERP-PBCC-22 and ERP-PBCC-33 cover code mapping**

ERP-PBCC mode achieves a 33 Mb/s data rate by using a 16.5 MHz clock for the data portion of the packet. The data portion is otherwise identical to the 22 Mb/s ERP-PBCC modulations. The structure and clock speed of the preamble is the same as in Clause 18. An extra clock switch section between the preamble and the data portion is added, with the format described below. The same pulse shape shall be used in each clock domain.

When the clock is switched from 11 MHz to 16.5 MHz, the clock switching structure in Figure 19-5 is used.

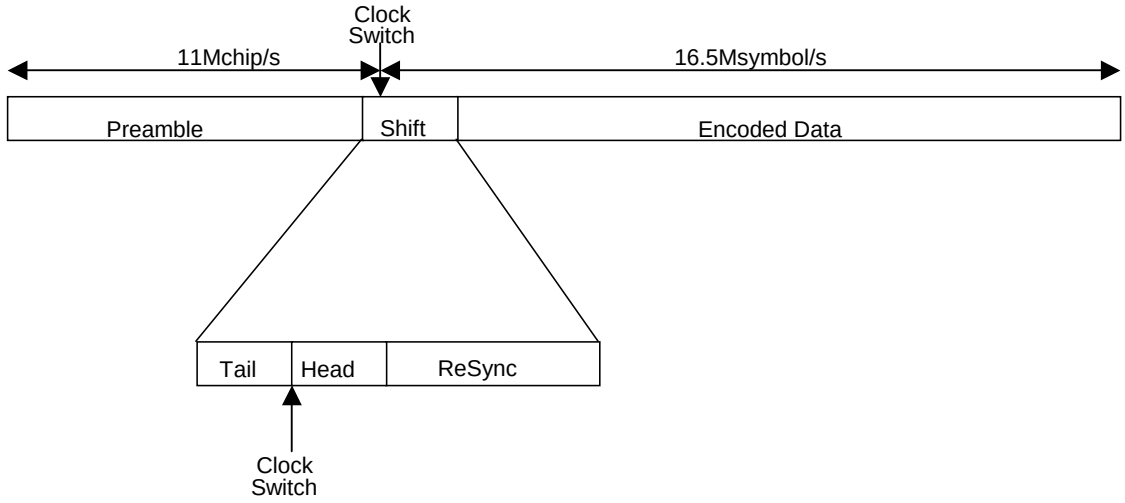



**Figure 19-5—33 Mb/s clock switching**

The tail is 3 clock cycles at 11 Mchip/s and the head is 3 clock cycles at 16.5 Msymbol/s (QPSK). The resync is 9 clock cycles at 16.5 Msymbol/s. The total clock switching time (tail and head and resync) is 1 μs. The tail bits are 1 1 1, the head bits are 0 0 0, and the resync bits are 1 0 0 0 1 1 1 0 1. The modulation is BPSK, which is phase synchronous with the previous symbol.

Copyright © 2007 IEEE. All rights reserved.

### 19.3.3.3 ERP-OFDM format

PLCP modulation and rate change for the ERP-OFDM frame format follows 17.3.7.

### 19.3.3.4 Long and short DSSS-OFDM PLCP format

The scrambler of 18.2.4 is used to scramble the DSSS-OFDM PLCP header, and the scrambler in 17.3.5.4 is used to scramble the data symbols in the OFDM segment.

#### 19.3.3.4.1 Overview of the DSSS-OFDM PLCP PSDU encoding process

This subclause contains the definitions and procedure for forming the PSDU portion of the DSSS-OFDM PLCP. Figure 19-6 shows an expanded view of the DSSS-OFDM PSDU. The PSDU is composed of four major sections. The first is the long sync training sequence that is used for acquisition of receiver parameters by the OFDM demodulator. The long sync training sequence for DSSS-OFDM is identical to the long training symbols described in 17.3.3. The second section is the OFDM SIGNAL field that provides the demodulator information on the OFDM data rate and length of the OFDM data section. The SIGNAL field for DSSS-OFDM is identical to the SIGNAL field described in 17.3.4. After the SIGNAL field is the data section of the PSDU. This is identical to the modulation procedure described in 17.3.2.1 in Step c) through Step m). After the data section, the PSDU for DSSS-OFDM appends a signal extension section to provide additional processing time for the OFDM demodulator. This signal extension is a period of no transmission as described in 19.3.3.4.5.

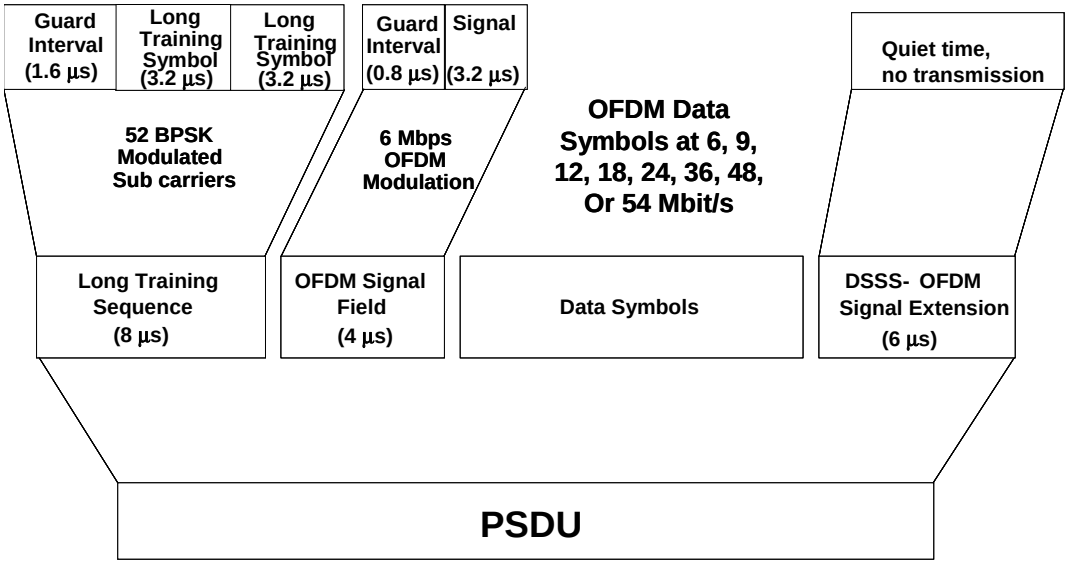



**Figure 19-6—DSSS-OFDM PSDU**

#### 19.3.3.4.2 Long sync training sequence definition

The long sync training sequence is defined in 17.3.3.

#### 19.3.3.4.3 OFDM signal field definition

The DSSS-OFDM SIGNAL field is defined in 17.3.4. Note that the length conveyed by the SIGNAL field is calculated as described in 17.3.4. That is, the length conveyed by this field does not include the signal extension described in 19.3.3.4.5.


Copyright © 2007 IEEE. All rights reserved.

#### 19.3.3.4.4 Data symbol definition

The same process as Step c) through Step m) of 17.3.2.1 is used to encode the data symbols' portion of the DSSS-OFDM PSDU.

#### 19.3.3.4.5 DSSS-OFDM signal extension

The DSSS-OFDM signal extension shall be a period of no transmission of 6 μs length. It is inserted to allow more time to finish the convolutional decoding of the OFDM segment waveform and still meet the 10 μs SIFS requirement of the ERP.

### 19.3.4 PLCP transmit procedure

The transmit procedure will depend on the data rate and modulation format requested. For data rates of 1, 2, 5.5, 11, 22, and 33 Mb/s, the PLCP transmit procedure shall follow 18.2.5. For the ERP_OFDM rates of 6, 12, and 24 Mb/s and the rates of 9, 18, 36, 48, and 54 Mb/s, the PLCP transmit procedure shall follow 17.3.11.

The transmit procedures for the optional DSSS-OFDM mode using the long or short PLCP preamble and header are the same as those described in 18.2.5, and they do not change apart from the ability to transmit a higher rate PSDU using DSSS-OFDM.

### 19.3.5 CCA

The PLCP shall provide the capability to perform a CCA and report the results of the assessment to the MAC. The CCA mechanism shall detect a "medium busy" condition for all supported preamble and header types. That is, the CCA mechanism shall detect that the medium is busy for the PLCP PPDUs specified in 17.3.3 and 18.2.2. The CCA mechanism performance requirements are given in 19.4.6.

The ERP shall provide the capability to perform CCA according to the following method:

> **CCA Mode** (ED and CS): A combination of CS and energy above threshold. CCA shall have a mechanism for CS that will detect all mandatory Clause 19 sync symbols. This CCA's mode's CS shall include both Barker code sync detection and OFDM sync symbol detection. CCA shall report busy at least while a PPDU with energy above the ED threshold is being received at the antenna.

The ED status shall be given by the PMD primitive, PMD_ED. The CS status shall be given by PMD_CS. The status of PMD_ED and PMD_CS is used in the PLCP convergence procedure to indicate activity to the MAC through the PHY interface primitive, PHY-CCA.indicate. A busy channel shall be indicated by PHY-CCA.indicate of class BUSY. A clear channel shall be indicated by PHY-CCA.indicate of class IDLE.

### 19.3.6 PLCP receive procedure

This subclause describes the procedure used by receivers of the ERP. An ERP receiver shall be capable of receiving 1, 2, 5.5, and 11 Mb/s PLCPs using either the long or short preamble formats described in Clause 18 and shall be capable of receiving 6, 12, and 24 Mb/s using the modulation and preamble described in Clause 17. The PHY may also implement the ERP-PBCC modulation at rates of 5.5, 11, 22, and 33 Mb/s; the ERP-OFDM modulations at rates of 9, 18, 36, 48, and 54 Mb/s; and/or the DSSS-OFDM modulation rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s. The receiver shall be capable of detecting the preamble type (ERP-OFDM, short preamble, or long preamble) and the modulation type. These values shall be reported in the RXVECTOR (see 19.2).

Upon the receipt of a PPDU, the receiver shall first distinguish between the ERP-OFDM preamble and the single carrier modulations (long or short preamble). In the case where the preamble is an ERP-OFDM preamble, the PLCP receive procedure shall follow the procedure described in 17.3.12. Otherwise, the

Copyright © 2007 IEEE. All rights reserved.

receiver shall then distinguish between the long preamble and short preamble as specified in 18.2.2. The receiver shall then demodulate the SERVICE field to determine the modulation type as specified in 19.3.2.1 or 19.3.2.2. For short preamble and long preamble using ERP-DSSS, ERP-CCK, or ERP-PBCC modulations, the receiver shall then follow the receive procedure described in 18.2.6.

A receiver that supports DSSS-OFDM is capable of receiving all rates specified in Clause 15 and all mandatory rates in Clause 17 and Clause 18. If the SIGNAL field indicates 3 Mb/s, the receiver shall attempt to receive a DSSS-OFDM packet. The remaining receive procedures for a DSSS-OFDM-capable receiver are the same as those described in 18.2.6, and they do not change apart from the ability to receive DSSS-OFDM in the PSDU. If DSSS-OFDM is being received, the receiver shall handle the modulation transition requirements as described in 19.7.2. The receiver shall then follow the receive procedure described in 17.3.12.

## 19.4 ERP PMD operating specifications (general)

Subclauses 19.4.1 through 19.4.7 provide general specifications for the ERP PMD sublayers. These specifications are based on 17.3.8 except where noted.

### 19.4.1 Regulatory requirements

All systems shall comply with the appropriate regulatory requirements for operation in the 2.4 GHz band.

### 19.4.2 Operating channel frequencies

The ERP shall operate in the frequency ranges specified in 18.4.6.2, as allocated by regulatory bodies in the United States, Europe, and Japan. OFDM operation in channel 14 may not be allowed in Japan. The channel numbering and the number of operating channels shall follow Table 18-9 of 18.4.6.2.

### 19.4.3 Transmit and receive in-band and out-of-band spurious emissions

The ERP shall conform to in-band and out-of-band spurious emissions as set by the appropriate regulatory bodies for the 2.4 GHz band.

### 19.4.4 Slot time

The slot time is 20 μs, except that an optional 9 μs slot time may be used when the BSS consists of only ERP STAs capable of supporting this option. The optional 9 μs slot time shall not be used if the network has one or more NonERP STAs associated. For IBSS, the Short Slot Time subfield shall be set to 0, corresponding to a 20 μs slot time.

### 19.4.5 SIFS value

The ERP shall use a SIFS of 10 μs.

### 19.4.6 CCA performance

The CCA shall indicate TRUE if there is no CCA “medium busy” indication. The CCA parameters are subject to the following criteria:

a) When a valid signal with a signal power of –76 dBm or greater at the receiver antenna connector is present at the start of the PHY slot, the receiver’s CCA indicator shall report the channel busy with probability CCA_Detect_Probabilty within a CCA_Time. CCA_Time is SlotTime – RxTxTurn-aroundTime. CCA_Detect_Probabilty is the probability that the CCA does respond correctly to a


Copyright © 2007 IEEE. All rights reserved.

valid signal. The values for these parameters are found in Table 19-5. Note that the CCA Detect Probability and the power level are performance requirements.

b) In the event that a correct PLCP header is received, the ERP shall hold the CCA signal inactive (channel busy) for the full duration, as indicated by the PLCP LENGTH field. Should a loss of CS occur in the middle of reception, the CCA shall indicate a busy medium for the intended duration of the transmitted PPDU.

**Table 19-5—CCA parameters**

| Parameter | Slot time = 20 μs | Slot time = 9 μs |
|---|---|---|
| SlotTime | 20 μs | 9 μs |
| RxTxTurnaroundTime | 5 μs | 5 μs |
| CCA_Time | 15 μs | 4 μs |
| CCA_Detect_Probability | > 99% | > 90% |

### 19.4.7 PMD transmit specifications

The PMD transmit specifications shall follow 17.3.9 with the exception of the transmit power level (17.3.9.1), the transmit center frequency tolerance (17.3.9.4), and the symbol clock frequency tolerance (17.3.9.5). Regulatory requirements may have an effect on the combination of maximum transmit power and spectral mask if the resulting signals violate restricted band emission limits.

#### 19.4.7.1 Transmit power levels

The maximum transmit power level shall meet the requirements of the local regulatory body.

#### 19.4.7.2 Transmit center frequency tolerance

The transmit center frequency tolerance shall be ± 25 PPM maximum. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked).

#### 19.4.7.3 Symbol clock frequency tolerance

The symbol clock frequency tolerance shall be ± 25 PPM maximum. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked oscillators). This means that the error in PPM for the carrier and the symbol timing shall be the same.

## 19.5 ERP operation specifications

This subclause describes the receive specifications for the PMD sublayer. The receive specification for the ERP-OFDM modes shall follow 17.3.10 with the exception of the receiver maximum input level (17.3.10.4) and the adjacent channel rejection (17.3.10.2). The receive specifications for the ERP-DSSS modes shall follow 18.4.8 with the exception of the receiver maximum input level (18.4.8.2).

### 19.5.1 Receiver minimum input level sensitivity

The PER of the ERP-OFDM modes shall be less than 10% at a PSDU length of 1000 octets for the input levels of Table 17-13 of 17.3.10. Input levels are specific for each data rate and are measured at the antenna

Copyright © 2007 IEEE. All rights reserved.

connector. A noise figure of 10 dB and an implementation loss of 5 dB are assumed. The PER of the ERP-DSSS modes shall be as specified in 18.4.8.1.

### 19.5.2 Adjacent channel rejection

Adjacent channels at 2.4 GHz are defined to be at ± 25 MHz spacing. The adjacent channel rejection shall be measured by setting the desired signal's strength 3 dB above the rate-dependent sensitivity specified in Table 17-13 of 17.3.10 and raising the power of the interfering signal until 10% PER is caused for a PSDU length of 1000 octets. The power difference between the interfering and the desired channel is the corresponding adjacent channel rejection. The interfering signal in the adjacent channel shall be a conformant OFDM signal, unsynchronized with the signal in the channel under test. For an OFDM PHY, the corresponding rejection shall be no less than specified in Table 17-13 of 17.3.10.

The alternative adjacent channel rejection of Table 17-13 shall not be required for the ERP.

The adjacent channel rejection of the ERP-DSSS modes shall follow 18.4.8.3.

### 19.5.3 Receive maximum input level capability

The PER shall be less than 10% at a PSDU length of 1000 octets for an input level of –20 dBm measured at the antenna connector for any supported modulation signal or data rate (i.e., 1, 2, 5.5, 6, 9, 11, 12, 18, 22, 24, 33, 36, 48, 54 Mb/s).

### 19.5.4 Transmit spectral mask

The transmit spectral mask for the ERP-OFDM modes shall follow I.2.3 in Annex I and is shown in Figure I.1 therein. The transmit spectral mask for the ERP-DSSS modes shall follow 18.4.7.3 and is shown in Figure 18-19 therein.

## 19.6 ERP-PBCC operation specifications

The ERP-PBCC receiver specifications shall follow 18.4.8 except as noted.

These optional modes provide systems with the ability to achieve data rates of 22 Mb/s and 33 Mb/s in modes that are fully backwards compatible with Clause 15 and Clause 18 BSSs without requiring additional coordination or protection mechanisms. In addition, the 22 Mb/s ERP-PBCC mode is spectrally identical to Clause 18 BSSs. Four optional ERP-PBCC modulation formats and data rates are specified for the ERP. They shall be based on PBCC 5.5, 11, 22, and 33 Mb/s modulations. The rates of 5.5 Mb/s and 11 Mb/s are described in 18.4.6.6.

### 19.6.1 Receiver minimum input level sensitivity

For the 22 Mb/s ERP-PBCC mode, the frame error ratio shall be less than $8 \times 10^{-2}$ at a PSDU length of 1024 octets for an input level of –76 dBm measured at the antenna connector. For the 33 Mb/s ERP-PBCC mode, the corresponding input level shall be –74 dBm.

### 19.6.2 Receiver adjacent channel rejection

The adjacent channel rejection shall be equal to or better than 35 dB, with an FER of $8 \times 10^{-2}$ using ERP-PBCC modulation and a PSDU length of 1024 octets. The adjacent channel rejection shall be measured using the following method. Input an ERP-PBCC modulated signal of the same rate at a level 6 dB greater than specified in 19.6.1. In an adjacent channel (25 MHz separation as defined by the channel numbering), input a signal modulated in a similar fashion, which adheres to the transmit mask specified in 18.4.7.3, to a


Copyright © 2007 IEEE. All rights reserved.

level 41 dB above the level specified in 19.6.1. The adjacent channel signal shall be derived from a separate signal source. It shall not be a frequency-shifted version of the reference channel. Under these conditions, the FER shall be no worse than $8 \times 10^{-2}$.

## 19.7 DSSS-OFDM operation specifications

This optional mode provides systems with the ability to use OFDM in a mode that is fully compatible with Clause 15 and Clause 18 BSSs without requiring additional coordination. That is, it does not need a protection mechanism. This compatibility requires the use of Clause 18 long and short preambles and inclusion of a signal extension field to match SIFS spacing of Clause 18 systems. By reusing the Clause 18 preambles, this optional mode ensures that the Clause 18 CCA and SIFS interval function properly when ERP and NonERP STAs interoperate. When this option is enabled, the same rates shall be supported in both ERP-OFDM and DSSS-OFDM. The DSSS-OFDM PMD transmit and receive specifications shall follow the related ERP-OFDM specifications in 19.5.

### 19.7.1 Overview

This optional extension of the DSSS system builds on the payload data rates of 1, 2, 5.5, and 11 Mb/s, as described in Clause 18, to provide 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s payload data rates while reusing the preambles (short and long) described by Clause 18. The capability described in this subclause is called DSSS-OFDM. This optional capability complements the Extended Rate OFDM mode described in Clause 19 by combining OFDM modulation with DSSS preambles. As a result, for DSSS-OFDM, the PPDU format described in 18.2.2 is relatively unchanged. The major change is to the format of the PSDU. The Clause 18 single carrier PSDU is replaced by a PSDU that is very similar to the PSDUs described in Clause 17. This subclause highlights the differences. In addition, 19.7.2 specifies the radio and physical layer behavior of the transition from the Barker symbol-modulated preamble and the OFDM-modulated data for PSDU.

### 19.7.2 Single carrier to multicarrier transition requirements

The spectrum mask for the DSSS-OFDM waveform shall meet the requirements as shown in Figure 17-13 of 17.3.9.2.

The single carrier signal segment of the packet shall have a coherent relationship with the multicarrier (OFDM) segment of the packet. All characteristics of the signal shall be transferable from one symbol to the next, even when transitioning to the OFDM segment. This enables high-performance, coherent receiver operation across the whole packet. This requirement is no different in nature than that stated in Clause 15, Clause 17, and Clause 18. The distinction is that those clauses use a signaling scheme that is either just single carrier or just multicarrier. In contrast, for this mode, both single carrier signaling and multicarrier signaling are used within the context of a single packet.

This subclause specifies the coherent relationship between the single carrier segment and the OFDM segment, so that the receiver has the opportunity to track through the transition without any forced parameter reacquisition. The single carrier preamble and header provide all parametric information required for demodulation of the OFDM segment to within conventional estimation-in-noise accuracy. Although multicarrier sync features are provided for convenience at OFDM segment onset, if and how to use the multicarrier sync for reacquisition is an implementer's decision. Multicarrier sync is not necessary. The packet is coherent throughout.

As shown in Figure 19-7, the ideal transition would provide a constant carrier frequency and phase, a constant power, a constant spectrum, and a constant timing relationship. Constant in this context means that the same clock crystal that sets the frequencies and timing of each part is the same through the transition. This allows the frequency and timing tracking loops to work undisturbed through the transition. Subclauses

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 753 of 1232

19.7.2.1 through 19.7.2.6 establish the ideal transition characteristics for the transmit signal. Subclause 19.7.2.7 specifies the required implementation fidelity or accuracy.

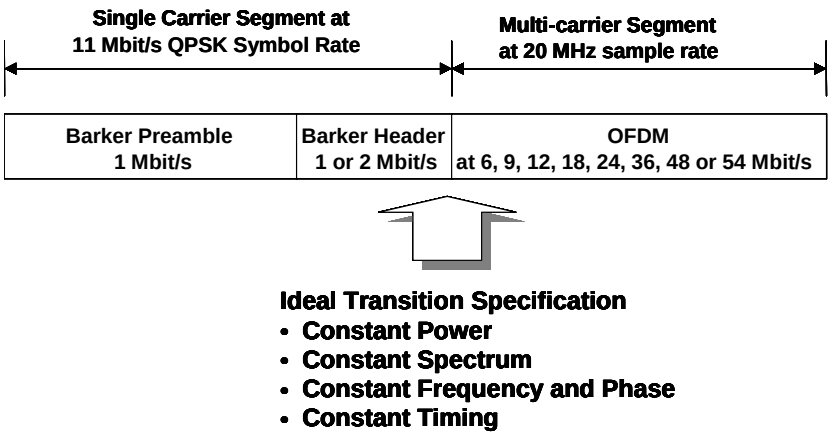



**Figure 19-7—Single carrier to multicarrier transition definition**

### 19.7.2.1 Spectral binding requirement

The spectral binding requirement allows the receiver's estimate of the channel state information to be transferred from the single carrier packet segment to the multicarrier packet segment. This requirement establishes a coherent relationship between the end-to-end frequency responses of the single carrier and multicarrier segments.

During reception of the single carrier preamble and header, the receiver may estimate the channel impulse response. In practice, this could be accomplished through Barker code correlation. The channel impulse response contains end-to-end frequency response information about the linear distortion experienced by the signal due to filters and multipath. This distortion can be mitigated with an equalizer or other commonly known techniques.

The channel impulse response estimate generated during the single carrier packet segment will include the single carrier's pulse-shaping, filter frequency response used to control the single carrier's transmit spectrum and transmit impulse response. The single carrier's pulse-shaping filter may be distinct from the shaping technique used for the multicarrier segment.

The spectral binding requirement states that the linear distortions experienced by the single carrier signal and the linear distortions experienced by the multicarrier signal have a known relationship. This relationship is defined by this specification and shall be manifested by all compliant transmit radios. This will allow any receiver to exploit channel information derived during the single carrier segment and reuse the channel information during the multicarrier segment, if desired.

Three elements have been itemized for this specification to achieve spectral binding. All three elements are necessary to achieve spectral binding, and they are discussed in the next three subclauses. The first element focuses on distortions common to both the single carrier packet segment and the multicarrier packet segment. The second element deals with pulse-shaping unique to the OFDM packet segment. The third element deals with pulse-shaping unique to the single carrier packet segment. The multicarrier pulse shape discussion precedes the single carrier's pulse shape discussion because it is believed this will be a more comfortable progression, due to similar multicarrier pulse-shaping considerations contained in Clause 17.


Copyright © 2007 IEEE. All rights reserved.

#### 19.7.2.1.1 Common linear distortions

Separate from the single carrier and the multicarrier pulse shaping, transmit signal generation will be designed to provide linear distortion continuity to the receiver's demodulation algorithms. The common linear distortion requirement is illustrated in Figure 19-8, where it is shown that the processing at the receiver assumes that the dominant linear distortions are induced on all waveform segments. The receiver observes the composite linear distortion due to imperfect transmit radio filters, due to multipath filtering, and due to imperfect receive radio filters. In general, the receiver is unable to decompose the distortion into separate physical components and is only able to observe the aggregate effect. This specification constrains only the linear distortions in the transmit radio, because that is what is necessary to ensure interoperability. The soft switch that appears in Figure 19-8 is a conceptual element to implement the transition, as described below.

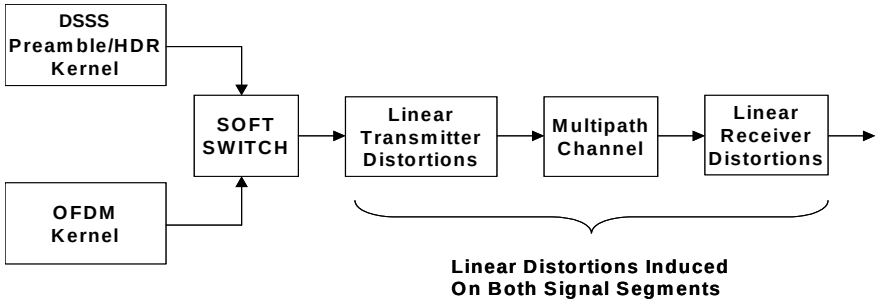



**Figure 19-8—Linear distortions common to the single carrier and multicarrier signal segments**

In short, this common linear distortion requirement states that the dominant filters in the transmit radio will stay invariant and common to all waveform segments. Once the receiver has determined the end-to-end impulse response, channel information is assumed to be common to both the single carrier signal and the multicarrier signal. This will enable receiver design of linear-distortion mitigation techniques that do not require a reacquisition after transitioning to OFDM.

#### 19.7.2.1.2 Symbol shaping unique to the DSSS-OFDM segment

OFDM spectral shaping may be achieved using two mechanisms: (1) Time-domain convolution filtering may be used to shape the spectrum. (2) Time-domain window tapering of OFDM symbol onset and termination may be used to shape the spectrum. This second mechanism can be viewed as frequency domain convolution. The first mechanism shall be common to both the OFDM and single carrier if it is a dominant distortion mechanism. The second mechanism may be unique to the OFDM segment, because it does not affect the frequency response of the 52 subcarriers.

The first spectral shaping mechanism using time-domain convolution filtering shall be common to both the single carrier and multicarrier segments for the reasons described in the preceding section. The receiver should not see intrapacket frequency response discontinuities.

Convolution filtering may be budgeted in various ways. One option would be to use a single filter that both the single carrier and multicarrier segments use. Another option would be to use two different physical filter realizations, one for the single carrier segment and a second for the multicarrier segment, say, for reason of distinct sample rates or bit precision. With this second implementation option, the designer shall ensure the frequency response of the filter is common to both packet segments.

The second shaping mechanism, which uses frequency-domain convolution through time-domain subcarrier onset-and-termination shaping, may be unique to the OFDM segment. This unique technique is acceptable because it does not modify the required frequency response of the 52 subcarriers.

Copyright © 2007 IEEE. All rights reserved.

Spectral shaping by tapering the OFDM symbol onset-and-termination using a time-domain window is described in Clause 17 and is equally germane to Clause 19 systems. For convenience, one of the relevant figures from Clause 17 is repeated here as Figure 19-9. Clause 17 suggested that the tapering transition duration is 0.1 µs.

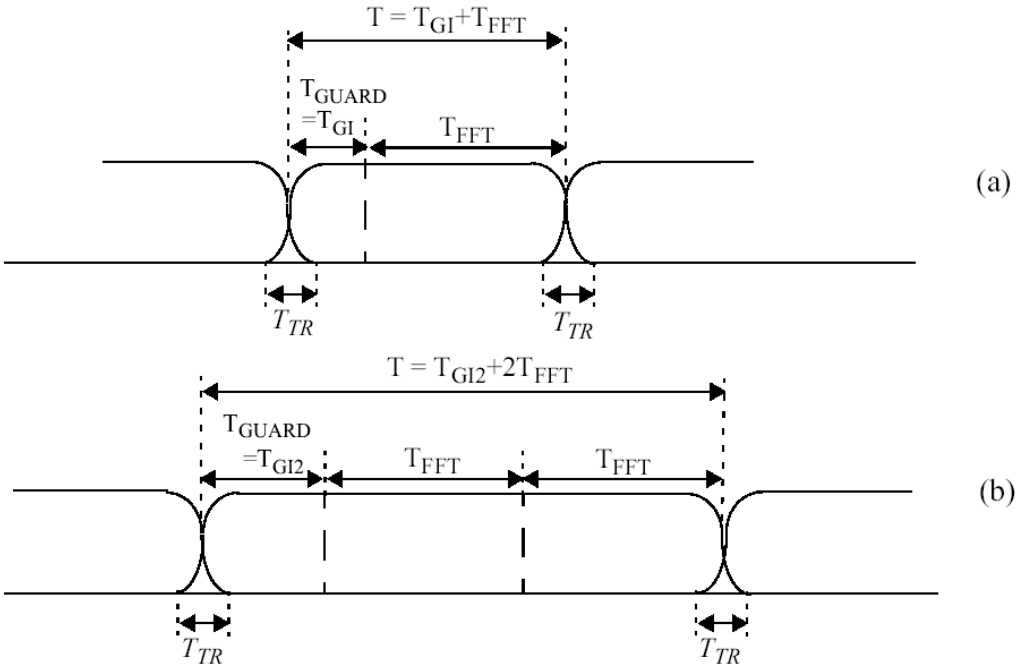



**Figure 19-9—Spectral shaping achieved by OFDM symbol onset and termination shaping**

The effect of time-domain windowing on a single subcarrier's power spectrum is shown in Figure 19-10 for two cases. The first case is rectangular time-domain windowing of an OFDM symbol. The second case is for the Clause 17 suggested time-domain windowing of an OFDM symbol with a 0.1 µs transition. Note the difference in frequency-dependent amplitude roll-off. Adding the 52 frequency-bin-centered individual subcarrier power spectral densities generates the composite 52 subcarrier power spectrum.

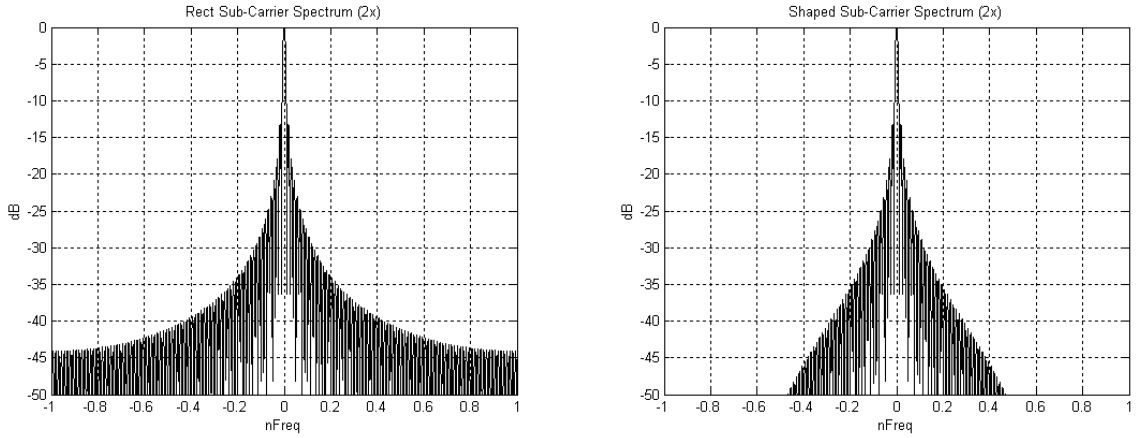



**Figure 19-10—Subcarrier spectrums for rectangular windowing and Clause 17 suggested windowing**


Copyright © 2007 IEEE. All rights reserved.

This type of OFDM spectrum control does not affect the relative amplitudes and phases of the individual subcarriers. Instead, it affects each subcarrier's power spectral density. Consequently, this type of spectrum control has a benign effect on the relative spectrums of single carrier and the multicarrier packet segments. That is why it can be unique.

To achieve the design goal, the implementer may budget spectral shaping in the transmit radio. Some of the spectral shaping may be achieved using time-domain convolution filtering, and some may be achieved through time-domain windowing of the OFDM. In any case, the transmit implementation shall provide frequency response coherency.

#### 19.7.2.1.3 Pulse shaping unique to the single carrier segment

This subclause describes the pulse-shaping requirements of the single carrier segment of the DSSS-OFDM packet. To establish frequency response coherency, it is necessary to specify the frequency response of the single carrier signal that establishes a coherent relationship to the frequency response of the OFDM.

The frequency response of the single carrier pulse is patterned after the tandem OFDM. The pattern is the OFDM signal as described in Clause 17, with an example provided in Annex G. The ideal OFDM signal has a flat amplitude response and zero-phase offset across 52 subcarriers. Clause 17 establishes the ideal frequency-response relationship among the 12 short SYNC subcarriers, 52 long SYNC subcarriers, the 52 SIGNAL field subcarriers, and the 52 data field subcarriers. Similarly, the ideal relationship to the single carrier frequency response is defined.

Relative to the ideal OFDM, the single carrier part of the DSSS-OFDM signal shall have the pulse shape established herein. In a particular implementation, it is acceptable to deviate from this ideal but only in a manner that is common to both the single carrier signal and the multicarrier signal across the passband of the OFDM signal. This requirement provides the required frequency response coherency.

The frequency response of the single carrier pulse is patterned after the OFDM that will be transmitted in tandem. The single carrier pulse is derived from a time-windowed sinc function as shown in Figure 19-11 and Equation (19-3). The sinc function is the time response of an ideal brickwall filter. The brickwall filter is set equal to the bandwidth of an ideal OFDM signal. In particular, the bandwidth of the brickwall filter has been set to 52 times the Clause 17 subcarrier spacing of 20/64 MHz, or 312.5 KHz.

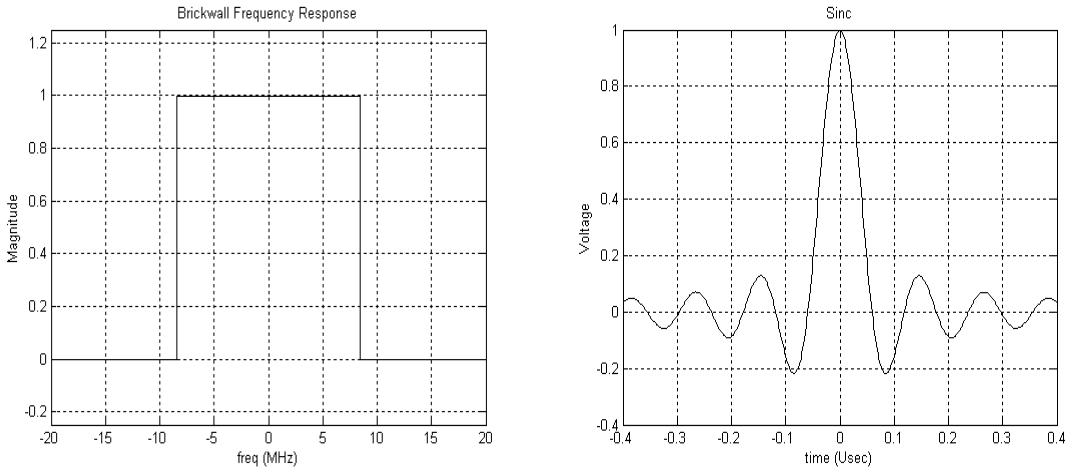



**Figure 19-11—Foundational brickwall filter**

$$h_{IdealBW}(t) = f_W \frac{\sin(\pi f_W t)}{\pi f_W t} = f_W \mathrm{sinc}(f_W t) \quad (19\text{-}3)$$

Copyright © 2007 IEEE. All rights reserved.

where

$f_W$ = 52(20/64) MHz

The infinite duration impulse response of the brickwall filter should be windowed to something practical. A continuous time version of the Hanning window may be used. The Hanning window and an overlay of the sinc function are shown in Equation (19-4) and Figure 19-12.

$$h_{\text{Window}}(t) = 0.5\left[1 + \cos\left(2\pi\frac{t}{t_{\text{SPAN}}}\right)\right] \quad (19\text{-}4)$$

where

$T_{\text{SPAN}}$ = 0.8 μs

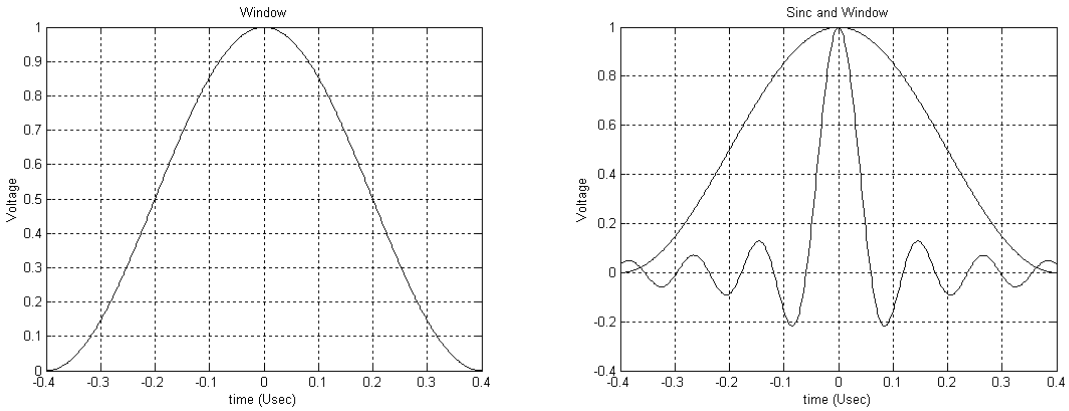



**Figure 19-12—Continuous time Hanning window**

The pulse specified for use with the single carrier packet segment is obtained by application of the window as shown in Equation (19-5) and Figure 19-13. Notice that its duration is equal to a Clause 17 short sync cycle, only 0.8 μs.

$$p(t) = h_{\text{Window}}h_{\text{IdealBW}}(t) \quad (19\text{-}5)$$

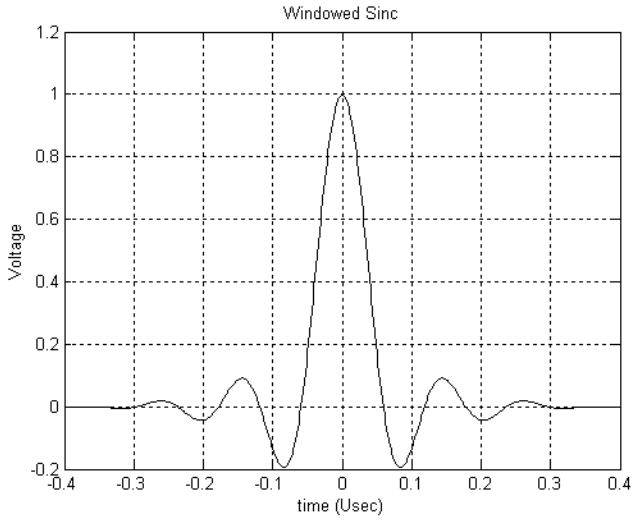



**Figure 19-13—Specified pulse**


Copyright © 2007 IEEE. All rights reserved.

The frequency response of the derived pulse is shown in Figure 19-14. This pulse generates a single carrier signal that has a spectrum nearly equal to that of the OFDM signal. This means that the receiver will experience essentially no change in receive signal power behavior even in the presence of multipath. At the point of the outermost subcarrier in the OFDM signal, the single carrier spectrum is down only about 4 dB. This is deemed adequate because the single carrier preamble-header is long in duration compared to the Clause 17 sync duration. Plenty of time is available to generate channel impulse response information that is sufficiently accurate.

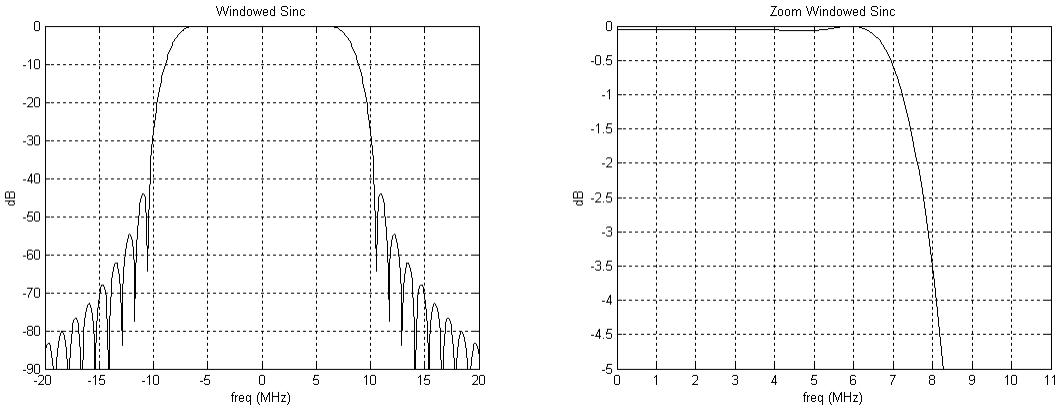



**Figure 19-14—Single carrier frequency response**

In summary, the specified single carrier pulse provides frequency response coherency between the single carrier and multicarrier segments of the packet. This does not mean that the spectrums are identical between segments. Rather, it means the ideal frequency responses of both are known. Beyond this, all linear distortion is common to both. It is not necessary to use this single carrier pulse during Clause 15 or Clause 18 packet transmissions.

### 19.7.2.2 Sample-power matching requirement

The transmit signal power shall be equal for the single carrier and multicarrier signal segments. The point of comparison is shown in Figure 19-15. The power measurement will be over the single carrier header and over the OFDM data symbols.

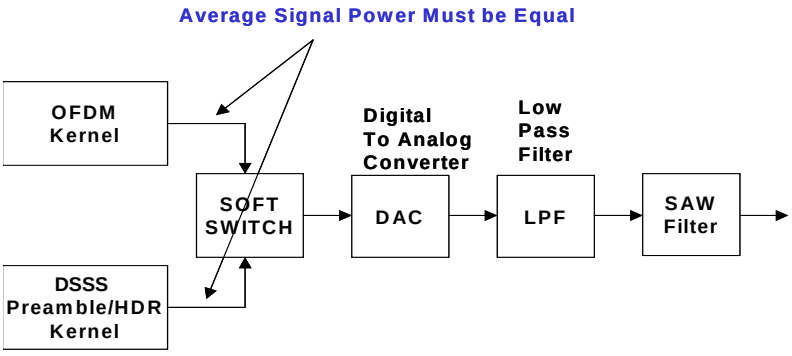



**Figure 19-15—Comparing signal power**

Copyright © 2007 IEEE. All rights reserved.

### 19.7.2.3 Transition time alignment

This subclause describes how the single carrier signal and the multicarrier signal are time aligned. The single carrier signal uses a chip rate of 11 MHz. The OFDM signal uses a fundamental sample rate of 20 MHz. The signals are easily aligned by first aligning the 11 MHz clock and the 20 MHz clock on 1 µs boundaries as shown in Figure 19-16.

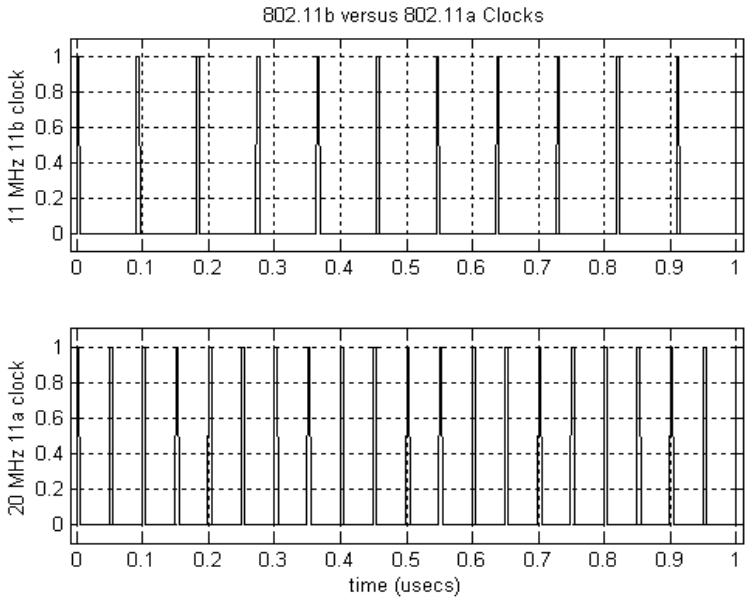



**Figure 19-16—Aligning the 11 MHz and 20 MHz clocks**

The 11 Barker chips of the preamble and header are transmitted aligned with this timing epoch. The first Barker chip is transmitted synchronous to the epoch, and then the remaining 10 chips follow. This is repeated over the duration of the preamble and header.

The peak of the continuous-time single carrier pulse shall be aligned to this epoch as shown in Figure 19-17.

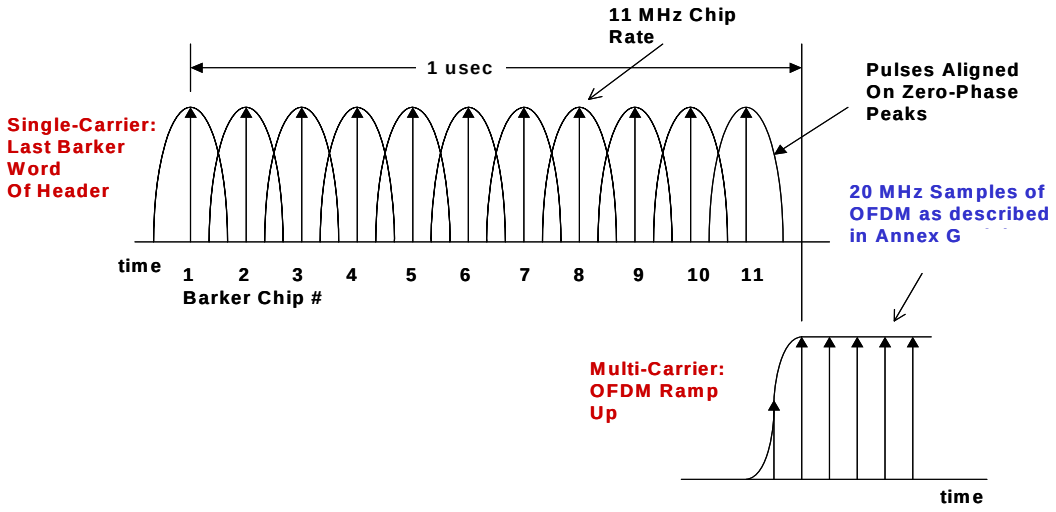



**Figure 19-17—Single carrier to OFDM time alignment**


Copyright © 2007 IEEE. All rights reserved.

The first full-strength OFDM sample is sent on the 1 µs epoch boundary, as illustrated in Figure 19-17. Tapering may precede this. The peak corresponds to the first full-strength sample described in Annex G.

### 19.7.2.4 Single carrier termination

The single carrier segment of a packet should terminate in nominally 0.1 µs with the same type shaping described for Clause 17. This is depicted in Figure 19-18. It is not necessary to completely flush the single carrier pulse-shaping filter. This minimizes the transition time overhead. This is informative as the basic requirement is to meet the spectral mask defined in 17.3.9.2.

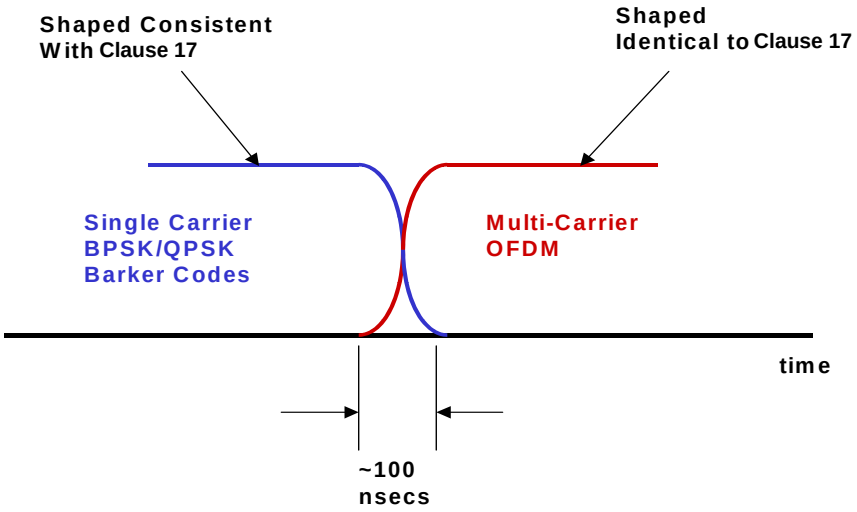



**Figure 19-18—Single carrier termination requirement**

This termination may be performed explicitly in the baseband processor, or it may be provided by filters in the transmit radio.

### 19.7.2.5 Transition carrier frequency requirement

The carrier frequency shall be coherent across the packet segments. This effect is depicted in Figure 19-19.

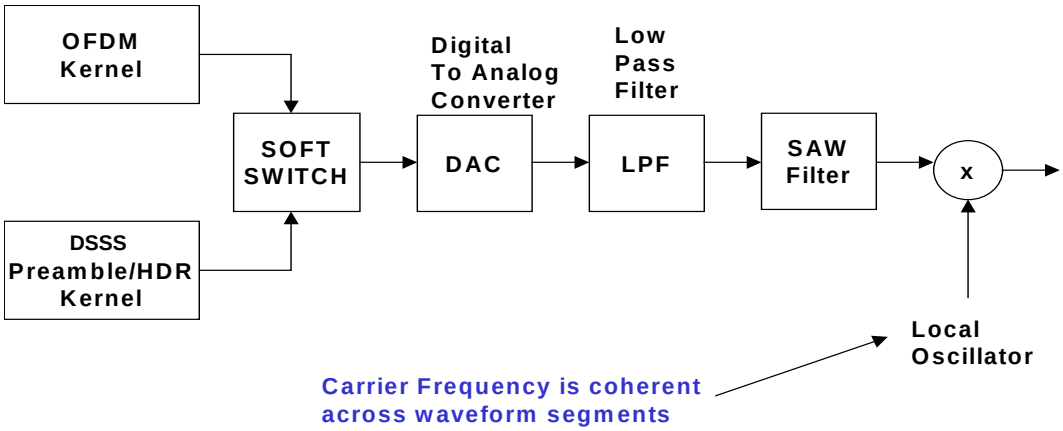



**Figure 19-19—Carrier frequency coherency shall be maintained**

Copyright © 2007 IEEE. All rights reserved.

### 19.7.2.6 Transition carrier phase requirement

The carrier phase shall be coherent across the single carrier to multicarrier transition. This coherency shall be differentially established relative to the phase of the last Barker symbol transmitted (the last 11 single carrier chips). The OFDM segment symbols shall be transmitted with one of four phases relative to the phase of OFDM symbols as described in Clause 17. These phases include 0, 90, 180, or 270 degrees, depending on the phase of the last Barker symbol. The phase of the first OFDM symbol (as referenced by the pilot tones) shall be 45 degrees more than the phase of the last Barker symbol. “More than” implies a clockwise rotation as shown in Figure 19-20.

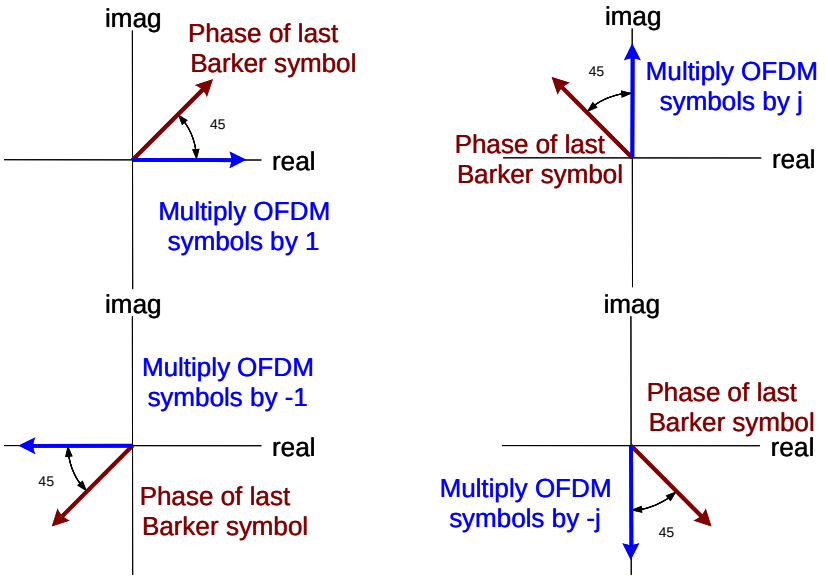



**Figure 19-20—The phase of the first OFDM segment symbol is established by the last Barker symbol**

In a transmit implementation using I/Q signaling, it is common to maximally energize in the I-and-Q channels concurrently for BPSK or QPSK signaling. The analog stages of the transmit radio tend to perform best with this configuration. To achieve this effect, typically the BPSK or QPSK I-and-Q alignment of the Barker symbols are at 45 degrees, 135 degrees, –135 degrees, and –45 degrees, as shown in Figure 19-21. This Barker symbol alignment is used to establish the phase of the OFDM signal.

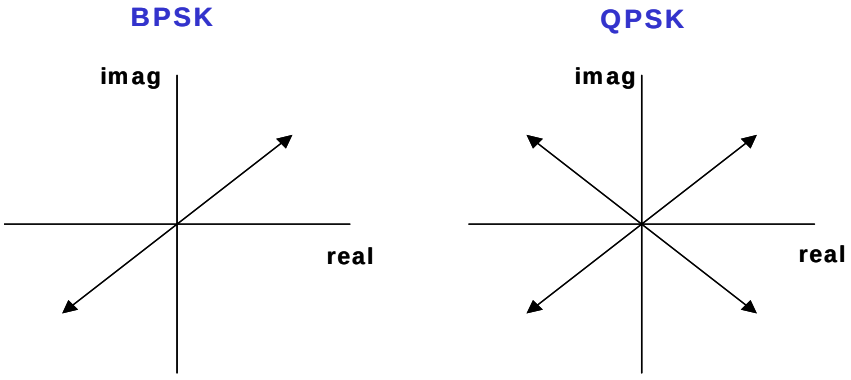



**Figure 19-21—BPSK and QPSK signaling with the I/Q channels maximally energized**


Copyright © 2007 IEEE. All rights reserved.

Figure 19-20 is a series of diagrams illustrating the phase relationship between the last Barker symbol (not the last chip) in the header and subsequent OFDM symbols. For example, if the phase of the last Barker symbol is in the first quadrant at 45 degrees, then the phase of the OFDM symbols will be transmitted as described in Annex G unmodified. However, if the phase of the last Barker symbol is in the second quadrant (135 degree phase), then the phase of the OFDM symbols will be rotated by +90 degrees relative to the phase of the samples in Annex G. If the phase of the last Barker symbol is in the third quadrant (–135 degree phase), then the phase of the OFDM symbols will be rotated by +180 degrees relative to the phase of the samples in Annex G. If the phase of the last Barker symbol is in the fourth quadrant (–45 degree phase), then the phase of the OFDM symbols will be rotated by +270 degrees relative to the phase of the samples in Annex G.

If the transmitter generates the Barker symbols at some other angular relationship to the I/Q axes, then the OFDM symbols shall be transmitted at a phase 45 degrees more than the phase of the last 11-chip Barker symbol.

#### 19.7.2.7 Transmit modulation accuracy requirement

The preceding subclauses establish transmit modulation requirements without mention of required accuracy. The accuracy is as described in 17.3.9.7.

The required accuracy for a given transmit packet is data rate dependent. The packet accuracy is set by the data rate of the OFDM portion of the packet. The preamble and header will be transmitted with the same fidelity requirement as the fidelity requirement levied on the OFDM portion of the packet. For the single carrier portion of the packet, the EVM is interpreted as normalized mean-squared error.

## 19.8 ERP PLME

### 19.8.1 PLME SAP

Table 19-6 lists the additional MIB attributes that may be accessed by the PHY sublayer entities and the intralayer of higher LMEs. These attributes are accessed via the PLME_GET, PLME_SET, PLME_RESET, and PLME_CHARACTERISTICS primitives defined in 10.4.

### 19.8.2 MIB

High Rate PHY MIB attributes are defined in Annex D with additions from this supplement and with specific values defined in Table 19-6.

**Table 19-6—MIB attribute default values/ranges**

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11 PHY Operation Table** | | |
| dot11PHYtype | ERP (X'06') | Static |
| dot11CurrentRegDomain | Implementation dependent | Static |
| dot11TempType | Implementation dependent | Static |
| **dot11 PHY Antenna Table** | | |
| dot11CurrentTxAntenna | Implementation dependent | Dynamic |
| dot11DiversitySupport | Implementation dependent | Static |
| dot11CurrentRxAntenna | Implementation dependent | Dynamic |

Copyright © 2007 IEEE. All rights reserved.

**Table 19-6—MIB attribute default values/ranges *(continued)***

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11 PHY Tx Power Table** | | |
| dot11NumberSupportedPowerLevels | Implementation dependent | Static |
| dot11TxPowerLevel1 | Implementation dependent | Static |
| dot11TxPowerLevel2 | Implementation dependent | Static |
| dot11TxPowerLevel3 | Implementation dependent | Static |
| dot11TxPowerLevel4 | Implementation dependent | Static |
| dot11TxPowerLevel5 | Implementation dependent | Static |
| dot11TxPowerLevel6 | Implementation dependent | Static |
| dot11TxPowerLevel7 | Implementation dependent | Static |
| dot11TxPowerLevel8 | Implementation dependent | Static |
| dot11CurrentTxPowerLevel | Implementation dependent | Dynamic |
| **dot11 Phy DSSS Table** | | |
| dot11CurrentChannel | Implementation dependent | Dynamic |
| **dot11 Reg Domains Supported Table** | | |
| dot11RegDomainsSupportedValue(s) | Implementation dependent | Static |
| **dot11 PHY Antennas List Table** | | |
| dot11SupportedTxAntenna | Implementation dependent | Static |
| dot11SupportedRxAntenna | Implementation dependent | Static |
| dot11DiversitySelectionRx | Implementation dependent | Dynamic |
| **dot11 Supported Data Rates Tx Table** | | |
| dot11SupportedDataratesTxValue | X'02' = 1 Mb/s<br>X'04' = 2 Mb/s<br>X'0B' = 5.5 Mb/s<br>X'16' = 11 Mb/s<br>X'0C' = 6 Mb/s<br>X'12' = 9 Mb/s<br>X'18' = 12 Mb/s<br>X'24' = 18 Mb/s<br>X'2C = 22 Mb/s<br>X'30' = 24 Mb/s<br>X'42 = 33 Mb/s<br>X'48' = 36 Mb/s<br>X'60' = 48 Mb/s<br>X'6C' = 54 Mb/s | Static |


Copyright © 2007 IEEE. All rights reserved.

**Table 19-6—MIB attribute default values/ranges *(continued)***

| Managed object | Default value/range | Operational semantics |
|---|---|---|
| **dot11 Supported Data Rates Rx Table** | | |
| dot11SupportedDataRatesRxValue | X'02' = 1 Mb/s<br>X'04' = 2 Mb/s<br>X'0B' = 5.5 Mb/s<br>X'16' = 11 Mb/s<br>X'0C' = 6 Mb/s<br>X'12' = 9 Mb/s<br>X'18' = 12 Mb/s<br>X'24' = 18 Mb/s<br>X'2C = 22 Mb/s<br>X'30' = 24 Mb/s<br>X'42 = 33 Mb/s<br>X'48' = 36 Mb/s<br>X'60' = 48 Mb/s<br>X'6C' = 54 Mb/s | Static |
| **dot11 HRDSSS PHY Table** | | |
| dot11ShortPreambleOptionImplemented | True | Static |
| dot11PBCCOptionImplemented | Implementation dependent | Static |
| dot11ChannelAgilityPresent | Implementation dependent | Static |
| dot11ChannelAgilityEnabled | False/Boolean | Dynamic |
| **dot11 PHY ERP Table** | | |
| dot11ERP-PBCCOptionImplemented | False/Boolean | Static |
| dot11DSSS-OFDMOptionImplemented | False/Boolean | Static |
| dot11DSSS-OFDMOptionEnabled | False/Boolean | Dynamic |
| dot11ShortSlotTimeOptionImplemented | False/Boolean | Static |
| dot11ShortSlotTimeOptionEnabled | False/Boolean | Dynamic |

### 19.8.3 TXTIME

The value of TXTIME is calculated for each modulation type based on parameters in the TXVECTOR. For the 1, 2, 5.5, and 11 Mb/s modes with DSSS, CCK, and PBCC modulation formats, the value shall be calculated as described in 18.3.4.

#### 19.8.3.1 ERP-OFDM TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated using the ERP-OFDM TXTIME calculation as shown in Equation (19-6).

$$\text{TXTIME} = T_{PREAMBLE} + T_{SIGNAL} + T_{SYM} \times \text{Ceiling}\left((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}\right) + \text{Signal Extension} \quad (19\text{-}6)$$

where

$T_{PREAMBLE}$, $T_{SIGNAL}$, and $T_{SYM}$ are defined in Table 17-4 in 17.3.2.3

$N_{DBPS}$ is the number of data bits per symbol and is derived from the DATARATE parameter in Table 17-3 in 17.3.2.2

Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| Ceiling | is a function that returns the smallest integer value greater than or equal to its argument value |
| Signal Extension | is 6 μs |

#### 19.8.3.2 ERP-PBCC TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated according to the following:

For PBCC 5.5 Mb/s and 11 Mb/s, see 18.3.4.

For ERP-PBCC-22 Mb/s, use Equation (19-7).

TXTIME = PreambleLength + PLCPHeaderTime + Ceiling(((LENGTH+PBCC) × 8) / DATARATE) (19-7)

For ERP-PBCC-33 Mb/s, use Equation (19-8).

TXTIME = PreambleLength + PLCPHeaderTime
+ Ceiling(((LENGTH+PBCC) × 8) / DATARATE) + ClkSwitchTime (19-8)

where

| | |
|---|---|
| LENGTH and DATARATE | are values from the TXVECTOR parameter of the corresponding PLME_TXTIME request primitive |
| PBCC | has a value of 1 if the SIGNAL value from the TXVECTOR parameter specifies ERP-PBCC and has a value of 0 otherwise |
| PreambleLength | is 144 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE” or 72 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE” |
| PLCPHeaderTime | is 48 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE” or 24 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE” |
| LENGTH | is in units of octets |
| DATARATE | is in units of Mb/s |
| ClkSwitchTime | is defined as 1 μs |
| Ceiling | is a function that returns the smallest integer value greater than or equal to its argument value |

#### 19.8.3.3 DSSS-OFDM TXTIME calculations

The value of the TXTIME parameter returned by the PLME_TXTIME.confirm primitive shall be calculated according to Equation (19-9):

TXTIME = PreambleLengthDSSS + PLCPHeaderTimeDSSS
+ PreambleLengthOFDM + PLCPSignalOFDM
+ 4 × Ceiling((PLCPServiceBits + 8 × (NumberOfOctets) + PadBits) / $N_{DBPS}$) + SignalExtension (19-9)

where

| | |
|---|---|
| PreambleLengthDSSS | is 144 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “LONGPREAMBLE,” or 72 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates “SHORTPREAMBLE” |


Copyright © 2007 IEEE. All rights reserved.

| | |
|---|---|
| PLCPHeaderTimeDSSS | is 48 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates "LONGPREAMBLE," or 24 μs if the PREAMBLE_TYPE value from the TXVECTOR parameter indicates "SHORTPREAMBLE" |
| Ceiling | is a function that returns the smallest integer value greater than or equal to its argument value |
| PreambleLengthOFDM | is 8 μs |
| PLCPSignalOFDM | is 4 μs |
| PLCPServiceBits | is 16 bits |
| NumberOfOctets | is the number of data octets in the PSDU |
| PadBits | is 6 bits |
| SignalExtension | is 6 μs |
| $N_{DBPS}$ | is the number of data bits per OFDM symbol |

### 19.8.4 ERP-OFDM PLCP PSDU definition

The DSSS PHY characteristics in Table 19-7 shall be used for the ERP for the purposes of MAC timing calculations.

**Table 19-7—ERP characteristics**

| Characteristic | Value |
|---|---|
| aSlotTime | Long = 20 μs, short = 9 μs |
| aSIFSTime | 10 μs |
| aCCATime | <15 μs for long slot time or <4 μs for Short Slot Time, see 19.4.6 |
| aPHY-RX-START-Delay | 24 μs for ERP-OFDM,<br>192 μs for ERP-DSSS/CCK with long preamble, and<br>96 μs for ERP-DSSS/CCK with short preamble |
| aRxTxTurnaroundTime | <5 μs |
| aTxRxTurnaroundTime | <10 μs |
| aTxPLCPDelay | Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met. |
| aRxPLCPDelay | Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met. |
| aRxTxSwitchTime | <<1 μs |
| aTxRampOnTime | Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met. |
| aTxRampOffTime | Implementation dependent as long as the requirements of aSIFSTime are met. |
| ATxRFDelay | Implementation dependent as long as the requirements of aRxTxTurnaround-Time are met. |
| ARxRFDelay | Implementation dependent as long as the requirements of aSIFSTime and aCCATime are met. |
| aAirPropagationTime | <<1 μs |
| aMACProcessingDelay | <2 μs |
| aPreambleLength | 20 μs |
| aPLCPHeaderLength | 4 μs |
| aMPDUMaxLength | 4095 |
| aCWmin(0) | 31 |

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 767 of 1232

**Table 19-7—ERP characteristics *(continued)***

| Characteristic | Value |
|---|---|
| aCWmin(1) | 15 |
| ACWmin | The set aCWmin() |
| aCWmax | 1023 |

The slot time shall be 20 μs, unless the BSS consists only of ERP STAs that support the Short Slot Time option. STAs indicate support for a short slot time by setting the Short Slot Time subfield to 1 when transmitting Association Request and Reassociation Request MMPDUs. If the BSS consists of only ERP STAs that support the Short Slot Time option, an optional 9 μs slot time may be used. APs indicate usage of a 9 μs slot time by setting the Short Slot Time subfield to 1 in all Beacon, Probe Response, Association Response, and Reassociation MMPDU transmissions as described in 7.3.1.4. STAs shall use short slot if the BSS indicates short slot.

## 19.9 Extended rate PMD sublayer

### 19.9.1 Scope and field of application

This subclause describes the PMD services provided to the PLCP for the ERP.

### 19.9.2 Overview of service

The ERP sublayer accepts PLCP sublayer service primitives and provides the actual means by which data are transmitted or received from the medium. The combined functions of the Extended Rate PMD sublayer primitives and parameters for the receive function result in a data stream, timing information, and associated received signal parameters being delivered to the PLCP sublayer. A similar functionality is provided for data transmission.

### 19.9.3 Overview of Interactions

The primitives associated with the PLCP sublayer to the ERP fall into two basic categories, as follows:

a) Service primitives that support PLCP peer-to-peer interactions
b) Service primitives that have local significance and that support sublayer-to-sublayer interactions

### 19.9.4 Basic service and options

All of the service primitives described in this subclause are considered mandatory, unless otherwise specified.

#### 19.9.4.1 PMD_SAP peer-to-peer service primitives

Table 19-8 indicates the primitives for peer-to-peer interactions.

**Table 19-8—PMD_SAP peer-to-peer services**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_Data | X | X | | |


Copyright © 2007 IEEE. All rights reserved.

#### 19.9.4.2 PMD_SAP sublayer-to-sublayer service primitives

Table 19-9 indicates the primitives for sublayer-to-sublayer interactions.

**Table 19-9—PMD_SAP sublayer-to-sublayer services**

| Primitive | Request | Indicate | Confirm | Response |
|---|---|---|---|---|
| PMD_TXSTART | X | | | |
| PMD_TXEND | X | | | |
| PMD_ANTSEL | X | | | |
| PMD_TXPWRLVL | X | | | |
| PMD_MODULATION | X | | | |
| PMD_PREAMBLE | X | | | |
| PMD_RATE | X | | | |
| PMD_RSSI | | X | | |
| PMD_SQ | | X | | |
| PMD_CS | | X | | |
| PMD_ED | | X | | |

#### 19.9.4.3 PMD_SAP service primitive parameters

Table 19-10 shows the parameters used by one or more of the PMD_SAP service primitives.

**Table 19-10—List of parameters for the PMD primitives**

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| TXD_UNIT | PMD_DATA.request | 0 to (2^*n*)–1, where *n* is the number of bits per symbol for the modulation and rate specified in PMD_ MODU-LATION.request and PMD_RATE.request primitives. | This parameter represents a single block of data, which, in turn, is used by the PMD to be encoded into a transmitted symbol. |
| RXD_UNIT | PMD_DATA.indicate | 0 to (2^*n*)–1, where *n* is the number of bits per symbol for the modulation and rate specified in PMD_ MODU-LATION.request and PMD_RATE.request primitives. | This parameter represents a single symbol that has been demodulated by the PMD entity. |
| MODULATION | PMD_MODULATION.request | ERP-DSSS, ERP-CCK, PBCC, ERP-PBCC, ERP-OFDM, DSSS-OFDM | The MODULATION parameter specifies to the PMD layer, which ERP modulation format is for transmission of the PSDU portion of the PPDU. |

Copyright © 2007 IEEE. All rights reserved.

**Table 19-10—List of parameters for the PMD primitives *(continued)***

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| PREAMBLE | PMD_PREAMBLE.request | 0 for long, 1 for short | PREAMBLE selects which of the ERP preamble types is used for PLCP transmission, when applicable. It is not applicable to ERP-OFDM format. |
| ANT_STATE | PMD_ANTSEL.request | 1 to 256 | ANT_STATE selects which of the available antennas is used for transmission. The number of available antennas is determined from the MIB table parameters. |
| TXPWR_LEVEL | PMD_TXPWRLVL.request | 1–8 (max of 8 levels) | TXPWR_LEVEL selects which of the optional transmit power levels should be used for the current PPDU transmission. The number of available power levels is determined from the MIB table parameters. |
| RATE | PMD_RATE.request | X'0A' for 1 Mb/s<br>X'14' for 2 Mb/s<br>X'37' for 5.5 Mb/s<br>X'6E' for 11 Mb/s<br>X'DC' for 22 Mb/s<br>X'21' for 33 Mb/s<br>X'75' for 12 Mb/s BPSK<br>X'E7' for 24 Mb/s QPSK<br>X'4B' for 48 Mb/s 16 QAM<br>X'AA' for 72 Mb/s 64QAM | RATE selects which of the ERP data rates is used for PSDU transmission. Note that the OFDM rates are the raw, uncoded rates as in 17.3.7 and 17.5.5 and represent the rates existing at this interface. |
| RSSI | PMD_RSSI.indicate | 8 bits of RSSI (256 levels) | The RSSI is a measure of the RF energy received. Mapping of the RSSI values to actual received power is implementation dependent. See 19.9.5.10. |
| SQ | PMD_SQ.indicate | 8 bits of SQ | This parameter is a measure of the signal quality received by the ERP during the PLCP preamble and header. It is not applicable to ERP-OFDM format. See 19.9.5.11. |
| CS | PMD_CS.indicate | 0 for DISABLED1 for ENABLED | The PMD_CS (preamble detect) primitive, in conjunction with the PMD_ED, provides the CCA status through the PLCP layer PHY-CCA primitive. PMD_CS indicates a binary status of ENABLED or DISABLED. PMD_CS is ENABLED upon detection of Barker code or OFDM sync signals. PMD_CS is DISABLED otherwise. |


Copyright © 2007 IEEE. All rights reserved.

**Table 19-10—List of parameters for the PMD primitives *(continued)***

| Parameter | Associated primitive | Value | Description |
|---|---|---|---|
| ED | PMD_ED.indicate | 0 for DISABLED1 for ENABLED | The PMD_ED primitive, along with the PMD_SQ, provides CCA status at the PLCP layer through the PHY-CCA primitive. PMD_ED indicates a binary status of ENABLED or DISABLED. PMD_ED is ENABLED when the RSSI indicated in the PMD_RSSI is greater than the detection threshold. PMD_ED is DISALBED otherwise. |

### 19.9.5 PMD_SAP detailed service specification

Subclauses 19.9.5.1 through 19.9.5.13 describe the services provided by each PMD primitive.

#### 19.9.5.1 PMD_DATA.request

This primitive is the same as that defined in 17.5.5.1 and 18.4.5.1 except that the parameter TXD_UNIT is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.3.

#### 19.9.5.2 PMD_DATA.indicate

This primitive is the same as that defined in 17.5.5.2 and 18.4.5.2 except that the parameter RXD_UNIT is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.3.

#### 19.9.5.3 PMD_MODULATION.request

This primitive is the same as that defined in 18.4.5.3 except that the parameter MODULATION is expanded in scope to reflect the supported modulation formats of ERP as defined in 19.9.4.3.

#### 19.9.5.4 PMD_PREAMBLE.request

This primitive is the same as that defined in 18.4.5.4, including the definition of the parameter PREAMBLE. This primitive is not used in association with transmission of ERP-OFDM modulations.

#### 19.9.5.5 PMD_TXSTART.request

This primitive is the same as that defined in 17.5.5.3 and 18.4.5.6.

#### 19.9.5.6 PMD_TXEND.request

This primitive is the same as that defined in 17.5.5.4 and 18.4.5.7.

#### 19.9.5.7 PMD_ANTSEL.request

This primitive is the same as that defined in 18.4.5.8, including the definition of the parameter ANT_STATE.

Copyright © 2007 IEEE. All rights reserved.

#### 19.9.5.8 PMD_TXPRWLVL.request

This primitive is the same as that defined in 17.5.5.5, including the definition of the parameter TXPWR_LEVEL.

#### 19.9.5.9 PMD_RATE.request

This primitive is the same as that defined in 17.5.5.6 and 18.4.5.10, except that the parameter RATE is expanded in scope to reflect the supported ERP transmission rates as defined in 19.9.4.3.

#### 19.9.5.10 PMD_RSSI.indicate

This primitive is the same as that defined in 17.5.5.7 and 18.4.5.11, including the parameter RSSI. This primitive is used to aid in link optimization algorithms such as roaming decisions.

#### 19.9.5.11 PMD_SQ.indicate

This primitive is the same as that defined in 18.4.5.12, including the parameter SQ. This primitive is not used in association with reception of ERP-OFDM modulations. This primitive is used to aid in link optimization algorithms such as roaming decisions.

#### 19.9.5.12 PMD_CS.indicate

This primitive is the same as that defined in 18.4.5.13, except that its use is expanded for use with all ERP modulation types as described in 19.3.5.

#### 19.9.5.13 PMD_ED.indicate

This primitive is the same as that defined in 18.4.5.14, except that its use is expanded for use with all ERP modulation types as described in 19.3.5.


Copyright © 2007 IEEE. All rights reserved.

# Annex A

(normative)

# Protocol Implementation Conformance Statement (PICS) proforma

## A.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.11-2007 shall complete the following protocol implementation conformance statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) By the supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) By the user, or potential user, of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS proformas);

d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

## A.2 Abbreviations and special symbols

### A.2.1 Symbols for Status column

M mandatory
O optional
O.<n> optional, but support of at least one of the group of options labeled by the same numeral <n> is required
pred: conditional symbol, including predicate identification

### A.2.2 General abbreviations for Item and Support columns

N/A not applicable
AD address function capability
CF implementation under test (IUT) configuration
DS direct sequence
FR medium access control (MAC) frame capability
FS frame sequence capability
FT frame transmission
HRDS High Rate direct sequence
MD multidomain
PC protocol capability

Copyright © 2007 IEEE. All rights reserved.

## A.3 Instructions for completing the PICS proforma

### A.3.1 General structure of the PICS proforma

The first parts of the PICS proforma, Implementation identification and Protocol summary, are to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed questionnaire, divided into subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) or by entering a value or a set or a range of values. (Note that there are some items where two or more choices from a set of possible answers may apply. All relevant choices are to be marked in these cases.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered. The third column contains the reference or references to the material that specifies the item in the main body of this standard. The remaining columns record the status of each item, i.e., whether support is mandatory, optional, or conditional, and provide the space for the answers (see also A.3.4). Marking an item as supported is to be interpreted as a statement that all relevant requirements of the subclauses and normative annexes, cited in the References column for the item, are met by the implementation.

A supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A<*I*> or X<*I*>, respectively, for cross-referencing purposes, where <*I*> is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format or presentation.

The PICS proforma for a STA consists of A.4.1 through A.4.4 inclusive, and at least one of A.4.5, A.4.6, A.4.7, A.4.8, A.4.9, or A.4.12 corresponding to the physical layer (PHY) implemented.

A completed PICS proforma, including any Additional Information and Exception Information, is the PICS for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's capabilities, if this makes for easier and clearer presentation of the information.

### A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large quantity of information will be supplied, and a PICS can be considered complete without any such information. Examples of such Additional Information might be an outline of the ways in which an (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

### A.3.3 Exception information

It may happen occasionally that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer


Copyright © 2007 IEEE. All rights reserved.

will be found in the Support column for this. Instead, the supplier shall write the missing answer into the Support column, together with an X<*I*> reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception Information item itself.

An implementation for which an Exception Information item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

### A.3.4 Conditional status

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply, mandatory or optional, are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the N/A answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form "<pred>:<S>", where "<pred>" is a predicate as described below, and "<S>" is one of the status symbols M or O.

If the value of the predicate is true, the conditional item is applicable, and its status is given by S: the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant and the N/A answer is to be marked.

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is true if the item is marked as supported, and is false otherwise.
b) A boolean expression constructed by combining item-references using the boolean operator OR: the value of the predicate is true if one or more of the items is marked as supported, and is false otherwise.

Each item referenced in a predicate, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

Copyright © 2007 IEEE. All rights reserved.

# A.4 PICS proforma—IEEE Std 802.11-2007[32]

## A.4.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for queries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification, e.g., name(s) and version(s) of the machines and/or operating systems(s), system names | |

NOTE 1—Only the first three items are required for all implementations. Other information may be completed as appropriate in meeting the requirement for full identification.

NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

## A.4.2 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.11-2007 |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | Amd. : Corr. :<br>Amd. : Corr. : |
| Have any exception items been required?<br>(See A.3.3; the answer Yes means that the implementation does not conform to IEEE Std 802.11-2007.) | Yes ☐ No ☐ |

| | |
|---|---|
| Date of statement (dd/mm/yy) | |

[32]*Copyright release for PICS proforma:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.


Copyright © 2007 IEEE. All rights reserved.

### A.4.3 IUT configuration

| Item | IUT configuration | References | Status | Support |
|---|---|---|---|---|
| | What is the configuration of the IUT? | | | |
| * CF1 | Access point (AP) | 5.2 | O.1 | Yes ☐ No ☐ |
| * CF2 | Independent station (*not* an AP) | 5.2 | O.1 | Yes ☐ No ☐ |
| * CF3 | Frequency-hopping spread spectrum (FHSS) PHY for the 2.4 GHz band | — | O.2 | Yes ☐ No ☐ |
| * CF4 | Direct sequence spread spectrum (DSSS) PHY for the 2.4 GHz band | — | O.2 | Yes ☐ No ☐ |
| * CF5 | Infrared (IR) PHY | — | O.2 | Yes ☐ No ☐ |
| * CF6 | Orthogonal frequency division multiplexing (OFDM) PHY for the 5 GHz band | — | O.2 | Yes ☐ No ☐ |
| * CF7 | High-speed PHY | — | O.2 | Yes ☐ No ☐ |
| * CF8 | Is multidomain operation capability implemented? | 7.3.2.10, 7.3.2.11, 9.8, 11.1.3.4 | O.3 | Yes ☐ No ☐ |
| * CF9 | Extended Rate PHY (ERP) | Clause 19 | O.2 | Yes ☐ No ☐ N/A ☐ |
| * CF10 | Is spectrum management operation supported? | 7.3.1.4, 11.6 | CF6:O | Yes ☐ No ☐ |
| *CF11 | Is regulatory classes capability implemented? | 7.3.2.12, 17.3.8.3.2, 17.3.8.6, 17.4.2, Annex I, Annex J | CF6&CF8& CF10:O | Yes ☐ No ☐ N/A ☐ |
| * CF12 | Quality of service (QoS) supported | 9.9, 9.10 | O | Yes ☐ No ☐ |

### A.4.4 MAC protocol

#### A.4.4.1 MAC protocol capabilities

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| | Are the following MAC protocol capabilities supported? | | | |
| PC1 | Authentication service | 5.4.3.1, 5.4.3.2, 8.1, Annex C | M | Yes ☐ No ☐ |
| PC1.1 | Authentication state | 11.3 | M | Yes ☐ No ☐ |
| PC1.2 | Open System authentication | 8.1.1 | M | Yes ☐ No ☐ |
| PC1.3 | Shared Key authentication | 8.1.2, 8.3 | PC2:M | Yes ☐ No ☐ N/A ☐ |
| * PC2 | Wired equivalent privacy (WEP) algorithm This capability is deprecated (applicable only to systems that are backward compatible). | 5.4.3.3, 8.2.1, Annex C | O | Yes ☐ No ☐ |
| PC2.1 | WEP encryption procedure | 8.2.1 | PC2:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.4.1 MAC protocol capabilities *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| PC2.2 | WEP decryption procedure | 8.2.1 | PC2:M | Yes ☐ No ☐ N/A ☐ |
| PC2.3 | Security services management | 8.4 | M | Yes ☐ No ☐ |
| PC3 | Distributed coordination function (DCF) | 9.1, 9.2, Annex C | M | Yes ☐ No ☐ |
| PC3.1 | Network allocation vector (NAV) function | 9.2.1, 9.2.5, 9.3.2.2 | M | Yes ☐ No ☐ |
| PC3.2 | Interframe space usage and timing | 9.2.3, 9.2.5, 9.2.10 | M | Yes ☐ No ☐ |
| PC3.3 | Random Backoff function | 9.2.4 | M | Yes ☐ No ☐ |
| PC3.4 | DCF Access procedure | 9.2.5.1, 9.2.5.5 | M | Yes ☐ No ☐ |
| PC3.5 | Random Backoff procedure | 9.2.5.2 | M | Yes ☐ No ☐ |
| PC3.6 | Recovery procedures and retransmit limits | 9.2.5.3 | M | Yes ☐ No ☐ |
| PC3.7 | Request to send (RTS)/clear to send (CTS) procedure | 9.2.5.4, 9.2.5.6, 9.2.5.7 | M | Yes ☐ No ☐ |
| PC3.8 | Directed MAC protocol data unit (MPDU) transfer | 9.2.6 | M | Yes ☐ No ☐ |
| PC3.9 | Broadcast and multicast MPDU transfer | 9.2.7 | M | Yes ☐ No ☐ |
| PC3.10 | MAC-level acknowledgment | 9.2.2, 9.2.8 | M | Yes ☐ No ☐ |
| PC3.11 | Duplicate detection and recovery | 9.2.9 | M | Yes ☐ No ☐ |
| * PC4 | Point coordinator (PC) | 9.1, 9.3, Annex C | CF1:O | Yes ☐ No ☐ N/A ☐ |
| PC4.1 | Maintenance of contention-free period (CFP) structure and timing | 9.3.1, 9.3.2 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| PC4.2 | Point coordination function (PCF) MPDU transfer from PC | 9.3.3 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| * PC4.3 | PCF MPDU transfer to PC | 9.3.3 | PC4:O | Yes ☐ No ☐ N/A ☐ |
| PC4.4 | Overlapping PC provisions | 9.3.3.2 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| PC4.5 | Polling list maintenance | 9.3.4 | PC4.3:M | Yes ☐ No ☐ N/A ☐ |
| * PC5 | Contention-free (CF)-Pollable | 9.1, 9.3, Annex C | CF2:O | Yes ☐ No ☐ N/A ☐ |
| PC5.1 | Interpretation of CFP structure and timing | 9.3.1, 9.3.2 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| PC5.2 | PCF MPDU transfer to/from and CF-Pollable station (STA) | 9.3.3 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| PC5.3 | Polling list update | 9.3.4 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| PC6 | Fragmentation | 9.2, 9.4, Annex C | M | Yes ☐ No ☐ |
| PC7 | Defragmentation | 9.2, 9.5, Annex C | M | Yes ☐ No ☐ |
| PC8 | MAC data service | 9.1.6, 9.7, Annex C | M | Yes ☐ No ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.4.1 MAC protocol capabilities *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| PC8.1 | Reorderable-Multicast service class | 9.7 | M | Yes ☐ No ☐ |
| PC8.2 | StrictlyOrdered service class | 9.7 | O | Yes ☐ No ☐ |
| PC9 | Multirate support | 9.6, Annex C | M | Yes ☐ No ☐ |
| * PC10 | Multiple outstanding MAC service data unit (MSDU) support | 9.7, Annex C | O | Yes ☐ No ☐ |
| PC10.1 | Multiple outstanding MSDU transmission restrictions | 9.7 | PC10:M | Yes ☐ No ☐ N/A ☐ |
| PC11 | Timing synchronization function (TSF) | 11.1, Annex C | M | Yes ☐ No ☐ |
| PC11.1 | Timing in an infrastructure network | 11.1.1.1, 11.1.4 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC11.2 | Timing in an independent basic service set (IBSS) | 11.1.1.2, 11.1.4 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC11.3 | Beacon generation function | 11.1.2 | M | Yes ☐ No ☐ N/A ☐ |
| PC11.4 | TSF synchronization and accuracy | 11.1.2 | M | Yes ☐ No ☐ |
| PC11.5 | Infrastructure basic service set (BSS) initialization | 11.1.3 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC11.6 | IBSS initialization | 11.1.3 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC11.7 | Passive scanning | 11.1.3 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC11.8 | Active scanning | 11.1.3 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC11.9 | Probe response | 11.1.3 | M | Yes ☐ No ☐ |
| PC11.10 | Hop Synchronization function | 11.1.5 | CF3:M | Yes ☐ No ☐ N/A ☐ |
| PC12 | Infrastructure power management | 11.2.1, Annex C | M | Yes ☐ No ☐ |
| PC12.1 | STA power management modes | 11.2.1.1, 11.2.1.8 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC12.2 | Traffic indication map (TIM) transmission | 11.2.1.2, 11.2.1.3 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC12.3 | AP function during contention period (CP) | 11.2.1.4 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC12.4 | AP function during CFP | 11.2.1.5 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| PC12.5 | Receive function during CP | 11.2.1.6 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC12.6 | Receive function during CFP | 11.2.1.7 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| PC12.7 | Aging function | 11.2.1.9 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC13 | IBSS power management | 11.2.2, Annex C | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC13.1 | Initialization of power management | 11.2.2.2 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC13.2 | STA power state transitions | 11.2.2.3 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC13.3 | Announcement traffic indication message (ATIM) and frame transmission | 11.2.2.4 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC14 | Association and reassociation | 5.4, 11.3, 11.3.2, Annex C | M | Yes ☐ No ☐ |
| PC14.1 | Association state | 11.3.2 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.4.1 MAC protocol capabilities *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| PC14.2 | STA association procedure | 11.3.2.1 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC14.3 | AP association procedure | 11.3.2.2 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC14.4 | STA reassociation procedure | 11.3.2.3 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| PC14.5 | AP reassociation procedure | 11.3.2.4 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| PC15 | Management information base (MIB) | Annex D | M | Yes ☐ No ☐ |
| PC15.1 | dot11SMTbase, dot11SmtAuthenticationAlgorithms | Annex D | M | Yes ☐ No ☐ |
| * PC15.2 | dot11SMTprivacy | Annex D | PC2:M | Yes ☐ No ☐ N/A ☐ |
| PC15.3 | dot11MACbase, dot11CountersGroup, dot11MacGroupAddresses | Annex D | M | Yes ☐ No ☐ |
| * PC15.4 | dot11MACStatistics | Annex D | O | Yes ☐ No ☐ |
| PC15.5 | dot11ResourceTypeID | Annex D | M | Yes ☐ No ☐ |
| PC16 | Set dot11ShortPreambleOptionImplemented to 1 | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC17 | Set packet binary convolutional code (PBCC) subfield as described in reference | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC18 | Set DSSS-OFDM subfield as described in reference | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC19 | Set channel agility subfield as described in reference | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC20 | Set Short Slot Time subfield as described in reference | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC21 | Monitor each received short time slot sub-field and take action as described in refer-ence. | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC22 | Transmit the ERP Information element in each transmitted Beacon or Probe Responses in the format and with content as described in reference | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC23 | Receive the ERP Information element and employ a protection mechanism when required prior to transmitting information using ERP-OFDM modulation | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC24 | Determine the value of aCWmin based on the characteristic rate set as described in the reference | 9.2.12 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC25 | Transmit control response frames at the largest basic rates less than equal to the rate received and with the same PHY options or use the highest mandatory rate if no basic rate meets the above criterion | 9.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC26 | Transmit broadcast or multicast frames at a rate contained in the BSSBasicRateSet parameter | 9.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC27 | Transmit unicast frames at any supported rate selected by a rate switching mechanism as long as it is supported by the destination STA | 9.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.4.1 MAC protocol capabilities *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| PC28 | Do not transmit at a data rate higher than the greatest rate in the OperationalRateSet | 9.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC29 | Use ERP Information element to control use of protection mechanism as described in the reference | 9.13 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC30 | Updated NAV is long enough to cover frame and any response | 9.13 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC31 | Support transmission of CTS-to-self sequence as described in the references | 9.2.11, 9.12 | CF9:O | Yes ☐ No ☐ N/A ☐ |
| PC32 | Support reception of CTS-to-self sequence as described in the references | 9.2.11, 9.12 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC33 | Update NAV | 9.13 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| PC34 | Robust security network association (RSNA) | 7.2.2, 7.3.1.4, 5.4.3.3, 8.7.2, 11.3.1, 11.3.2, 8.3.3 | O | Yes ☐ No ☐ |
| PC34.1 | RSN Information Element (IE) | 7.3.2.25 | PC34:M | Yes ☐ No ☐ |
| PC34.1.1 | Group cipher suite | 7.3.2.25 | PC34.1:M | Yes ☐ No ☐ |
| PC34.1.2 | Pairwise cipher suite list | 7.3.2.25 | PC34.1:M | Yes ☐ No ☐ |
| PC34.1.2.1 | Counter mode with Cipher-block chaining Message authentication code Protocol (CCMP) data confidentiality protocol | 8.3.3 | PC34:M | Yes ☐ No ☐ |
| PC34.1.2.1.1 | CCMP cryptographic encapsulation procedure | 8.3.3.3 | PC34.1.2.1:M | Yes ☐ No ☐ |
| PC34.1.2.1.2 | CCMP decapsulation procedure | 8.3.3.4 | PC34.1.2.1:M | Yes ☐ No ☐ |
| PC34.1.2.2 | Temporal Key Integrity Protocol (TKIP) data confidentiality protocol | 8.3.2 | PC34:O | Yes ☐ No ☐ |
| PC34.1.2.2.1 | TKIP cryptographic encapsulation procedure | 8.3.2.1.1 | PC34.1.2.2:M | Yes ☐ No ☐ |
| PC34.1.2.2.2 | TKIP decapsulation procedure | 8.3.2.1.2 | PC34.1.2.2:M | Yes ☐ No ☐ |
| PC34.1.2.2.3 | TKIP countermeasures | 8.3.2.4 | PC34.1.2.2:M | Yes ☐ No ☐ |
| PC34.1.2.2.4 | TKIP security services management | 8.3.2.3 | PC34.1.2.2:M | Yes ☐ No ☐ |
| PC34.1.3 | Authentication key management (AKM) suite list | 7.3.2.25, 8.3.1 | PC34.1:M | Yes ☐ No ☐ |
| PC34.1.3.1 | IEEE 802.1X-defined/ RSNA key management | 7.3.2.25 | PC34.1:M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.4.1 MAC protocol capabilities *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| PC34.1.3.2 | Preshared key (PSK)/ RSNA key management | 7.3.2.25 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3 | RSNA key management | 8.5 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3.1 | Key hierarchy | 8.5, 8.6 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3.1.1 | Pairwise key hierarchy | 8.5.1.2 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3.1.2 | Group key hierarchy | 8.5.1.3 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3.2 | 4-Way Handshake | 8.5.3 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.3.3.3 | Group Key Handshake | 8.5.4 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.4 | RSN capabilities | 7.3.2.25, 8.1.2 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.5 | RSNA preauthentication | 8.4.6.1 | PC34.1: O | Yes ☐ No ☐ |
| PC34.1.6 | RSNA security association management | 8.4 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.7 | RSNA pairwise master key security association (PMKSA) caching | 8.4.1, 8.4.6.2 | PC34.1: M | Yes ☐ No ☐ |
| PC34.1.8 | RSNA extended service set (ESS) | 8.4.6, 8.4.8 | (PC34.1 and CF1):M | Yes ☐ No ☐ |
| PC34.1.8.1 | RSNA PeerKey Handshake | 8.5.8 | PC34.1.8:O | Yes ☐ No ☐ |
| PC34.1.9 | RSNA IBSS | 8.4.4, 8.4.7, 8.4.9 | (PC34.1 and CF2):O | Yes ☐ No ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.4.2 MAC frames

| Item | MAC frame | References | Status | Support |
|---|---|---|---|---|
| | Is transmission of the following MAC frames supported? | Clause 7, Annex C | | |
| FT1 | Association request | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FT2 | Association response | Clause 7 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| FT3 | Reassociation request | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FT4 | Reassociation response | Clause 7 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| FT5 | Probe request | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FT6 | Probe response | Clause 7 | M | Yes ☐ No ☐ |
| FT7 | Beacon | Clause 7 | M | Yes ☐ No ☐ |
| FT8 | ATIM | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FT9 | Disassociation | Clause 7 | M | Yes ☐ No ☐ |
| FT10 | Authentication | Clause 7 | M | Yes ☐ No ☐ |
| FT11 | Deauthentication | Clause 7 | M | Yes ☐ No ☐ |
| FT12 | Power save (PS)-Poll | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FT13 | RTS | Clause 7 | M | Yes ☐ No ☐ |
| FT14 | CTS | Clause 7 | M | Yes ☐ No ☐ |
| FT15 | Acknowledgment (ACK) | Clause 7 | M | Yes ☐ No ☐ |
| FT16 | CF-End | Clause 7 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| FT17 | CF End+CF-Ack | Clause 7 | PC4:M | Yes ☐ No ☐ N/A ☐ |
| FT18 | Data | Clause 7 | M | Yes ☐ No ☐ |
| FT19 | Data + CF-Ack | Clause 7 | (PC4 OR PC5):M | Yes ☐ No ☐ N/A ☐ |
| FT20 | Data + CF-Poll | Clause 7 | PC4.3:M | Yes ☐ No ☐ N/A ☐ |
| FT21 | Data + CF-Ack+CF-Poll | Clause 7 | PC4.3:M | Yes ☐ No ☐ N/A ☐ |
| FT22 | Null | Clause 7 | M | Yes ☐ No ☐ |
| FT23 | CF-Ack (no data) | Clause 7 | (PC4 OR PC5):M | Yes ☐ No ☐ N/A ☐ |
| FT24 | CF-Poll (no data) | Clause 7 | PC4.3:M | Yes ☐ No ☐ N/A ☐ |
| FT25 | CF-Ack+CF-Poll (no data) | Clause 7 | PC4.3:M | Yes ☐ No ☐ N/A ☐ |
| | Is reception of the following MAC frames supported? | Clause 7, Annex C | | |
| FR1 | Association request | Clause 7 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| FR2 | Association response | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FR3 | Reassociation request | Clause 7 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| FR4 | Reassociation response | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FR5 | Probe request | Clause 7 | M | Yes ☐ No ☐ |
| FR6 | Probe response | Clause 7 | M | Yes ☐ No ☐ |
| FR7 | Beacon | Clause 7 | M | Yes ☐ No ☐ |
| FR8 | ATIM | Clause 7 | CF2:M | Yes ☐ No ☐ N/A ☐ |
| FR9 | Disassociation | Clause 7 | M | Yes ☐ No ☐ |
| FR10 | Authentication | Clause 7 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.4.2 MAC frames *(continued)*

| Item | MAC frame | References | Status | Support |
|---|---|---|---|---|
| FR11 | Deauthentication | Clause 7 | M | Yes ☐ No ☐ |
| FR12 | PS-Poll | Clause 7 | CF1:M | Yes ☐ No ☐ N/A ☐ |
| FR13 | RTS | Clause 7 | M | Yes ☐ No ☐ |
| FR14 | CTS | Clause 7 | M | Yes ☐ No ☐ |
| FR15 | ACK | Clause 7 | M | Yes ☐ No ☐ |
| FR16 | CF-End | Clause 7 | M | Yes ☐ No ☐ |
| FR17 | CF End+CF-Ack | Clause 7 | M | Yes ☐ No ☐ |
| FR18 | Data | Clause 7 | M | Yes ☐ No ☐ |
| FR19 | Data + CF-Ack | Clause 7 | M | Yes ☐ No ☐ |
| FR20 | Data + CF-Poll | Clause 7 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| FR21 | Data + CF-Ack+CF-Poll | Clause 7 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| FR22 | Null | Clause 7 | M | Yes ☐ No ☐ |
| FR23 | CF-Ack (no data) | Clause 7 | (PC4 OR PC5):M | Yes ☐ No ☐ N/A ☐ |
| FR24 | CF-Poll (no data) | Clause 7 | PC5:M | Yes ☐ No ☐ N/A ☐ |
| FR25 | CF-Ack+CF-Poll (no data) | Clause 7 | PC5:M | Yes ☐ No ☐ N/A ☐ |

### A.4.4.3 Frame exchange sequences

| Item | Frame exchange sequence | References | Status | Support |
|---|---|---|---|---|
| | Are the following frame sequences supported? | | | |
| FS1 | Basic frame sequences | 9.12, Annex C | M | Yes ☐ No ☐ |
| FS2 | CF-Frame sequences | 9.12, Annex C | (PC4 OR PC5):M | Yes ☐ No ☐ N/A ☐ |

### A.4.4.4 MAC addressing functions

| Item | MAC Address function | References | Status | Support |
|---|---|---|---|---|
| | Are the following MAC Addressing functions supported? | | | |
| AD1 | STA universal individual IEEE 802 address | 7.1.3.3 | M | Yes ☐ No ☐ |
| AD2 | BSS identification (BSSID) generation | 7.1.3.3, 11.1.3, Annex C | M | Yes ☐ No ☐ |
| AD3 | Receive address matching | 7.1.3.3, 7.2.2, Annex C | M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.5 Frequency hopping (FH) PHY functions

| Item | Protocol feature | References | Status | Support |
|---|---|---|---|---|
| | Which requirements and options does the PHY support? | | | |
| FH1 | PHY service primitive parameters | | | |
| FH1.1 | TXVECTOR parameter: LENGTH | 14.2.2.1 | M | Yes ☐ No ☐ |
| FH1.2 | TXVECTOR parameter: PLCPBITRATE | 14.2.2.2 | M | Yes ☐ No ☐ |
| FH1.2.1 | PLCPBITRATE = X'00' (1.0 Mb/s) | 14.2.2.2 | M | Yes ☐ No ☐ |
| * FH1.2.2 | PLCPBITRATE = X'02' (2.0 Mb/s) | 14.2.2.2 | O | Yes ☐ No ☐ |
| FH1.3 | RXVECTOR parameter: LENGTH | 14.2.3.1 | M | Yes ☐ No ☐ |
| FH1.4 | RXVECTOR parameter: Receive signal strength indicator (RSSI) | 14.2.3.2 | O | Yes ☐ No ☐ |
| FH2 | Physical layer convergence procedure (PLCP) frame format | | | |
| FH2.1 | PLCP preamble: Sync | 14.3.2.1.1 | M | Yes ☐ No ☐ |
| FH2.2 | PLCP preamble: Start frame delimiter (SFD) | 14.3.2.1.2 | M | Yes ☐ No ☐ |
| FH2.3 | PLCP header: PSDU length word (PLW) | 14.3.2.2.1 | M | Yes ☐ No ☐ |
| FH2.4 | PLCP header: PLCP Signaling field (PSF) | 14.3.2.2.2 | M | Yes ☐ No ☐ |
| FH2.5 | PLCP header: Header error check (HEC) | 14.3.2.2.3 | M | Yes ☐ No ☐ |
| FH2.6 | PLCP data whitener: Scrambling and bias suppression encoding | 14.3.2.3, 14.3.3.1.1 | M | Yes ☐ No ☐ |
| FH3 | Transmit PLCP | | | |
| FH3.1 | Transmit: transmit on MAC request | 14.3.3.1.1 | M | Yes ☐ No ☐ |
| FH3.2 | Transmit: format and whiten frame | 14.3.3.1.1 | M | Yes ☐ No ☐ |
| FH3.3 | Transmit: Timing | 14.3.3.1.1 | M | Yes ☐ No ☐ |
| FH4 | Carrier sense (CS)/clear channel assessment (CCA) procedure | | | |
| FH4.1 | CS/CCA: perform on a minimum of one antenna | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.2. | CS/CCA: Detect preamble starting up to 20 μs after start of slot time | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.3 | CS/CCA: Detect preamble starting at least 16 μs prior to end of slot time | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.4 | CS/CCA: Detect random data | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.5 | CS/CCA: Perform on antenna with essentially same gain and pattern as transmit antenna | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.6 | CS/CCA: Detect valid SFD and PLCP header | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH4.7 | CS/CCA: Maintain BUSY indication until end of length contained in valid PLCP header | 14.3.3.2.1 | M | Yes ☐ No ☐ |
| FH5 | Receive PLCP | | | |
| FH5.1 | Receive: Receive and dewhiten frame | 14.3.3.3.1 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.5 Frequency hopping (FH) PHY functions *(continued)*

| Item | Protocol feature | References | Status | Support |
|---|---|---|---|---|
| FH6 | Physical layer management entity (PLME) | | | |
| FH6.1 | PLME: Support FH sync | 14.4.2.2 | M | Yes ☐ No ☐ |
| FH6.2 | PLME: Support PLME primitives | 14.4.3.2 | O | Yes ☐ No ☐ |
| FH7 | Geographic area specific requirements | | | |
| * FH7.1 | Geographic areas | | | |
| FH7.1.1 | North America | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.1.2 | Most of Europe | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.1.3 | Japan | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.1.4 | Spain | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.1.5 | France | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.1.6 | China | 14.6.2 | O.1 | Yes ☐ No ☐ |
| FH7.2 | Operating frequency range | 14.6.3 | FH7.1:M | Yes ☐ No ☐ |
| FH7.3 | Number of operating channels | 14.6.4 | FH7.1:M | Yes ☐ No ☐ |
| FH7.4 | Operating channel frequencies | 14.6.5 | FH7.1:M | Yes ☐ No ☐ |
| FH7.5 | Occupied channel bandwidth | 14.6.6 | FH7.1:M | Yes ☐ No ☐ |
| FH7.6 | Minimum hop rate | 14.6.7 | FH7.1:M | Yes ☐ No ☐ |
| FH7.7 | Hop sequences | 14.6.8 | FH7.1:M | Yes ☐ No ☐ |
| FH7.8 | Unwanted emissions | 14.6.9 | FH7.1:M | Yes ☐ No ☐ |
| FH8 | 1 Mb/s physical medium dependent (PMD) | | | |
| FH8.1 | Modulation 2GFSK, bit time (BT) = 0.5, 1 = positive frequency deviation, 0 = negative frequency deviation | 14.6.10 | M | Yes ☐ No ☐ |
| FH8.2 | Peak frequency deviation | 14.6.10 | M | Yes ☐ No ☐ |
| FH8.3 | Zero-Crossing error | 14.6.10 | M | Yes ☐ No ☐ |
| FH8.4 | Nominal channel data rate | 14.6.11 | M | Yes ☐ No ☐ |
| FH8.5 | Channel switching/settling time | 14.6.12 | M | Yes ☐ No ☐ |
| FH8.6 | Receive to transmit switch time | 14.6.13 | M | Yes ☐ No ☐ |
| FH8.7 | Nominal transmit power | 14.6.14.1 | M | Yes ☐ No ☐ |
| FH8.8 | Transmit power levels | 14.6.14.2 | M | Yes ☐ No ☐ |
| FH8.9 | Transmit power level control to < 100 mW | 14.6.14.3 | M | Yes ☐ No ☐ |
| FH8.10 | Transmit spectrum shape | 14.6.14.4 | M | Yes ☐ No ☐ |
| FH8.11 | Transmit center frequency tolerance | 14.6.14.5 | M | Yes ☐ No ☐ |
| FH8.12 | Transmitter ramp periods | 14.6.14.6 | M | Yes ☐ No ☐ |
| FH8.13 | Receiver input dynamic range | 14.6.15.1 | M | Yes ☐ No ☐ |
| FH8.14 | Receiver center frequency acceptance range | 14.6.15.2 | M | Yes ☐ No ☐ |
| FH8.15 | CCA power threshold for a probability of detection of 90% (preamble)/70% (random data) for 100 mW units | 14.6.15.3 | M | Yes ☐ No ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.5 Frequency hopping (FH) PHY functions *(continued)*

| Item | Protocol feature | References | Status | Support |
|---|---|---|---|---|
| FH8.16 | CCA power threshold for units > 100 mW; sensitivity threshold is 1/2 dB lower for every dB above 20 dBm | 14.6.15.3 | M | Yes ☐ No ☐ |
| FH8.17 | Minimum receiver sensitivity at frame error ratio (FER) = 3% with 400 octet frames | 14.6.15.4 | M | Yes ☐ No ☐ |
| FH8.18 | Intermodulation protection (IMp) | 14.6.15.5 | M | Yes ☐ No ☐ |
| FH8.19 | Desensitization (Dp) | 14.6.15.6 | M | Yes ☐ No ☐ |
| FH8.20 | Operating temperature range | 14.6.16 | M | Yes ☐ No ☐ |
| FH8.20.1 | Temperature type 1 | 14.6.16 | O | Yes ☐ No ☐ |
| FH8.20.2 | Temperature type 2 | 14.6.16 | O | Yes ☐ No ☐ |
| FH8.20.3 | Temperature type 3 | 14.6.16 | O | Yes ☐ No ☐ |
| FH9 | 2 Mb/s PMD | | | |
| FH9.1 | All 1M PMD requirements | 14.7.1 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.2 | Modulation 4GFSK, BT = 0.5 | 14.7.2 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.3 | Frame structure for 2M PHY | 14.7.2.1 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.4 | Nominal channel data rate | 14.7.3 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.5 | Input dynamic range | 14.7.3.1 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.6 | Minimum receiver sensitivity at FER = 3% with 400 octet frames | 14.7.3.2 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.7 | IMp | 14.7.3.3 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH9.8 | Dp | 14.7.3.4 | FH1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| FH10 | MIB | 14.8, Annex D | M | Yes ☐ No ☐ |
| FH10.1 | dot11PhyFHSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup | 14.8 | M | Yes ☐ No ☐ |

### A.4.6 Direct sequence PHY functions

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| | PLCP sublayer procedures | 15.2 | | |
| DS1 | Preamble prepend on transmit (TX) | 15.2.1 | M | Yes ☐ No ☐ |
| DS1.1 | PLCP frame format | 15.2.2, 15.2.3 | M | Yes ☐ No ☐ |
| DS1.2 | PLCP integrity check generation | 15.2.3, 15.2.3.6 | M | Yes ☐ No ☐ |
| DS1.3 | TX rate change capability | 15.2.3.3, 15.2.5 | M | Yes ☐ No ☐ |
| DS1.4 | Supported data rates | 15.1, 15.2.3.3 | M | Yes ☐ No ☐ |
| DS1.5 | Data whitener scrambler | 15.2.4 | M | Yes ☐ No ☐ |
| DS1.6 | Scrambler initialization | 15.2.4 | M | Yes ☐ No ☐ |
| DS2 | Preamble process on receive (RX) | 15.2.1 | | |
| DS2.1 | PLCP frame format | 15.2.2, 15.2.3 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.6 Direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| DS2.2 | PLCP integrity check verify | 15.2.3, 15.2.3.6 | M | Yes ☐ No ☐ |
| DS2.3 | RX Rate change capability | 15.2.3.3, 15.2.5 | M | Yes ☐ No ☐ |
| DS2.4 | Data whitener descrambler | 15.2.4 | M | Yes ☐ No ☐ |
| DS3 | Pseudonoise (PN) code sequence | 15.4.6.3 | M | Yes ☐ No ☐ |
| DS4 | Chipping continue on power-down | 15.2.6 | O | Yes ☐ No ☐ |
| *DS5 | Operating channel capability | 15.2.6, 15.4.6.2 | | |
| * DS5.1 | North America (FCC) | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.1.1 | Channel 1 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.2 | Channel 2 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.3 | Channel 3 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.4 | Channel 4 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.5 | Channel 5 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.6 | Channel 6 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.7 | Channel 7 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.8 | Channel 8 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.9 | Channel 9 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.10 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| DS5.1.11 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.1:M | Yes ☐ No ☐ N/A ☐ |
| * DS5.2 | Canada (IC) | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.2.1 | Channel 1 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.2 | Channel 2 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.3 | Channel 3 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.4 | Channel 4 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.5 | Channel 5 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.6 | Channel 6 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.7 | Channel 7 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.8 | Channel 8 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.9 | Channel 9 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.10 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| DS5.2.11 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.2:M | Yes ☐ No ☐ N/A ☐ |
| * DS5.3 | Europe (ETSI) | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.3.1 | Channel 1 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.2 | Channel 2 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.3 | Channel 3 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.4 | Channel 4 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.5 | Channel 5 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.6 | Channel 6 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.7 | Channel 7 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.8 | Channel 8 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.9 | Channel 9 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.6 Direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| DS5.3.10 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.11 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.12 | Channel 12 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| DS5.3.13 | Channel 13 | 15.2.6, 15.4.6.2 | DS5.3:M | Yes ☐ No ☐ N/A ☐ |
| * DS5.4 | France | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.4.1 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.4:M | Yes ☐ No ☐ N/A ☐ |
| DS5.4.2 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.4:M | Yes ☐ No ☐ N/A ☐ |
| DS5.4.3 | Channel 12 | 15.2.6, 15.4.6.2 | DS5.4:M | Yes ☐ No ☐ N/A ☐ |
| DS5.4.4 | Channel 13 | 15.2.6, 15.4.6.2 | DS5.4:M | Yes ☐ No ☐ N/A ☐ |
| * DS5.5 | Spain | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.5.1 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.5:M | Yes ☐ No ☐ N/A ☐ |
| DS5.5.2 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.5:M | Yes ☐ No ☐ N/A ☐ |
| * DS5.6 | Japan | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| * DS5.7 | China | 15.2.6, 15.4.6.2 | DS5:O.1 | Yes ☐ No ☐ N/A ☐ |
| DS5.7.1 | Channel 1 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.2 | Channel 2 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.3 | Channel 3 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.4 | Channel 4 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.5 | Channel 5 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.6 | Channel 6 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.7 | Channel 7 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.8 | Channel 8 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.9 | Channel 9 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.10 | Channel 10 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.11 | Channel 11 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.12 | Channel 12 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS5.7.13 | Channel 13 | 15.2.6, 15.4.6.2 | DS5.7:M | Yes ☐ No ☐ N/A ☐ |
| DS6 | Bits to symbol mapping | 15.4.6.4 | | |
| DS6.1 | 1 Mb/s | 15.4.6.4 | M | Yes ☐ No ☐ |
| DS6.2 | 2 Mb/s | 15.4.6.4 | M | Yes ☐ No ☐ |
| *DS7 | CCA functionality | 15.4.8.4 | | |
| DS7.1 | Energy Only (RSSI above threshold) | 15.4.8.4 | DS7:O.2 | Yes ☐ No ☐ |
| DS7.2 | IEEE 802.11 DSSS correlation | 15.4.8.4 | DS7:O.2 | Yes ☐ No ☐ |
| DS7.3 | Both methods | 15.4.8.4 | DS7:O.2 | Yes ☐ No ☐ |
| DS7.4 | Hold CCA busy for packet duration of a correctly received PLCP but carrier lost during reception of MPDU | 15.2.7 | M | Yes ☐ No ☐ |
| DS7.5 | Hold CCA busy for packet duration of a correctly received but out of specification PLCP | 15.2.7 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.6 Direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| DS8 | Transmit antenna selection | 15.4.5.5, 15.4.5.6 | O | Yes ☐ No ☐ |
| DS9 | Receive antenna diversity | 15.4.5.5, 15.4.5.6, 15.4.5.7 | O | Yes ☐ No ☐ |
| *DS10 | Antenna port(s) availability | 15.4.6.9 | O | Yes ☐ No ☐ |
| DS10.1 | 50 ¾ impedance | 15.4.6.9 | DS10:M | Yes ☐ No ☐ N/A ☐ |
| *DS11 | Transmit power level support | 15.4.5.8, 15.4.7.3 | O | Yes ☐ No ☐ |
| DS11.1 | If greater than 100 mW capability | 15.4.7.3 | DS11:M | Yes ☐ No ☐ N/A ☐ |
| *DS12 | Radio type (temperature range) | 15.4.6.10 | | |
| DS12.1 | Type 1 | 15.4.6.10 | DS12:O.3 | Yes ☐ No ☐ N/A ☐ |
| DS12.2 | Type 2 | 15.4.6.10 | DS12:O.3 | Yes ☐ No ☐ N/A ☐ |
| DS13 | Spurious emissions conformance | 15.4.6.5 | M | Yes ☐ No ☐ |
| DS14 | TX-to-RX turnaround time | 15.4.6.6 | M | Yes ☐ No ☐ |
| DS15 | RX-to-TX turnaround time | 15.4.6.7 | M | Yes ☐ No ☐ |
| DS16 | Slot time | 15.4.6.8 | M | Yes ☐ No ☐ |
| DS17 | Energy detection (ED) reporting time | 15.4.6.8, 15.4.8.4 | M | Yes ☐ No ☐ |
| DS18 | Minimum transmit power level | 15.4.7.2 | M | Yes ☐ No ☐ |
| DS19 | Transmit spectral mask conformance | 15.4.7.4 | M | Yes ☐ No ☐ |
| DS20 | Transmitted center frequency tolerance | 15.4.7.5 | M | Yes ☐ No ☐ |
| DS21 | Chip clock frequency tolerance | 15.4.7.6 | M | Yes ☐ No ☐ |
| DS22 | Transmit power-on ramp | 15.4.7.7 | M | Yes ☐ No ☐ |
| DS23 | Transmit power-down ramp | 15.4.7.7 | M | Yes ☐ No ☐ |
| DS24 | Radio frequency (RF) carrier suppression | 15.4.7.8 | M | Yes ☐ No ☐ |
| DS25 | Transmit modulation accuracy | 15.4.7.9 | M | Yes ☐ No ☐ |
| DS26 | Receiver minimum input level sensitivity | 15.4.8.1 | M | Yes ☐ No ☐ |
| DS27 | Receiver maximum input level | 15.4.8.2 | M | Yes ☐ No ☐ |
| DS28 | Receiver adjacent channel rejection | 15.4.8.3 | M | Yes ☐ No ☐ |
| DS29 | MIB | 15.3.2, Annex D | M | Yes ☐ No ☐ |
| DS29.1 | dot11PhyDSSSComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup | 15.3.2 | M | Yes ☐ No ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.7 IR baseband PHY functions

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| IR1 | Is the transmitted synchronization (SYNC) field length in the range of required number of pulse position modulation (PPM) slots, with the absence of a pulse in the last slot of the field? | 16.2.4.1 | M | Yes ☐ |
| IR2 | Is the transmitted SYNC field entirely populated by alternating presence and absence of pulses in consecutive PPM slots, with the absence of a pulse in the last slot of the field? | 16.2.4.1 | M | Yes ☐ |
| IR3 | Is the transmitted SFD field the binary sequence 1001, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? | 16.2.4.2 | M | Yes ☐ |
| IR4 | Is the transmitted data rate (DR) field pulse sequence equal to the correct value for the data rate provided by the TXVECTOR parameter PLCP BITRATE, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot? | 16.2.4.3 | M | Yes ☐ |
| IR5 | Is the transmitted dc level adjustment (DCLA) field 32 PPM slots long with the specified sequence for 1 Mb/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?<br>1 Mb/s:<br>00000000100000000000000010000000 | 16.2.4.4 | M | Yes ☐ |
| * IR5a | Does the unit support 2 Mb/s transmission? | 16.2.4.4 | O | Yes ☐ No ☐ |
| IR5b | If the unit supports 2 Mb/s transmission, is the transmitted DCLA field 32 PPM slots long with the specified sequence for 2 Mb/s, where 1 indicates a pulse in the PPM slot and 0 indicates no pulse in the PPM slot?<br>2 Mb/s:<br>00100010001000100010001000100010 | 16.2.4.4 | IR5a:M | Yes ☐ No ☐ N/A ☐ |
| IR6 | Is the transmitted LENGTH field the correct PPM representation of the unsigned 16-bit binary integer, least significant bit (LSB) transmitted first, equal to the correct value provided by the TXVECTOR parameter LENGTH? | 16.2.4.5 | M | Yes ☐ |
| IR7 | Is the transmitted cyclic redundancy code (CRC) field the correct PPM representation of the CRC value calculated according to the reference subclause, transmitted LSB first? | 16.2.4.6 | M | Yes ☐ |
| IR8 | Is the transmitted PLCP service data unit (PSDU) field the correct PPM representation of the PSDU, transmitted LSB first? | 16.2.4.7 | M | Yes ☐ |
| IR9 | When the CCA is false does transmission begin based on PHYTXSTART.request? | 16.2.5.1 | M | Yes ☐ |
| IR10 | Does the PHY issue a PHYTXSTART.confirm after the transmission of the PLCP header? | 16.2.5.1 | M | Yes ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.7 IR baseband PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| IR11 | Does the PHY accept each octet of the PSDU in a PHYDATA.request and answer with a PHYDATA.confirm? | 16.2.5.1 | M | Yes ☐ |
| IR12 | Does the PHY cease transmission in response to a PHYTXEND.request and answer with a PHYTXEND.confirm? | 16.2.5.1 | M | Yes ☐ |
| IR13 | Does the PHY of a receiving STA send a PHY-CCA.indicate during reception of the SYNC field? | 16.2.5.2 | M | Yes ☐ |
| IR14 | Does the PHY of a receiving STA properly receive a transmission that changes data rate according to the DR field? | 16.2.5.2 | M | Yes ☐ |
| IR15 | Does the PHY of a receiving STA properly reject an incorrect CRC? | 16.2.5.2 | M | Yes ☐ |
| IR16 | Does the PHY of a receiving STA properly reject a DR field other than those specified in reference subclause? | 16.2.5.2, 16.2.4.3 | M | Yes ☐ |
| IR17 | Does the PHY of a receiving STA send PHYRXSTART.indicate with correct RATE and LENGTH parameters after proper reception of PLCP preamble and PLCP header? | 16.2.5.2 | M | Yes ☐ |
| IR18 | Does the PHY of a receiving STA forward receive octets in PHYDATA.indicate primitives? | 16.2.5.2 | M | Yes ☐ |
| IR19 | Does the PHY of a receiving STA send a PHYRXEND.indicate after the final octet indicated by the LENGTH field? | 16.2.5.2 | M | Yes ☐ |
| IR20 | Does the PHY of a receiving STA send a PHY-CCA.indicate with a state value of IDLE after the PHYRXEND.indicate? | 16.2.5.2 | M | Yes ☐ |
| IR21 | Does the PHY reset its CCA detection mechanism upon receiving a PHY-CCARST.request, and respond with a PHY-CCARST.indicate? | 16.2.5.3 | M | Yes ☐ |
| IR22 | When transmitting at 1 Mb/s does the PHY transmit PPM symbols according to the 16-PPM Basic Rate Mapping table, transmitting from left to right? | 16.3.2.1, 16.3.2.2 | M | Yes ☐ |
| IR23 | When transmitting at 2 Mb/s does the PHY transmit PPM symbols according to the 4-PPM Enhanced Rate Mapping table, transmitting from left to right? | 16.3.2.1, 16.3.2.2 | IR5a:M | Yes ☐ |
| IR24 | Does the PHY operate over a temperature range of 0 °C to 40 °C? | 16.3.2.4 | M | Yes ☐ |
| * IR25 | If the unit is conformant to emitter radiation mask 1, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width? | 16.3.3.1 | O.1 | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.7 IR baseband PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| * IR26 | If the unit is conformant to emitter radiation mask 2, is the peak optical power of an emitted pulse within the specification range averaged over the pulse width? | 16.3.3.1 | O.1 | Yes ☐ No ☐ N/A ☐ |
| IR27 | Does the transmitted pulse shape conform to the description of the reference subclause? | 16.3.3.2 | M | Yes ☐ |
| IR28 | Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation Mask 1? | 16.3.3.3 | IR25:M | Yes ☐ No ☐ N/A ☐ |
| IR28a | Does the emitter radiation pattern as a function of angle conform to the requirements of the reference subclause as applicable based on conformance to emitter radiation Mask 2? | 16.3.3.3 | IR26:M | Yes ☐ No ☐ N/A ☐ |
| IR29 | Is the peak emitter optical output as a function of wavelength in the range specified? | 16.3.3.4 | M | Yes ☐ |
| IR30 | Does the spectrum of the transmit signal amplitude as a voltage or current meet the requirements of the reference subclause? | 16.3.3.5 | M | Yes ☐ |
| IR31 | Does the receiver sensitivity meet the requirements of the reference subclause for receive signals of both 1 Mb/s and 2 Mb/s? | 16.3.4.1 | M | Yes ☐ |
| IR32 | Does the receiver exhibit a dynamic range as specified in reference subclause? | 16.3.4.2 | M | Yes ☐ |
| IR33 | Does the receiver field of view (FOV) conform to the requirements of the reference subclause? | 16.3.4.3 | M | Yes ☐ |
| IR34 | When it is known that the conditions are such that the carrier detect signal and the ED signal are false, is the CCA asserted IDLE? | 16.3.5.1 | M | Yes ☐ |
| IR35 | When the conditions are such that ED is true for greater than the time defined in reference subclause, does CCA become IDLE? | 16.3.5.1 | M | Yes ☐ |
| IR36 | When conditions are such that either carrier detect or ED go true, does CCA go BUSY? | 16.3.5.1 | M | Yes ☐ |
| IR37 | Are these compliance groups implemented? dot11PhyIRComplianceGroup, dot11PhyRegDomainsSupportGroup, and dot11PhyOperationComplianceGroup | 16.4 | M | Yes ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.8 OFDM PHY functions

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| **OF1: OFDM PHY Specific Service Parameters** | | | | |
| OF1.1 | TXVECTOR parameter: LENGTH | 17.2.2.1 | M | Yes ☐ No ☐ |
| OF1.2 | TXVECTOR parameter: DATARATE | 17.2.2.2 | M | Yes ☐ No ☐ |
| OF1.2.1 | DATARATE = 6.0 Mb/s | 17.2.2.2 | M | Yes ☐ No ☐ |
| *OF1.2.2 | DATARATE = 9.0 Mb/s | 17.2.2.2 | O | Yes ☐ No ☐ |
| OF1.2.3 | DATARATE = 12.0 Mb/s | 17.2.2.2 | M | Yes ☐ No ☐ |
| *OF1.2.4 | DATARATE = 18.0 Mb/s | 17.2.2.2 | O | Yes ☐ No ☐ |
| OF1.2.5 | DATARATE = 24.0 Mb/s | 17.2.2.2 | M | Yes ☐ No ☐ |
| *OF1.2.6 | DATARATE = 36.0 Mb/s | 17.2.2.2 | O | Yes ☐ No ☐ |
| *OF1.2.7 | DATARATE = 48.0 Mb/s | 17.2.2.2 | O | Yes ☐ No ☐ |
| *OF1.2.8 | DATARATE = 54.0 Mb/s | 17.2.2.2 | O | Yes ☐ No ☐ |
| OF1.3 | TXVECTOR parameter: SERVICE | 17.2.2.3 | M | Yes ☐ No ☐ |
| OF1.4 | TXVECTOR parameter: TXPWR_LEVEL | 17.2.2.4 | M | Yes ☐ No ☐ |
| OF1.5 | RXVECTOR parameter: LENGTH | 17.2.3.1 | M | Yes ☐ No ☐ |
| OF1.6 | RXVECTOR parameter: RSSI | 17.2.3.2 | M | Yes ☐ No ☐ |
| *OF1.7 | 10 MHz Channel spacing | 17.2.2, 17.2.3, 17.2.3.3 | CF11:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.1 | DATARATE = 3 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.2 | DATARATE = 4.5 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.3 | DATARATE = 6 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.4 | DATARATE = 9 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.5 | DATARATE = 12 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.6 | DATARATE = 18 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.7 | DATARATE = 24 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.7.8 | DATARATE = 27 Mb/s (10 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.8 | 5 MHz Channel spacing | 17.2.2, 17.2.3, 17.2.3.3 | CF11:O | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| *OF1.8.1 | DATARATE = 1.5Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.2 | DATARATE = 2.25 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.3 | DATARATE = 3Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.4 | DATARATE = 4.5 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.5 | DATARATE = 6 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.6 | DATARATE = 9 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.7 | DATARATE = 12 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| *OF1.8.8 | DATARATE = 13.5 Mb/s (5 MHz channel spacing) | 17.2.2, 17.2.3, 17.2.3.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| **OF2: OFDM PLCP Sublayer** | | | | |
| OF2.1 | RATE-dependent parameters | 17.3.2.2 | M | Yes ☐ No ☐ |
| OF2.2 | Timing related parameters | 17.3.2.3 | M | Yes ☐ No ☐ |
| OF2.3 | PLCP preamble: SYNC | 17.3.3 | M | Yes ☐ No ☐ |
| OF2.4 | PLCP header: SIGNAL | 17.3.4 | M | Yes ☐ No ☐ |
| OF2.5 | PLCP header: LENGTH | 17.3.4.1 | M | Yes ☐ No ☐ |
| OF2.6 | PLCP header: RATE | 17.3.4.2 | M | Yes ☐ No ☐ |
| OF2.7 | PLCP header: parity, reserve | 17.3.4.3 | M | Yes ☐ No ☐ |
| OF2.8 | PLCP header: SIGNAL TAIL | 17.3.4.3 | M | Yes ☐ No ☐ |
| OF2.9 | PLCP header: SERVICE | 17.3.5.1 | M | Yes ☐ No ☐ |
| OF2.10 | PLCP protocol data unit (PPDU): TAIL | 17.3.5.2 | M | Yes ☐ No ☐ |
| OF2.11 | PPDU: PAD | 17.3.5.3 | M | Yes ☐ No ☐ |
| OF2.12 | PLCP/OFDM PHY data scrambler and descrambler | 17.3.5.4 | M | Yes ☐ No ☐ |
| OF2.13 | Convolutional encoder | 17.3.5.5 | M | Yes ☐ No ☐ |
| OF2.13.1 | Rate R = 1/2 | 17.3.5.5 | M | Yes ☐ No ☐ |
| OF2.13.2 | Punctured coding R = 2/3 | 17.3.5.5 | OF1.2.7:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| OF2.13.3 | Punctured coding R = 3/4 | 17.3.5.5 | OF1.2.2 OR OF1.2.4 OR OF1.2.6 OR OF1.2.8:M | Yes ☐ No ☐ N/A ☐ |
| OF2.14 | Data interleaving | 17.3.5.6 | M | Yes ☐ No ☐ |
| OF2.15 | Subcarrier modulation mapping | 17.3.5.7 | M | Yes ☐ No ☐ |
| OF2.15.1 | Binary phase shift keying (BPSK) | 17.3.5.7 | M | Yes ☐ No ☐ |
| OF2.15.2 | Quadrature phase shift keying (QPSK) | 17.3.5.7 | M | Yes ☐ No ☐ |
| OF2.15.3 | 16-quadrature amplitude modulation (QAM) | 17.3.5.7 | M | Yes ☐ No ☐ |
| OF2.15.4 | 64-QAM | 17.3.5.7 | OF1.2.7 OR OF1.2.8:M | Yes ☐ No ☐ N/A ☐ |
| OF2.16 | Pilot subcarriers | 17.3.5.8 | M | Yes ☐ No ☐ |
| OF2.17 | OFDM modulation | 17.3.5.9 | M | Yes ☐ No ☐ |
| OF2.18 | Packet duration calculation | | M | Yes ☐ No ☐ |
| OF2.19 | CCA | | | |
| OF2.19.1 | CCA: RSSI | 17.3.6 | M | Yes ☐ No ☐ |
| OF2.19.2 | CCA: indication to MAC sublayer | 17.3.6 | M | Yes ☐ No ☐ |
| OF2.20 | PLCP data modulation and modulation rate change | 17.3.7 | M | Yes ☐ No ☐ |
| OF2.21 | Modulation-dependent parameters (10 MHz channel spacing) | 17.3.2.2 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF2.22 | Timing-related parameters (10 MHz channel spacing) | 17.3.2.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF2.23 | PLCP header: RATE (10 MHz channel spacing) | 17.3.4.1 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF2.24 | Modulation-dependent parameters (5 MHz channel spacing) | 17.3.2.2 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF2.25 | Timing-related parameters (5 MHz channel spacing) | 17.3.2.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF2.26 | PLCP header: RATE (5 MHz channel spacing) | 17.3.4.1 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| **OF3: PDM Operating Specification General** | | | | |
| OF3.1 | Occupied channel bandwidth | | | |
| OF3.1.1 | 20 MHz channel spacing | 17.3.8.1 | M | Yes ☐ No ☐ |
| OF3.1.2 | 10 MHz channel spacing | 17.3.8.1 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF3.1.3 | 5 MHz channel spacing | 17.3.8.1 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF3.2 | Operating frequency range | 17.3.8.3.1 | | |
| *OF3.2.1 | 4.9 GHz band | Annex J | CF11:O | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| *OF3.2.2 | 5.0 GHz band | Annex J | CF11:M | Yes ☐ No ☐ N/A ☐ |
| OF3.2.3 | 5.15–5.25 GHz band | 17.3.8.3 | O.1 | Yes ☐ No ☐ N/A ☐ |
| OF3.2.4 | 5.25–5.35 GHz band | 17.3.8.3 | O.1 | Yes ☐ No ☐ N/A ☐ |
| *OF3.2.5 | 5.47–5.725 GHz band | Annex J | CF10:M | Yes ☐ No ☐ N/A ☐ |
| OF3.2.6 | 5.725–5.85 GHz band | 17.3.8.3 | O.1 | Yes ☐ No ☐ N/A ☐ |
| OF3.3 | Channelization | | | |
| OF3.3.1 | 5.15–5.25 GHz (20 MHz channel spacing) | Annex J | O.1 | Yes ☐ No ☐ |
| OF3.3.2 | 5.25–5.35 GHz (20 MHz channel spacing) | Annex J | O.1 | Yes ☐ No ☐ |
| OF3.3.3 | 5.725–5.825 GHz (20 MHz channel spacing) | Annex J | O.1 | Yes ☐ No ☐ |
| OF3.3.4 | 5.15–5.25 GHz band in Japan (20 MHz channel spacing) | Annex J | CF11:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.5 | 5.47–5.725 GHz (20 MHz channel spacing) | Annex J | CF10& OF3.2.5:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.6 | 5.725–5.85 GHz (20 MHz channel spacing) | Annex J | O.1 | Yes ☐ No ☐ |
| OF3.3.7 | 4.9 GHz band (20 MHz channel spacing) | Annex J | CF11& OF3.2.1:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.8 | 5.0 GHz band (20 MHz channel spacing) | Annex J | CF11& OF3.2.2:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.9 | 4.9 GHz band (10 MHz channel spacing) | Annex J | CF11& OF3.2.1& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.10 | 5.0 GHz band (10 MHz channel spacing) | Annex J | CF11& OF3.2.2& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.11 | 4.9 GHz band (5 MHz channel spacing) | Annex J | CF11& OF3.2.1& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF3.3.12 | 5.0 GHz band (5 MHz channel spacing) | Annex J | CF11& OF3.2.2& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF3.4 | Number of operating channels | Annex J | M | Yes ☐ No ☐ |
| OF3.5 | Operating channel frequencies | Annex J | M | Yes ☐ No ☐ |
| OF3.6 | Transmit and receive in band and out of band spurious emission | Annex J | M | Yes ☐ No ☐ |
| OF3.6.1 | Interference-limited areas, 4.9 GHz band (20 MHz channel spacing) | Annex J | CF11& OF3.2.1:M | Yes ☐ No ☐ N/A ☐ |
| OF3.6.2 | Interference-limited areas, 5.0 GHz band (20 MHz channel spacing) | Annex J | CF11& OF3.2.2:M | Yes ☐ No ☐ N/A ☐ |
| OF3.6.3 | Interference-limited areas, 4.9 GHz band (10 MHz channel spacing) | Annex J | CF11& OF3.2.1& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| OF3.6.4 | Interference-limited areas, 5.0 GHz band (10 MHz channel spacing) | Annex J | CF11& OF3.2.2& OF1.7:O | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| OF3.6.5 | Interference-limited areas, 4.9 GHz band (5 MHz channel spacing) | Annex J | CF11& OF3.2.1& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| OF3.6.6 | Interference-limited areas, 5.0 GHz band (5 MHz channel spacing) | Annex J | CF11& OF3.2.2& OF1.8:O | Yes ☐ No ☐ N/A ☐ |
| OF3.7 | TX RF delay | 17.3.8.5 | M | Yes ☐ No ☐ |
| OF3.8 | Slot time | 17.3.8.6 | M | Yes ☐ No ☐ |
| OF3.8.1 | Slot time (20 MHz channel spacing) | 17.3.8.6 | CF11& RC2:M | Yes ☐ No ☐ N/A ☐ |
| OF3.8.2 | Slot time (10 MHz channel spacing) | 17.3.8.6 | CF11& RC3& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF3.8.3 | Slot time (5 MHz channel spacing) | 17.3.8.6 | CF11& RC4& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF3.9 | Transmit and receive antenna port impedance | 17.3.8.7 | M | Yes ☐ No ☐ |
| OF3.10 | Transmit and receive operating temperature range | 17.3.8.8 | M | Yes ☐ No ☐ |
| OF3.10.1 | Type 1 (0 °C to 40 °C) | 17.3.8.8 | M | Yes ☐ No ☐ |
| OF3.10.2 | Type 2 (–20 °C to 50 °C) | 17.3.8.8 | O | Yes ☐ No ☐ |
| OF3.10.3 | Type 3 (–30 °C to 70 °C) | 17.3.8.8 | O | Yes ☐ No ☐ |
| **OF4: PMD Transmit Specification** | | | | |
| OF4.1 | Transmit power levels | | M | Yes ☐ No ☐ |
| OF4.1.1 | Power level (5.15–5.25 GHz) | 17.3.9.1 | OF3.3.1:M | Yes ☐ No ☐ N/A ☐ |
| OF4.1.2 | Power level (5.25–5.35 GHz) | 17.3.9.1 | OF3.3.2:M | Yes ☐ No ☐ N/A ☐ |
| OF4.1.3 | Power level (5.725–5.825 GHz) | 17.3.9.1 | OF3.3.3:M | Yes ☐ No ☐ N/A ☐ |
| OF4.2 | Spectrum mask | 17.3.9.2 | M | Yes ☐ No ☐ |
| OF4.3 | Spurious | 17.3.9.3 | M | Yes ☐ No ☐ |
| OF4.4 | Center frequency tolerance | 17.3.9.4 | M | Yes ☐ No ☐ |
| OF4.5 | Clock frequency tolerance | 17.3.9.5 | M | Yes ☐ No ☐ |
| OF4.6 | Modulation accuracy | | | Yes ☐ No ☐ |
| OF4.6.1 | Center frequency leakage | 17.3.9.6.1 | M | Yes ☐ No ☐ |
| OF4.6.2 | Spectral flatness | 17.3.9.6.2 | M | Yes ☐ No ☐ |
| OF4.6.3 | Transmitter constellation error < –5 dB | 17.3.9.6.3 | M | Yes ☐ No ☐ |
| OF4.6.4 | Transmitter constellation error < –8 dB | 17.3.9.6.3 | OF1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| OF4.6.5 | Transmitter constellation error < –10 dB | 17.3.9.6.3 | M | Yes ☐ No ☐ |
| OF4.6.6 | Transmitter constellation error < –13 dB | 17.3.9.6.3 | OF1.2.4:M | Yes ☐ No ☐ N/A ☐ |
| OF4.6.7 | Transmitter constellation error < –16 dB | 17.3.9.6.3 | M | Yes ☐ No ☐ |
| OF4.6.8 | Transmitter constellation error < –19 dB | 17.3.9.6.3 | OF1.2.6:M | Yes ☐ No ☐ N/A ☐ |
| OF4.6.9 | Transmitter constellation error < –22 db | 17.3.9.6.3 | OF1.2.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.6.10 | Transmitter constellation error < –25 dB | 17.3.9.6.3 | OF1.2.8:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| OF4.7 | Power level, 4.9 GHz band (20 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.1: M | Yes ☐ No ☐ N/A ☐ |
| OF4.8 | Power level, 5.0 GHz band (20 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.2: M | Yes ☐ No ☐ N/A ☐ |
| OF4.9 | Power level, 5.47–5.725 GHz band | 17.3.9.1 | CF11& OF3.12.3: M | Yes ☐ No ☐ N/A ☐ |
| OF4.10 | Power level, 4.9 GHz band (10 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.1& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.11 | Power level, 5.0 GHz band (10 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.2& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.12 | Power level, 4.9 GHz band (5 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.1& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF4.13 | Power level, 5.0 GHz band (5 MHz channel spacing) | 17.3.9.1 | CF11& OF3.12.2& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF4.14 | Spectrum mask (20 MHz channel spacing) | 17.3.9.2 | CF11:M | Yes ☐ No ☐ N/A ☐ |
| OF4.1 | Spectrum mask (10 MHz channel spacing) | 17.3.9.2 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.16 | Spectrum mask (5 MHz channel spacing) | 17.3.9.2 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17 | Transmitter constellation error (10 MHz channel spacing) | | | |
| OF4.17.1 | Transmitter constellation error < –5 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.1:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.2 | Transmitter constellation error < –8 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.2:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.3 | Transmitter constellation error < –10 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.3:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.4 | Transmitter constellation error < –13 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.4:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.5 | Transmitter constellation error < –16 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.5:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.6 | Transmitter constellation error < –19 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.6:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.7 | Transmitter constellation error < –22 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.17.8 | Transmitter constellation error < –25 dB (10 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.7.8:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18 | Transmitter constellation error (5 MHz channel spacing) | | | |
| OF4.18.1 | Transmitter constellation error < –5 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.1:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| OF4.18.2 | Transmitter constellation error < –8 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.2:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.3 | Transmitter constellation error < –10 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.3:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.4 | Transmitter constellation error < –13 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.4:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.5 | Transmitter constellation error < –16 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.5:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.6 | Transmitter constellation error < –19 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.6:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.7 | Transmitter constellation error < –22 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.7:M | Yes ☐ No ☐ N/A ☐ |
| OF4.18.8 | Transmitter constellation error < –25 dB (5 MHz channel spacing) | 17.3.9.6.3 | CF11& OF1.8.8:M | Yes ☐ No ☐ N/A ☐ |
| **OF5: PMD Receiver Specifications** | | | | |
| OF5.1 | Minimum input level sensitivity at packet error rate (PER) = 10% with 1000 octet frames | | | |
| OF5.1.1 | –82 dBm for 6 Mb/s | 17.3.10.1 | M | Yes ☐ No ☐ |
| OF5.1.2 | –81 dBm for 9 Mb/s | 17.3.10.1 | OF1.2.2:M | Yes ☐ No ☐ N/A ☐ |
| OF5.1.3 | –79 dBm for 12 Mb/s | 17.3.10.1 | M | Yes ☐ No ☐ |
| OF5.1.4 | –77 dBm for 18 Mb/s | 17.3.10.1 | OF1.2.4:M | Yes ☐ No ☐ N/A ☐ |
| OF5.1.5 | –74 dBm for 24 Mb/s | 17.3.10.1 | M | Yes ☐ No ☐ |
| OF5.1.6 | –70 dBm for 36 Mb/s | 17.3.10.1 | OF1.2.6:M | Yes ☐ No ☐ N/A ☐ |
| OF5.1.7 | –66 dBm for 48 Mb/s | 17.3.10.1 | OF1.2.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.1.8 | –65 dBm for 54 Mb/s | 17.3.10.1 | OF1.2.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.2 | Adjacent channel rejection | 17.3.10.2 | M | Yes ☐ No ☐ |
| OF5.3 | Nonadjacent channel rejection | 17.3.10.3 | M | Yes ☐ No ☐ |
| OF5.4 | Maximum input level | 17.3.10.4 | M | Yes ☐ No ☐ |
| OF5.5 | CCA sensitivity | 17.3.10.5 | M | Yes ☐ No ☐ |
| OF5.6 | Maximum input level sensitivity at packet error rate (PER) = 10% with 1000 octet frames (10 MHz channel spacing) | | | |
| OF5.6.1 | –85 dBm for 3 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.1:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.2 | –84 dBm for 4.5 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.2:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.3 | –82 dBm for 6 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.3:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.4 | –80 dBm for 9 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.4:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.5 | –77 dBm for 12 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.5:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.6 | –73 dBm for 18 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.6:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| OF5.6.7 | –69 dBm for 24 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.6.8 | –68 dBm for 27 Mb/s (10 MHz channel spacing) | 17.3.10.1 | CF11& OF1.7.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.7 | Adjacent channel rejection (10 MHz channel spacing) | 17.3.10.2 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.8 | Nonadjacent channel rejection (10 MHz channel spacing) | 17.3.10.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.9 | Maximum input level (10 MHz channel spacing) | 17.3.10.4 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.10 | CCA sensitivity (10 MHz channel spacing) | 17.3.10.5 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11 | Maximum input level sensitivity at packet error rate (PER) = 10% with 1000 octet frames (5 MHz channel spacing) | | | |
| OF5.11.1 | –85 dBm for 3 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.1:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.2 | –84 dBm for 4.5 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.2:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.3 | –82 dBm for 6 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.3:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.4 | –80 dBm for 9 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.4:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.5 | –77 dBm for 12 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.5:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.6 | –73 dBm for 18 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.6:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.7 | –69 dBm for 24 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF.1.8.7:M | Yes ☐ No ☐ N/A ☐ |
| OF5.11.8 | –68 dBm for 27 Mb/s (5 MHz channel spacing) | 17.3.10.1 | CF11& OF1.8.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.12 | Adjacent channel rejection (5 MHz channel spacing) | 17.3.10.2 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.13 | Nonadjacent channel rejection (5 MHz channel spacing) | 17.3.10.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.14 | Maximum input level (5 MHz channel spacing) | 17.3.10.4 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF5.15 | CCA sensitivity (5 MHz channel spacing) | 17.3.10.5 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| **OF6: Transmit PLCP** | | | | |
| OF6.1 | Transmit: transmit on MAC request | 17.3.11 | M | Yes ☐ No ☐ |
| OF6.2 | Transmit: format and data encoding | 17.3.11 | M | Yes ☐ No ☐ |
| OF6.3 | Transmit: timing | 17.3.11 | M | Yes ☐ No ☐ |
| **OF7: Receive PLCP** | | | | |
| OF7.1 | Receive: receive and data decoding | 17.3.12 | M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| **OF8: PLME** | | | | |
| OF8.1 | PLME: support PLME_SAP management primitives | 17.4.1 | M | Yes ☐ No ☐ |
| OF8.2 | PLME: support PHY MIB | 17.4.2 | M | Yes ☐ No ☐ |
| OF8.3 | PLME: support PHY characteristics | 17.4.3 | M | Yes ☐ No ☐ |
| OF8.4 | PLME:support PHY characteristics (dot11ChannelStartingFactor) | 17.4.2 | CF11:M | Yes ☐ No ☐ N/A ☐ |
| **OF9: OFDM PMD Sublayer** | | | | |
| OF9.1 | PMD: support PMD_SAP peer-to-peer service primitives | 17.5.4.1, 17.5.5.1, 17.5.5.2 | M | Yes ☐ No ☐ |
| OF9.2 | PMD: support PMD_SAP sublayer-to-sublayer service primitives | 17.5.4.2, 17.5.5.3, 17.5.5.4, 17.5.5.5, 17.5.5.6, 17.5.5.7 | M | Yes ☐ No ☐ |
| OF9.3 | PMD_SAP service primitive parameters | | | |
| OF9.3.1 | Parameter: TXD_UNIT | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.2 | Parameter: RXD_UNIT | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.3 | Parameter: TXPWR_LEVEL | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.4 | Parameter: RATE (12 Mb/s) | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.5 | Parameter: RATE (24 Mb/s) | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.6 | Parameter: RATE (48 Mb/s) | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.3.7 | Parameter: RATE (72 Mb/s) | 17.5.4.3 | O | Yes ☐ No ☐ |
| OF9.3.8 | Parameter: RSSI | 17.5.4.3 | M | Yes ☐ No ☐ |
| OF9.4 | | | | |
| OF9.4.1 | Parameter: RATE (6 Mb/s for 10 MHz channel spacing) | 17.5.4.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.2 | Parameter: RATE (12 Mb/s for 10 MHz channel spacing) | 17.5.4.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.3 | Parameter: RATE (24 Mb/s for 10 MHz channel spacing) | 17.5.4.3 | CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.4 | Parameter: RATE (36 Mb/s for 10 MHz channel spacing) | 17.5.4.3 | CF11& OF1.7:O | Yes ☐ No ☐ N/A ☐ |
| OF9.4.5 | Parameter: RATE (3 Mb/s for 5MHz channel spacing) | 17.5.4.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.6 | Parameter: RATE (6 Mb/s for 5 MHz channel spacing) | 17.5.4.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.7 | Parameter: RATE (12 Mb/s for 5 MHz channel spacing) | 17.5.4.3 | CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |
| OF9.4.8 | Parameter: RATE (18 Mb/s for 5 MHz channel spacing) | 17.5.4.3 | CF11& OF1.8:O | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.8 OFDM PHY functions *(continued)*

| Item | Feature | References | Status | Support |
|---|---|---|---|---|
| **OF10: Geographic Area Specific Requirements** | | | | |
| *OF10.1 | Geographic areas | 17.3.8.2, 17.3.8.3, 17.3.8.4, 17.3.9.3 | M | Yes ☐ No ☐ |
| OF10.2 | Regulatory domain extensions | 17.3.8.3.3, 17.3.8.4, 17.3.9.1, 17.3.9.2, Annex J | CF11:M | Yes ☐ No ☐ N/A ☐ |

### A.4.9 High Rate, direct sequence PHY functions

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| | Are the following PHY features supported? | | | |
| HRDS1 | Long preamble and header procedures | 18.2 | M | Yes ☐ No ☐ |
| HRDS1.1 | Long direct sequence preamble prepended on TX | 18.2.1 | M | Yes ☐ No ☐ |
| HRDS1.2 | Long PLCP integrity check generation | 18.2.3, 18.2.3.6 | M | Yes ☐ No ☐ |
| HRDS1.3 | TX rate change capability | 18.2.3.3 | M | Yes ☐ No ☐ |
| HRDS1.4 | Supported data rates | 18.1, 18.2.3.3 | M | Yes ☐ No ☐ |
| HRDS1.5 | Data scrambler | 18.2.4 | M | Yes ☐ No ☐ |
| HRDS1.6 | Scrambler initialization | 18.2.4 | M | Yes ☐ No ☐ |
| *HRDS2 | Channel Agility option | 18.3.2 | O | Yes ☐ No ☐ |
| *HRDS3 | Short preamble and header procedures | 18.2 | O | Yes ☐ No ☐ |
| HRDS3.1 | Short preamble prepended on TX | 18.2.2 | HRDS3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS3.2 | Short header transmission | 18.2.3.8, 18.2.3.9, 18.2.3.10, 18.2.3.11, 18.2.3.12, 18.2.3.13, 18.2.3.14 | HRDS3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS4 | Long preamble process on RX | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS4.1 | PLCP format | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS4.2 | PLCP integrity check verify | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS4.3 | RX Rate change capability | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS4.4 | Data whitener descrambler | 18.2.6 | M | Yes ☐ No ☐ |
| *HRDS5 | Short preamble process on RX | 18.2.6 | HRDS3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS5.1 | PLCP format | 18.2.6 | HRDS5:M | Yes ☐ No ☐ N/A ☐ |
| HRDS5.2 | PLCP integrity check verify | 18.2.6 | HRDS5:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.9 High Rate, direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| HRDS5.3 | RX rate change capability | 18.2.6 | HRDS5:M | Yes ☐ No ☐ N/A ☐ |
| HRDS5.4 | Data whitener descrambler | 18.2.6 | HRDS5:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6 | Operating channel capability | — | — | — |
| *HRDS6.1 | North America (FCC) | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.1 | Channel 1 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.2 | Channel 2 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.3 | Channel 3 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.4 | Channel 4 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.5 | Channel 5 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.6 | Channel 6 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.7 | Channel 7 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.8 | Channel 8 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.9 | Channel 9 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.10 | Channel 10 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.1.11 | Channel 11 | 18.4.6.2 | HRDS6.1:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.2 | Canada (IC) | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.1 | Channel 1 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.2 | Channel 2 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.3 | Channel 3 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.4 | Channel 4 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.5 | Channel 5 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.6 | Channel 6 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.7 | Channel 7 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.8 | Channel 8 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.9 High Rate, direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| HRDS6.2.9 | Channel 9 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.10 | Channel 10 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.2.11 | Channel 11 | 18.4.6.2 | HRDS6.2:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.3 | Europe (ETSI) | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.1 | Channel 1 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.2 | Channel 2 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.3 | Channel 3 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.4 | Channel 4 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.5 | Channel 5 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.6 | Channel 6 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.7 | Channel 7 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.8 | Channel 8 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.9 | Channel 9 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.10 | Channel 10 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.11 | Channel 11 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.12 | Channel 12 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.3.13 | Channel 13 | 18.4.6.2 | HRDS6.3:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.4 | France | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.4.1 | Channel 10 | 18.4.6.2 | HRDS6.4:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.4.2 | Channel 11 | 18.4.6.2 | HRDS6.4:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.4.3 | Channel 12 | 18.4.6.2 | HRDS6.4:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.4.4 | Channel 13 | 18.4.6.2 | HRDS6.4:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.5 | Spain | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.5.1 | Channel 10 | 18.4.6.2 | HRDS6.5:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.9 High Rate, direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| HRDS6.5.2 | Channel 11 | 18.4.6.2 | HRDS6.5:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.6 | Japan | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.1 | Channel 1 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.2 | Channel 2 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.3 | Channel 3 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.4 | Channel 4 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.5 | Channel 5 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.6 | Channel 6 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.7 | Channel 7 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.8 | Channel 8 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.9 | Channel 9 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.10 | Channel 10 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.11 | Channel 11 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.12 | Channel 12 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.13 | Channel 13 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.6.14 | Channel 14 | 18.4.6.2 | HRDS6.6:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS6.7 | China (Radio Administration The Radio Administration of P.R.China) | 18.4.6.2 | HRDS6:O.3 | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.1 | Channel 1 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.2 | Channel 2 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.3 | Channel 3 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.4 | Channel 4 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.5 | Channel 5 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.6 | Channel 6 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.7 | Channel 7 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.9 High Rate, direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| HRDS6.7.8 | Channel 8 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.9 | Channel 9 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.10 | Channel 10 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.11 | Channel 11 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.12 | Channel 12 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS6.7.13 | Channel 13 | 18.4.6.2 | HRDS6.7:M | Yes ☐ No ☐ N/A ☐ |
| HRDS7 | Hop sequences | | HRDS2:M | Yes ☐ No ☐ N/A ☐ |
| HRDS8 | Complementary code keying (CCK) bits to symbol mapping | | | |
| HRDS8.1 | 5.5 Mb/s | 18.4.6.5 | M | Yes ☐ No ☐ |
| HRDS8.2 | 11 Mb/s | 18.4.6.5 | M | Yes ☐ No ☐ |
| *HRDS9 | PBCC bits to symbol mappings | 18.4.6.6 | O | |
| HRDS9.1 | 5.5 Mb/s | 18.4.6.6 | HRDS9:M | Yes ☐ No ☐ |
| HRDS9.2 | 11 Mb/s | 18.4.6.6 | HRDS9:M | Yes ☐ No ☐ |
| *HRDS10 | CCA functionality | 18.4.8.4 | | |
| HRDS10.1 | CCA Mode 1, energy only (RSSI above threshold) | 18.4.8.4 | HRDS10:O.4 | Yes ☐ No ☐ |
| HRDS10.2 | CCA Mode 4, CS with timer | 18.4.8.4 | HRDS10:O.4 | Yes ☐ No ☐ |
| HRDS10.3 | CCA Mode 5, energy detect with High Rate CS | 18.4.8.4 | HRDS10:O.4 | Yes ☐ No ☐ |
| HRDS10.4 | Hold CCA busy for packet duration of a correctly received PLCP, but carrier lost during reception of MPDU. | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS10.5 | Hold CCA busy for packet duration of a correctly received, but out of spec, PLCP. | 18.2.6 | M | Yes ☐ No ☐ |
| HRDS11 | Transmit antenna selection | 18.4.5.8 | O | Yes ☐ No ☐ |
| HRDS12 | Receive antenna diversity | 18.4.5.8, 18.4.5.9 | O | Yes ☐ No ☐ |
| *HRDS13 | Antenna port(s) availability | 18.4.6.8 | O | Yes ☐ No ☐ |
| HRDS13.1 | If available (50 ¾ impedance) | 18.4.6.8 | HRDS13:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS14 | Transmit power level support | 18.4.5.9, 18.4.7.2 | O | Yes ☐ No ☐ |
| HRDS14.1 | If greater than 100 mW capability | 18.4.7.2 | HRDS14:M | Yes ☐ No ☐ N/A ☐ |
| *HRDS15 | Radio type (temperature range) | 18.4.6.14 | | |
| HRDS15.1 | Type 1 | 18.4.6.14 | HRDS15:O.5 | Yes ☐ No ☐ N/A ☐ |
| HRDS15.2 | Type 2 | 18.4.6.14 | HRDS15:O.5 | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.9 High Rate, direct sequence PHY functions *(continued)*

| Item | PHY feature | References | Status | Support |
|---|---|---|---|---|
| HRDS16 | Spurious emissions conformance | 18.4.6.8 | M | Yes ☐ No ☐ |
| HRDS17 | TX-to-RX turnaround time | 18.4.6.9 | M | Yes ☐ No ☐ |
| HRDS18 | RX-to-TX turnaround time | 18.4.6.10 | M | Yes ☐ No ☐ |
| HRDS19 | Slot time | 18.4.6.11 | M | Yes ☐ No ☐ |
| HRDS20 | ED reporting time | 18.4.6.10, 18.4.8.4 | M | Yes ☐ No ☐ |
| HRDS21 | Minimum transmit power level | 18.4.7.2 | M | Yes ☐ No ☐ |
| HRDS22 | Transmit spectral mask conformance | 18.4.7.3 | M | Yes ☐ No ☐ |
| HRDS23 | Transmitted center frequency tolerance | 18.4.7.4 | M | Yes ☐ No ☐ |
| HRDS24 | Chip clock frequency tolerance | 18.4.7.5 | M | Yes ☐ No ☐ |
| HRDS25 | Transmit power-on ramp | 18.4.7.6 | M | Yes ☐ No ☐ |
| HRDS26 | Transmit power-down ramp | 18.4.7.6 | M | Yes ☐ No ☐ |
| HRDS27 | RF carrier suppression | 18.4.7.7 | M | Yes ☐ No ☐ |
| HRDS28 | Transmit modulation accuracy | 18.4.7.8 | M | Yes ☐ No ☐ |
| HRDS29 | Receiver minimum input level sensitivity | 18.4.8.1 | M | Yes ☐ No ☐ |
| HRDS30 | Receiver maximum input level | 18.4.8.2 | M | Yes ☐ No ☐ |
| HRDS31 | Receiver adjacent channel rejection | 18.4.8.3 | M | Yes ☐ No ☐ |
| HRDS32 | MIB | 18.3.2, Annex C | M | Yes ☐ No ☐ |
| HRDS32.1 | PHY object class | 18.3.3 | M | Yes ☐ No ☐ |

### A.4.10 Regulatory Domain Extensions

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| MD1 | Country information element<br><br>Length<br>CountryString<br>FieldChannel Number<br>First Channel Number<br>Maximum Transmit Power Level<br>Number of Channels | 7.2.3.1,<br>7.2.3.9<br>7.3.2.9 | CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD2 | Inclusion of the Request information in the Probe Request frame | 7.2.3.8 | CF8:O | Yes ☐ No ☐ N/A ☐ |
| MD3 | Hopping Pattern Parameters<br><br>Element ID<br>Prime Radix<br>Number of Channels | 7.3.2.10 | (CF3 or (CF7 and HRDS2)) and CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD4 | Hopping Pattern information element<br><br>Format<br>Element ID<br>Random table method | 7.3.2.11 | (CF3 or (CF7 and HRDS2)) and CF8:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

### A.4.10 Regulatory Domain Extensions *(continued)*

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| MD5 | Request information element<br><br>Format<br>Element ID<br>Order of the Requested Elemented IDs | 7.3.2.12 | CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD6 | Entering a Regulatory Domain<br>Lost Connectivity with its extended service set (ESS)<br>Passive Scanning to learn Beacon information<br>Transmit Probe Request | 9.8.1 | CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD7 | Determination of the Hopping Patterns<br>[Hop Index Method without table, Hop Index Method with table, and hyperbolic congruence code (HCC)/extended HCC (EHCC)] | 7.3.2.11, 9.8.2.1 | (CF3 or (CF7 and HRDS2)) and CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD8 | Roaming requires Beacon frame with country information element | 11.1.3.3 | CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD9 | Actions to be taken upon the receipt of the Beacon frame | 11.1.3.4 | CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD10 | Ignore improperly formed Request information element | 7.2.3.9 | CF8:O | Yes ☐ No ☐ N/A ☐ |
| MD11 | Hopping Pattern set attribute | 14.8.2.20 | (CF3 or (CF7 and HRDS2)) and CF8:M | Yes ☐ No ☐ N/A ☐ |
| MD12 | Regulatory and Coverage classes | 7.3.2.12 | RC1:M | Yes ☐ No ☐ N/A ☐ |

### A.4.11 ERP functions

| Item | PHY features | References | Status | Support |
|---|---|---|---|---|
| *ERP1 | Transmit and Receive ERP-DSSS data rates 1 and 2 Mb/s and ERP-CCK data rates 5.5 and 11 Mb/s | 19.3.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP1.1 | Transmit and receive ERP-OFDM data rates of 6, 12, and 24 Mb/s | 19.3.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP1.2 | Transmit and receive ERP-OFDM data rate of 9 Mb/s | 19.3.2 | ERP1:O | Yes ☐ No ☐ N/A ☐ |
| ERP1.3 | Transmit and receive ERP-OFDM data rate of 18 Mb/s | 19.3.2 | ERP1:O | Yes ☐ No ☐ N/A ☐ |
| ERP1.4 | Transmit and receive ERP-OFDM data rate of 36 Mb/s | 19.3.2 | ERP1:O | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

## A.4.11 ERP functions *(continued)*

| Item | PHY features | References | Status | Support |
|---|---|---|---|---|
| ERP1.5 | Transmit and receive ERP-OFDM data rate of 48 Mb/s | 19.3.2 | ERP1:O | Yes ☐ No ☐ N/A ☐ |
| ERP1.6 | Transmit and receive ERP-OFDM data rate of 54 Mb/s | 19.3.2 | ERP1:O | Yes ☐ No ☐ N/A ☐ |
| *ERP2 | Transmit and receive ERP-PBCC data rate of 22 Mb/s | 19.3.2 | CF9&HRDS9.1&HRDS9.2:O | Yes ☐ No ☐ N/A ☐ |
| ERP2.1 | Transmit and receive ERP-PBCC data rate of 33 Mb/s | 19.3.2 | ERP2:O | Yes ☐ No ☐ N/A ☐ |
| *ERP3 | Transmit and receive DSSS-OFDM data at same rates as ERP-OFDM | 19.3.2 | CF9:O | Yes ☐ No ☐ N/A ☐ |
| ERP4 | Support of ERP3 required PPDU formats as described in reference | 19.3.2 | CF9:O | Yes ☐ No ☐ N/A ☐ |
| ERP5 | Able to transmit and receive long and short DSSS as well as OFDM preambles | 19.3.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP6 | Set SERVICE field bits for DSSS-OFDM, ERP-PBCC, locked clocks, and length extension (b0, b2, b3, b5, b6, and b7) | 19.3.2.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP7 | Set b1 & b4 of long and short preamble PPDU SERVICE field to 0 | 19.3.2.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP8 | b2 shall be set to 1 in all long and short preamble PPDU SERVICE fields | 19.3.2.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP9 | Set bits b5, b6, and b7 of the long and short preamble PPDU SERVICE fields as described in the reference | 19.3.2.1, 19.3.2.1.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP10 | Use Clause 15 or Clause 18 rates when using protection mechanisms | 9.13 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP11 | SIGNAL field set to 3 Mb/s in all long and short DSSS-OFDM PPDU formats as described in the reference | 19.3.2.4 | ERP3:M | Yes ☐ No ☐ N/A ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.11 ERP functions *(continued)*

| Item | PHY features | References | Status | Support |
|---|---|---|---|---|
| ERP12 | Calculate DSSS-OFDM length with signal extension | 19.3.2.4.1 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP13 | Set ERP-PBCC encoder in state 0 at beginning of PPDU | 19.3.3.2 | ERP2:M | Yes ☐ No ☐ N/A ☐ |
| ERP14 | Set phase of ERP-PBCC relative to header | 19.3.3.2 | ERP2:M | Yes ☐ No ☐ N/A ☐ |
| ERP15 | Use same pulse shape for 22 and 33 Mb/s | 19.3.3.2 | ERP2:M | Yes ☐ No ☐ N/A ☐ |
| ERP16 | Add signal extension of 6 μs | 19.3.2.3 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP17 | Simultaneous CCA on long preamble Barker, short preamble Barker, and OFDM | 19.3.5 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP18 | CCA with energy detect above threshold and CS | 19.3.5 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP19 | Decode as DSSS-OFDM if signal field indicates 3 Mb/s | 19.3.6 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP20 | Able to automatically detect format of long preamble Barker, short preamble Barker, and OFDM and receive appropriately | 19.3.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP21 | Comply with local regulatory frequency allocation requirements | 19.4.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP22 | Use frequency plan for 2.4 GHz | 19.4.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP23 | Comply with regulatory spurious emissions regulations | 19.4.3 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP24 | Slot time requirements | 19.4.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP25 | Implement Short Slot Time option | 19.4.4 | CF9:O | Yes ☐ No ☐ N/A ☐ |
| ERP26 | Use 10 μs short interframe space (SIFS) time | 19.4.6 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP27 | Comply with regulatory transmit power requirements | 19.4.7.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP28 | ± 25 PPM frequency tolerance | 19.4.7.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP29 | Use locked clocks | 19.4.7.2, 19.4.7.3 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP30 | Tolerate input level of –20 dBm | 19.5.3 | CF9:M | Yes ☐ No ☐ N/A ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.11 ERP functions *(continued)*

| Item | PHY features | References | Status | Support |
|---|---|---|---|---|
| ERP31 | Use specified transmit mask | 19.5.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP32 | Meet sensitivity for all supported data rates | 19.5.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP33 | Reject adjacent channels as in Table 17-13 in 17.3.10.1 or in 18.4.8.3 as appropriate | 19.5.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP34 | Coherent transition of ERP-DSSS to OFDM | 19.7.2, 19.7.2.7 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP35 | Same signal shaping of ERP-DSSS and OFDM | 19.7.2.1 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP36 | Transmit power equal for ERP-DSSS and OFDM segments | 19.7.2.2 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP37 | Align transition time | 19.7.2.3 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP38 | Set transition phase to 45 degrees | 19.7.2.3 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP39 | Calculate ERP-OFDM TXTIME | 19.8.3.1 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP40 | Calculate ERP-PBCC TXTIME | 19.8.3.2 | ERP2:M | Yes ☐ No ☐ N/A ☐ |
| ERP41 | Calculate DSSS-OFDM TXTIME | 19.8.3.3 | ERP3:M | Yes ☐ No ☐ N/A ☐ |
| ERP42 | Revert to 20 ms slot time when establishing association with a long slot time STA | 7.3.1.4 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP43 | Support TXVECTOR and RXVECTOR as described in reference | 9.2 | CF9:M | Yes ☐ No ☐ N/A ☐ |
| ERP44 | Terminate single carrier segment smoothly | 19.7.2.4 | ERP3:M | Yes ☐ No ☐ N/A ☐ |

### A.4.12 Spectrum management extensions

| Item | IUT configuration | References | Status | Support |
|---|---|---|---|---|
| SM1 | Country, Power Constraint, and transmit power control (TPC) Report elements included in Beacon and Probe Response frames | 7.2.3.1, 7.2.3.9, 7.3.2.9, 7.3.2.13, 7.3.2.16 | CF10: M | Yes ☐ No ☐ |
| SM2 | Spectrum Management Capability bit | 7.3.1.4 | CF10:M | Yes ☐ No ☐ |
| SM3 | Power Capability and Supported Channels elements in Association and Reassociation frames | 7.2.3.4, 7.2.3.5, 11.6.1 | CF10:M | Yes ☐ No ☐ |


Copyright © 2007 IEEE. All rights reserved.

## A.4.12 Spectrum management extensions *(continued)*

| Item | IUT configuration | References | Status | Support |
|---|---|---|---|---|
| SM4 | Action frame protocol for spectrum management actions | 7.3.1.11, 7.4 | CF10:M | Yes ☐ No ☐ |
| | Measurement Request frame | 7.4.1.1 | CF10:M | Yes ☐ No ☐ |
| | Measurement Report frame | 7.4.1.2 | CF10:M | Yes ☐ No ☐ |
| | TPC Request frame | 7.4.1.3 | CF10:M | Yes ☐ No ☐ |
| | TPC Report frame | 7.4.1.4 | CF10:M | Yes ☐ No ☐ |
| | Channel Switch Announcement frame | 7.4.1.5 | CF10:M | Yes ☐ No ☐ |
| SM5 | Measurement requests | | | |
| | Basic request | 7.3.2.21.1 | CF10:M | Yes ☐ No ☐ |
| | CCA request | 7.3.2.21.2 | CF10:O | Yes ☐ No ☐ N/A ☐ |
| | Receive power indication (RPI) histogram | 7.3.2.21.3 | CF10:O | Yes ☐ No ☐ N/A ☐ |
| | Enabling/disabling requests and reports | 7.3.2.21 | CF10:M | Yes ☐ No ☐ |
| SM6 | Measurement reports | | | |
| | Basic report | 7.3.2.22.1 | CF10:M | Yes ☐ No ☐ |
| | CCA report | 7.3.2.22.2 | CF10:O | Yes ☐ No ☐ N/A ☐ |
| | RPI histogram report | 7.3.2.22.3 | CF10:O | Yes ☐ No ☐ N/A ☐ |
| | Refusal to measure | 7.3.2.22 | CF10:M | Yes ☐ No ☐ |
| SM7 | Quiet interval | | | |
| | AP-defined Quiet interval | 7.2.3.1, 7.2.3.9, 7.3.2.23, 11.6.2 | (CF1 and CF10):M | Yes ☐ No ☐ |
| | STA-defined Quiet interval | 7.2.3.1, 7.2.3.9, 7.3.2.23, 11.6.2 | (CF2 and CF10):M | Yes ☐ No ☐ |
| | STA support for Quiet interval | 7.2.3.1, 7.2.3.9, 7.3.2.23, 11.6.2 | CF10:M | Yes ☐ No ☐ |
| SM8 | Association control based on spectrum management capability | 11.5, 11.6 | (CF1 and CF10):M | Yes ☐ No ☐ |
| SM9 | Association control based on transmit power capability | 11.8.1 | (CF1 and CF10):M | Yes ☐ No ☐ |
| SM10 | Maximum transmit power levels | | | |
| | AP determination and communication of local maximum transmit power level | 11.8.2 | (CF1 and CF10):M | Yes ☐ No ☐ |
| | STA determination and communication of local maximum transmit power level | 11.8.2 | (CF2 and CF10):M | Yes ☐ No ☐ |
| SM11 | Selection of transmit power | 11.8.3 | CF10:M | Yes ☐ No ☐ |
| SM12 | Adaptation of transmit power | | | |
| | TPC report in Beacon and Probe Response frames | 11.8.4 | CF10:M | Yes ☐ No ☐ |
| | Dynamic transmit power adaptation | 11.8.4 | CF10:O | Yes ☐ No ☐ N/A ☐ |
| SM13 | Testing channels for radars | 11.9.3 | CF10:M | Yes ☐ No ☐ |
| SM14 | Detecting and discontinuing operations after detection of a radar | 11.9.4 | CF10:M | Yes ☐ No ☐ |
| SM15 | Requesting and reporting of measurements | 11.9.6 | CF10:M | Yes ☐ No ☐ |
| SM16 | Autonomous reporting of radars | 11.9.6 | CF10:M | Yes ☐ No ☐ |

Copyright © 2007 IEEE. All rights reserved.

### A.4.12 Spectrum management extensions *(continued)*

| Item | IUT configuration | References | Status | Support |
|---|---|---|---|---|
| SM17 | IBSS dynamic frequency selection (DFS) element including channel map | 7.3.2.24 | (CF2 and CF10):M | Yes ☐ No ☐ |
| SM18 | DFS owner function | 11.9.7 | (CF2 and CF10):M | Yes ☐ No ☐ |
| SM19 | DFS owner recovery procedure | 11.9.7 | (CF2 and CF10):M | Yes ☐ No ☐ |
| SM20 | Channel switch procedure | | | |
| | Transmission of channel switch announcement and channel switch procedure by an AP | 11.9.7 | (CF1 and CF10):M | Yes ☐ No ☐ |
| | Transmission of channel switch announcement and channel switch procedure by a STA | 11.9.7 | (CF2 and CF10):M | Yes ☐ No ☐ |
| | Reception of channel switch announcement and channel switch procedure by a STA | 11.9.7 | CF10:M | Yes ☐ No ☐ |

### A.4.13 Regulatory classes extensions

| Item | Protocol capability | References | Status | Support |
|---|---|---|---|---|
| RC1 | Regulatory and coverage classes | 7.3.2.9 | CF8&CF11:M | Yes ☐ No ☐ N/A ☐ |
| RC2 | Regulatory and coverage classes (20 MHz channel spacing) | 7.3.2.9, 17.3.8.6 | CF8&CF11:M | Yes ☐ No ☐ N/A ☐ |
| RC3 | Regulatory and coverage classes (10 MHz channel spacing) | 7.3.2.9, 17.3.8.6 | CF8&CF11& OF1.7:M | Yes ☐ No ☐ N/A ☐ |
| RC4 | Regulatory and coverage classes (5 MHz channel spacing) | 7.3.2.9, 17.3.8.6 | CF8&CF11& OF1.8:M | Yes ☐ No ☐ N/A ☐ |

### 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, 11.2.1 | 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 ☐ |


Copyright © 2007 IEEE. All rights reserved.

### 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 ☐ |
| 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, 7.3.2.15, 7.3.2.16, 7.4.2.1–7.4.2.3, 11.4 | (CF1 & CF12):O<br>(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 ☐ |

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex B

(informative)

# Hopping sequences

The following tables pertain to the hopping sequences for China, North America and ETSI.

Copyright © 2007 IEEE. All rights reserved.

**Table B.1—Hopping sequence set 1**

| index | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 2 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 3 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 4 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |
| 5 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 6 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 7 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 8 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |
| 9 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 10 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 11 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 12 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 13 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 14 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 15 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |
| 16 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 17 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 18 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 19 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 20 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 21 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 22 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 23 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 24 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 25 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 26 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 27 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 28 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 29 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 30 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 31 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 32 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 33 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |
| 34 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 35 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 36 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 37 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 38 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 39 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.1—Hopping sequence set 1 *(continued)***

| index | 0 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 41 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |
| 42 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 43 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 44 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 45 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 46 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 47 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 48 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 49 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 50 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 51 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 52 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 53 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 54 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 55 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |
| 56 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 57 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 58 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 59 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 60 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 61 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |
| 62 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 63 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 64 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 65 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |
| 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 67 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 68 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 69 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 70 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 71 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 72 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 73 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |
| 74 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 75 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 76 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |
| 77 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 78 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 79 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |

Copyright © 2007 IEEE. All rights reserved.

**Table B.1—Hopping sequence set 1 *(continued)***

| index | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 2 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 3 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 4 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |
| 5 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 6 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 7 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 8 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |
| 9 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 10 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 11 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 12 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 13 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 14 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 15 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |
| 16 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 17 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 18 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 19 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 20 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 21 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 22 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 23 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 24 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 25 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |
| 26 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 27 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 28 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 29 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 30 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 31 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 32 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 33 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |
| 34 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 35 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 36 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 37 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 38 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 39 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.1—Hopping sequence set 1 *(continued)***

| index | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 41 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |
| 42 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 43 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 44 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 45 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 46 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |
| 48 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 49 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 50 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 51 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 52 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 53 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 54 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 55 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |
| 56 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 57 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 58 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 59 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 60 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 61 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |
| 62 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 63 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 64 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 65 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 66 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 67 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 68 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 69 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 70 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 71 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 72 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 73 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |
| 74 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 75 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 76 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 77 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 78 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 79 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |

Copyright © 2007 IEEE. All rights reserved.

**Table B.2—Hopping sequence set 2**

| index | 1 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 2 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 3 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 4 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 5 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 6 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 7 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 8 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |
| 9 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 10 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 11 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 12 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 13 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 14 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 15 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |
| 16 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 17 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 18 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 19 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 20 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 21 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 22 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |
| 23 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 24 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 25 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 26 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 27 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 28 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 29 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 30 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 31 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 32 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 33 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |
| 34 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 35 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 36 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 37 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 38 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 39 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.2—Hopping sequence set 2 *(continued)***

| index | 1 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 41 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 42 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 43 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 44 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 45 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 46 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 47 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 48 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 49 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 50 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |
| 51 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 52 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 53 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 54 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 55 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |
| 56 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 57 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 58 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |
| 59 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 60 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 61 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 62 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 63 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |
| 64 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 65 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |
| 66 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 67 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 68 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 69 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 70 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 71 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 72 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 73 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 74 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 75 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 76 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |
| 77 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 78 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 79 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |

Copyright © 2007 IEEE. All rights reserved.

**Table B.2—Hopping sequence set 2 *(continued)***

| index | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 2 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 3 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 4 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |
| 5 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 6 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 7 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 8 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |
| 9 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 10 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 11 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 12 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 13 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 14 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 15 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |
| 16 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 17 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 18 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 19 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 20 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 21 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 22 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |
| 23 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 24 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 25 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 26 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 27 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 28 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 29 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 30 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 31 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 32 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 33 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |
| 34 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 36 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 37 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 38 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 39 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.2—Hopping sequence set 2 *(continued)***

| index | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 41 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 42 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 43 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 44 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 45 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 46 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 47 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 48 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 49 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 50 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |
| 51 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 52 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 53 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 54 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 55 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |
| 56 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 57 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 58 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 59 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 60 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 61 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 62 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 63 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 64 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 65 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |
| 66 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 67 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 68 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 69 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 70 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 71 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 72 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 73 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |
| 74 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 75 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 76 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |
| 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 78 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 79 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |

Copyright © 2007 IEEE. All rights reserved.

**Table B.3—Hopping sequence set 3**

| index | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 2 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 3 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 4 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 5 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 6 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 7 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 8 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 9 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 10 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 11 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 12 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 13 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 14 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 15 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 16 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 17 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 18 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 19 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 20 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 21 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 22 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |
| 23 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 24 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 25 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 26 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 27 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 28 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 29 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 30 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |
| 31 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 32 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 33 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 34 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 35 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 36 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |
| 37 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 38 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 39 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.3—Hopping sequence set 3 *(continued)***

| index | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 41 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 42 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 43 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 44 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 45 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 46 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 47 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 48 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 49 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 50 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |
| 51 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 52 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 53 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 54 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 55 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 56 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 57 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 58 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |
| 59 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 60 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 61 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 62 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 63 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |
| 64 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 65 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |
| 66 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 67 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |
| 68 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 69 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 70 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 71 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 72 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 73 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 74 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 75 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |
| 76 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |
| 77 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 78 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 79 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |

Copyright © 2007 IEEE. All rights reserved.

**Table B.3—Hopping sequence set 3 *(continued)***

| index | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 |
| 2 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 |
| 3 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 |
| 4 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 |
| 5 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 |
| 6 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 |
| 7 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 |
| 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 |
| 9 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 |
| 10 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 |
| 11 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 |
| 12 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 |
| 13 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 |
| 14 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 |
| 15 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 |
| 16 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 |
| 17 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 |
| 18 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
| 19 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 |
| 20 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 |
| 21 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 |
| 22 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 |
| 23 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| 24 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 |
| 25 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 |
| 26 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 |
| 27 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 |
| 28 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 |
| 29 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 |
| 30 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 |
| 31 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 |
| 32 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 |
| 33 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 |
| 34 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 |
| 35 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 |
| 36 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 |
| 37 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 | 60 |
| 38 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 39 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 |


Copyright © 2007 IEEE. All rights reserved.

**Table B.3—Hopping sequence set 3 *(continued)***

| index | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 |
| 41 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 |
| 42 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 | 54 | 57 |
| 43 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 |
| 44 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 |
| 45 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 |
| 46 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 |
| 47 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 |
| 48 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 |
| 49 | 42 | 45 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 |
| 50 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 |
| 51 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 |
| 52 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 |
| 53 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 |
| 54 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 |
| 55 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 |
| 56 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 |
| 57 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 |
| 58 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 |
| 59 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 | 56 |
| 60 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 |
| 61 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 |
| 62 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 |
| 63 | 48 | 51 | 54 | 57 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 |
| 64 | 60 | 63 | 66 | 69 | 72 | 75 | 78 | 2 | 5 | 8 | 11 | 14 | 17 |
| 65 | 49 | 52 | 55 | 58 | 61 | 64 | 67 | 70 | 73 | 76 | 79 | 3 | 6 |
| 66 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 | 58 | 61 | 64 | 67 |
| 67 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 |
| 68 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 |
| 69 | 44 | 47 | 50 | 53 | 56 | 59 | 62 | 65 | 68 | 71 | 74 | 77 | 80 |
| 70 | 71 | 74 | 77 | 80 | 4 | 7 | 10 | 13 | 16 | 19 | 22 | 25 | 28 |
| 71 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 | 49 | 52 | 55 |
| 72 | 78 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 |
| 73 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 | 47 | 50 | 53 |
| 74 | 67 | 70 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 |
| 75 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | 41 | 44 |
| 76 | 15 | 18 | 21 | 24 | 27 | 30 | 33 | 36 | 39 | 42 | 45 | 48 | 51 |
| 77 | 2 | 5 | 8 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 |
| 78 | 73 | 76 | 79 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 79 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 34 | 37 | 40 | 43 | 46 |

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex C

(informative)

This clause is no longer maintained and may not be compatible with or describe all features of this standard.

## Formal description of a subset of MAC operation

This annex contains formal descriptions of the behavior of a subset of MAC STA and AP entities. These descriptions also describe the frame formats and the generation and interpretation of information encoded in MAC frames, in the parameters of service primitives supported by the MAC, and in MIB attributes used or generated by the MAC. The MAC is described using the 1992 version of the ITU Specification and Description Language (SDL-92). SDL-92 is defined in ITU-T Recommendation Z.100 (03/93). An update to ITU-T Recommendation Z.100 was approved in 1996 (SDL-96), but none of the SDL facilities used in this annex were modified. An introduction to the MAC formal description is provided in C.1. Definitions of the data types and operators used by the MAC state machines are provided in C.2. An SDL system describing MAC operation at an IEEE 802.11 STA is contained in C.3. Finally, a subset of an SDL system describing the aspects of MAC operation at an IEEE 802.11 AP that differ from operation at a non-AP STA is provided in C.4.

In Annex D, the MAC and PHY MIBs are described in Abstract Syntax Notation One (ASN.1), defined in ISO/IEC 8824-1:1995, ISO/IEC 8824-2:1995, ISO/IEC 8824-3:1995, ISO/IEC 8824-4:1995, ISO/IEC 8825-1:1995, and ISO/IEC 8825-2:1996. ITU-T Recommendation Z.105 (03/95) defines the use of SDL in conjunction with ASN.1, allowing system behavior to be defined using SDL and data types to be defined using ASN.1. Incomplete tool support precluded the use of ITU-T Recommendation Z.105 in this annex. However, within the limits of ITU-T Recommendation Z.100 (referred to subsequently as Z.100), the data types in C.2 are defined in a similar manner to ITU-T Recommendation Z.105 (referred to subsequently as Z.105). Annex P contains a listing of available documentation.

NOTE 1—The SDL definitions in this annex should be usable with any SDL tool that supports the 1993 version or 1996 update of ITU-T Recommendation Z.100. Software for generating, analyzing, verifying, and simulating SDL system descriptions is available from several sources.

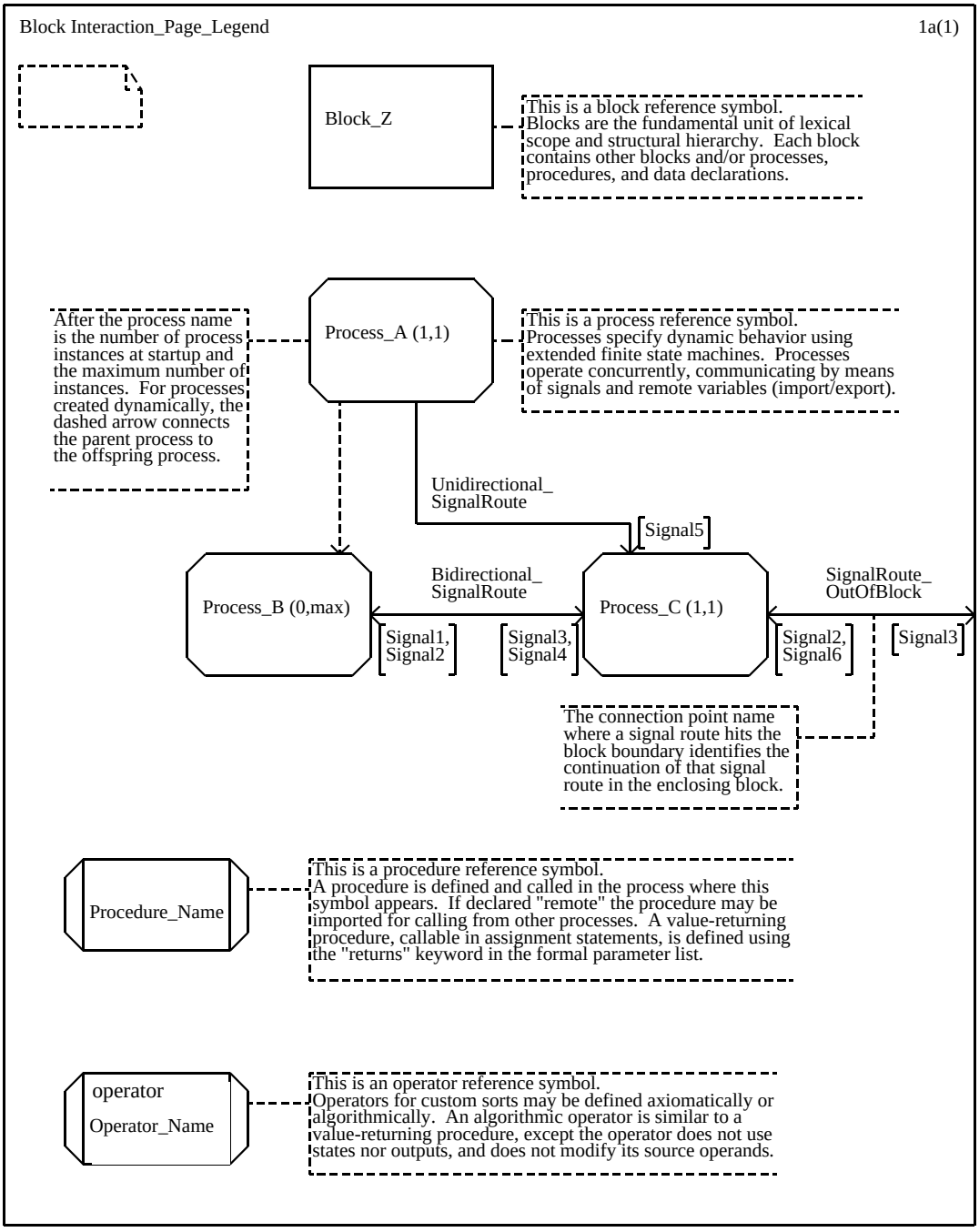
NOTE 2—The SDL code in this annex was generated using *SDT/PC version 3.02*; from Telelogic AB, Malmo, Sweden (+46-40-174700; internet: telelogic.se); U.S. office in Princeton, NJ (+1-609-520-1935; internet: telelogic.com). Telelogic offers SDT for several workstation platforms in addition to SDT/PC.

NOTE 3— The use of Telelogic's product to prepare this annex does not constitute an endorsement of SDT by the IEEE LAN MAN Standards Committee or by the IEEE.

NOTE 4—The diagrams on the next two pages show most of the symbols of SDL graphical syntax (SDL-GR) used in the MAC formal description. The symbols in these diagrams have labels and comments that explain their meanings. These diagrams are intended to serve as a legend for the SDL-GR symbols that comprise most of the process interaction and state transition diagrams. These diagrams are neither a complete SDL system, nor a complete presentation of SDL-GR symbology. Also, this state machine fragment exists to illustrate the SDL graphical syntax and does not describe any useful behavior.

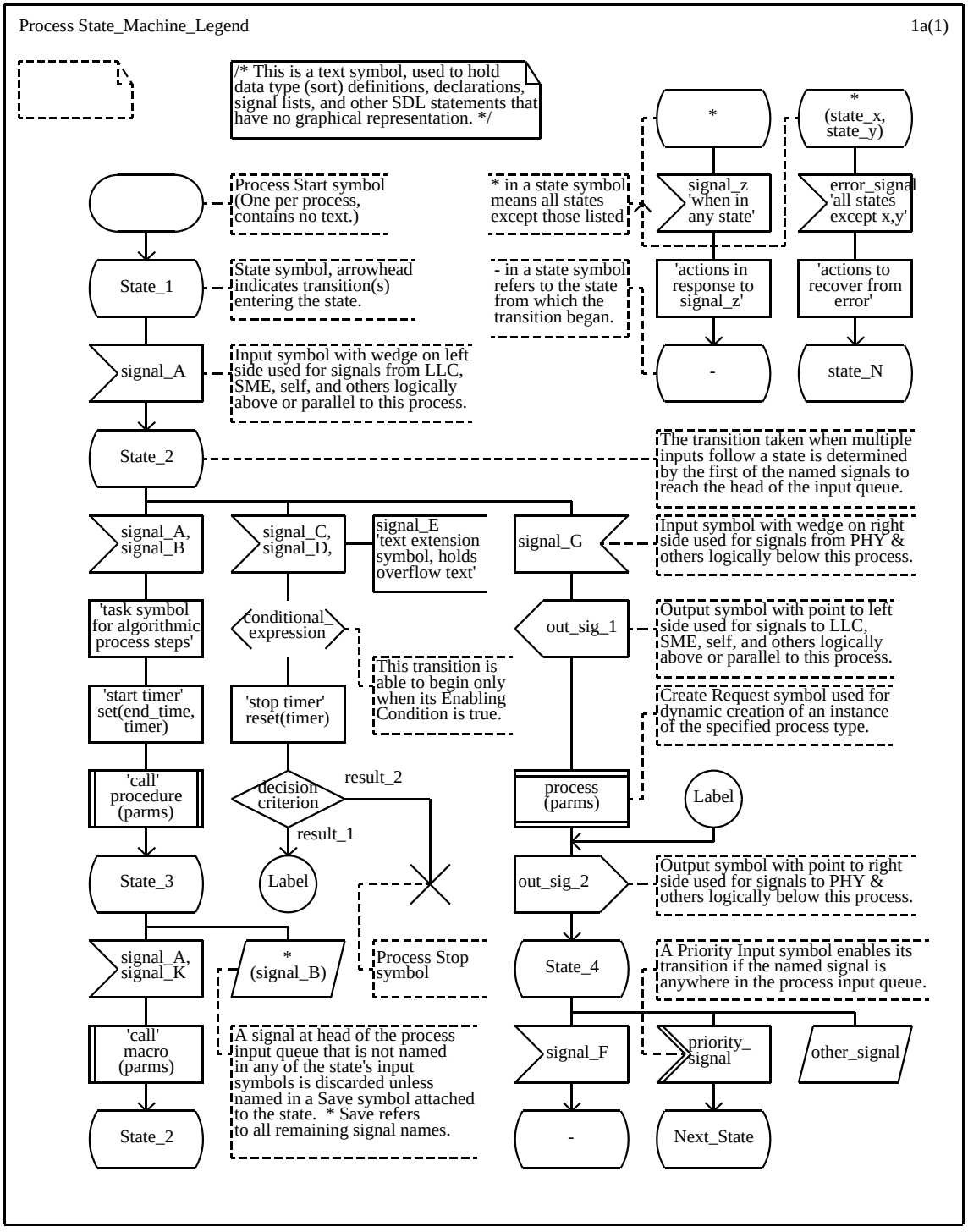
Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 831 of 1232





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

## C.1 Introduction to the MAC formal description

This formal description defines the behavior of IEEE 802.11 MAC entities. The MAC protocol functional decomposition used herein facilitates explicit description of the reference points and durations of the various timed intervals; the bases for generation and/or validation of header fields, service parameters, and MIB attributes; and the interpretation of each value in cases where enumerated data types are used in service parameters.

### C.1.1 Fundamental assumptions

The MAC protocol is described as an SDL system, which is a set of extended finite state machines. Each state machine is a set of independent processes, all of which operate concurrently. All variable data-holding entities and procedures exist solely within the context of a single process. In SDL all interprocess communication is done with signals (there are no global variables). Signals may be sent and received explicitly, using SDL's output and input symbols, or implicitly, using SDL's export/import mechanism (only if the variables or procedures are declared "remote"). By default, signals incur delays when traversing channels between blocks; however, only nondelaying channels and signal routes are used in the MAC state machines, and all remote variables and procedures are declared with the "nodelay" property.

State transitions, procedure calls, and tasks (assignment statements and other algorithmic processing steps) are assumed to require zero time. This permits the time intervals that are part of the normative MAC behavior to be defined explicitly, using SDL timers. One unit of system time (a 1.0 change in the value of "now") is assumed to represent 1 μs of real time. Usec (microsecond) and TU data types are defined, with operators to convert Usec and TU values to SDL time or duration when necessary.

The SDL system boundary encloses the MAC entities. The LLC, SME, PHY, and DS are part of the environment. SDL generally assumes that entities in the environment operate as specified; however, the MAC state machines that communicate with the various SAPs attempt to validate inputs from the environment, and to handle cases where a pair of communicating entities, one within the system and the other outside the system boundary, have different local views of the medium, STA, or service state. All STAs in an IEEE 802.11 service set are assumed to exhibit the behaviors described herein. Nevertheless, because of the open nature of the WM, the MAC state machines check for error cases that can arise only when an entity on the WM is transmitting IEEE 802.11 protocol data units (PDUs), but is not obeying the communication protocols specified by this standard.

### C.1.2 Notation conventions

When practical, names used in the clauses of this standard are spelled identically in this annex. The principal exceptions are those names that conflict with one of SDL's reserved words (such as power management mode "active," which is renamed "sta_active" in SDL). To help fit the SDL text into the graphic symbols, acronyms with multiple, sequential capital letters are written with only the first letter capitalized (e.g., "MSDU" is written "Msdu" and "MLMEJoin.request" is written "MlmeJoin.request").

SDL reserved words and the names of variables and synonyms (named constants) begin with lowercase letters. The names of sorts (data types), signals, signal routes, channels, blocks, and processes begin with uppercase letters. The names of certain groups of variables and/or synonyms begin with a particular lowercase letter, followed by the remainder of the name, beginning with an uppercase letter. These groups are

"aNameOfAttribute"PHY operational parameters.
"cNameOfCapability"Capability bits, also used for internal values exported as MIB counters.
"dNameOfDuration"Duration (relative time) values, declared as Usec, TU, or Duration.
"dot11NameOfAttribute"MIB attributes.


Copyright © 2007 IEEE. All rights reserved.

"eNameOfElement"Element ID values.

"mNameOfVariable"Remote variables used for intra-MAC communication, but not part of the MIB. Most of these variables are exported from the MLME block.

"sNameOfStaticValue"Synonyms for static data values used within the MAC.

"tNameOfTime"Time (absolute time) values, declared as Usec, TU, or Time. The names of timers begin with "T."

### C.1.3 Modeling techniques

State machines are grouped according to defined function sets that are visible, directly or indirectly, at an exposed interface. The emphasis in the organization of the state machines is explicitly to show initiation of and response to events at the exposed interfaces, and time-related actions, including those dependent on the absence of external events (e.g., response timeouts) and intervals measured in derived units (e.g., backoff "time" in units of slots during which the WM is idle). The operations associated with the various state transitions emphasize communication functions. Most of the details regarding insertion, extraction, and encoding of information in fields of the PDUs is encapsulated with the definitions of those fields. This approach, which relies heavily on SDL's abstract data type and inheritance mechanisms, permits the behavior of the data-holding entities to be precisely defined, without obscuring process flow by adding in-line complexity to the individual state transitions.

The modeling of PDUs and service data units (SDUs) requires sorts such as octet strings, and operators such as bitwise boolean functions, which are not predefined in SDL. These sorts and operators are defined in Package macsorts, which appears in C.2.

PDU and SDU sorts are based on the **Bit** sort. Bit is a subtype of SDL's predefined Boolean sort. As a result, Bit literals 0 and 1 are alternative names for "false" and "true" and have no numeric significance. To use 0 or 1 as integer values requires a conversion operation. Items of the **Bitstring** sort are 0-origin, variable-length strings of Bits. With Bitstring operands, operators "and," "or," "xor," and "not" operate bitwise, with the length of the result equal to the length of the longest (or only) source string. The **Octet** sort is a subtype of Bitstring that adds conversion operators to and from Integer. Each item of the Octet sort has length=8 {by usage convention in Z.100, enforced in Z.105}. Items of the **Octetstring** sort are 0-origin, variable-length strings of Octets. The **Frame** sort is a subtype of Octetstring that adds operators to extract and to modify all MAC header fields and most other MAC frame fields and elements. Most MAC fields and elements that contain named values with specific value assignments or enumerations are defined as subtypes of Frame, Octetstring, or Bitstring with the names added as literals or synonyms, so that the state machines can refer to the names without introducing ambiguity about the value encodings.

Where communication at a SAP or between processes is strictly first in first out (FIFO), the (implicit) input queue of the SDL processes is used. When more sophisticated queue management is needed, a queue whose entries are instances of one, specified sort is created using the **Queue** generator. Entries on Queue sorts may be added and removed at either the tail or the head, and the number of queue entries may be determined. The contents of a Queue may also be searched to locate entries with particular parameter values.

In C.2 is an SDL-92 Package (a named collection of SDL definitions that can be included by reference into an SDL System specification), which is a formal description of the formats and data encodings used in IEEE 802.11 SDUs, PDUs, and the parameters of the service primitives used at each of the SAPs supported by the IEEE 802.11 MAC. This package also contains definitions for some data structures and operators used internally by one or more of the MAC state machines.

The behaviors of many intra-MAC operators are part of the normative description of the MAC protocol because results of the specified operations are visible, directly or indirectly, at exposed interfaces. For example, custom operators are used to define the generation of the CRC-32 value used in the FCS field (operator crc32, page 424), the calculation of frame transmission time used as part of the value in the

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 835 of 1232

Duration/ID field in certain types of frames (operator calcDur, page 440), the comparison of the values of particular fields of a received MAC header with cached data values as part of the procedure for detecting duplicate frames (operator searchTupleCache, page 412), and numerous other aspects of frame formats and information encoding. On the other hand, data structures used solely for intra-MAC storage or for transferring of information between different state machines of a single STA or AP, are only normative to the extent that they define items of internal state and the temporal sequence necessary for proper operation of the MAC protocol. The specific structures and encodings used for internal data storage and communication functions in this formal description do *not* constrain MAC implementations, provided those implementations exhibit the specified behaviors at the defined SAPs and, in conjunction with an appropriate PHY, on the WM.

## C.2 Data type and operator definitions for the MAC state machines

This clause is in SDL/PR (phrase notation), with the exception of procedural operators, which are defined in SDL/GR (graphic notation). Package macsorts contains the definitions of the sorts (data types with associated operators and literals) and synonyms (named constants) used by the MAC state machines. Package macmib defines data types for attributes in the MAC MIB, and portions of the PHY MIB, accessed by the MAC state machines. Package macmib exists solely to satisfy SDL's strong type checking in the absence of an SDL tool that fully supports Z.105 (the combined use of SDL with ASN.1).


Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3101_d\MacEnum(31)



```
/*    PACKAGE MACSORTS    */
/* This package contains definitions of the custom sorts (data types), operators,
  literals, and synonyms (named constants) used by the MAC state machines. */
```

```
/*************************************************************************
 *       Enumerated types used within the MAC state machines
 *************************************************************************/
newtype ChangeType    /* type of change due at the next boundary */
  literals  dwell,    /* dwell (only with FH PHY) */
            mocp;     /* medium occupancy (only with PCF) */
endnewtype ChangeType;
newtype Imed    /* priority for queuing MMPDUs, relative to MSDUs */
  literals  head,    /* place MMPDU at head of transmit queue */
            norm;    /* place MMPDU at tail of transmit queue */
endnewtype Imed;
newtype NavSrc    /* source of duration in SetNav & ClearNav signals */
  literals  rts,    /* RTS frame */
            cfpBss,  cfendBss,    /* start/end of CFP in own BSS */
            cfpOther,  cfendOther,    /* start/end of CFP in other BSS */
            cswitch,    /* channel switch */
            misc,    /* durId from other frame types */
            nosrc;    /* nonreception events */
endnewtype NavSrc;
newtype PsMode    /* power save mode of a station (PsResponse signal) */
  literals  sta_active,  power_save,  unknown;   endnewtype PsMode;
newtype PsState    /* power save state of this station */
  literals  awake,  doze;   endnewtype PsState;
newtype StateErr    /* requests disasoc or deauth (MmIndicate signal) */
  literals  noerr,  class2,  class3;   endnewtype StateErr;
newtype StationState    /* asoc/auth state of sta (SsResponse signal) */
  literals  not_auth,  auth_open,  auth_key,  asoc,  dis_asoc;
endnewtype StationState;
newtype TxResult    /* transmission attempt status (PduConfirm signal) */
  literals  successful,  partial,  retryLimit,  txLifetime,
  atimAck,  atimNak;   endnewtype TxResult;
```

```
/*************************************************************************
 *       Enumerated types used in PHY service primitives
 *************************************************************************/
newtype CcaStatus    /* <state> parameter of PhyCca.indication */
  literals  busy,  idle;   endnewtype CcaStatus;
newtype PhyRxStat    /* <rxerror> parameter of PhyRxEnd.indication */
  literals  no_error,  fmt_violation,  carrier_lost,  unsupt_rate;
endnewtype PhyRxStat;
```

```
/*************************************************************************
 *       Placeholders for Mlme/Plme Get/Set Parameter Values
 *************************************************************************/
   /* MibAtrib (placeholder in MlmeGet/Set definitions) */
syntype MibAtrib = Charstring   endsyntype MibAtrib;
   /* MibValue (placeholder in MlmeGet/Set definitions) */
syntype MibValue = Integer   endsyntype MibValue;
```

Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3102_d\LmeEnum(31)



```
/*************************************************************************
 *       Enumerated types used in Mac and Mlme service primitives
 *************************************************************************/
newtype AuthType    /* <authentication type> parm in Mlme primitives */
  inherits Octetstring  operators all;
  adding  literals open_system,  shared_key;
  axioms  open_system == mkOS(0, 2);  shared_key == mkOS(1, 2);
endnewtype AuthType;
newtype AuthTypeSet  powerset( AuthType);   endnewtype AuthTypeSet;
newtype BssType    /* <BSS type> parameter & BSS description element */
  literals infrastructure,  independent,  any_bss;   endnewtype BssType;
newtype BssTypeSet  powerset( BssType);   endnewtype BssTypeSet;
newtype CfPriority    /* <priority> parameter of various requests */
  literals  contention,  contentionFree;   endnewtype CfPriority;
newtype MibStatus    /* <status> parm of Mlme/Plme Get/Set.confirm */
  literals  success,   invalid,   write_only,   read_only;
endnewtype MibStatus;
newtype MlmeStatus    /* <status> parm of Mlme operation confirm */
  literals  success,  invalid,  timeout,  refused,
          tomany_req,  already_bss;   endnewtype MlmeStatus;
newtype PwrSave    /* <power save mode> parameter of MlmePowerMgt */
  literals  sta_active,  power_save;   endnewtype PwrSave;
newtype Routing    /* <routing info> parameter for MAC data service */
  literals  null_rt;   endnewtype Routing;
newtype RxStatus    /* <reception status> parm of MaUnitdata indication */
  literals  rx_success,  rx_failure;   endnewtype RxStatus;
newtype ScanType    /* <scan type> parameter of MlmeScan.request */
  literals  active_scan,  passive_scan;   endnewtype ScanType;
newtype ServiceClass    /* <service class> parameter for MaUnitdata */
  literals  reorderable,  strictlyOrdered;   endnewtype ServiceClass;
newtype TxStatus    /* <transmission status> parm of MaUnitdataStatus */
  literals  successful,   retryLimit,   txLifetime,   noBss,
     excessiveDataLength,  nonNullSourceRouting,
     unsupportedPriority,  unavailablePriority,
     unsupportedServiceClass,  unavailableServiceClass,
     unavailableKeyMapping;  endnewtype TxStatus;
```


Copyright © 2007 IEEE. All rights reserved.



Package macsorts

3103_e\IntraMac(31)



```
/*************************************************************************
 *       Intra-MAC remote variables (names of form mXYZ)
 *************************************************************************/
remote mActingAsAp Boolean nodelay;    /* =true if STA started BSS */
remote mAId  AsocId nodelay;    /* AID assigned to STA by AP */
remote mAssoc  Boolean nodelay;    /* =true if STA associated w/BSS */
remote mAtimW  Boolean nodelay;    /* =true if ATIM window in prog */
remote mBkIP  Boolean nodelay;    /* =true if backoff in prog */
remote mBrates Ratestring nodelay;    /* basic rate set for this sta */
remote mBssId  MacAddr nodelay;    /* identifier of current (I)BSS */
remote mCap  Octetstring nodelay;    /* capability info from MlmeJoin */
remote mCfp  Boolean nodelay;    /* =true if CF period in progress */
remote mDisable  Boolean nodelay;    /* =true if not in any BSS; then */
     /* TX only sends probe_req; RX only accepts beacon, probe_rsp */
remote mDtimCount  Integer nodelay;    /* =0 at Tbtt of Beacon with DTIM */
remote mFxIP  Boolean nodelay;    /* =true during frame exchange seq */
remote mIbss  Boolean nodelay;    /* =true if STA is member of IBSS */
remote mListenInt  Integer nodelay;    /* beacons between wake up @TBTT */
remote mNavEnd  Time nodelay;    /* NAV end Time, <=now when idle */
remote mNextBdry  Time nodelay;    /* next boundary Time; =0 if none */
remote mNextTbtt  Time nodelay;    /* Time next beacon due to occur */
remote mPcAvail  Boolean nodelay;    /* =true if point coord in BSS */
remote mPcDlvr  Boolean nodelay;    /* =true if CF delivery only */
remote mPcPoll  Boolean nodelay;    /* =true if CF delivery & polling */
remote mPdly  Usec nodelay;    /* probe delay from start or join */
remote mPss  PsState nodelay;    /* power save state of STA */
remote mReceiveDTIMs Boolean nodelay;  /* =true if DTIMs received */
remote mRxA  Boolean nodelay;    /* =true if RX indicated by PHY */
remote mSsId  Octetstring nodelay;    /* name of the current (I)BSS */
remote procedure TSF nodelay;    /* read & update 64-bit TSF timer */
   fpar Integer, Boolean;   returns Integer;
```

Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3104_d\StaticData(31)



```
/*************************************************************************
 *       Named static data values    (names of form sXYZ)
 *************************************************************************/
synonym sMaxMsduLng Integer = 2304;    /* max octets in an MSDU */
synonym sMacHdrLng  Integer = 24;    /* octets in data header, no WEP */
synonym sWepHdrLng  Integer = 28;    /* octets in data header with WEP */
synonym sWepAddLng  Integer = 8;    /* octets added for WEP */
synonym sWdsAddLng  Integer = 6;    /* octets added for WDS (addr4) */
synonym sCrcLng  Integer = 4;    /* octets for crc32 (FCS, ICV) */
synonym sMaxMpduLng  Integer =        /* max octets in an MPDU */
   (sMaxMsduLng + sMacHdrLng + sWdsAddLng + sWepAddLng + sCrcLng);
syntype FrameIndexRange = Integer    /* index range for octets in MPDU */
   constants 0 : sMaxMpduLng   endsyntype FrameIndexRange;
synonym sTsOctet  Integer = 24;    /* first octet of Timestamp field */
synonym sMinFragLng Integer = 256;    /* min value for aMpduMaxLength */
synonym sMaxFragNum Integer =        /* maximum fragment number */
   (sMaxMsduLng / (sMinFragLng - sMacHdrLng - sCrcLng));
synonym sAckCtsLng Integer = 112;  /* bits in ACK and CTS frames */
```

```
/*************************************************************************
 *       Station configuration flags (static, supplementary to MIB)
 *************************************************************************/
synonym sVersion  Integer = 0;    /* supported Protocol Version */
synonym sCanBeAp  Boolean = false;    /* =true if STA can operate as AP */
synonym sCanBePc  Boolean = false;    /* =true if AP can be Point Coord */
synonym sCfPollable Boolean =true;    /* =true if responds to CF-polls */
```


Copyright © 2007 IEEE. All rights reserved.



Package macsorts

3105_d\Usec_TU(31)



```
/*************************************************************************
 *       Discrete microsecond and Time Unit sorts
 *************************************************************************/
/* SDL does not define the relationship between its concept */
/* of Time and physical time in the system being described. */
/* An abstraction is needed to establish this relationship, */
/* because Time in SDL uses the semantics of Real, whereas */
/* time in the MAC protocol is discrete, with the semantics */
/* of Natural and a step size (resolution) of 1 micosecond. */
/* Most MAC times are defined using the subtypes of Integer */
/* Usec and TU.  These have operators for explicit conversion */
/* to SDL Time (tUsec, tTU), SDL Duration (dUsec, dTU), and */
/* from SDL Time (uTime, tuTime) as needed to comply with SDL's */
/* strong type checking.  Where the MAC state machines need to */
/* access the contents of the TSF timer, SDL's 'now' (current */
/* time) is used.  This yields readable time-dependent code, */
/* but the value of 'now' cannot be modified by an SDL program, */
/* so adopting the TSF time from timestamps in received Beacons */
/* or Probe Responses is shown as an informal task symbol. */
/* Microsecond sort -- also has operators tmin and tmax */
newtype Usec   inherits Integer   operators all;
 adding  operators
   dUsec : Usec -> Duration;
   tUsec : Usec -> Time;
   uTime : Time -> Usec;
   tmax  : Usec, Usec -> Usec;
   tmin  : Usec, Usec -> Usec;
 axioms    for all u, w in Usec(
     u >= w ==> tmax(u, w) == u;      u < w ==> tmax(u, w) == w;
     u >= w ==> tmin(u, w) == w;      u < w ==> tmin(u, w) == u;
     for all t in Time(     for all r in Real(
         r = float(u) ==> tUsec(u) == Time!(Duration!(r));
         t = Time!(Duration!(r)) and u = fix(r) ==> u == uTime(t);));
     for all d in Duration(     for all r in Real(
         r = float(u) ==> dUsec(u) == Duration!(r); )));
 constants >= 0 /* constrain value range to be non-negative */
endnewtype Usec;
/* Time Unit sort -- (1 * TU) = (1024 * Usec) */
newtype TU  inherits Integer  operators all;
 adding  operators
   dTU    : TU -> Duration;
   tTU    : TU -> Time;
   tuTime : Time -> TU;
   u2TU   : Usec -> TU;
   tu2U   : TU -> Usec;
 axioms    for all k in TU(   for all t in Time(   for all r in Real(
         r = float(k) ==> tTU(k) == Time!(Duration!(1024 * r));
         t = Time!(Duration!(r)) and k = (fix(r) / 1024) ==> k == tuTime(t);));
     for all d in Duration(   for all r in Real(
         r = float(k) ==> dTU(k) == Duration!(1024 * r);));
     for all u in Usec(   u2TU(u) == u / 1024;   tu2U(k) == k * 1024; ));
 constants >= 0 /* constrain value range to be non-negative */
endnewtype TU;
```

Copyright © 2007 IEEE. All rights reserved.



Package macsorts

3106_d\String0(31)



```
/*****************************************************************
 *       Generator for 0-origin String sorts (adapted from Z.105, Annex A)
 *****************************************************************/
/* String0(sort, nullSymbol) can define strings of any sort. */
/* These strings are indexed starting from 0 rather than 1. */
/* Sorts defined by String0 have the normal String operators, plus */
/* Tail (all but first item), Head (all but last item), and */
/* aggregators S2, S3, S4, S6, S8 (make fixed length strings). */
generator String0(type Item, literal Emptystring)
  literals Emptystring;
  operators
    MkString : Item -> String0;    /* make a string from an item */
    Length  : String0 -> Integer;    /* length of string */
    First  : String0 -> Item;    /* first item in string */
    Tail  : String0 -> String0;    /* all but first item in string */
    Last  : String0 -> Item;    /* last item in string */
    head  : String0 -> String0;    /* all but last item in string */
    "//"  : String0, String0 -> String0;    /* concatenation */
    Extract! : String0, Integer -> Item;    /* get item from string */
    Modify!  : String0, Integer, Item -> String0;  /* modify string */
    SubStr  : String0, Integer, Integer -> String0;
       /* SubStr(s,i,j) is string0 of length j starting at string0(i) */
    S2 : Item, Item -> String0;     S3 : Item, Item, Item -> String0;
    S4 : Item, Item, Item, Item -> String0;
    S6 : Item, Item, Item, Item, Item, Item -> String0;
    S8 : Item, Item, Item, Item, Item, Item, Item, Item -> String0;
 /* axioms  continued on next page... */

endgenerator String0;
```



Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3107_a\String0(31)



```
/* String0 axioms  */
/*   for all item0,item1,item2,item3,item4,item5,item6,item7 in Item(
     for all s, s1, S2, S3 in String0(     for all i, j in Integer(
     constructors are Emptystring, MkString, and "//";
      equalities between constructor terms
        s // Emptystring == s;          Emptystring // s == s;
        (s1 // S2) // S3 == s1 // (S2 // S3);
      definition of Length by applying it to all constructors
         type String Length(Emptystring) == 0;
         type String Length(MkString(item0)) == 1;
         type String Length(s1 // S2) == Length(s1) + Length(S2);
      definition of Extract! by applying it to all constructors,
        Extract!(MkString(item0), 0) == item0;
        i < Length(s1) ==> Extract!(s1 // S2, i) == Extract!(s1, i);
        i >= Length(s1) ==> Extract!(s1 // S2, i) == Extract!(S2, i - Length(s1));
        i < 0 or i >= Length(s) ==> Extract!(s, i) == error!;
      definition of First and Last by other operations
        First(s) == Extract!(s, 0);
        Last(s) == Extract!(s, Length(s) - 1);
      definition of substr(s,i,j) by induction on j,
        i >= 0 and i <= Length(s) ==> SubStr(s, i, 0) == Emptystring;
        i >= 0 and j > 0 and i + j <= Length(s) ==> SubStr(s, i, j) ==
                SubStr(s, i, j - 1) // MkString(Extract!(s, i + j - 1));
        i < 0 or j < 0 or i + j > Length(s) ==> SubStr(s, i, j) == error!;
      definition of Modify!, Head, Tail, Sx by other operations
        Modify!(s, i, item0) == SubStr(s, 0, i) // MkString(item0) //
           SubStr(s, i + 1, Length(s) - i - 1);
        head(s) == SubStr(s, 0, Length(s) - 1);
        Tail(s) == SubStr(s, 1, Length(s) - 1);
        S2(item0, item1) == MkString(item0) // MkString(item1);
        S3(item0, item1, item2) ==
          MkString(item0) // MkString(item1) // MkString(item2);
        S4(item0, item1, item2, item3) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3);
        S6(item0, item1, item2, item3, item4, item5) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3) // MkString(item4) // MkString(item5);
        S8(item0, item1, item2, item3, item4, item5, item6, item7) ==
          MkString(item0) // MkString(item1) // MkString(item2) //
            MkString(item3) // MkString(item4) // MkString(item5) //
            MkString(item6) // MkString(item7); )));
*/
```

Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3108_d\Bitstring(31)



```
/*****************************************************************
 *       ASN.1-style BIT sort    (from Z.105, Annex A)
 *****************************************************************/
   /* Bit is a subtype of Boolean -- bit values 0 and 1 are
   /* not numerals and cannot be used with Integer operators */
newtype Bit   inherits Boolean
   literals  0 = false,  1 = true;   operators all;   endnewtype Bit;
```

```
/*****************************************************************
 *       ASN.1-style BIT STRING sort    (adapted from Z.105, Annex A)
 *****************************************************************/
/* Bitstrings are 0-origin strings of Bit.  Z.105 uses ASN.1-style */
/* literals in binary ('1011'B) or hexadecimal ('D3'H), but this */
/* syntax is not accepted for Z.100 string literals.  Therefore, */
/* this version provides only hexadecimal literals 0x00-0xFF. */
/* Bitstring operators '=>', 'not', 'and', 'or', and 'xor' act */
/* bitwise, with the length of the result string equal to the */
/* length of the longest (or only) source string. */
newtype Bitstring String0(Bit, '')
 adding  literals     macro Hex_Literals;
 operators
   "not" : Bitstring -> Bitstring;
   "and" : Bitstring, Bitstring -> Bitstring;
   "or"  : Bitstring, Bitstring -> Bitstring;
   "xor" : Bitstring, Bitstring -> Bitstring;
   "=>"  : Bitstring, Bitstring -> Bitstring;     noequality;
 axioms     macro Hex_Axioms;
   for all s, s1, S2, S3 in Bitstring(
     s = s == true;      s1 = S2 == S2 = s1;
     s1 /= S2 == not (s1 = S2);      s1 = S2 == true ==> s1 == S2;
     ((s1 = S2) and (S2 = S3)) ==> s1 = S3 == true;
     ((s1 = S2) and (S2 /= S3)) ==> s1 = S3 == false;
     for all b, b1, b2 in Bit(
       not (') == ';
       not (MkString(b) // s) == MkString(not (b)) // not (s);
       " and " == ";
       Length(s) > 0 ==> " and s == MkString(0) and s;
       Length(s) > 0 ==> s and " == s and MkString(0);
       (MkString(b1) // s1) and (MkString(b2) // S2) ==
         MkString(b1 and b2) // (s1 and S2);
       s1 or S2 == not (not s1 and not S2);
       s1 xor S2 == (s1 or S2) and not (s1 and S2);
       s1 => S2 == not (not s1 and S2);));
 map  for all b1, b2 in Bitstring literals(
    for all bs1, bs2 in Charstring literals(
/* connection to the String generator */
      for all b in Bit literals(
         spelling(b1) = "" // bs1 // bs2 // "",
        spelling(b2) = "" // bs2 // "",  spelling(b) = bs1
          ==> b1 == MkString(b) // b2; )));
endnewtype Bitstring;
```


Copyright © 2007 IEEE. All rights reserved.



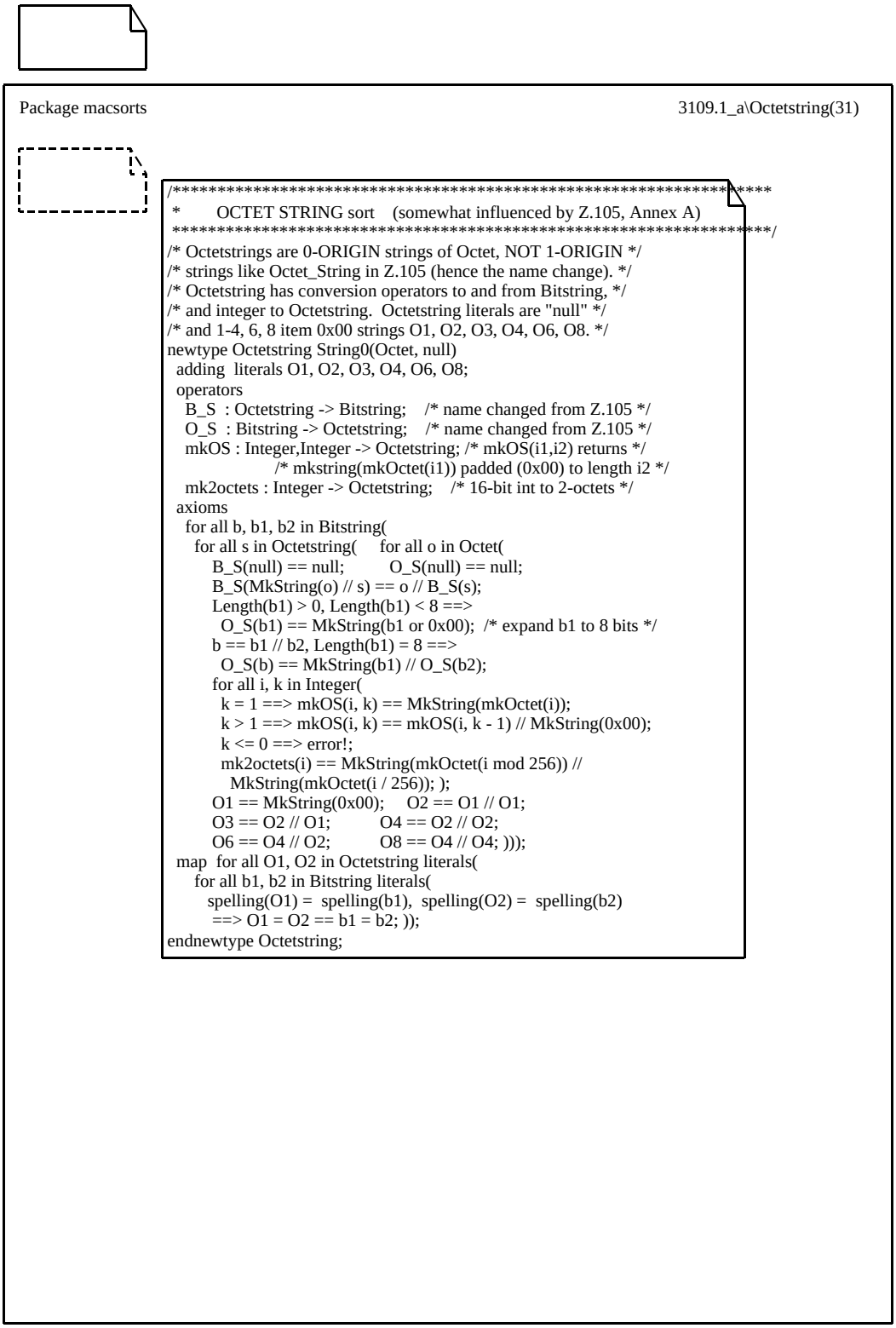
Package macsorts

3109_d\Octetstring(31)



```
/*************************************************************************
 *       OCTET sort    (influenced by Z.105, Annex A)
 *************************************************************************/
/* Octet is a subtype of Bitstring where length always =8. */
/* Z.105 adds a "size" keyword to SDL and defines Octet with */
/* "... constants size (8) ..." to impose this length constraint. */
/* Here Octet relies on proper use maintain lengths as multiples */
/* of 8.  Proper length strings are created by the hexadecimal */
/* Bitstring literals (e.g. 0xD5) and operator mkOctet: */
/*    o:= mkOctet(i)    converts a non-negative Integer (mod 256) */
/*                          to an Octet (exactly 8 bits) */
/*    i:= octetVal(o)    converts an Octet to an Integer (0:255) */
/*    o:= flip(o)    reverses bit order of the Octet */
/*                          (0<-->7, 1<-->6, 2<-->5, 3<-->4) */
newtype Octet   inherits Bitstring   operators all;
  adding  operators
    mkOctet  : Integer -> Octet;
    octetVal : Octet -> Integer;
    flip     : Octet -> Octet;
  axioms
    for all i in Integer(     for all z in Octet(
        i = 0 ==> mkOctet(i) == S8(0, 0, 0, 0, 0, 0, 0, 0);
        i = 1 ==> mkOctet(i) == S8(1, 0, 0, 0, 0, 0, 0, 0);
        i > 1 and i <= 255 ==> mkOctet(i) ==
            SubStr((First(mkOctet(i mod 2)) // mkOctet(i / 2)), 0, 8);
        i > 255 ==> mkOctet(i) == mkOctet(i mod 256);
        i < 0 ==> mkOctet(i) == error!;
        z = MkString(0) ==> octetVal(z) == 0;
        z = MkString(1) ==> octetVal(z) == 1;
        Length(z) > 1 and Length(z) <= 8 ==>
            octetVal(z) == octetVal(First(z)) +
                (2 * (octetVal(SubStr(z, 1, Length(z) - 1))));
        Length(z) > 8 ==> octetVal(z) == error!;
        flip(z) == S8(z(7),z(6),z(5),z(4),z(3),z(2),z(1),z(0)); ));
endnewtype Octet;
```

Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3110_d\MacAddr(31)



```
/*********************************************************************
 *       MAC Address sorts
 *********************************************************************/
/* MacAddr is a subtype of Octetstring with added operators: */
/*    isGroup(m) =true if given a group address */
/*    isBcst(m)  =true if given the broadcast address */
/*    isLocal(m) =true if given a locally-administered address */
/*    adrOs(m)   converts MacAddr to Octetstring */
/* MAC addresses must be defined to be exactly 6 octets long, */
/* typically using the S6 operator or nullAddr synonym. */
newtype MacAddr   inherits Octetstring   operators all;
 adding  operators
   isGroup : MacAddr -> Boolean;
   isBcst  : MacAddr -> Boolean;
   isLocal : MacAddr -> Boolean;
   adrOs   : MacAddr -> Octetstring;
 axioms
   for all m in MacAddr(
     (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x01) ==> isGroup(m) == true;
     (Length(m) = 6) and ((Extract!(m,0) and 0x01) = 0x00) ==> isGroup(m) == false;
     (Length(m) = 6) and (m = S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == true;
     (Length(m) = 6) and (m /= S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF)) ==> isBcst == false;
     (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x02) ==> isLocal == true;
     (Length(m) = 6) and ((Extract!(m,0) and 0x02) = 0x00) ==> isLocal == false;
     Length(m) /= 6 ==> error! /* common error! term */;
     for all o in Octetstring(m = MacAddr!(o) == adrOs(m) = o; ));
endnewtype MacAddr;
newtype MacAddrSet   powerset( MacAddr)   endnewtype MacAddrSet;
synonym bcstAddr  MacAddr =    /* Broadcast Address */
          <<type MacAddr>>  S6(0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
synonym nullAddr  MacAddr =    /* Null Address */
          << type MacAddr>>  S6(0x00,0x00,0x00,0x00,0x00,0x00);
```

```
/*********************************************************************
 *       BSS description sorts
 *********************************************************************/
   /* BssDscr is used with MlmeScan.confirm and MlmeJoin.request */
newtype BssDscr  struct
   bdBssId    MacAddr;
   bdSsId    Octetstring;    /* 1 <= length <= 32 */
   bdType    BssType;
   bdBcnPer    TU;    /* beacon period in Time Units */
   bdDtimPer    Integer;    /* DTIM period in beacon periods */
   bdTstamp    Octetstring;    /* 8 Octets from ProbeRsp/Beacon */
   bdStartTs    Octetstring;    /* 8 Octets TSF when rx Tstamp */
   bdPhyParms    PhyParms;    /* empty if not needed by PHY */
   bdCfParms    CfParms;    /* empty if not CfPollable/no PCF */
   bdIbssParms    IbssParms;    /* empty if infrastructure BSS */
   bdCap    Capability;    /* capability information */
   bdBrates    Ratestring;    /* BSS basic rate set */
endnewtype BssDscr;
newtype BssDscrSet   powerset( BssDscr)   endnewtype BssDscrSet;
```

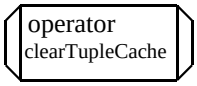
Copyright © 2007 IEEE. All rights reserved.

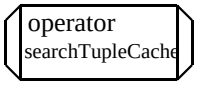


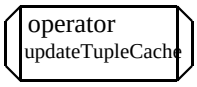
Package macsorts 3111_d\TupleCache(31)



```
/*************************************************************************
 *       Duplicate filtering support sorts
 *************************************************************************/
syntype FragNum = Integer    /* Range of possible fragment numbers */
       constants 0:sMaxFragNum  endsyntype FragNum;
syntype SeqNum = Integer    /* Range of possible sequence numbers */
       constants 0:4095  endsyntype SeqNum;
newtype Tuple  struct    /* for duplicate filtering & defragmentation */
   full    Boolean;    /* =true if Tuple contains valid info */
   ta    MacAddr;    /* transmitting station address (Addr2) */
   sn    SeqNum;    /* Msdu/Mmpdu sequence number */
   fn    FragNum;    /* most recent Mpdu fragment number */
   tRx    Time;    /* reception time (endRx of fragment) */
 default (. false, nullAddr, 0, 0, 0 .);
endnewtype Tuple;
```

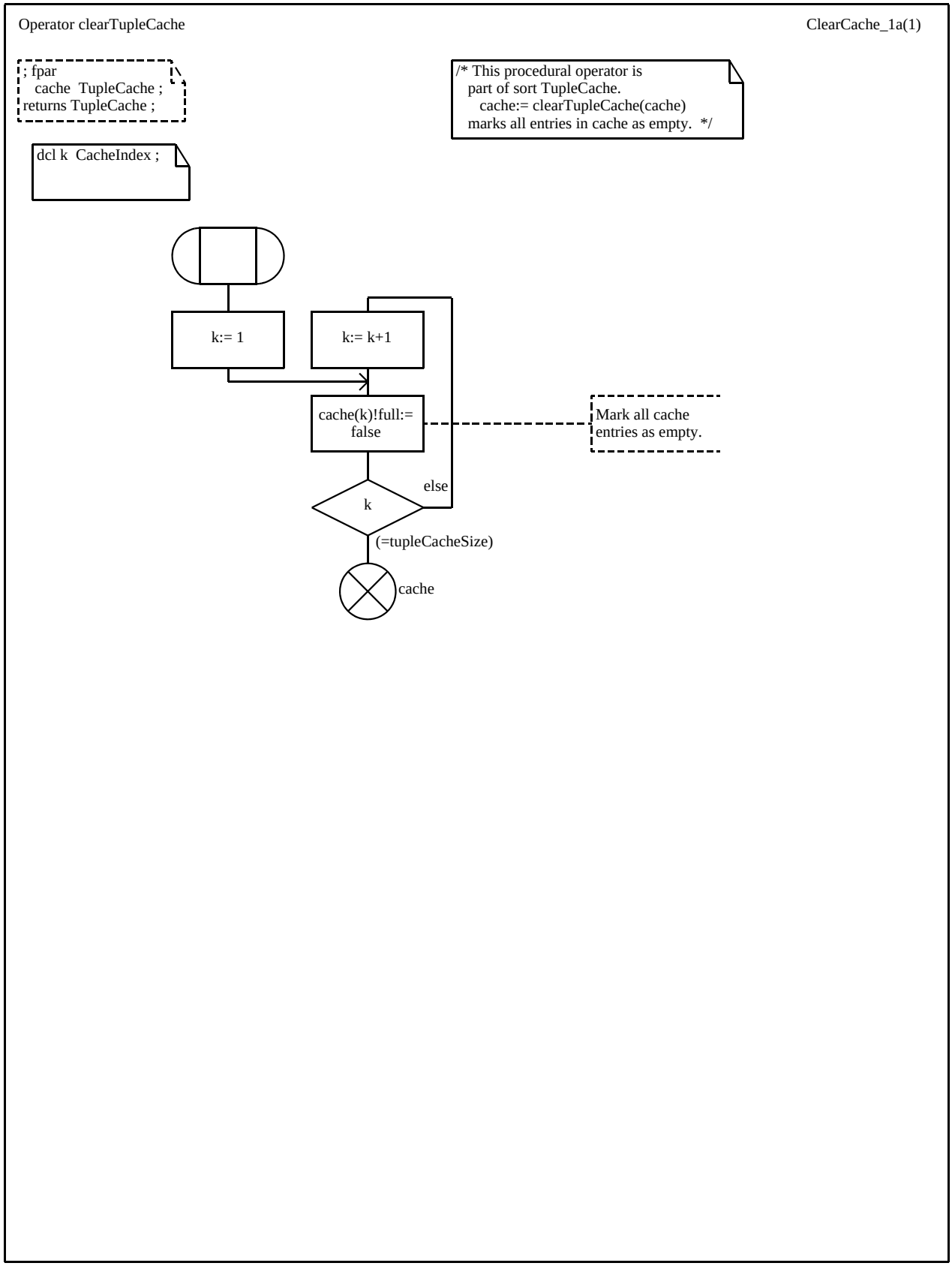


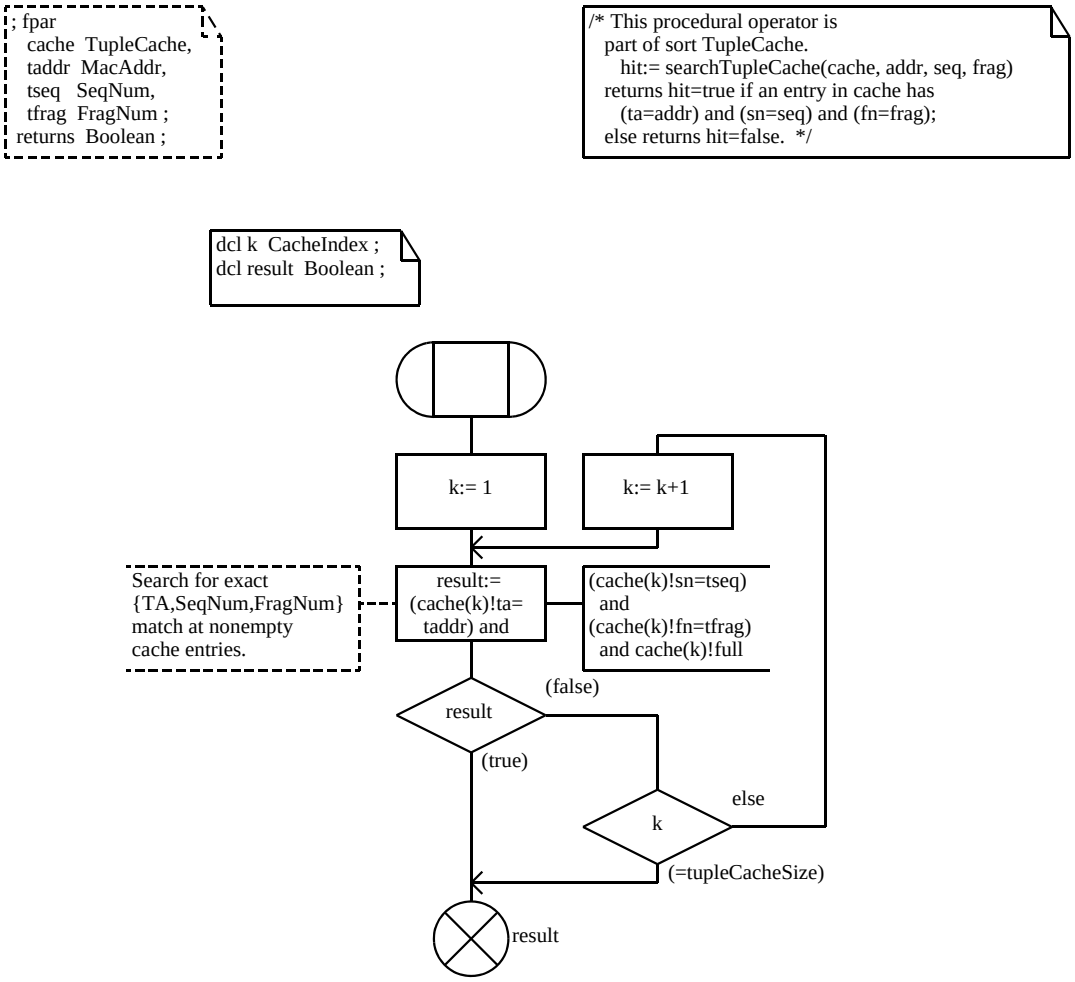

```
/*************************************************************************
 *       TupleCache support sorts
 *************************************************************************/
   /* Number of TupleCache entries and associated index range */
synonym tupleCacheSize Integer = 32;    /* this value is an example,
                  TupleCache size is implementation dependent */
syntype CacheIndex = Integer  constants 1:tupleCacheSize
  endsyntype CacheIndex;
/* TupleCache array */
/*    cache:= ClearTupleCache(cache)  to initialize cache */
/*    cache:= UpdateTupleCache(cache, addr, seq, frag, endRx) */
/*        if <addr,seq> is already cached, updates frag */
/*        if <addr,seq> not cached, fills an empty entry */
/*              or replaces an entry using an unspecified algorithm */
/*    SearchTupleCache(cache, addr, seq, frag) */
/*              returns true if specified <addr,seq,frag> in cache */
newtype TupleCache  Array( CacheIndex, Tuple);
  adding  operators
    ClearTupleCache  : TupleCache -> TupleCache;
    SearchTupleCache : TupleCache, MacAddr, SeqNum, FragNum -> Boolean;
    UpdateTupleCache : TupleCache, MacAddr, SeqNum, FragNum, Time ->
       TupleCache;
  operator ClearTupleCache;
    fpar cache TupleCache;  returns TupleCache;  referenced;
  operator SearchTupleCache;
    fpar cache TupleCache,  taddr MacAddr,  tseq SeqNum,  tfrag FragNum;
    returns Boolean;  referenced;
  operator UpdateTupleCache;
    fpar cache TupleCache,  taddr MacAddr,  tseq SeqNum,  tfrag FragNum,
    tnow Time;  returns TupleCache;  referenced;
endnewtype TupleCache;
```


Copyright © 2007 IEEE. All rights reserved.



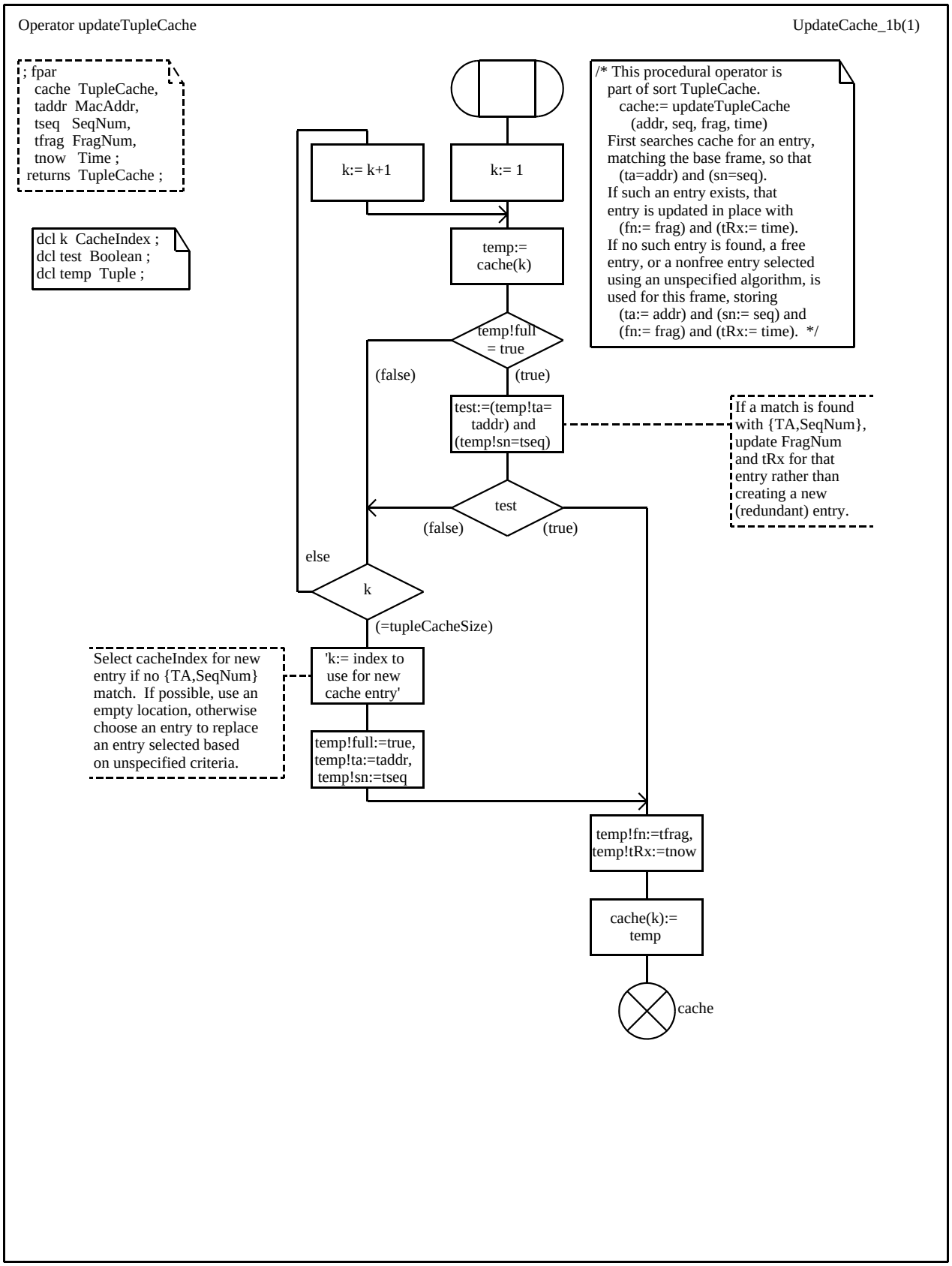

Copyright © 2007 IEEE. All rights reserved.

Operator searchTupleCache SearchCache_1a(1)





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3112_d\Counter(31)



```
/*****************************************************************
 *       32-bit Counter sort and Integer string sort
 *****************************************************************/
/* This sort used for MIB counters, needed because SDL Integers */
/* have no specified maximum value.  inc(counter) increments the */
/* counter value by 1, with wraparound from (2^32)-1 to 0. */

newtype Counter32   inherits Integer   operators all;
  adding  operators
    inc : Counter32 -> Counter32;
  axioms
    for all c in Counter32 (
      c < 4294967295 ==> inc(c) == c + 1;
      c >= 4294967295 ==> inc(c) == 0; );
endnewtype Counter32;
       /* String (1-origin) of Integer */
newtype Intstring   String( Integer, noInt);   endnewtype Intstring;
```


Copyright © 2007 IEEE. All rights reserved.



Package macsorts

3113_d\Queue(31)



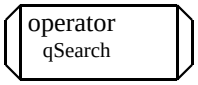
```
/*************************************************************************
 *       Generator for Queue sorts
 *************************************************************************/
/* The Queue generator is derived from the String0 generator */
/* to create Queues of any sort.  Queues operators are: */
/*    Qfirst(queue,item)  adds item as the first queue element */
/*    Qlast(queue,item)   adds item as the last queue element */
/* and the String0 operators Length, //, First, Last, Head, Tail */
/* Because operators can only return a single value, removing an */
/* element from a queue is a 2-step process: */
/*    dequeue first:  item:=First(queue);  queue:=Tail(queue); */
/*    dequeue last:   item:=Last(queue);   queue:=Head(queue); */
generator Queue(type Item, literal Emptyqueue)
 literals Emptyqueue;
 operators
   MkQ  : Item -> Queue;    /* make a queue from an item */
   Length  : Queue -> Integer;    /* number of items on queue */
   First  : Queue -> Item;    /* first item on queue */
   Qfirst  : Queue,Item -> Queue;    /* add item as first on queue */
   Tail  : Queue -> Queue;    /* all but first item on queue */
   Last  : Queue -> Item;    /* last item on queue */
   Qlast  : Queue,Item -> Queue;    /* add item as last on queue */
   head  : Queue -> Queue;    /* all but last item on queue */
    "//"  : Queue, Queue -> Queue;    /* concatenation */
   Extract! : Queue,Integer -> Item;    /* copy item from queue */
   Modify!  : Queue,Integer,Item -> Queue; /* modify item in queue */
   SubQ  : Queue,Integer,Integer -> Queue;
          /* SubQ(q,i,j) queue of length j starting from queue(i) */
 axioms
   for all item0 in Item(     for all q, q1, q2, q3 in Queue(
      for all i, j in Integer(
   /* constructors are Emptyqueue, MkQueue, and "//"; */
   /* equalities between constructor terms */
        q // Emptyqueue == q;          Emptyqueue // q == q;
        (q1 // q2) // q3 == q1 // (q2 // q3);
   /* definition of Length by applying it to all constructors */
         type Queue Length(Emptyqueue) == 0;
         type Queue Length(MkQueue(item0)) == 1;
         type Queue Length(q1 // q2) == Length(q1) + Length(q2);
   /* definition of Extract! by applying it to all constructors, */
        Extract!(MkQueue(item0), 0) == item0;
        i < Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q1, i);
        i >= Length(q1) ==> Extract!(q1 // q2, i) == Extract!(q2, i - Length(q1));
        i < 0 or i >= Length(q) ==> Extract!(q, i) == error!;
   /* definition of First and Last by other operations */
        First(q) == Extract!(q, 0);     Last(q) == Extract!(q, Length(q) - 1);
   /* definition of SubQ(q,i,j) by induction on j, */
        i >= 0 and i <= Length(q) ==> SubQ(q, i, 0) == Emptyqueue;
        i >= 0 and j > 0 and i + j <= Length(q) ==> SubQ(q, i, j) ==
           SubQ(q, i, j - 1) // MkQueue(Extract!(q, i + j - 1));
        i < 0 or j < 0 or i + j > Length(q) ==> SubQ(q,i,j) == error!;
   /* define Modify!, Head, Tail, Qfirst, Qlast by other ops */
        Modify!(q, i, item0) == SubQ(q, 0, I) //
           MkQueue(item0) // SubQ(q, i + 1, Length(q) - i - 1);
        head(q) == SubQ(q, 0, Length(q) - 1);
        Tail(q) == SubQ(q, 1, Length(q) - 1);
        Qfirst(q, item0) == MkQueue(item0) // q;
        Qlast(q, item0) == q // MkQueue(item0); )));
endgenerator Queue;
```

Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3114_d\Fragment(31)



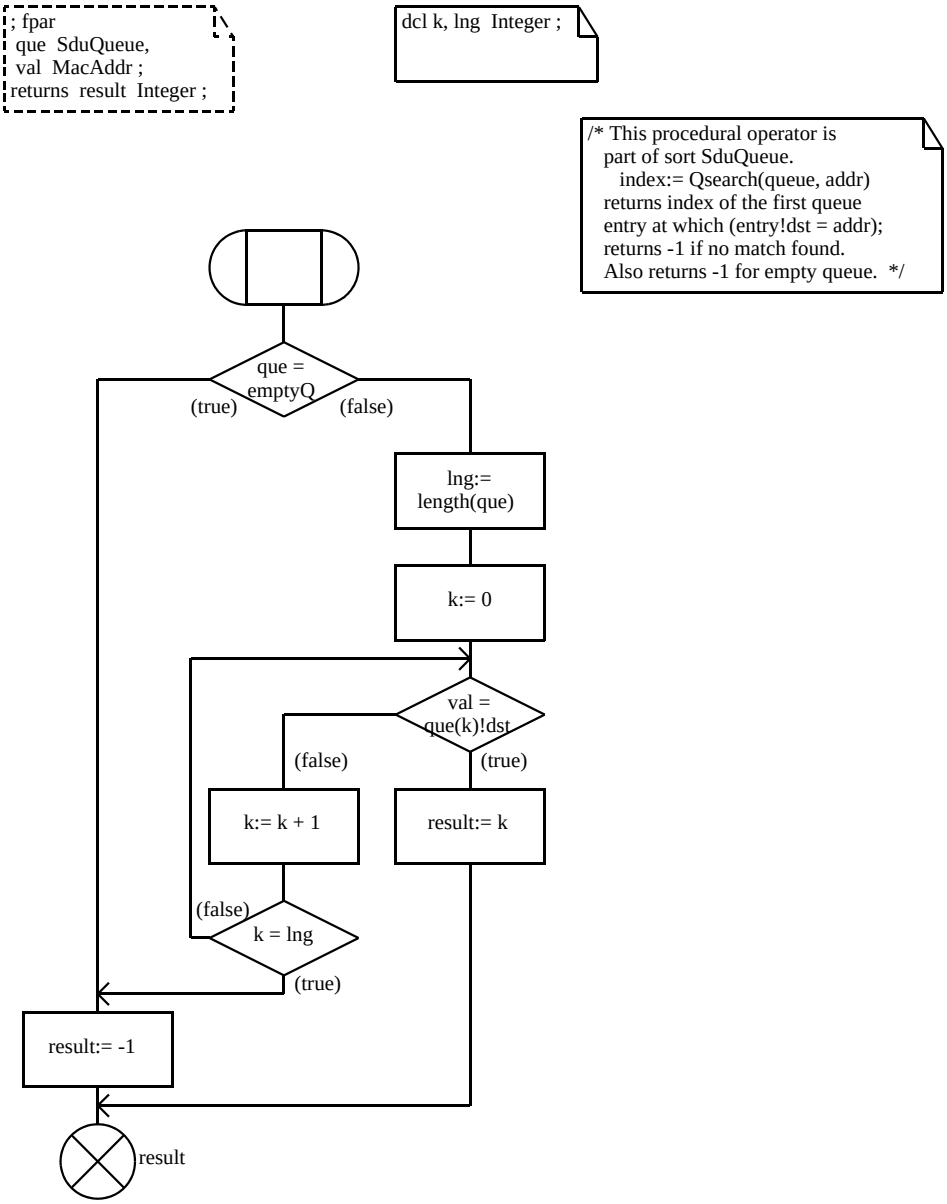



```
/*****************************************************************
 *       Fragmentation support sorts
 *****************************************************************/
/* Array to hold up to FragNum fragments of an Msdu/Mmpdu */
newtype FragArray  Array(FragNum, Frame);   endnewtype FragArray;
/* FragSdu structure is for OUTGOING MSDUs/MMPDUs (called SDUs) */
/* Each SDU, even if not fragmented, is held in an instance of */
/* this structure awaiting its (re)transmission attempt(s). */
/* Transmit queue(s) are ordered lists of FragSdu instances. */
newtype FragSdu struct
   fTot    FragNum;    /* number of fragments in pdus FragArray */
   fCur    FragNum;    /* next fragment number to send */
   fAnc    FragNum;    /* next fragment to announce in ATIM or TIM
                          when fAnc > fCur, pdus(fCur)+ may be sent */
   eol    Time;    /* set to (now + dUsec(aMaxTxMsduLifetime))
                          when the entry is created */
   sqf    SeqNum;    /* SDU sequence number, set at 1st Tx attempt */
   src    Integer;    /* short retry counter for this SDU */
   lrc    Integer;    /* long retry counter for this SDU */
   dst    MacAddr;    /* destinaton address */
   grpa    Boolean;    /* =true if RA (not DA) is a group address */
   psm    Boolean;    /* =true if RA (not DA) may be in pwr_save */
   resume    Boolean;    /* =true if fragment burst being resumed */
   cnfTo    PId;    /* address to which confirmation is sent */
   txrate    Rate;    /* data rate used for initial fragment */
   cf    CfPriority;    /* requested priority (from LLC) */
   pdus    FragArray;    /* array of Frame to hold fragments */
endnewtype FragSdu;
/* Queue of FragSdu */
/* for power save buffers, etc., searchable with Qsearch operator: */
/*    index:= Qsearch(queue, addr)     where queue is an SduQueue, */
/* index identifies the first queue entry at which */
/* entry!dst = addr; or as -1 if no match (or queue empty). */
newtype SduQueue Queue(FragSdu, emptyQ);
 adding  operators
   qSearch : SduQueue, MacAddr -> Integer;
 operator qSearch;
   fpar que SduQueue,  val MacAddr;  returns Integer; referenced;
endnewtype SduQueue;
```

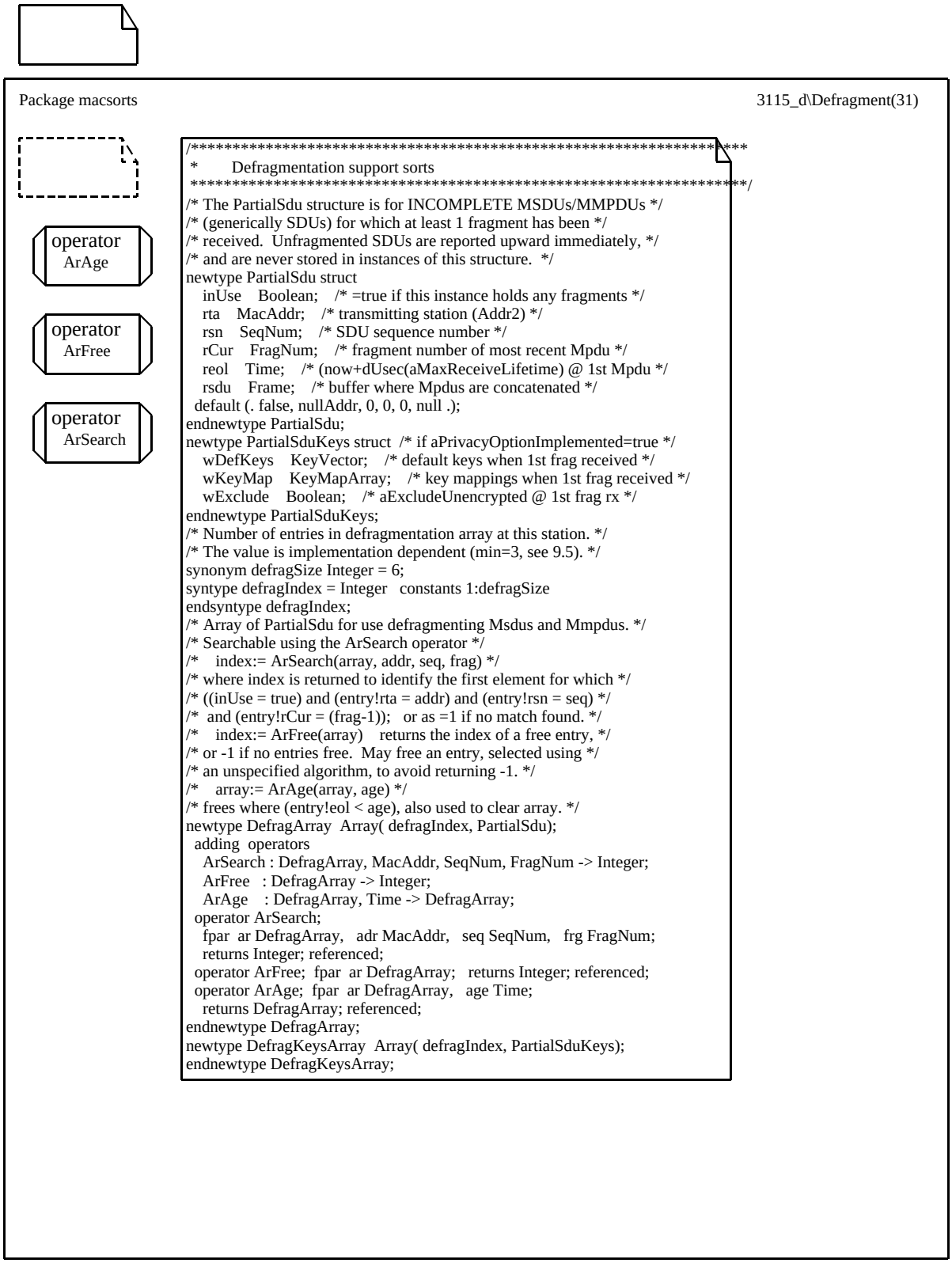

Copyright © 2007 IEEE. All rights reserved.

Operator Qsearch

Qsearch_1a(1)

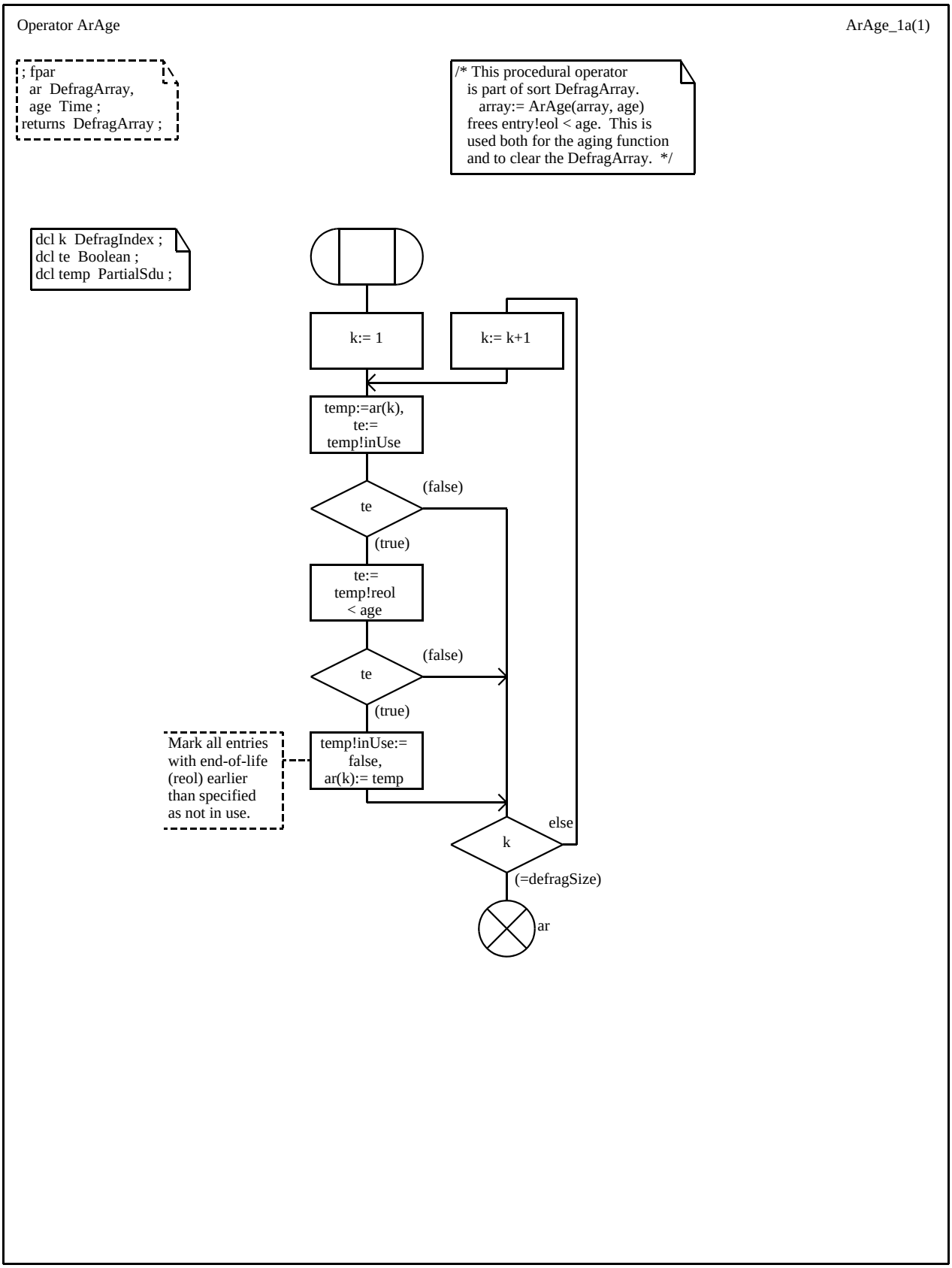



Copyright © 2007 IEEE. All rights reserved.

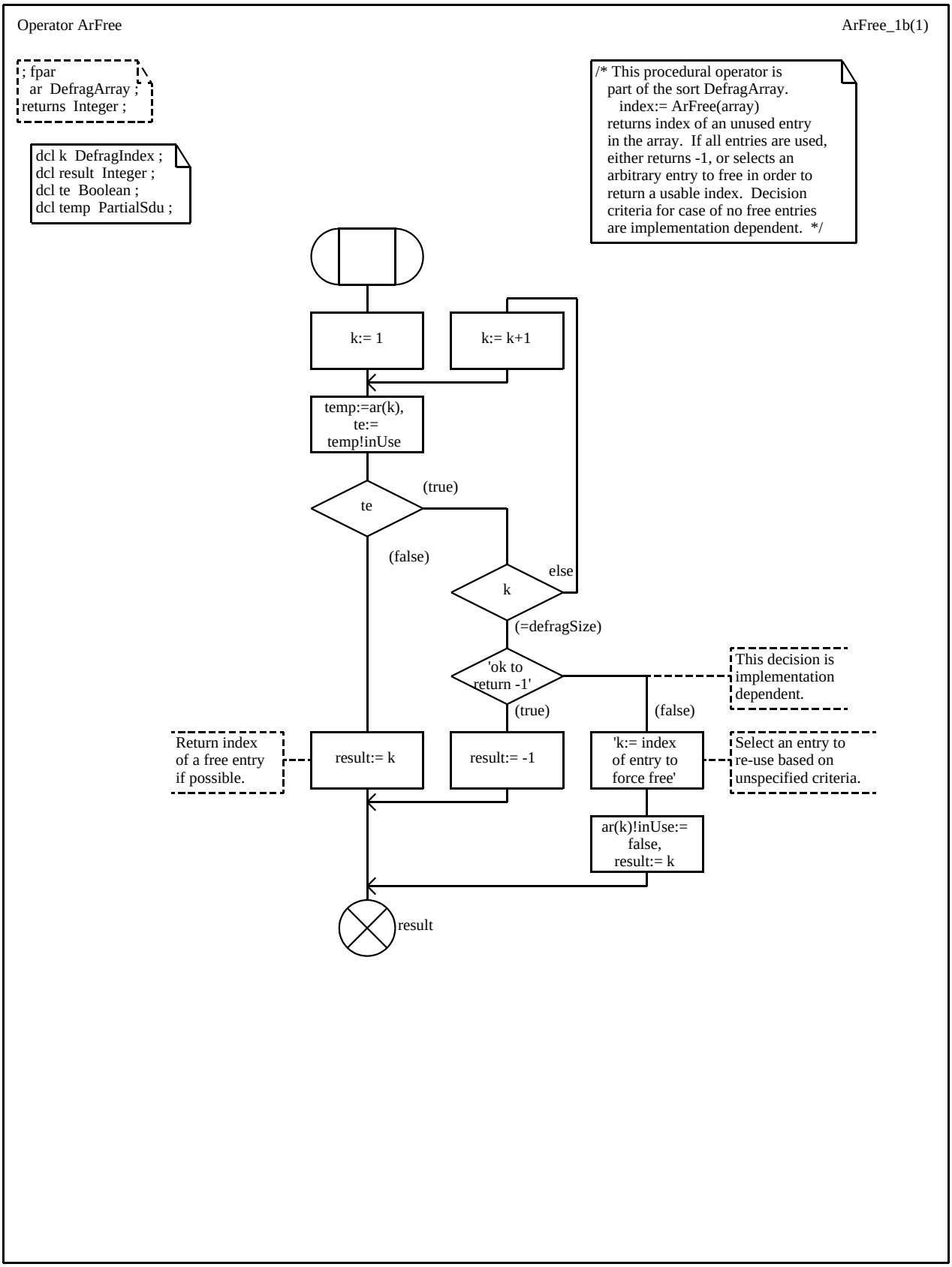




Copyright © 2007 IEEE. All rights reserved.

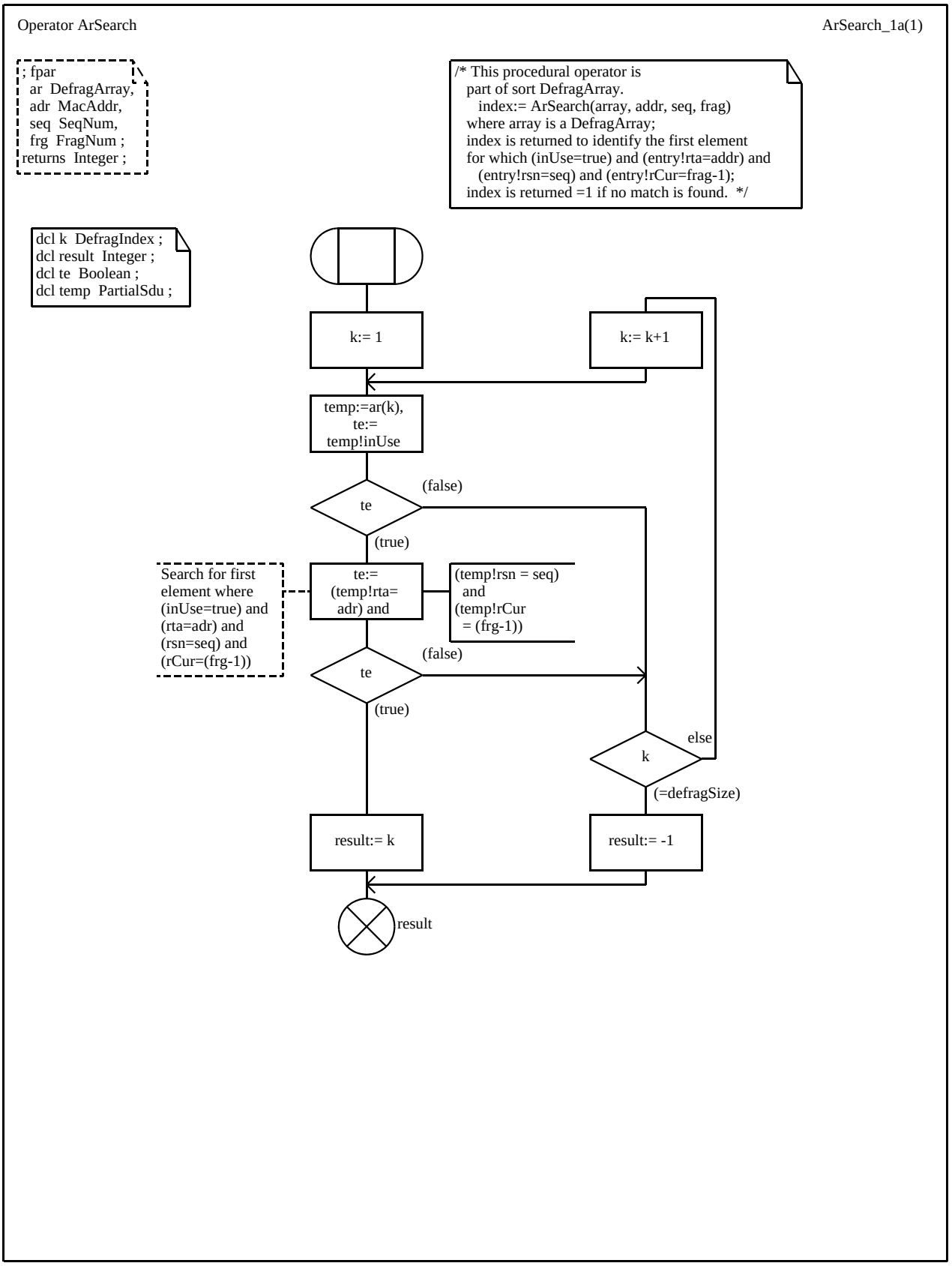



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3116_d\Crc_Wep(31)



operator
crc32

```
/*********************************************************************
 *       CRC-32 sorts (for FCS and ICV)
 *********************************************************************/
/* Crc is a subtype of Octetstring with added operators: */
/*    crc:= Crc32(crc,octet) */
/* updates the crc value to include the new octet, and */
/*    Mirror(crc), which returns a Crc value with the order */
/* of the octets, and of the bits in each octet, reversed for */
/* MSb-first transmission (see 7.1.1).  Crc variables must have */
/* exactly 4 octets, which is done using initCrc or S4. */
newtype Crc   inherits Octetstring   operators all;
  adding  operators
    Crc32  : Crc, Octet -> Crc;
    mirror : Crc -> Octetstring;
  operator Crc32;  fpar crcin Crc, val Octet;  returns Crc;  referenced;
  axioms     for all c in Crc(
      mirror(c) == S4(flip(c(3)),flip(c(2)),flip(c(1)), flip(c(0))); );
endnewtype Crc;
synonym initCrc Crc =    /* Initial Crc value (all 1s) */
          << type Crc>> S4(0xFF,0xFF,0xFF,0xFF);
synonym goodCrc Crc =    /* Unique remainder for valid CRC-32 */
          << type Crc>> S4(0x7B,0xDD,0x04,0xC7);
```

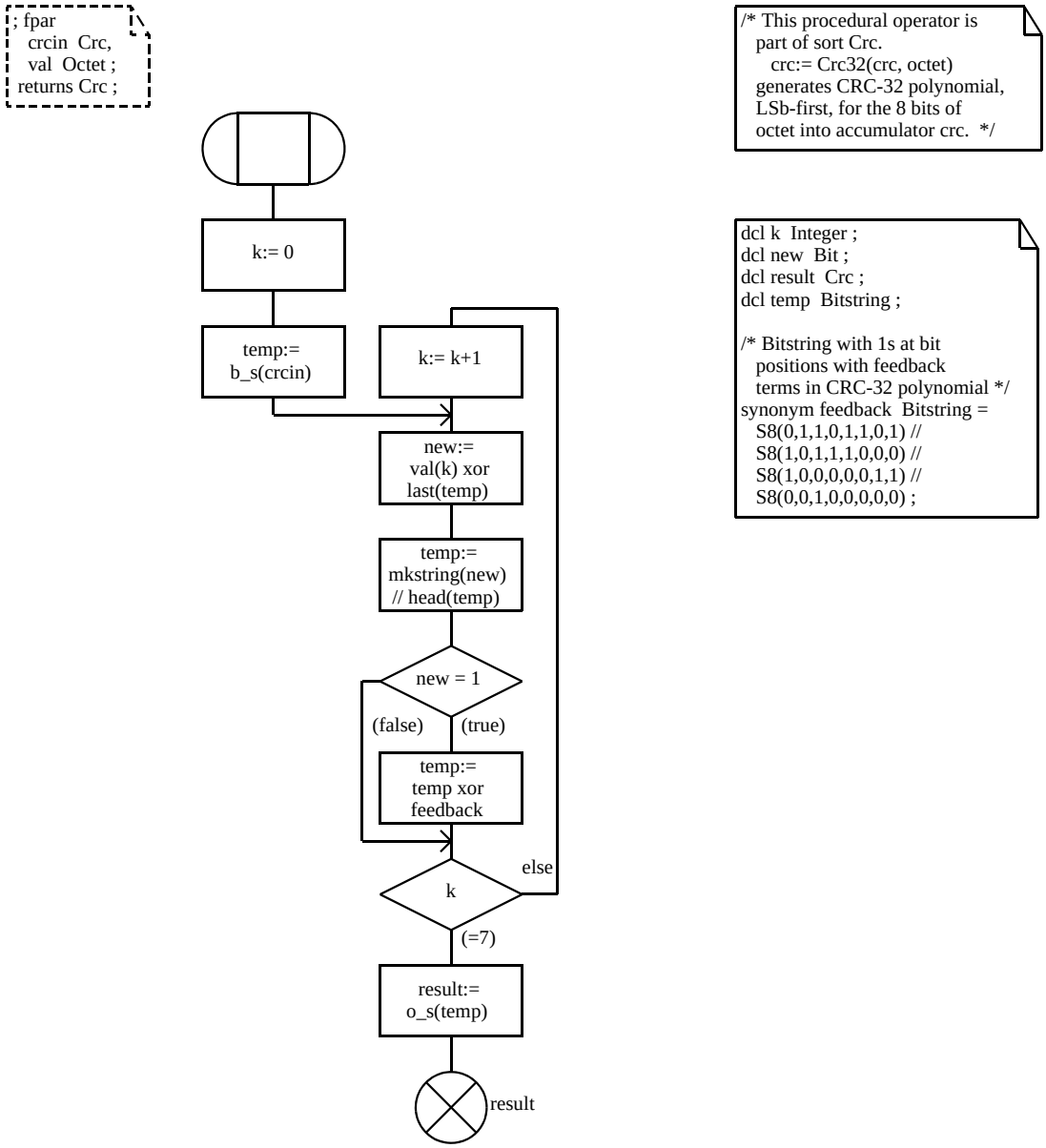


operator
keyLookup

```
/*********************************************************************
 *    WEP support sorts
 *********************************************************************/
syntype KeyIndex = Integer   constants 0:3   endsyntype KeyIndex;
newtype PrngKey inherits Octetstring   operators all;
  adding literals nullKey;  /* nullKey is not any of 2^40 key values */
  axioms nullKey == null;   default nullKey;   endnewtype PrngKey;
newtype KeyVector  /* vector of default WEP keys */
  Array( KeyIndex, PrngKey);   endnewtype KeyVector;
    /* Number of entries in aWepKeyMappings array at this station.
    /* implementation dependent value, minimum=10 (see 8.3.2). */
synonym sWepKeyMappingLength Integer = 10;
syntype KeyMappingRange = Integer
    constants 1:sWepKeyMappingLength   endsyntype KeyMappingRange;
newtype KeyMap  struct    /* structure used for entries in KeyMapArray */
    mappedAddr    MacAddr;
    wepOn    Boolean;
    wepKey    PrngKey;
endnewtype KeyMap;
/* KeyMapArray -- used for aWepKeyMapping table; */
/* an array of KeyMap indexed by KeyMappingRange, with operator */
/*    KeyMap := keyLookup(addr, keyMapArray, keyMapArrayLength) */
/* returns the KeyMap entry for the specified addr, or */
/* (. nullAddr, false, nullKey .) if no mapping for addr. */
newtype KeyMapArray  Array( KeyMappingRange, KeyMap);
  adding  operators
    keyLookup : MacAddr, KeyMapArray, Integer -> KeyMap;
  operator keyLookup;
    fpar  luadr MacAddr,   kma KeyMapArray,   kml Integer;
    returns KeyMap; referenced;
endnewtype KeyMapArray;
```

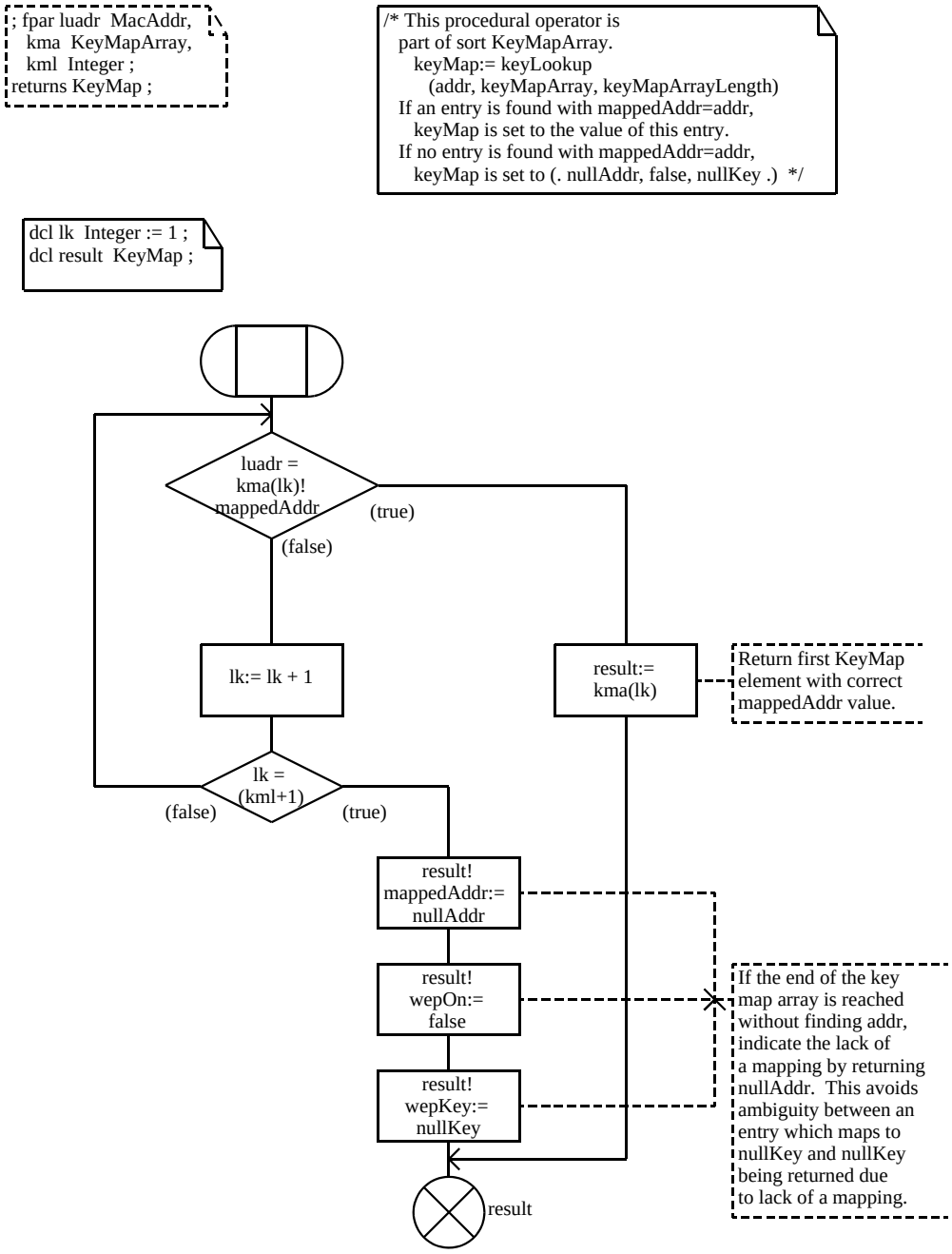

Copyright © 2007 IEEE. All rights reserved.

Operator Crc32 crc32_1a(1)




Copyright © 2007 IEEE. All rights reserved.

Operator keyLookup KeyLookup_1a(1)





Copyright © 2007 IEEE. All rights reserved.



Package macsorts

3117_d\Frame_1(31)



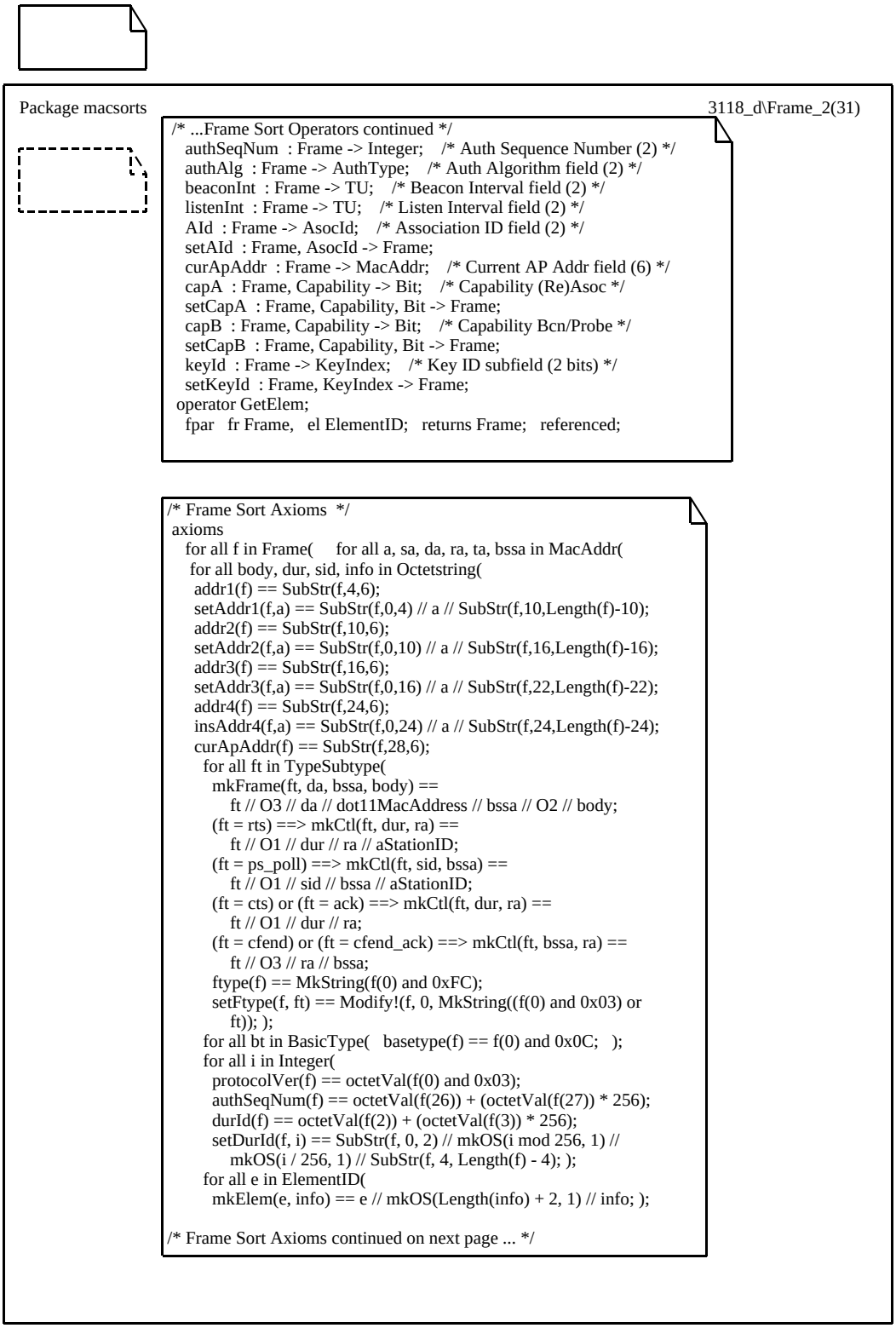
```
/*********************************************************************
 *       FRAME sort (the basic definition of fields in MAC frames)
 *********************************************************************/
/* Frame is a subtype of Octetstring with operators for creating
/* MAC headers, extracting each of the header fields and some
/* management frame fields, and modifying most of these fields.
/* There are operators to create and extract management frame
/* elements, but no operators for the frame body, IV, ICV, and FCS
/* fields, which are handled directly as Octetstrings. */
newtype Frame   inherits Octetstring   operators all;
  adding  operators
    mkFrame  : TypeSubtype, MacAddr, MacAddr, Octetstring -> Frame;
    mkCtl  : TypeSubtype, Octetstring, MacAddr -> Frame;
    protocolVer  : Frame -> Integer;   /* Protocol version (2 bits) */
    basetype  : Frame -> BasicType;    /* Type field (2 bits) */
    ftype  : Frame -> TypeSubtype;    /* Type & Subtype (6 bits) */
    setFtype  : Frame, TypeSubtype -> Frame;
    toDs  : Frame -> Bit;    /* To DS bit (1 bit) */
    setToDs  : Frame, Bit -> Frame;
    frDs  : Frame -> Bit;    /* From DS bit (1 bit) */
    setFrDs  : Frame, Bit -> Frame;
    moreFrag  : Frame -> Bit;    /* More Fragments bit (1 bit) */
    setMoreFrag  : Frame, Bit -> Frame;
    retryBit  : Frame -> Bit;    /* Retry bit (1 bit) */
    setRetryBit  : Frame, Bit -> Frame;
    pwrMgt   : Frame -> Bit;    /* Power Management bit (1 bit) */
    setPwrMgt  : Frame, Bit -> Frame;
    moreData  : Frame -> Bit;    /* More Data bit (1 bit) */
    setMoreData  : Frame, Bit -> Frame;
    wepBit  : Frame -> Bit;    /* WEP bit (1 bit) */
    setWepBit  : Frame, Bit -> Frame;
    orderBit  : Frame -> Bit;    /* {strictly}Order{ed} (1 bit) */
    setOrderBit  : Frame, Bit -> Frame;
    durId  : Frame -> Integer;    /* Duration/ID field (2) */
    setDurId  : Frame, Integer -> Frame;
    addr1  : Frame -> MacAddr;    /* Address 1 [DA/RA] field (6) */
    setAddr1  : Frame, MacAddr -> Frame;
    addr2  : Frame -> MacAddr;    /* Address 2 [SA/TA] field (6) */
    setAddr2  : Frame, MacAddr -> Frame;
    addr3  : Frame -> MacAddr;    /* Address 3 [Bss/DA/SA] field */
    setAddr3  : Frame, MacAddr -> Frame;
    addr4  : Frame -> MacAddr;    /* Address 4 [WDS-SA] field (6) */
    insAddr4  : Frame, MacAddr -> Frame;
    seq  : Frame -> SeqNum;    /* Sequence Number (12 bits) */
    setSeq  : Frame, SeqNum -> Frame;
    frag  : Frame -> FragNum;    /* Fragment Number (4 bits) */
    setFrag  : Frame, FragNum -> Frame;
    ts  : Frame -> Time;    /* Timestamp field (8) */
    setTs  : Frame, Time -> Frame;
    mkElem  : ElementID, Octetstring -> Frame;    /* make element */
    GetElem  : Frame, ElementID -> Frame;    /* get element if aval */
    status  : Frame -> StatusCode;    /* Status Code field (2) */
    setStatus  : Frame, StatusCode -> Frame;
    authStat  : Frame -> StatusCode;    /* Status Code in Auth frame */
    reason  : Frame -> ReasonCode;    /* Reason Code field (2) */

/* Frame operators continued on next page ...*/
```

operator
getElem

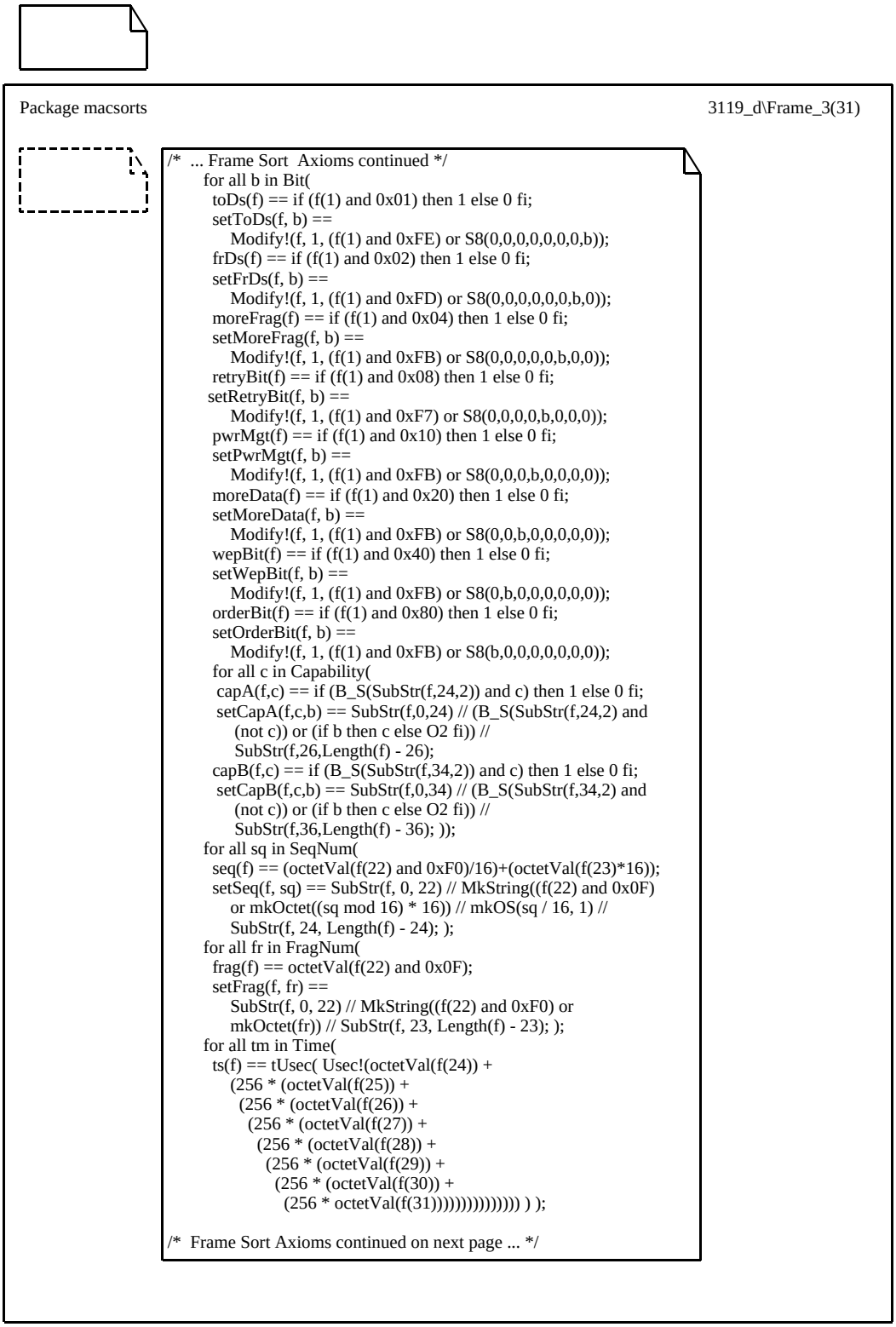
Gets element from body of Management frame. If the target element is not present an Octetstring of length zero is returned.

Copyright © 2007 IEEE. All rights reserved.

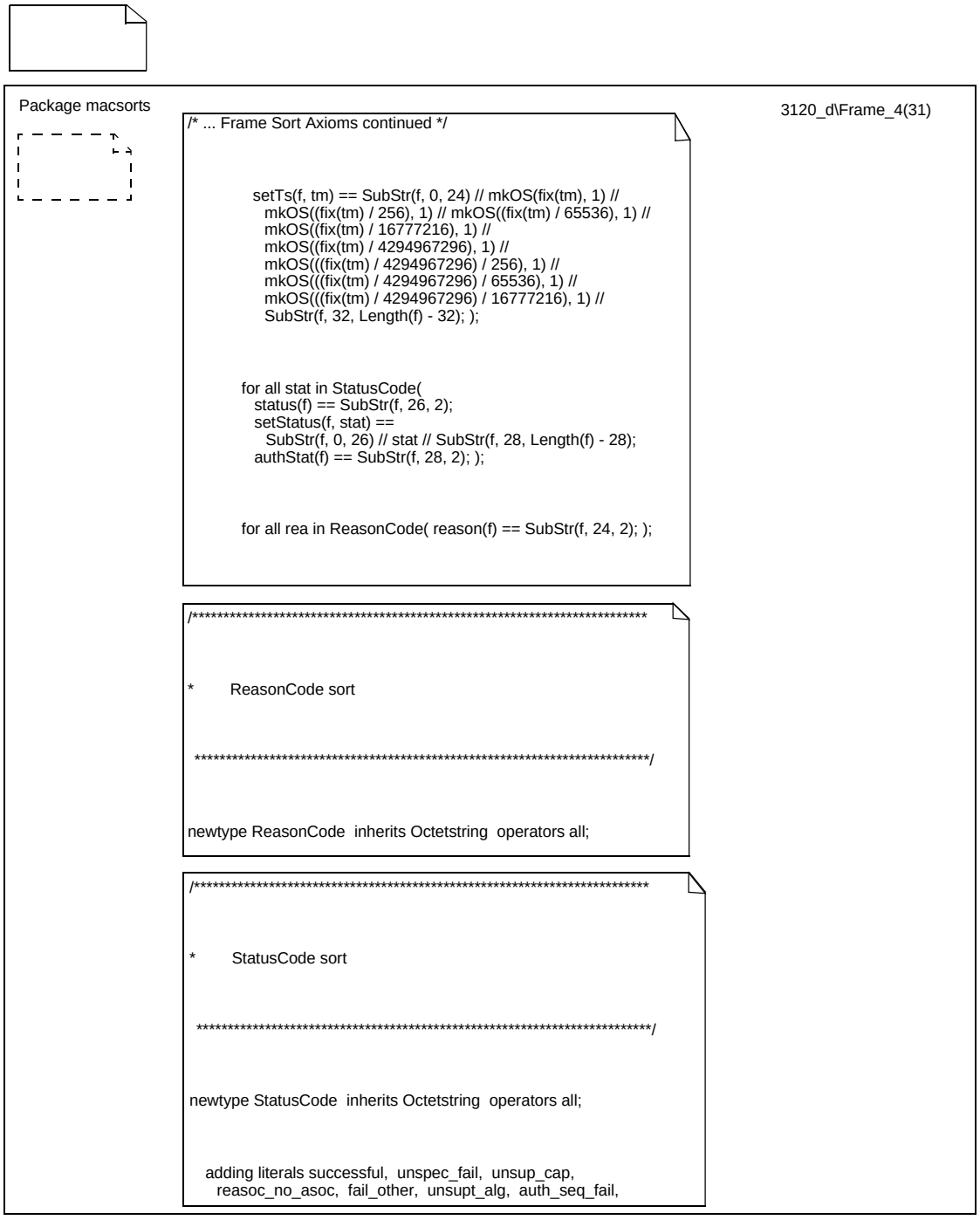




Copyright © 2007 IEEE. All rights reserved.

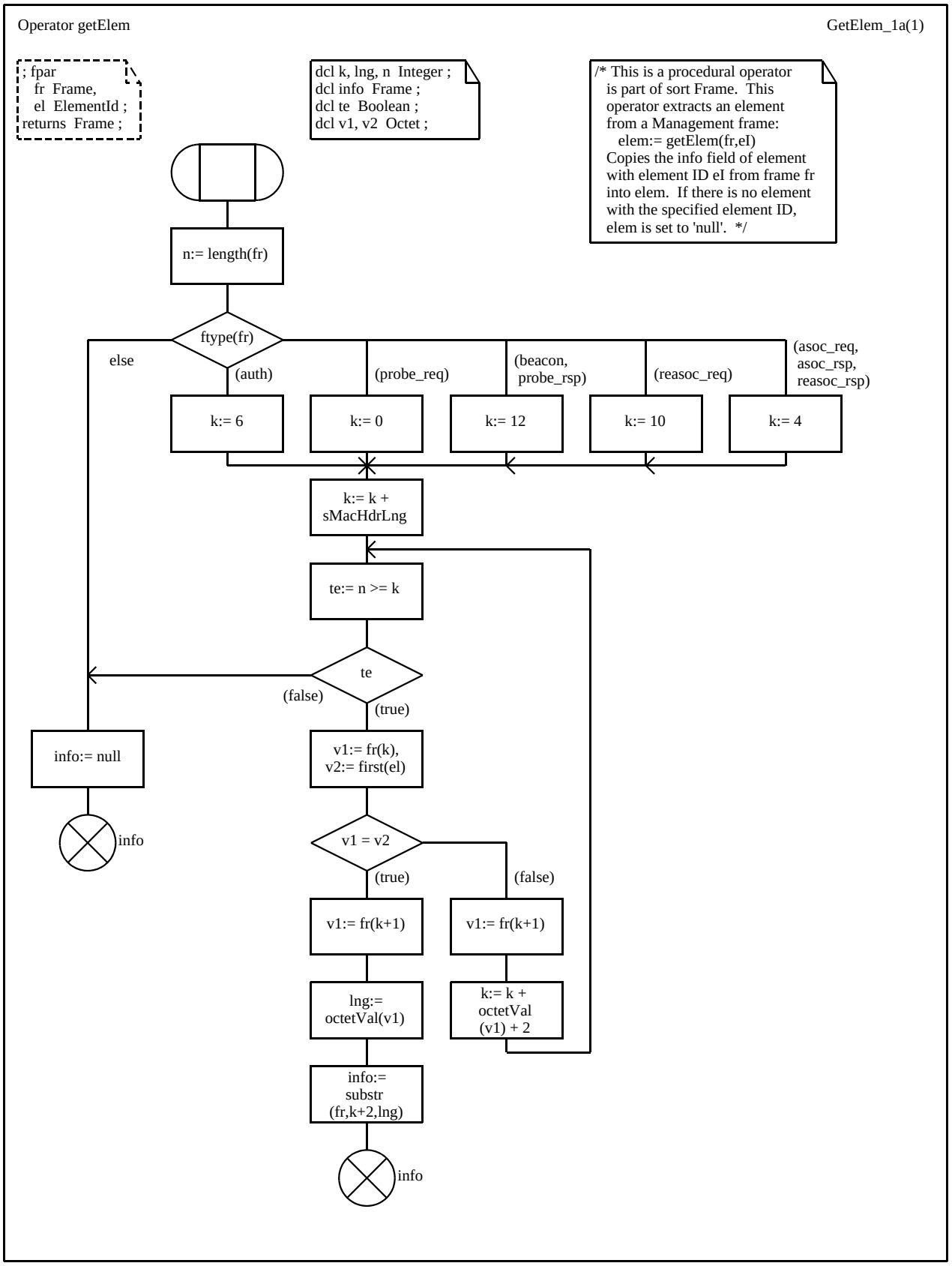



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

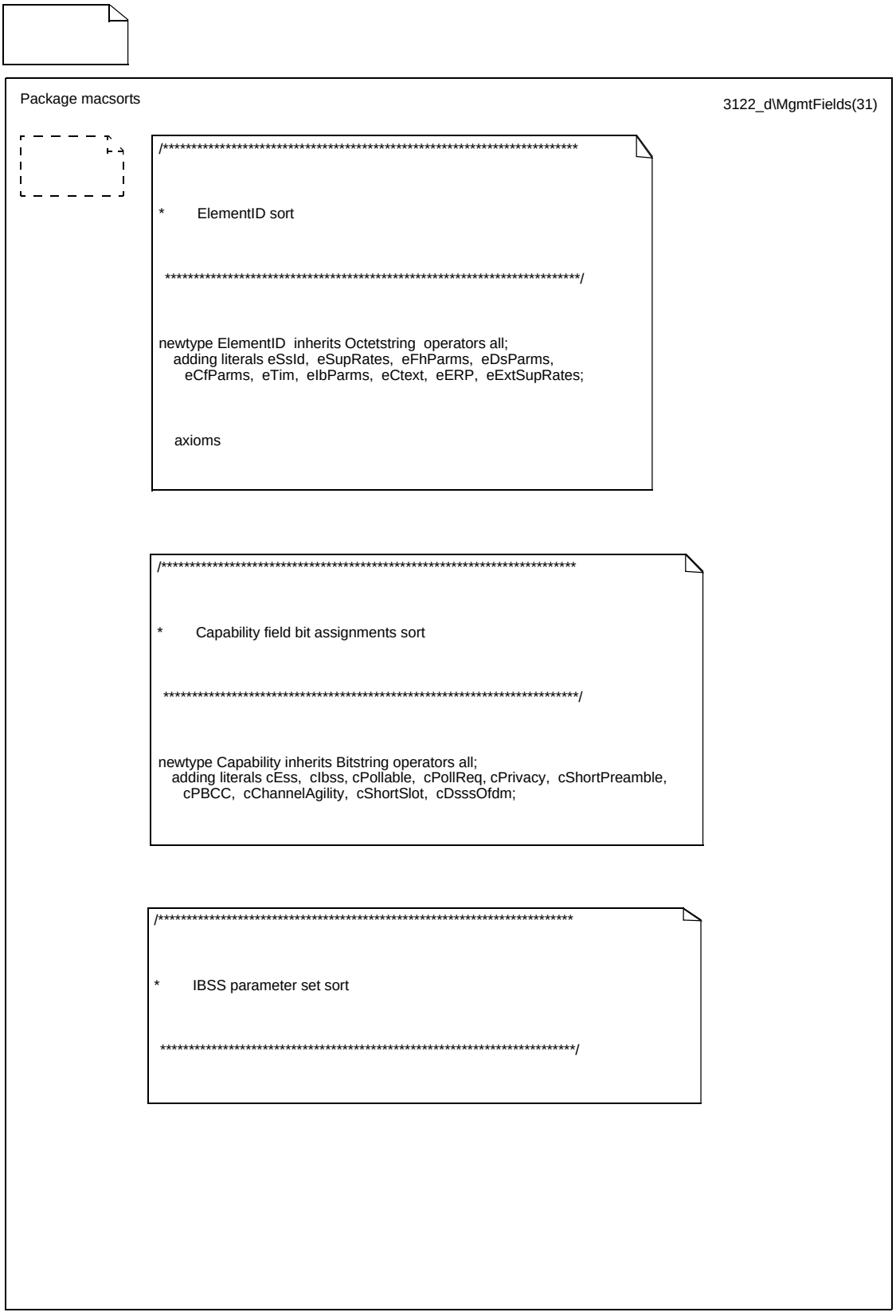


Package macsorts 3121_d\FrameType(31)



```
/*************************************************************************
 *       Frame Type sorts
 *************************************************************************/
/* TypeSubtype defines the full, 6-bit frame type identifiers. */
/* These values are useful with ftype operator of Frame sort. */
newtype TypeSubtype   inherits Octetstring   operators all;
 adding literals asoc_req, asoc_rsp, reasoc_req, reasoc_rsp,
   probe_req, probe_rsp, beacon, atim, disasoc, auth, deauth,
   ps_poll, rts, cts, ack, cfend, cfend_ack, data, data_ack,
   data_poll, data_poll_ack, null_frame, cfack, cfpoll, cfpoll_ack;
 axioms
   asoc_req == MkString(S8(0,0,0,0,0,0,0,0));
   asoc_rsp == MkString(S8(0,0,0,0,1,0,0,0));
   reasoc_req == MkString(S8(0,0,0,0,0,1,0,0));
   reasoc_rsp == MkString(S8(0,0,0,0,1,1,0,0));
   probe_req == MkString(S8(0,0,0,0,0,0,1,0));
   probe_rsp == MkString(S8(0,0,0,0,1,0,1,0));
   beacon == MkString(S8(0,0,0,0,0,0,0,1));
   atim == MkString(S8(0,0,0,0,1,0,0,1));
   disasoc == MkString(S8(0,0,0,0,0,1,0,1));
   auth == MkString(S8(0,0,0,0,1,1,0,1));
   deauth == MkString(S8(0,0,0,0,0,0,1,1));
   ps_poll == MkString(S8(0,0,1,0,0,1,0,1));
   rts == MkString(S8(0,0,1,0,1,1,0,1));
   cts == MkString(S8(0,0,1,0,0,0,1,1));
   ack == MkString(S8(0,0,1,0,1,0,1,1));
   cfend == MkString(S8(0,0,1,0,0,1,1,1));
   cfend_ack == MkString(S8(0,0,1,0,1,1,1,1));
   data == MkString(S8(0,0,0,1,0,0,0,0));
   data_ack == MkString(S8(0,0,0,1,1,0,0,0));
   data_poll == MkString(S8(0,0,0,1,0,1,0,0));
   data_poll_ack == MkString(S8(0,0,0,1,1,1,0,0));
   null_frame == MkString(S8(0,0,0,1,0,0,1,0));
   cfack == MkString(S8(0,0,0,1,1,0,1,0));
   cfpoll == MkString(S8(0,0,0,1,0,1,1,0));
   cfpoll_ack == MkString(S8(0,0,0,1,1,1,1,0));
endnewtype TypeSubtype;
/* BasicTypes defines the 2-bit frame type groups */
newtype BasicType   inherits Bitstring   operators all;
 adding literals   control,   data,   management,   reserved;
 axioms
   control == S8(0,0,1,0,0,0,0,0);    data == S8(0,0,0,1,0,0,0,0);
   management == S8(0,0,0,0,0,0,0,0);    reserved == S8(0,0,1,1,0,0,0,0);
endnewtype BasicType;
```


Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3123_d\CF_And_AsocParams(31)



```
/*************************************************************************
 *       CF parameter set sort
 *************************************************************************/
newtype CfParms   inherits Octetstring   operators all;
 adding  operators
   cfpCount  : CfParms -> Integer; /* CfpCount field (1) */
   setCfpCount  : CfParms, Integer -> CfParms;
   cfpPeriod  : CfParms -> Integer; /* CfpPeriod field (1) */
   setCfpPeriod  : CfParms, Integer -> CfParms;
   cfpMaxDur  : CfParms -> TU;    /* CfpMaxDuration field (2) */
   setCfpMaxDur  : CfParms, TU -> CfParms;
   cfpDurRem  : CfParms -> TU;    /* CfpDurRemaining field (2) */
   setCfpDurRem  : CfParms, TU -> CfParms;
 axioms   for all cf in CfParms(   for all i in Integer(   for all u in TU(
        cfpCount(cf) == octetVal(cf(0));
        setCfpCount(cf, i) == mkOS(i, 1) // Tail(cf);
        cfpPeriod(cf) == octetVal(cf(1));
        setCfpPeriod(cf, i) == cf(0) // mkOS(i, 1) // SubStr(cf,2,4);
        cfpMaxDur(cf) == octetVal(cf(2)) + (octetVal(cf(3)) * 256);
        setCfpMaxDur(cf, u) == SubStr(cf, 0, 2) // mkOS(u mod 256, 1)
           // mkOS(u / 256, 1) // SubStr(cf, 4, 2);
        cfpDurRem(cf) == octetVal(cf(4)) + (octetVal(cf(5)) * 256);
        setCfpDurRem(cf, u) == SubStr(cf, 0, 4) // mkOS(u mod 256, 1)
           // mkOS(u / 256, 1); )));
endnewtype CfParms;
```

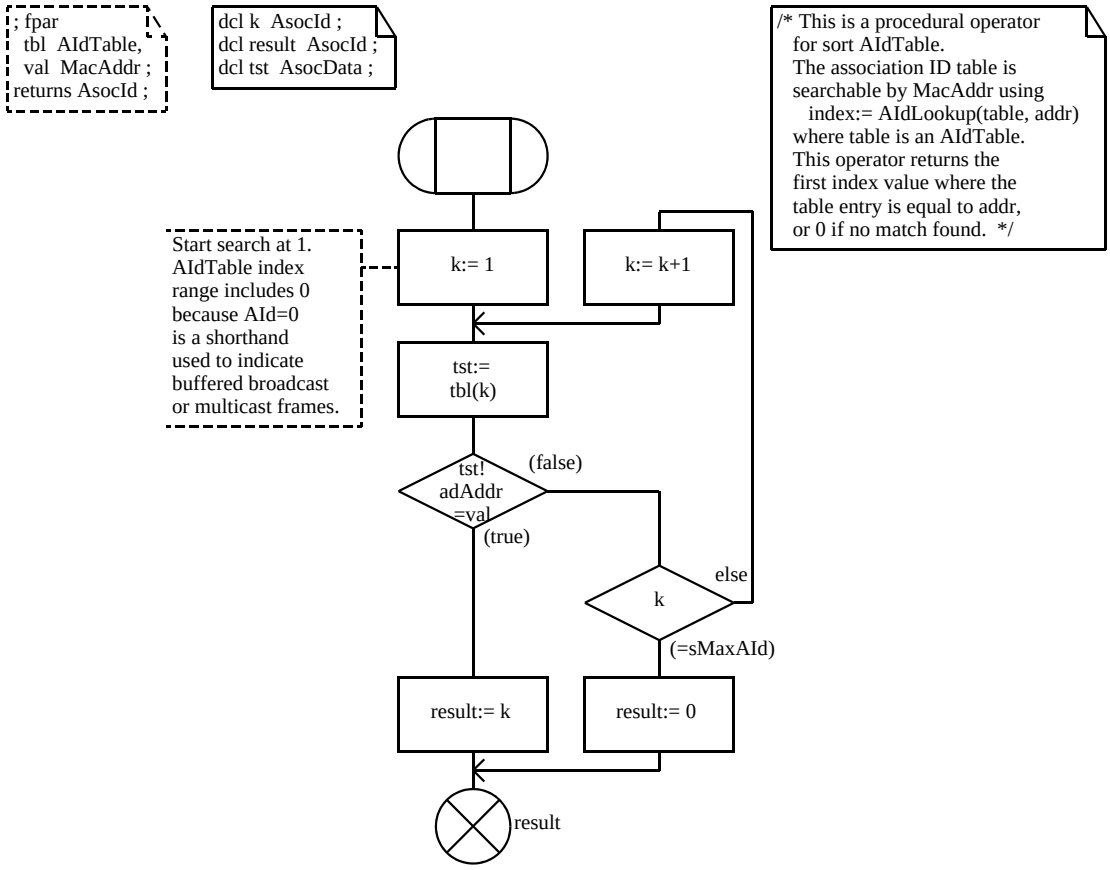



```
/*************************************************************************
 *       Sorts for association management at AP
 *************************************************************************/
synonym sMaxAId Integer = 2007;  /* 2007 is largest allowable value */
                          /* implementation limit may be lower */
syntype AsocId = Integer   constants 0:sMaxAId   endsyntype AsocId;
       /* Station Association Record -- only used at APs */
newtype AsocData struct
   adAddr    MacAddr;    /* address of associated station */
   adPsm     PwrSave;    /* power save mode of the station */
   adCfPoll  Boolean;    /* true if station is CfPollable */
   adPollRq  Boolean;    /* true if station requested polling */
   adNoPoll  Boolean;    /* true if station requested no polling */
   adMsduIP  Boolean;    /* true if partial Msdu outstanding to sta */
   adAuth    AuthType;   /* authentication type used by station */
   adRates   RateSet;    /* supported rates from association request */
   adAge     Time;    /* time of association */
endnewtype AsocData;
/* Association table -- array of AsocData, only used at APs */
/*    index:= AIdLookup(table, addr) */
/* returns the index of location where table(x)!adAddr=addr */
/* or 0 if no such location found. */
newtype AIdTable   Array(AsocId, AsocData);
 adding  operators
   AIdLookup : AIdTable, MacAddr -> AsocId;
 operator AIdLookup;
   fpar tbl AIdTable,   val MacAddr;   returns AsocId;   referenced;
endnewtype AIdTable;
```


Copyright © 2007 IEEE. All rights reserved.

Operator AIdLookup

AIdLookup_1a(1)




Copyright © 2007 IEEE. All rights reserved.



Package macsorts 3124_d\TIM(31)

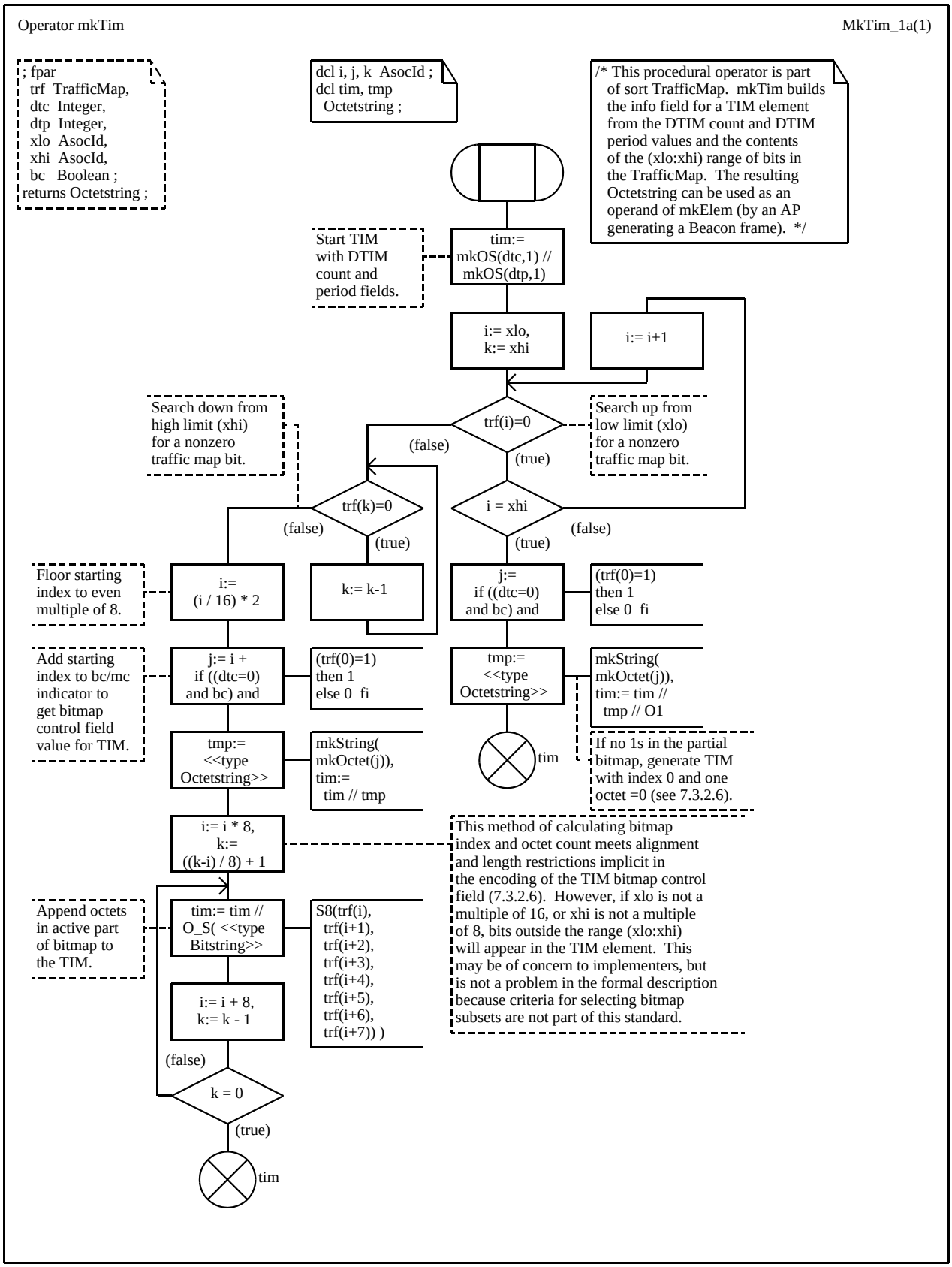


operator
mkTim

operator
nextAId

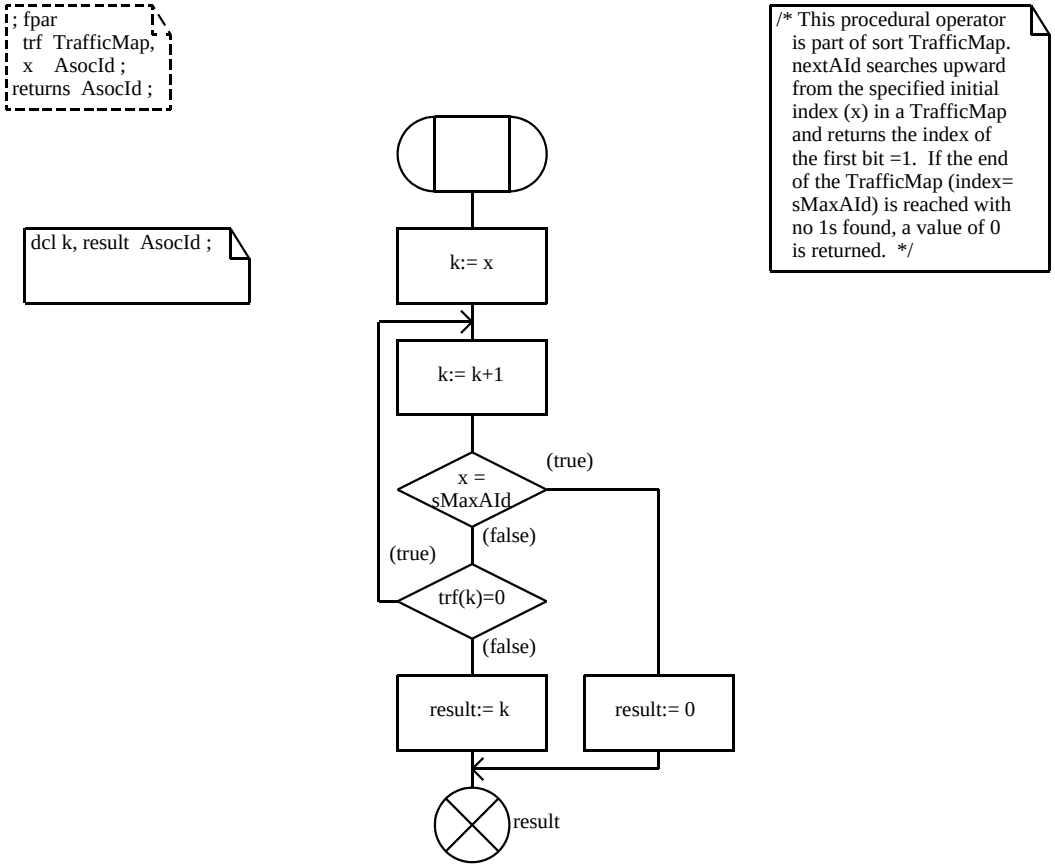
```
/*************************************************************************
 *       Traffic Information Map (TIM) support sorts
 *************************************************************************/
/* TrafficMap is an Array of Bit indexed by AId. */
/* Bits =1 in TrafficMap denote the presence of buffered frame(s) */
/* for the station assigned that AId.  TrafficMap operators are: */
/*    mkTim(trafficMap, dtimCnt, dtimPer, lowAId, highAId, bcst) */
/* returns Octetstring to use as the info field of a TIM element */
/* The TIM will contain bits =1 for TrafficMap locations in the */
/* range (lowAId):(highAId).  Buffered broadcasts and multicasts */
/* (AId 0) are indicated if dtimCnt=0 and if bcst=true. */
/*    nextAId(trafficMap, currentAId) */
/* returns index greater than currentAId at which TrafficMap=1. */
/* If no locations before sMaxAId are =1, returns 0. */
newtype TrafficMap  Array( AsocId, Bit);
  adding  operators
    mkTim : TrafficMap, Integer, Integer, AsocId, AsocId, Boolean -> Octetstring;
    nextAId : TrafficMap, AsocId -> AsocId;
  operator mkTim;
    fpar trf TrafficMap,  dtc Integer,  dtp Integer,  xlo AsocId,
    xhi AsocId,  bc Boolean;  returns Octetstring;  referenced;
  operator nextAId;
    fpar trf TrafficMap,  x AsocId;  returns AsocId;  referenced;
endnewtype TrafficMap;
/* TIM is a subtype of Octetstring with operators: */
/*    bufFrame(tim,AId)   returns true if the TIM info field */
/*        (obtained using getElem) is =1 at tim(AId). */
/*    bufBcst(tim)   returns true if the TIM info field */
/*        indicates buffered broadcast/multicast traffic */
/*    dtCount(tim)   returns DTIM count value from TIM */
/*    dtPeriod(tim)  returns DTIM period value from TIM */
newtype TIM   inherits Octetstring   operators all;
  adding  operators
    bufFrame : TIM, AsocId -> Boolean;
    bufBcst  : TIM -> Boolean;
    dtCount  : TIM -> Integer;
    dtPeriod : TIM -> Integer;
  axioms
    for all el in TIM(     for all a in AsocId(
      bufFrame(el, a) ==
        if a < (octetVal(el(2) and 0xFE) * 8) then false
          else
          if a >= ((octetVal(el(2) and 0xFE)*8) + ((Length(el)-3)*8))
            then false
            else
              Extract!(B_S(el), (a-(octetVal(el(2) and 0xFE)*8)+24)) = 1
          fi fi;
      bufBcst(el) == (el(2) and 0x01) = 0x01;
      dtCount(el) == octetVal(el(0));
      dtPeriod(el) == octetVal(el(1)); ));
endnewtype TIM;
```


Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

Operator nextAId NextAId_1a(1)





Copyright © 2007 IEEE. All rights reserved.



Package macsorts

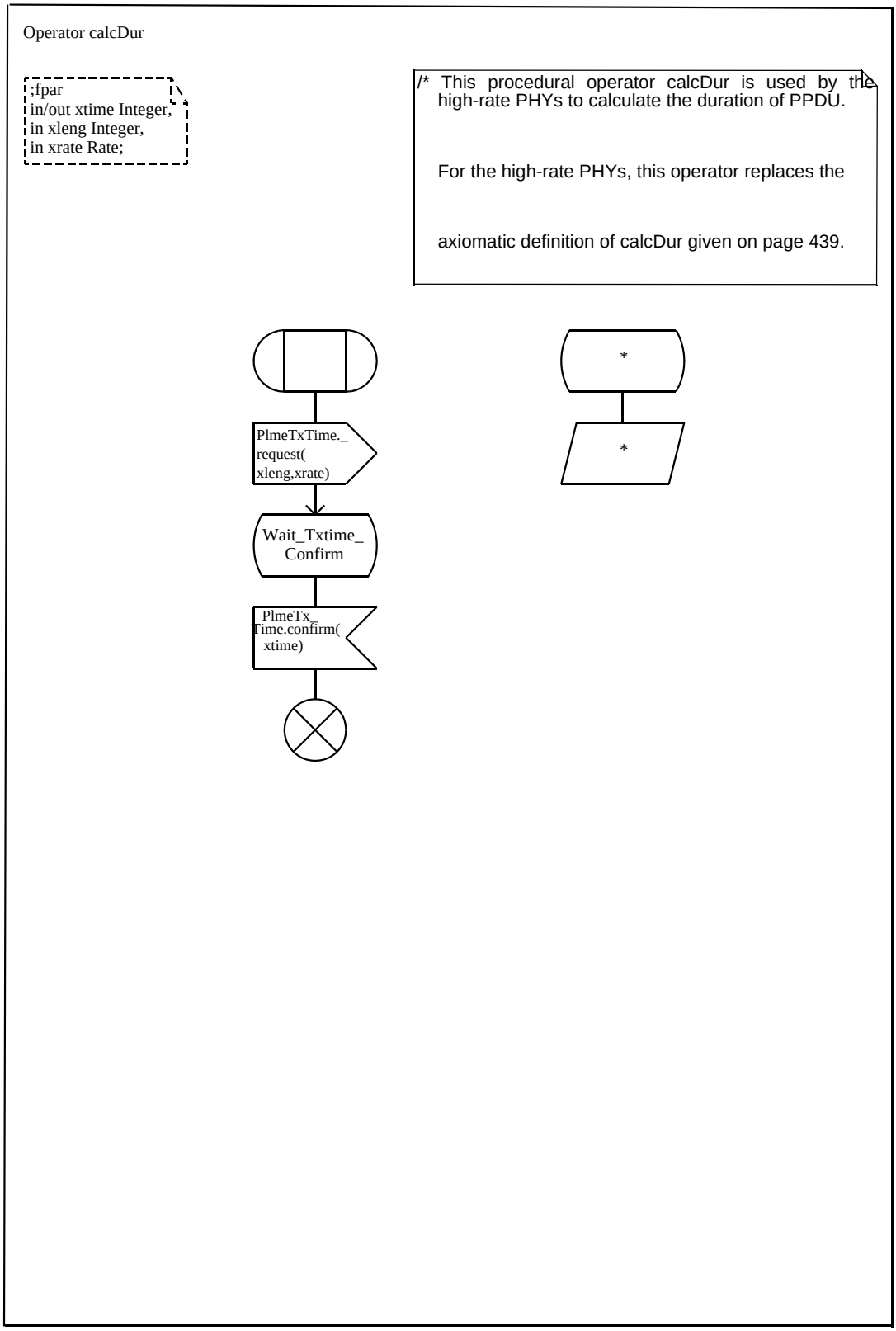
3125_d\RateAndDurationSorts(31)



```
/*********************************************************************
 *       Multi-rate support sorts
 *********************************************************************/
newtype Rate   inherits Octet   operators all;
 adding  operators
   calcDur : Rate, Integer -> Integer; /* converts (rate,bitCount) to integer usec */
   rateVal : Rate -> Rate; /* clears high-order bit */
   basicRate : Rate -> Rate; /* sets high-order bit */
   isBasic : Rate -> Boolean; /* true if high-order bit set */
 axioms
   for all r in Rate(    for all i in Integer(    for all b in Boolean(
        calcDur(r, i) == ((((10000000 + (octetVal(r and 0x7F) - 1)) /
            (500 * octetVal(r and 0x7F))) * i) + 9999) / 10000;
        rateVal(r) == r and 0x7F;          basicRate(r) == r or 0x80;
        isBasic(r) == (r and 0x80) = 0x80; )));
endnewtype Rate;
syntype RateString = Octetstring   endsyntype RateString;
```

```
/*********************************************************************
 *       MPDU duration factor support sort
 *********************************************************************/
/* These operators support the encoding used to allow */
/* an Integer to represent the value of aMpduDurationFactor. */
/*    calcDF(PlcpBits, MpduBits)  returns an Integer which is */
/* the fractional part of ((PlcpBits/MpduBits)-1)*(1e9). */
/*    stuff(durFactor, MpduBits)  returns the number of PlcpBits */
/* which result from MpduBits at the specified durFactor. */
newtype DurFactor   inherits Integer   operators all;
 adding  operators
   calcDF : Integer, Integer -> DurFactor;
   stuff  : DurFactor, Integer -> Integer;
 axioms
   for all df in DurFactor(     for all mb, pb in Integer(
       calcDF(pb, mb) == ((pb * 1000000000) / mb) - 1000000000;
       stuff(df, mb) == ((mb * df) + (mb - 1)) / 1000000000; ));
endnewtype DurFactor;
```

Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.



Package macsorts

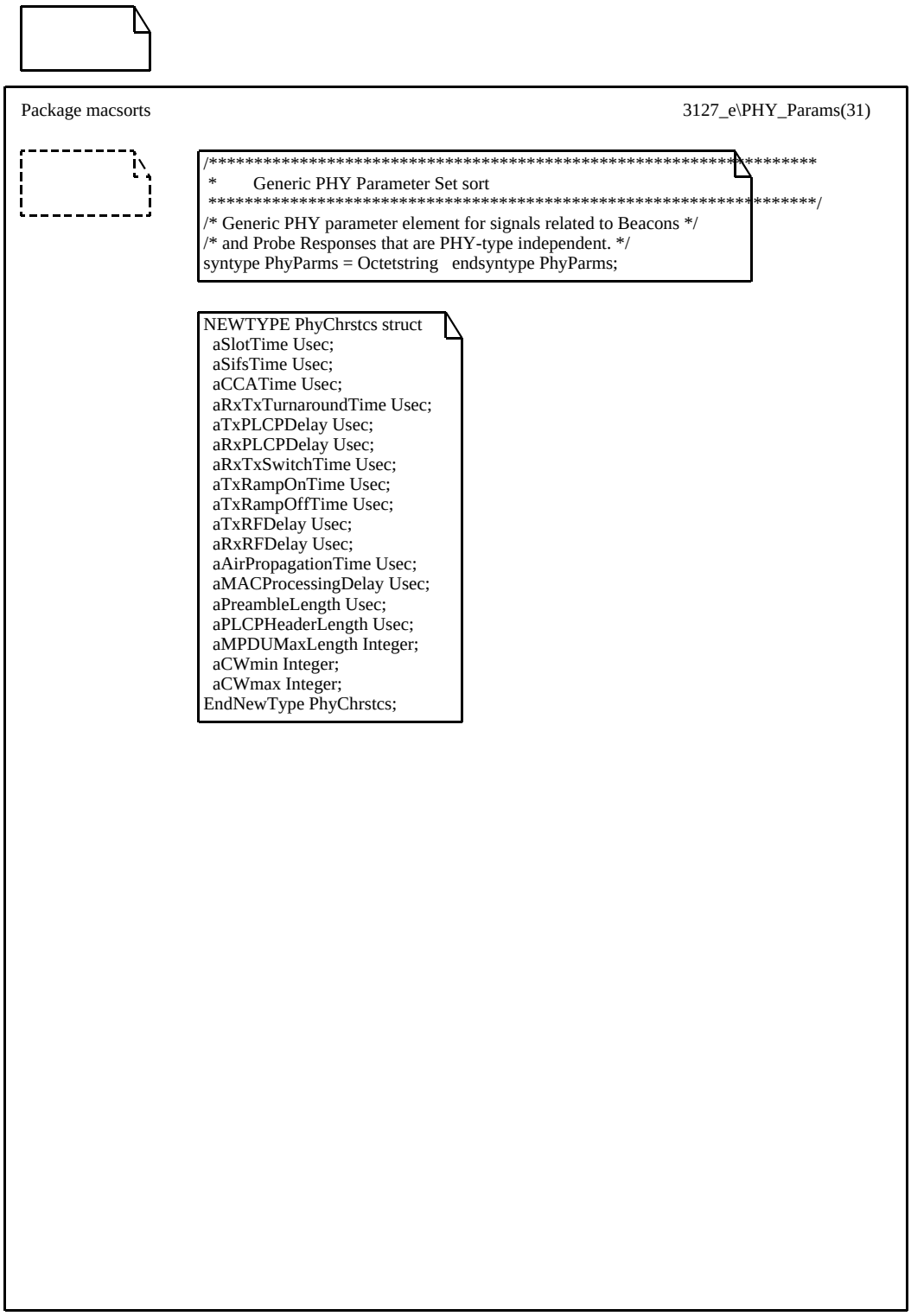
3126_d\FH_DS_Params(31)



```
/*************************************************************************
 *       FH parameter set sort
 *************************************************************************/
newtype FhParms   inherits Octetstring   operators all;
 adding  operators
   dwellTime  : FhParms -> TU;    /* Dwell Time field (2) */
   setDwellTime  : FhParms, TU -> FhParms;
   hopSet  : FhParms -> Integer; /* Hop Set field (1) */
   setHopSet  : FhParms, Integer -> FhParms;
   hopPattern  : FhParms -> Integer; /* Hop Pattern field (1) */
   setHopPattern  : FhParms, Integer -> FhParms;
   hopIndex  : FhParms -> Integer; /* Hop Index field (1) */
   setHopIndex  : FhParms, Integer -> FhParms;
 axioms
   for all fh in FhParms(   for all i in Integer(   for all u in TU(
     dwellTime(fh) == octetVal(fh(0)) + (octetVal(fh(1)) * 256);
     setDwellTime(fh, u) == mkOS(u mod 256, 1) // mkOS(u / 256, 1) // SubStr(fh, 2, 3);
     hopSet(fh) == octetVal(fh(2));
     setHopSet(fh,i) == SubStr(fh,0,2) // mkOS(i,1) // SubStr(fh,3,2);
     hopPattern(fh) == octetVal(fh(3));
     setHopPattern(fh, i) == SubStr(fh,0,3) // mkOS(i,1) // Last(fh);
     hopIndex(fh) == octetVal(fh(4));
     setHopIndex(fh, i) == SubStr(fh, 0, 4) // mkOS(i, 1);)));
endnewtype FhParms;
```

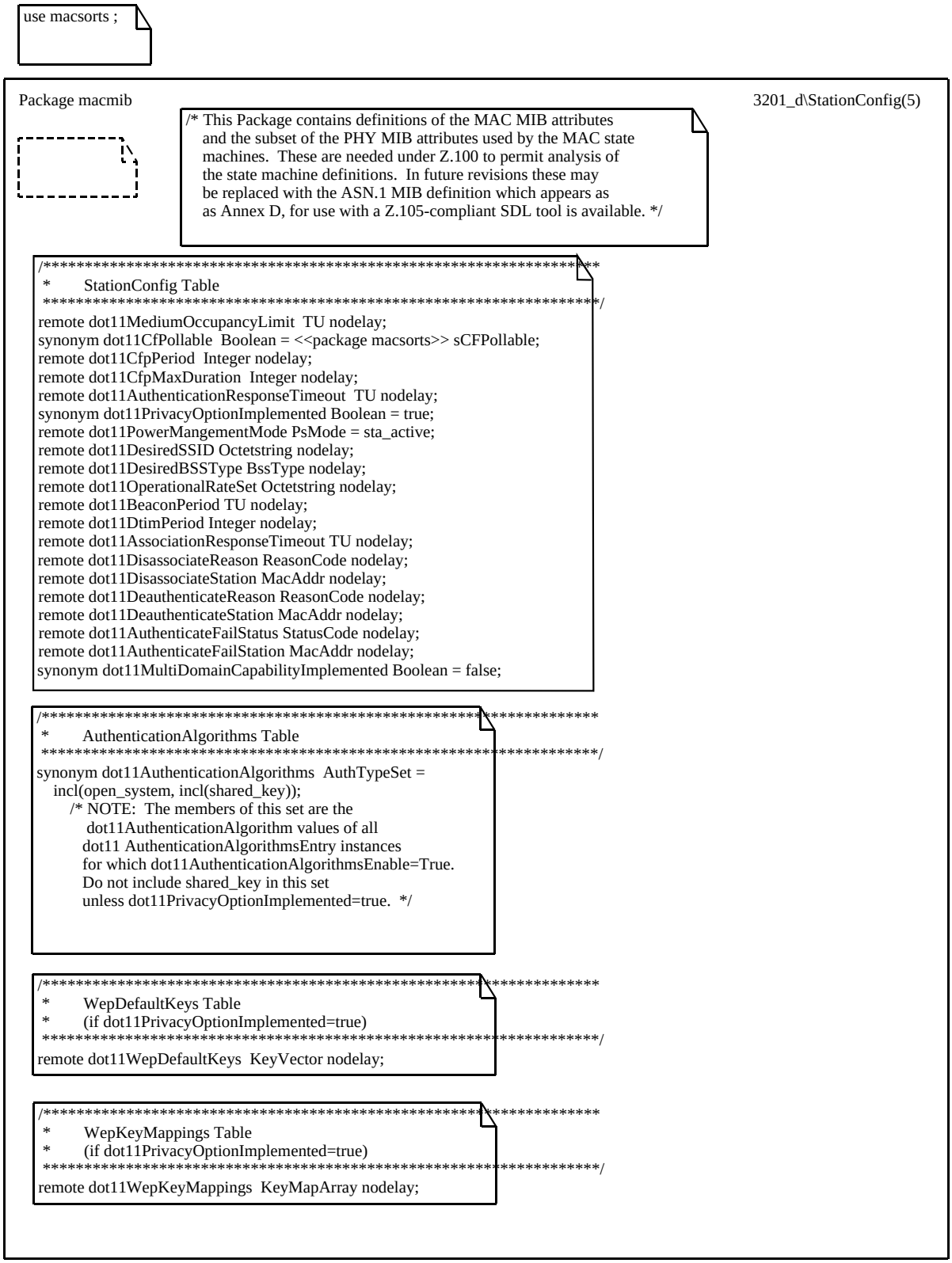
```
/*************************************************************************
 *       DS parameter set sort
 *************************************************************************/
newtype DsParms   inherits Octetstring   operators all;
 adding  operators
   curChannel  : DsParms -> Integer; /* Current Channel (1) */
   setCurChannel  : DsParms, Integer -> DsParms;
 axioms
   for all ds in DsParms(     for all i in Integer(
      curChannel(ds) == octetVal(ds(0));
      setCurChannel(ds, i) == mkOS(i); ));
endnewtype DsParms;
```

Copyright © 2007 IEEE. All rights reserved.

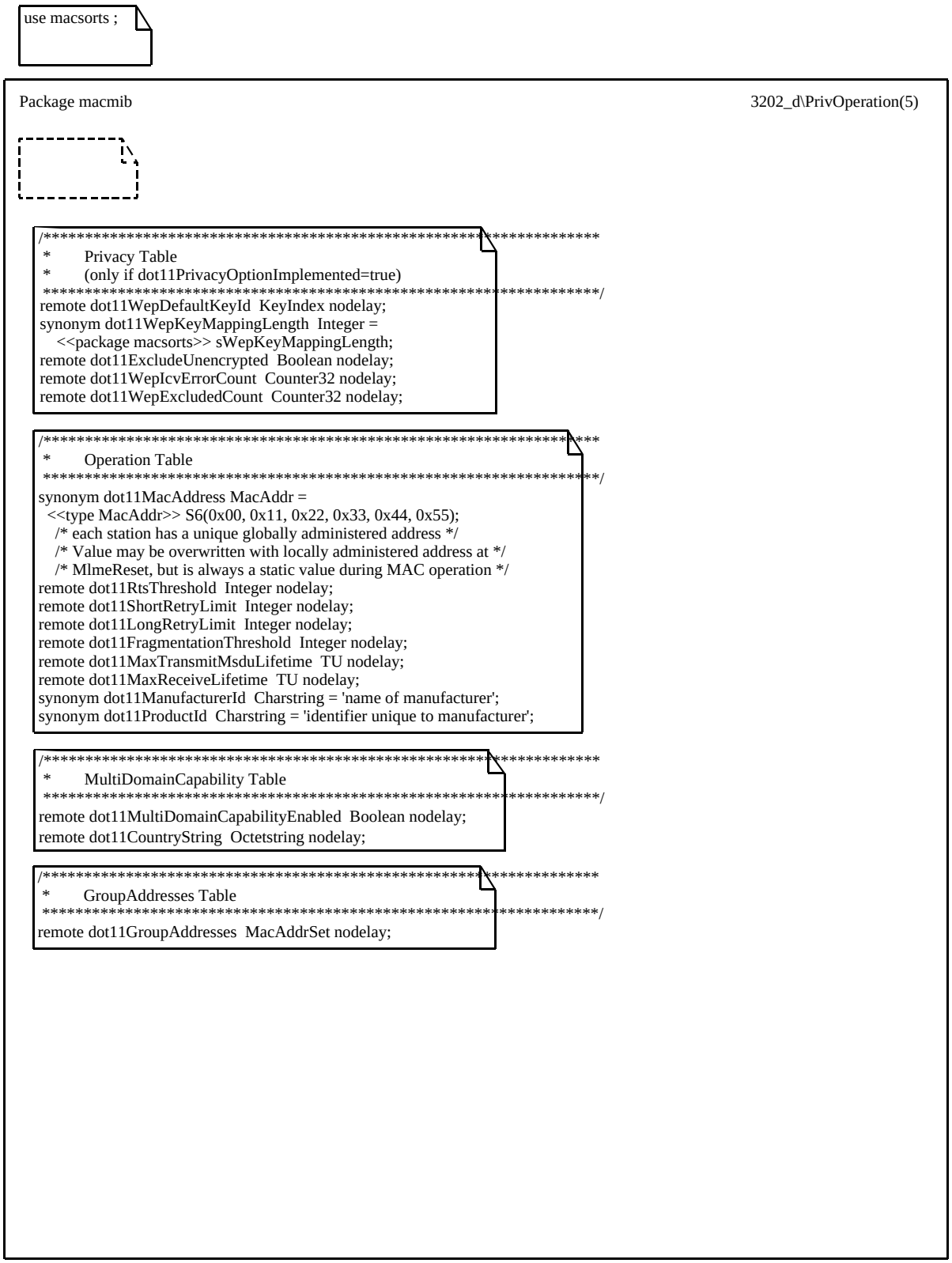




Copyright © 2007 IEEE. All rights reserved.

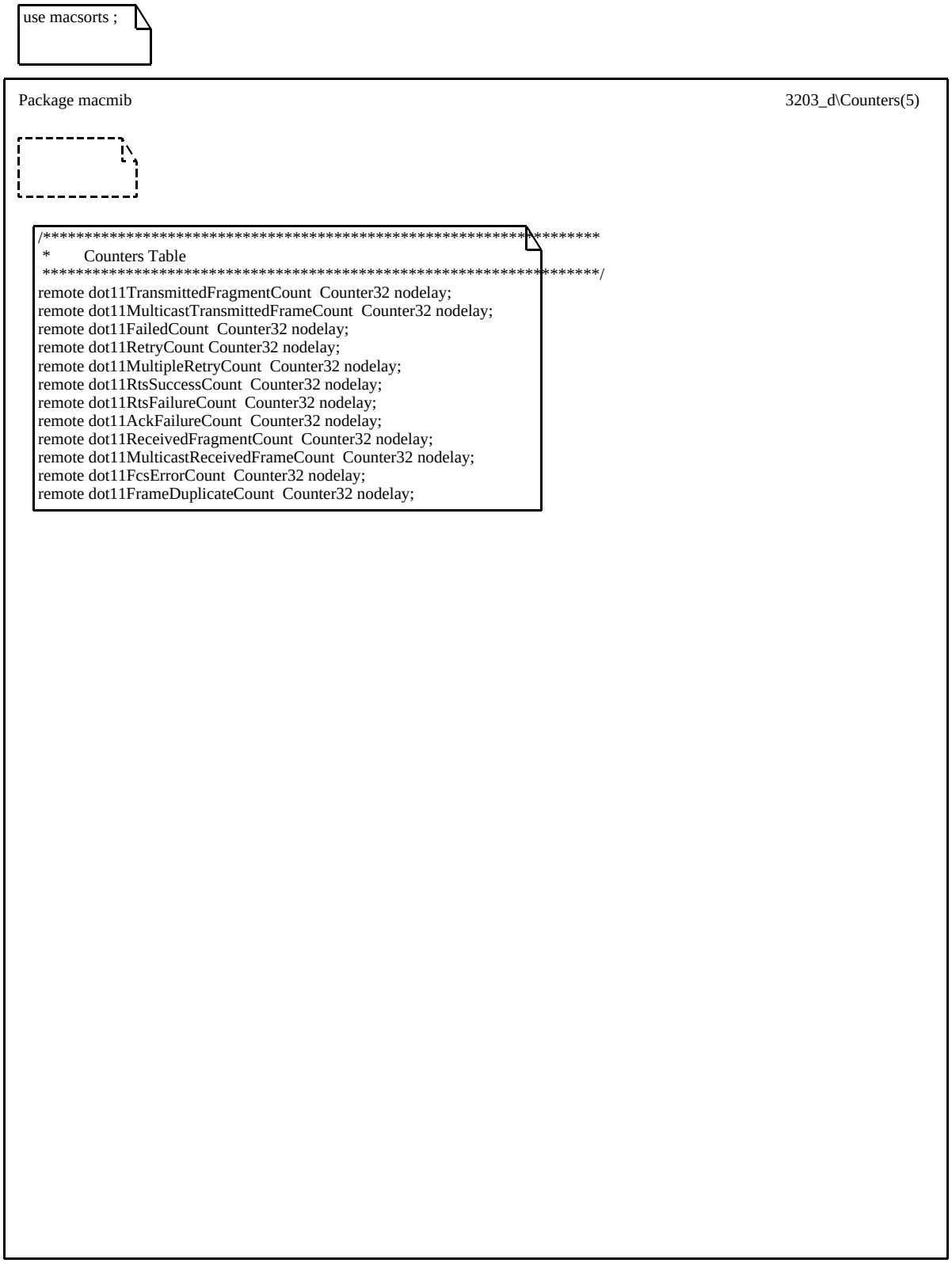



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

use macsorts ;

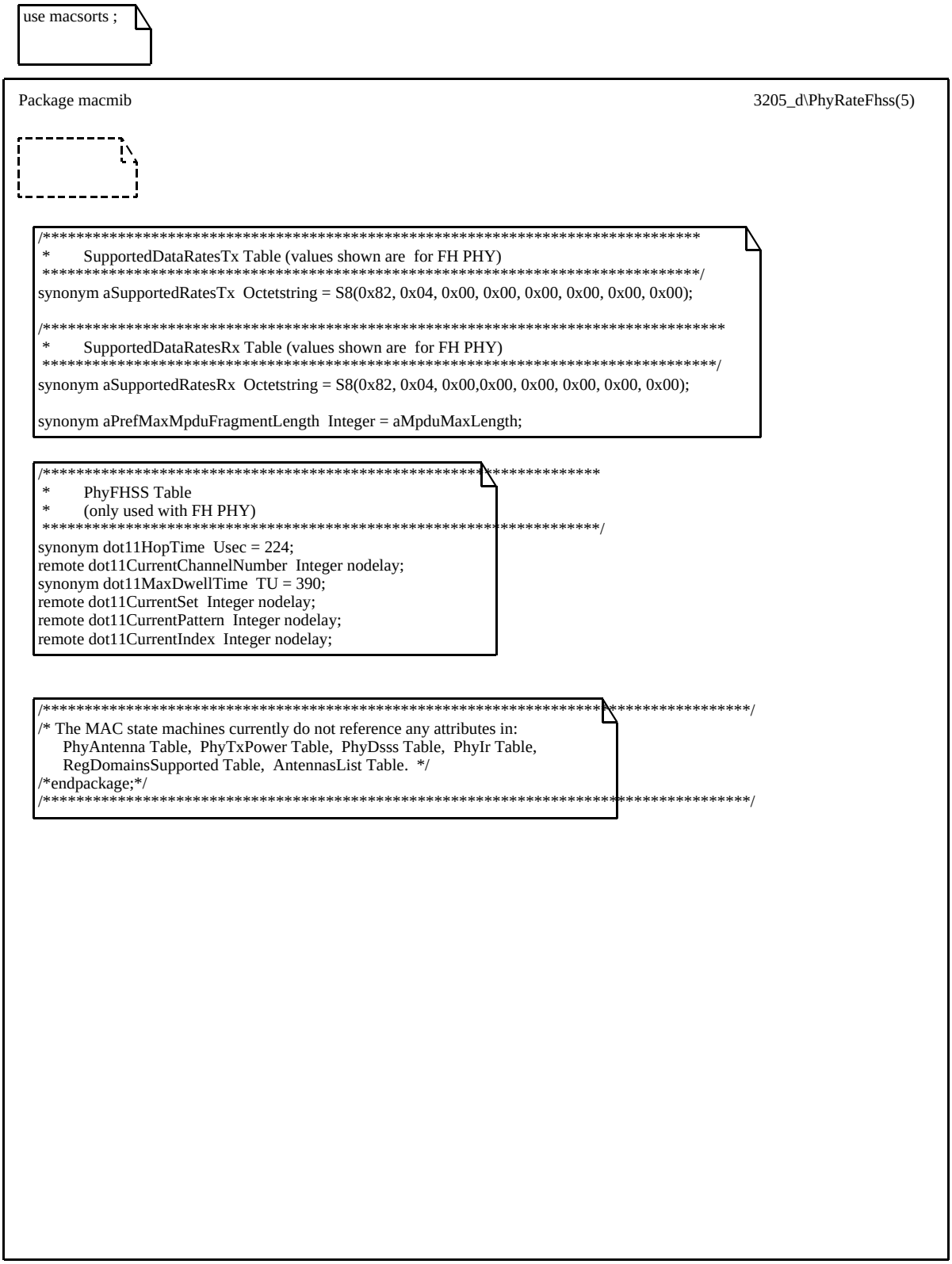
Package macmib

3204_e\PhyOperation(5)

```
/*****************************************************************
 *       PhyOperation Table
 *       (values shown are mostly for FH PHY)
 *****************************************************************/
synonym FHphy Integer = 01; /* enumerated dot11PHYType value */
synonym DSphy Integer = 02; /* enumerated dot11PHYType value */
synonym IRPhy Integer = 03; /* enumerated dot11PHYType value */
synonym dot11PHYType  Integer = FHphy;
remote dot11CurrentRegDomain  Integer nodelay;
synonym dot11TempType Integer = 01;

/*************************************************************************************
*  PhyCharacteristic Parameters (values shown are mostly for FH PHY )
*************************************************************************************/
/* NOTE:  The PhyCharacteristics are defined as synonyms because
   their values are static during MAC operation.  It is assumed
   that , during each initialization of MAC operation, current
   values for each of these parameters are obtained from the
   PHY using the PlmeCharacteristics primitive. */
remote procedure TxTime;  returns Integer;
synonym aSlotTime  Usec = (aCcaTime + aRxTxTurnaroundTime +
   aAirPropagationTime + aMacProcessingTime);
synonym aCcaTime  Usec = 27;
synonym aRxTxTurnaroundTime Usec = (aTxPlcpDelay + aRxTxSwitchTime +
   aTxRampOnTime + aTxRfDelay);
synonym aTxPlcpDelay  Usec = 1;
synonym aRxTxSwitchTime  Usec = 10;
synonym aTxRampOnTime  Usec = 8;
synonym aTxRfDelay  Usec = 1;
synonym aSifsTime Usec = (aRxRfDelay + aRxPlcpDelay +
   aMacProcessingTime + aRxTxTurnaroundTime);
synonym aRxRfDelay  Usec = 4;
synonym aRxPlcpDelay  Usec = 2;
synonym aMacProcessingTime  Usec = 2;
synonym aTxRampOffTime  Usec = 8;
synonym aPreambleLength  Usec = 96;
synonym aPlcpHeaderLength  Usec = 32;
synonym aMpduMaxLength Integer = 4095;
synonym aAirPropagationTime  Usec = 1;
synonym aCWmax  Integer = 1023;
synonym aCWmin  Integer = 15;
```


Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

## C.3 State machines for MAC STAs

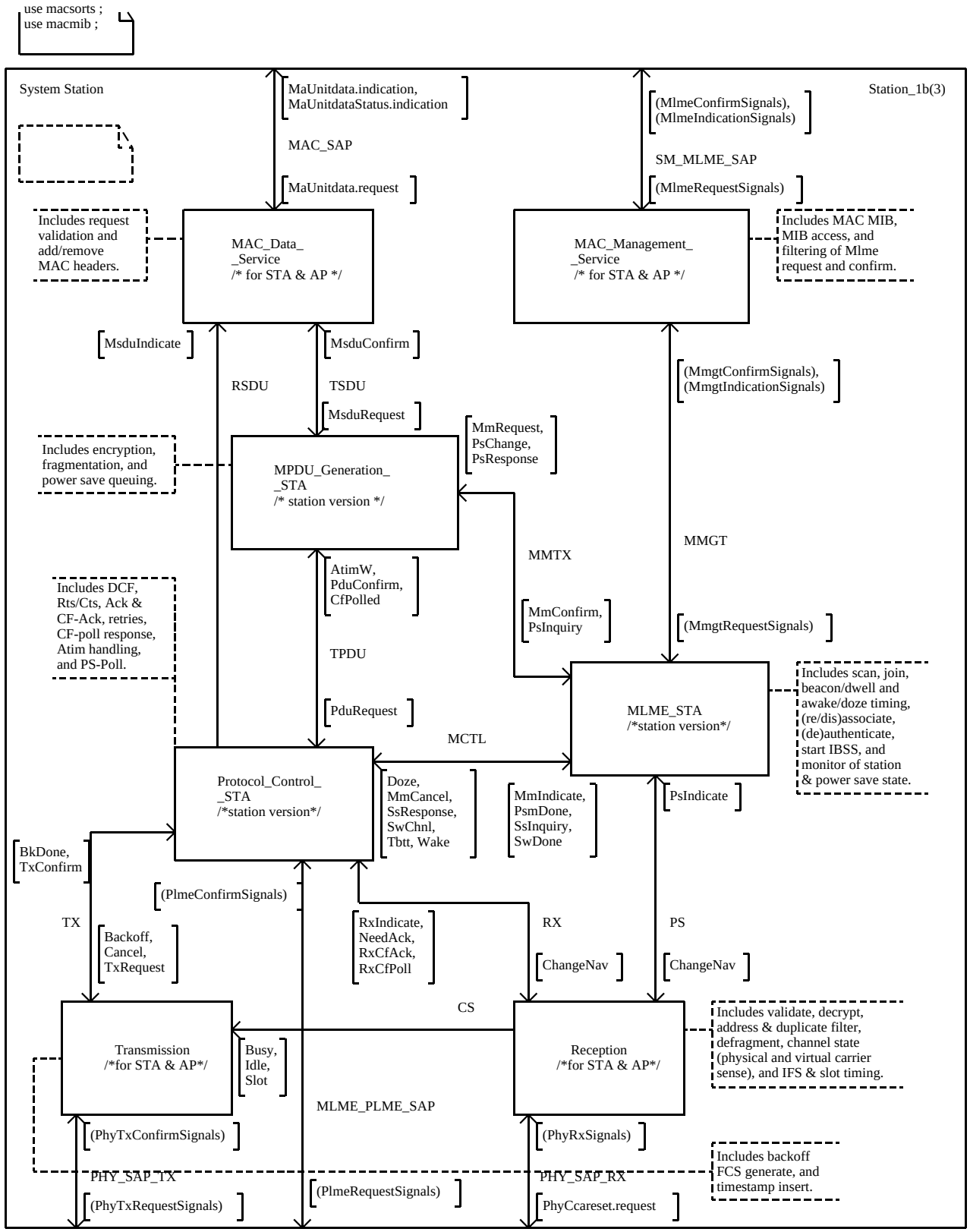
The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 STA. Many aspects of STA operation also apply to AP operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment /* for STA & AP */ below the block or process name. Blocks and processes specific to STA operation are identifiable by the SDL comment /* station version */ below the block or process name. The definitions of all blocks and processes referenced in the STA system specification appear in C.3.

The remainder of C.3 is the formal description, in SDL/GR, of an IEEE 802.11 STA.

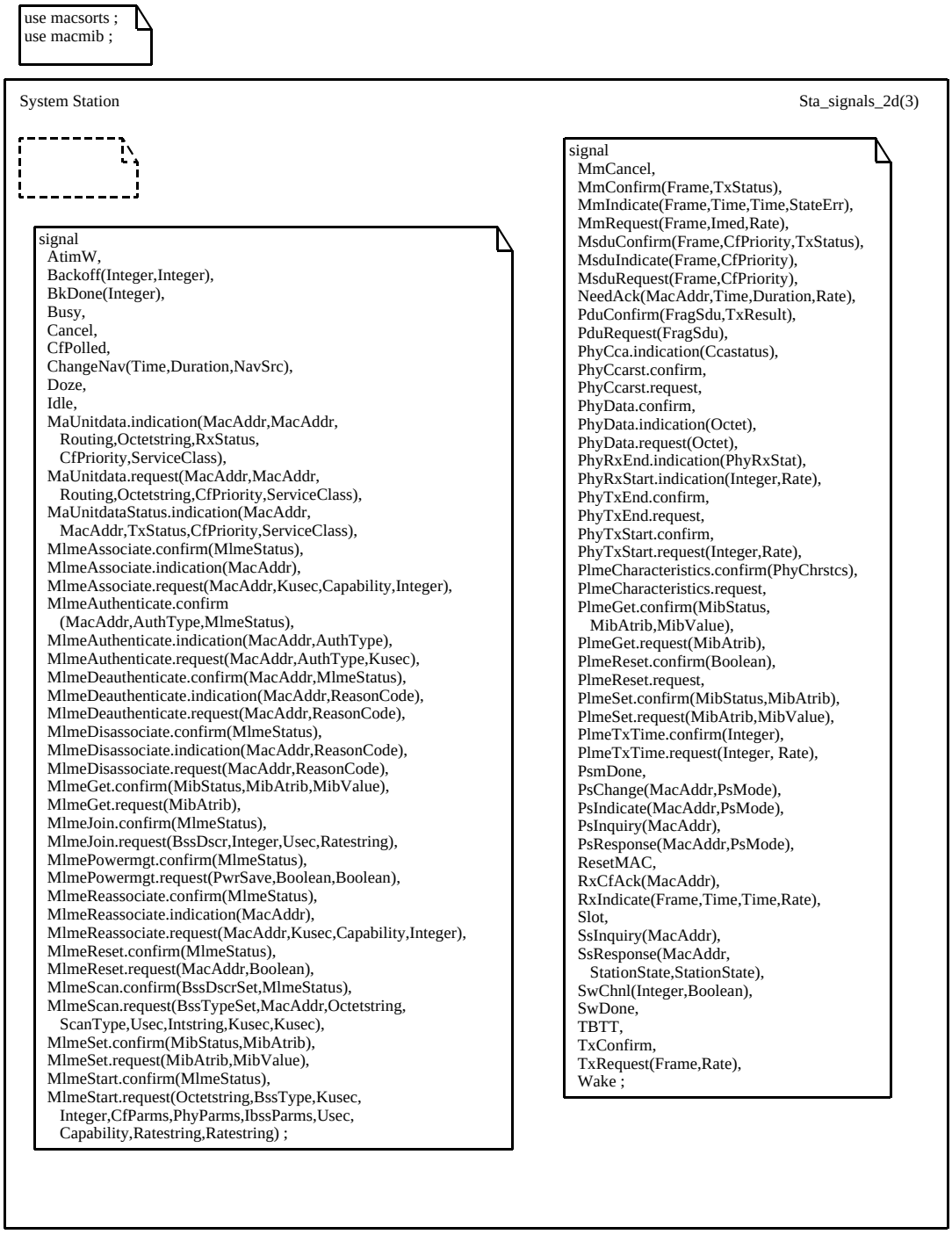
This subclause describes the security behavior of only 8.2.1 and 8.2.2.

This subclause does not describe the behavior of a STA with QoS facility.


Copyright © 2007 IEEE. All rights reserved.

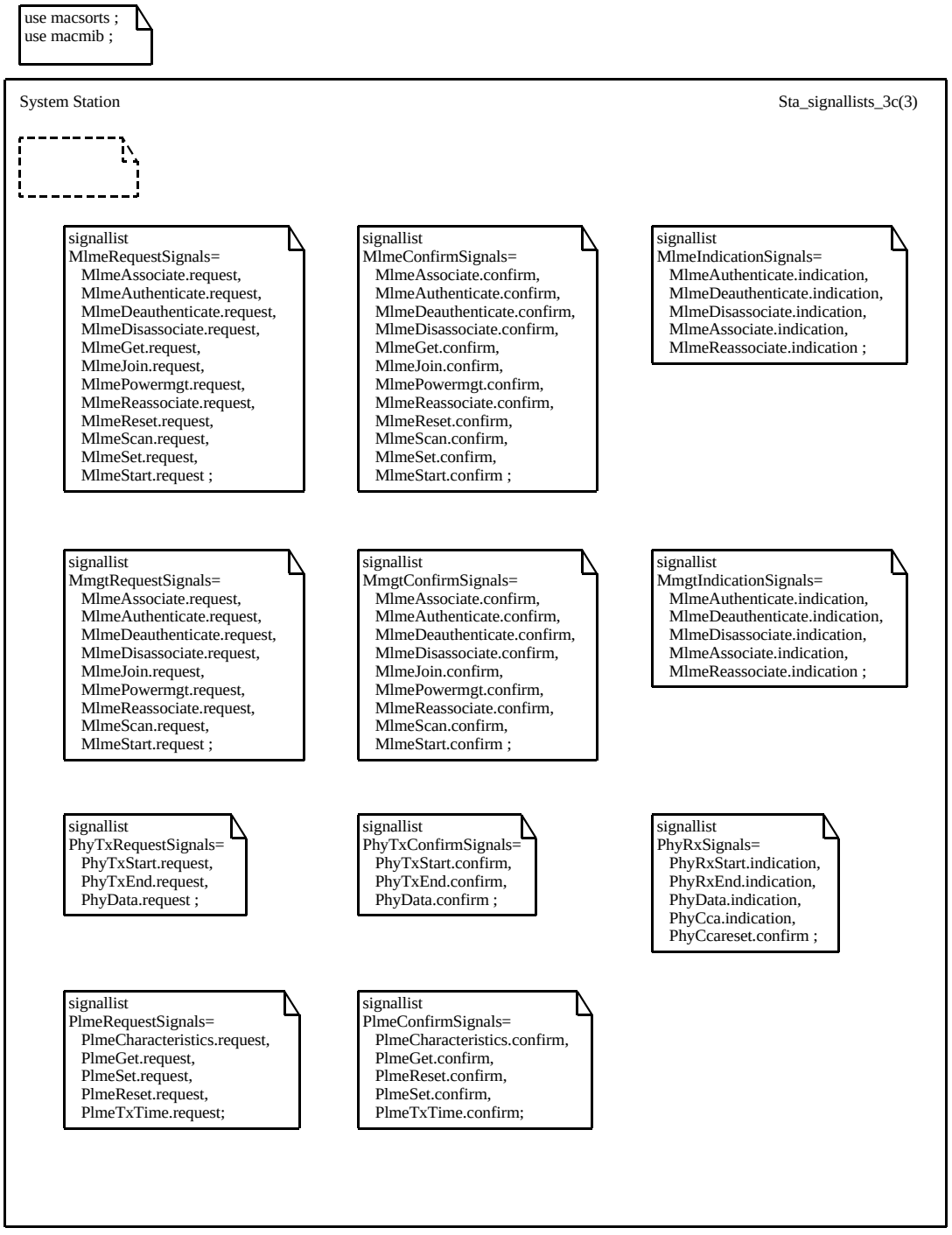



Copyright © 2007 IEEE. All rights reserved.

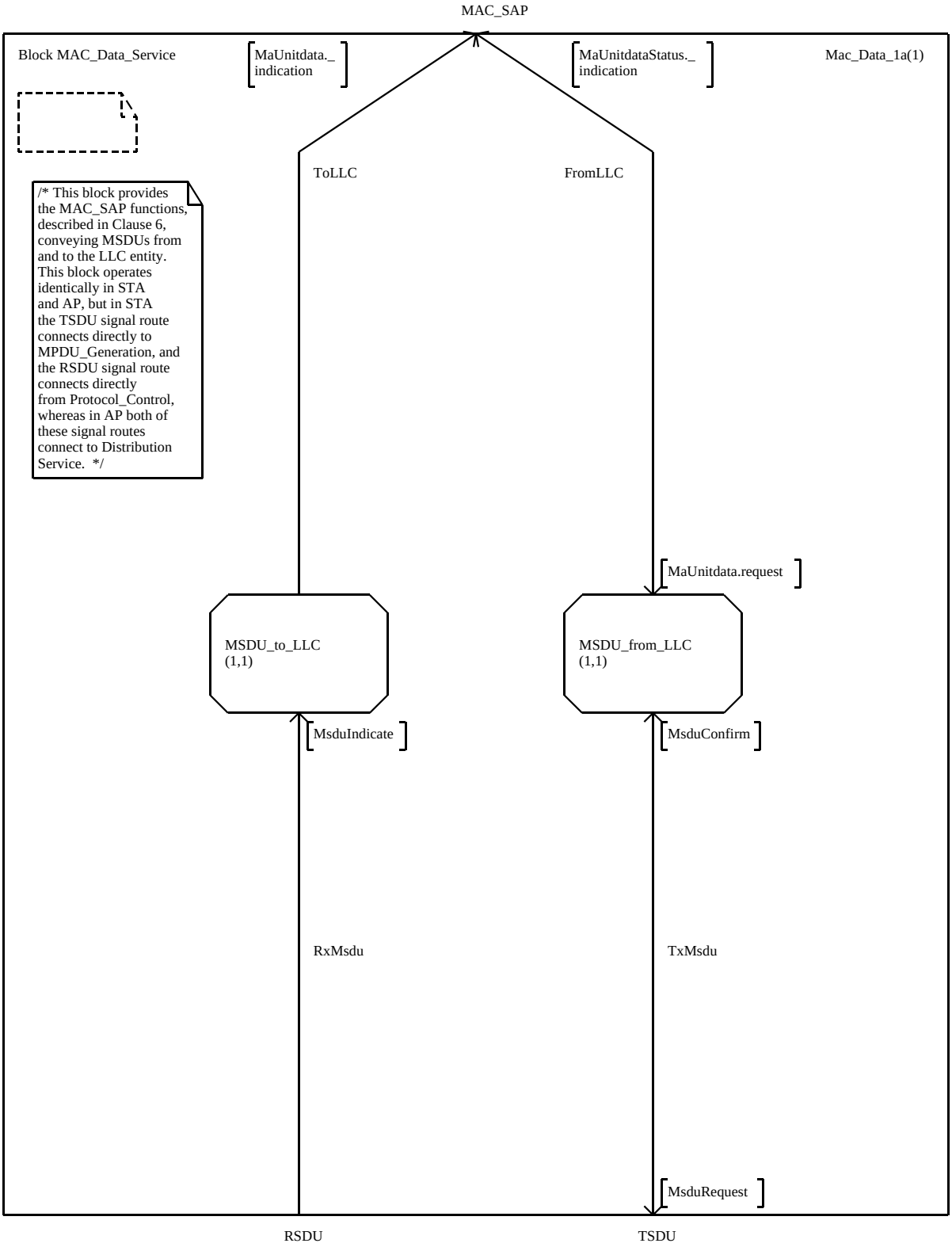




Copyright © 2007 IEEE. All rights reserved.

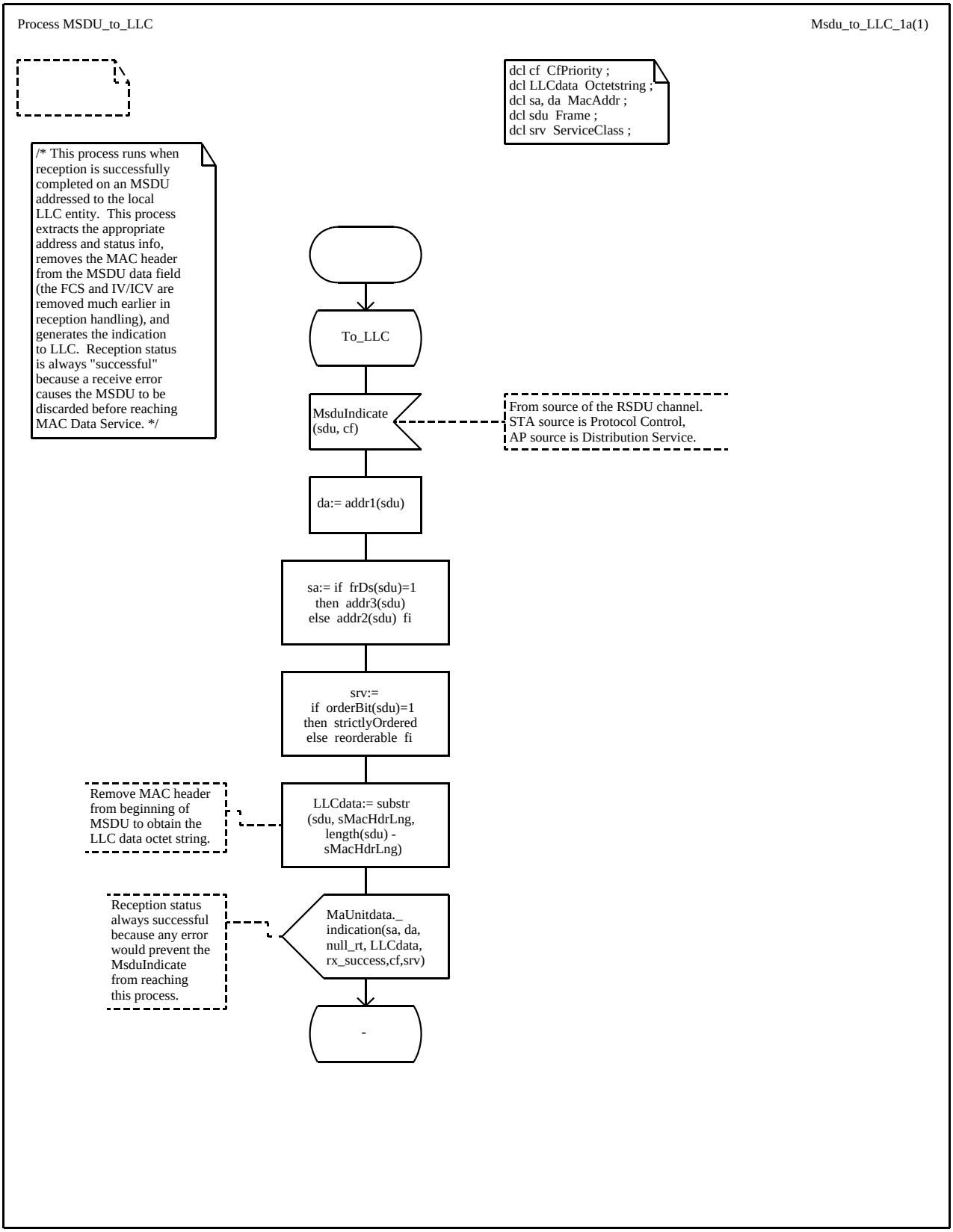



Copyright © 2007 IEEE. All rights reserved.

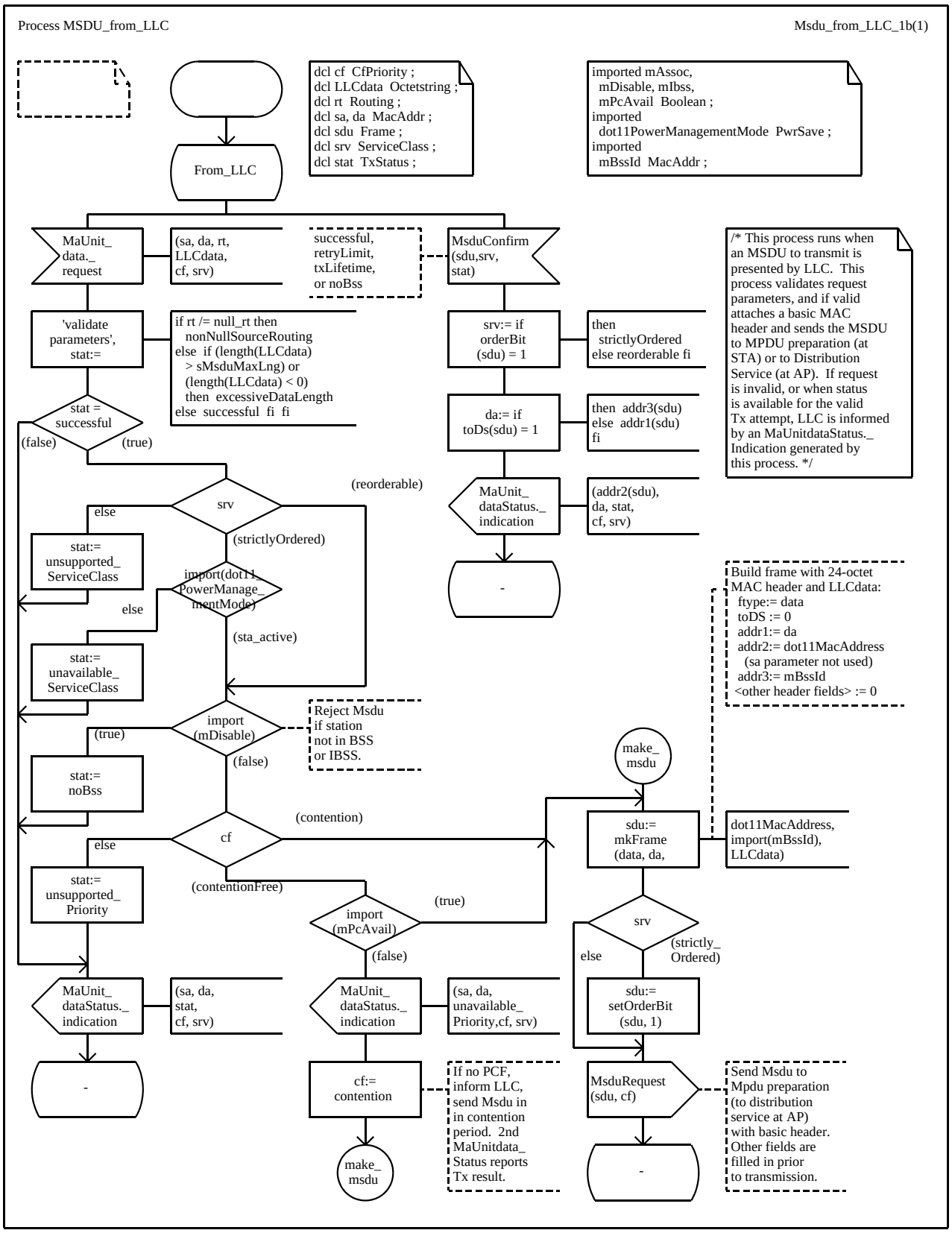




Copyright © 2007 IEEE. All rights reserved.

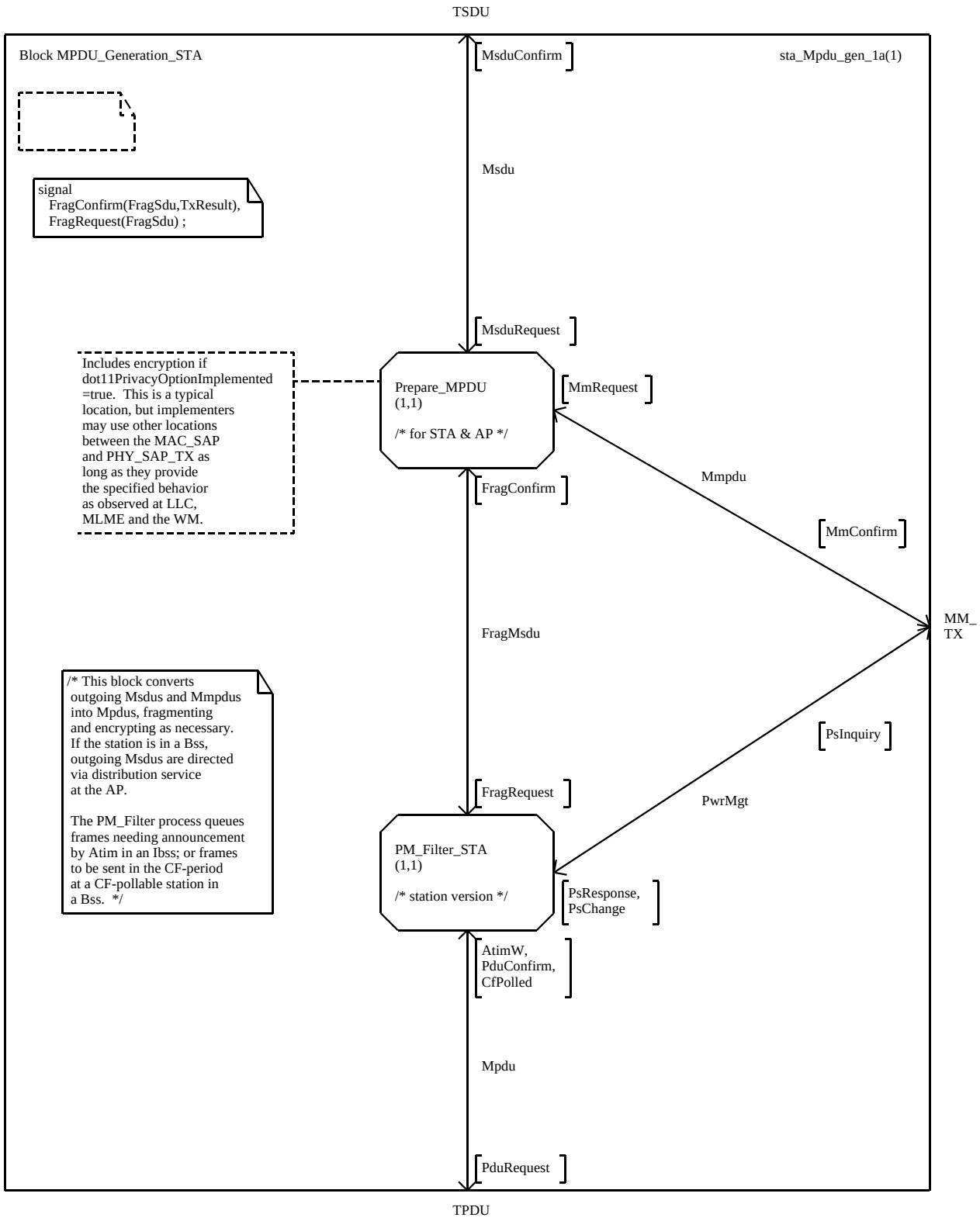



Copyright © 2007 IEEE. All rights reserved.

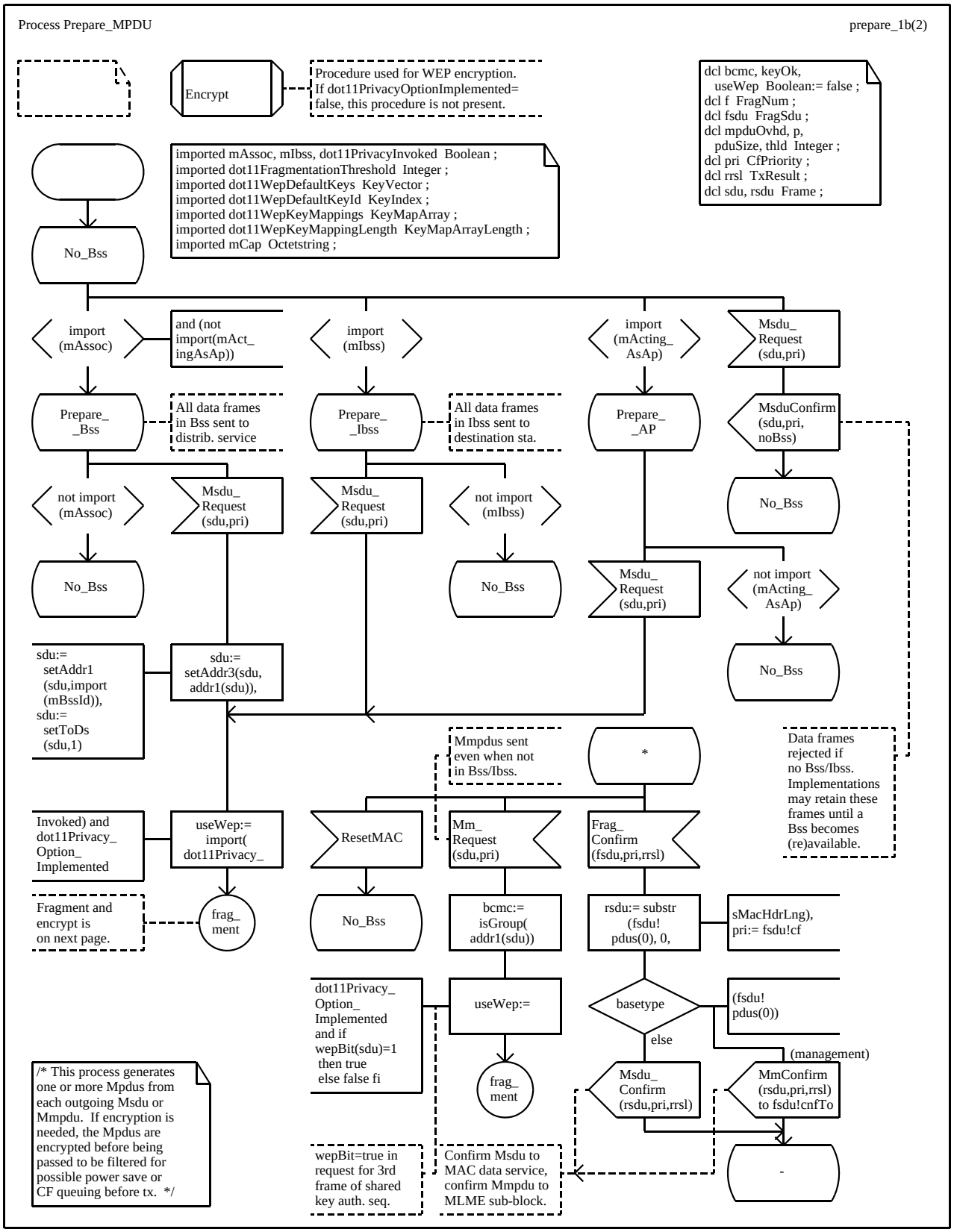




Copyright © 2007 IEEE. All rights reserved.

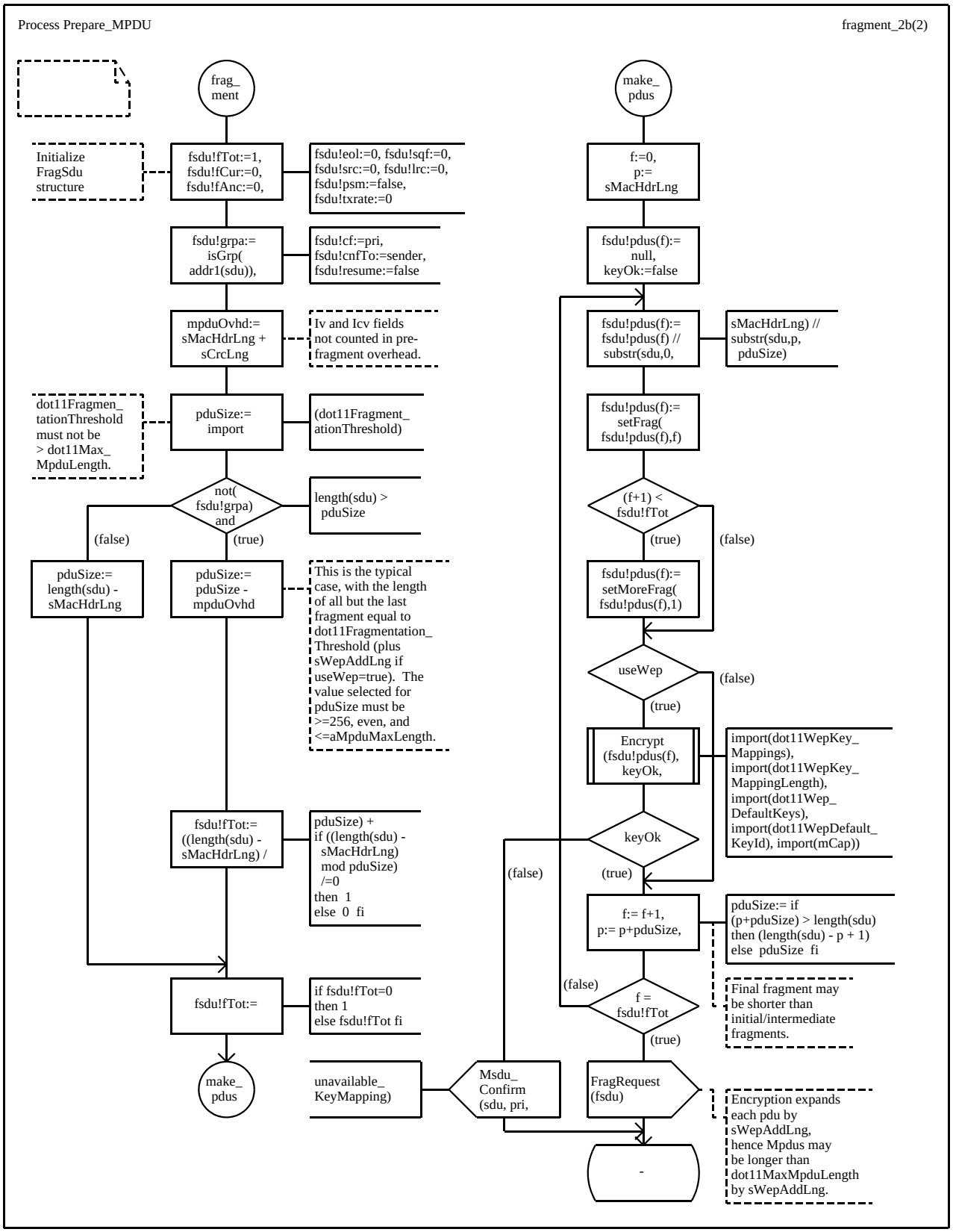



Copyright © 2007 IEEE. All rights reserved.

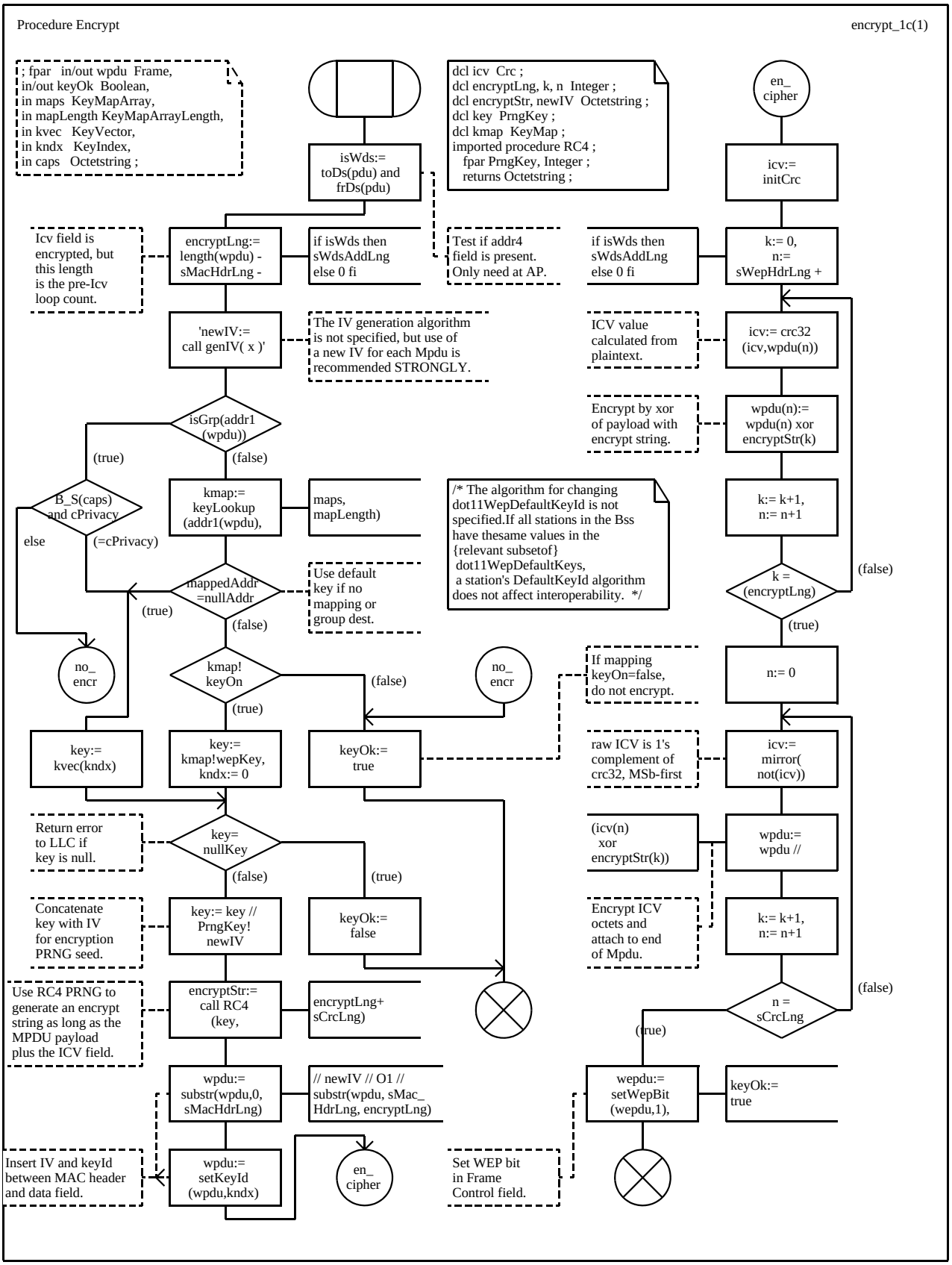




Copyright © 2007 IEEE. All rights reserved.

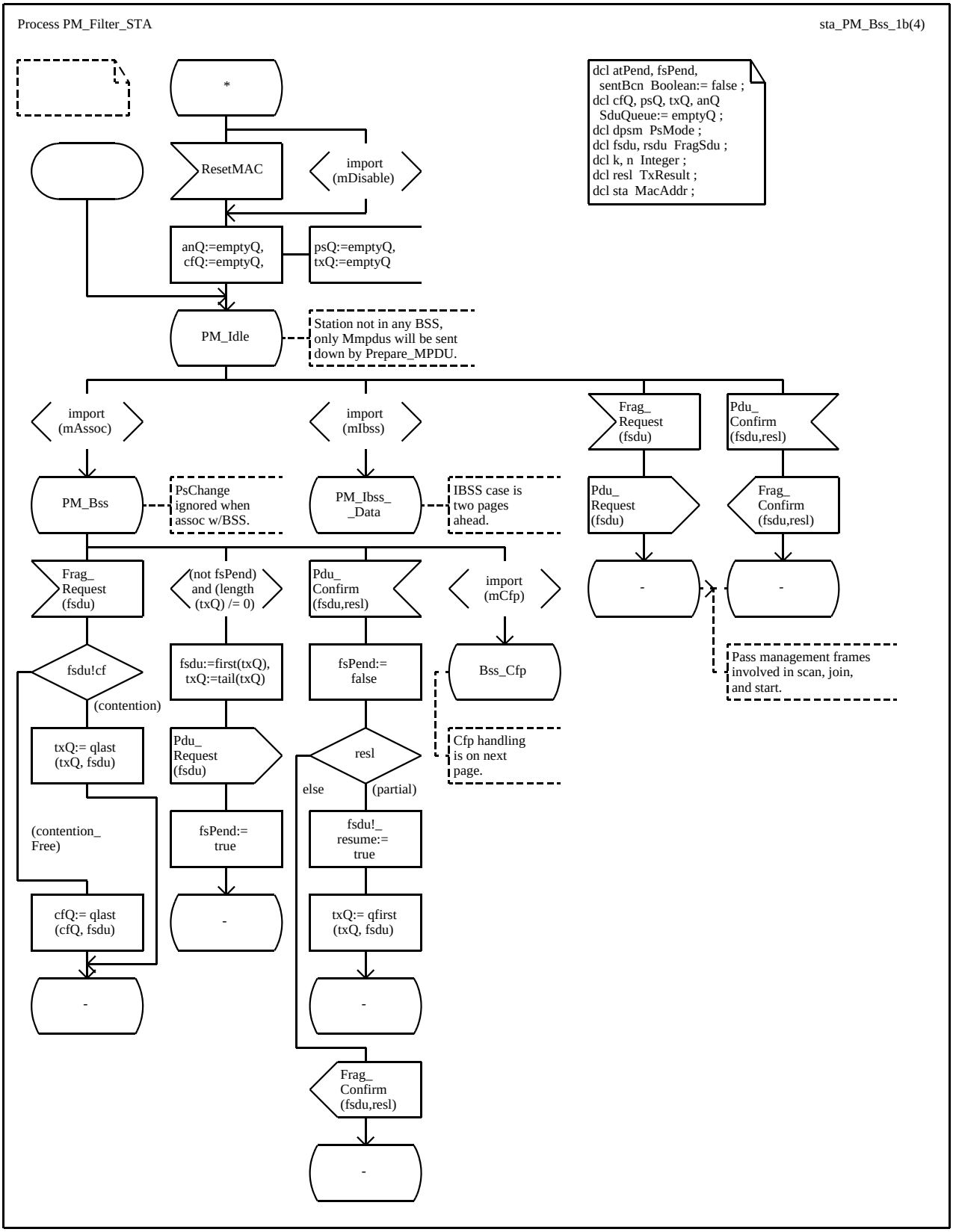



Copyright © 2007 IEEE. All rights reserved.

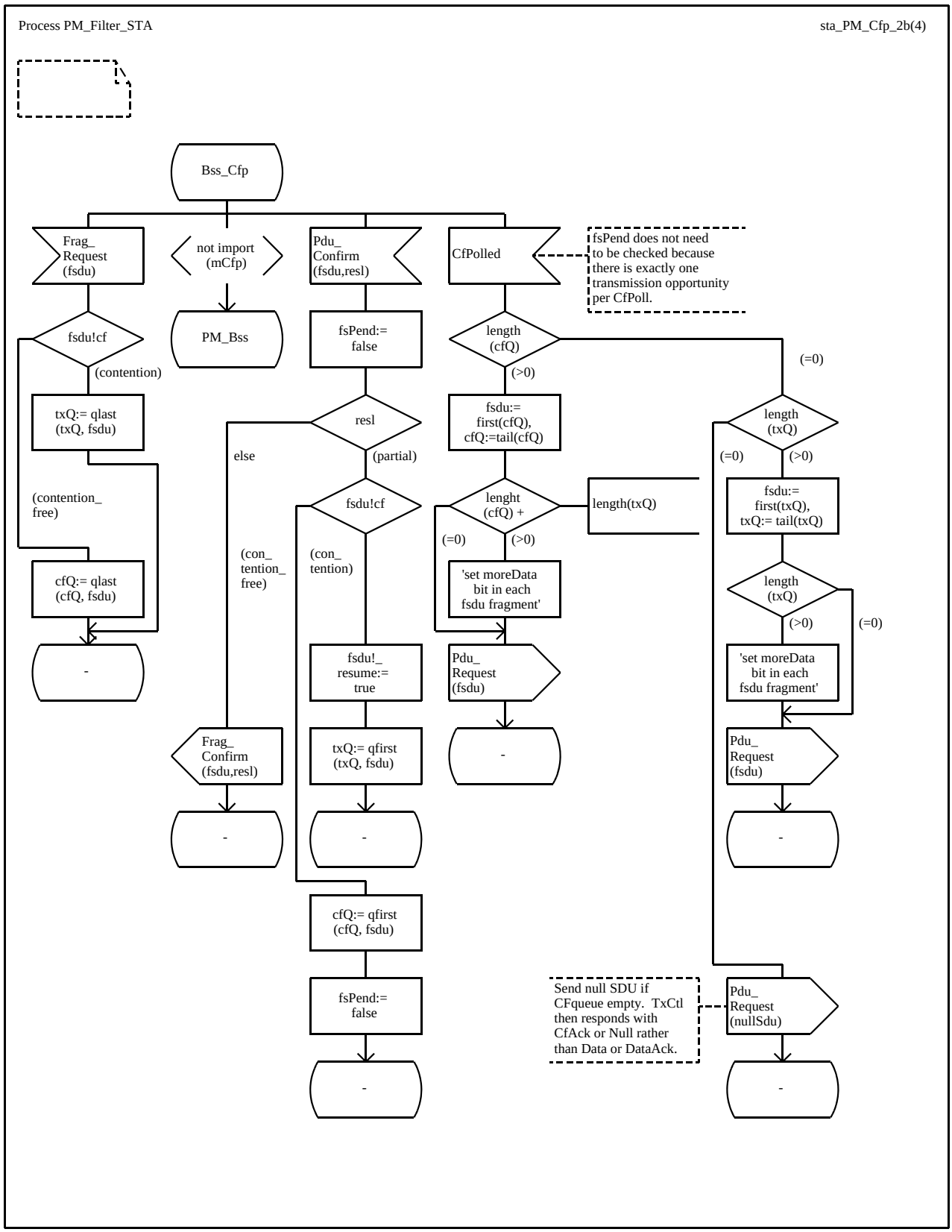




Copyright © 2007 IEEE. All rights reserved.

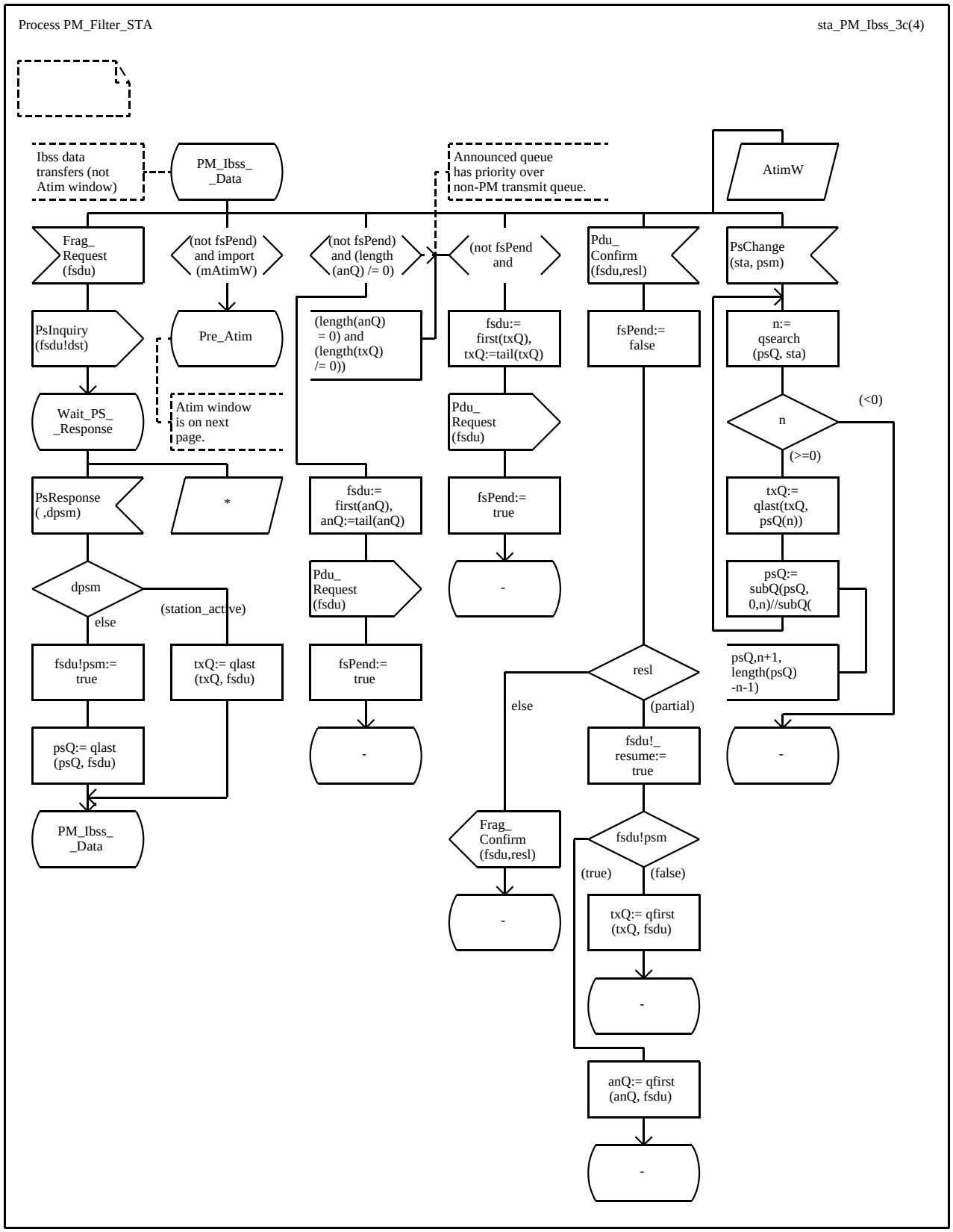



Copyright © 2007 IEEE. All rights reserved.

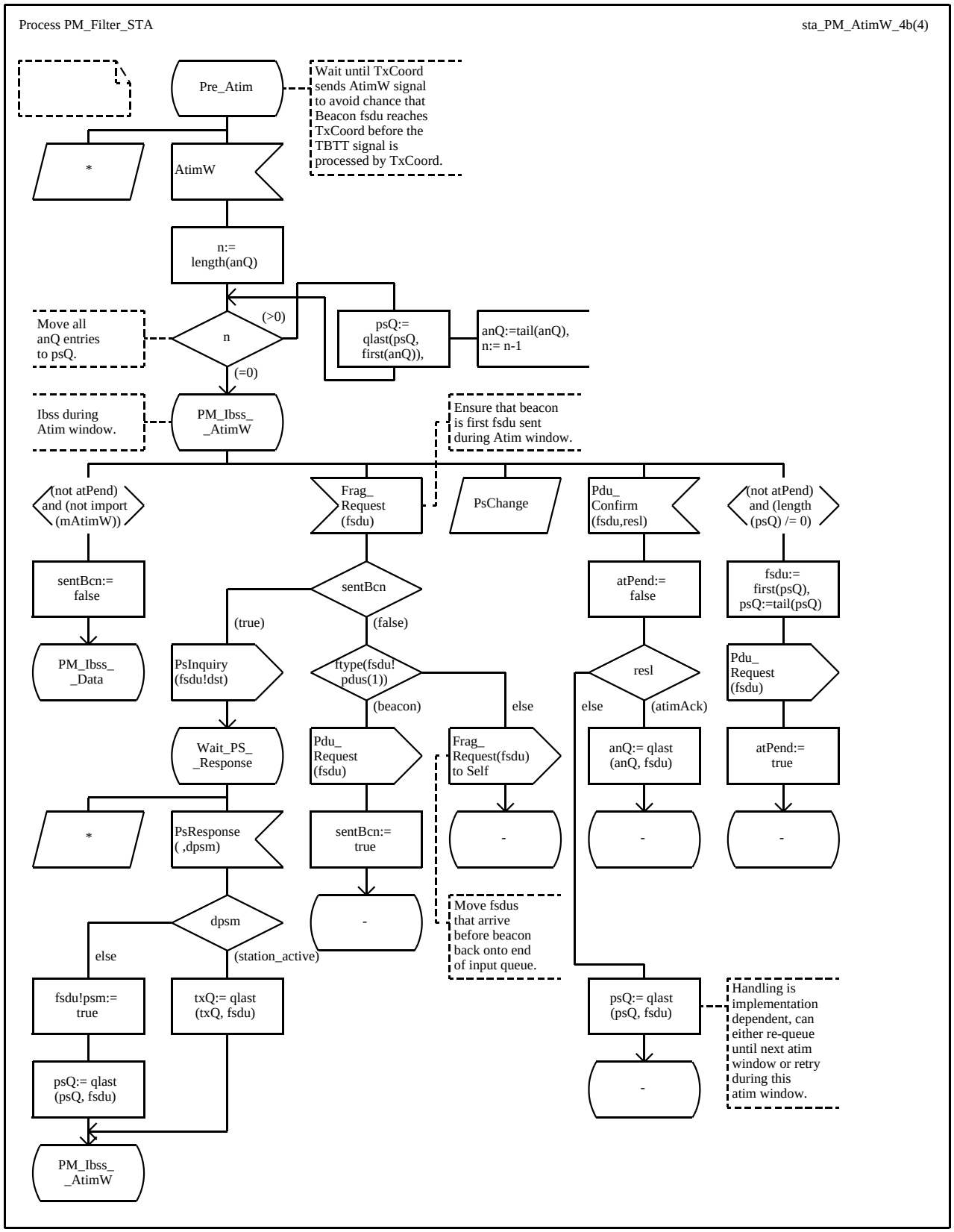




Copyright © 2007 IEEE. All rights reserved.

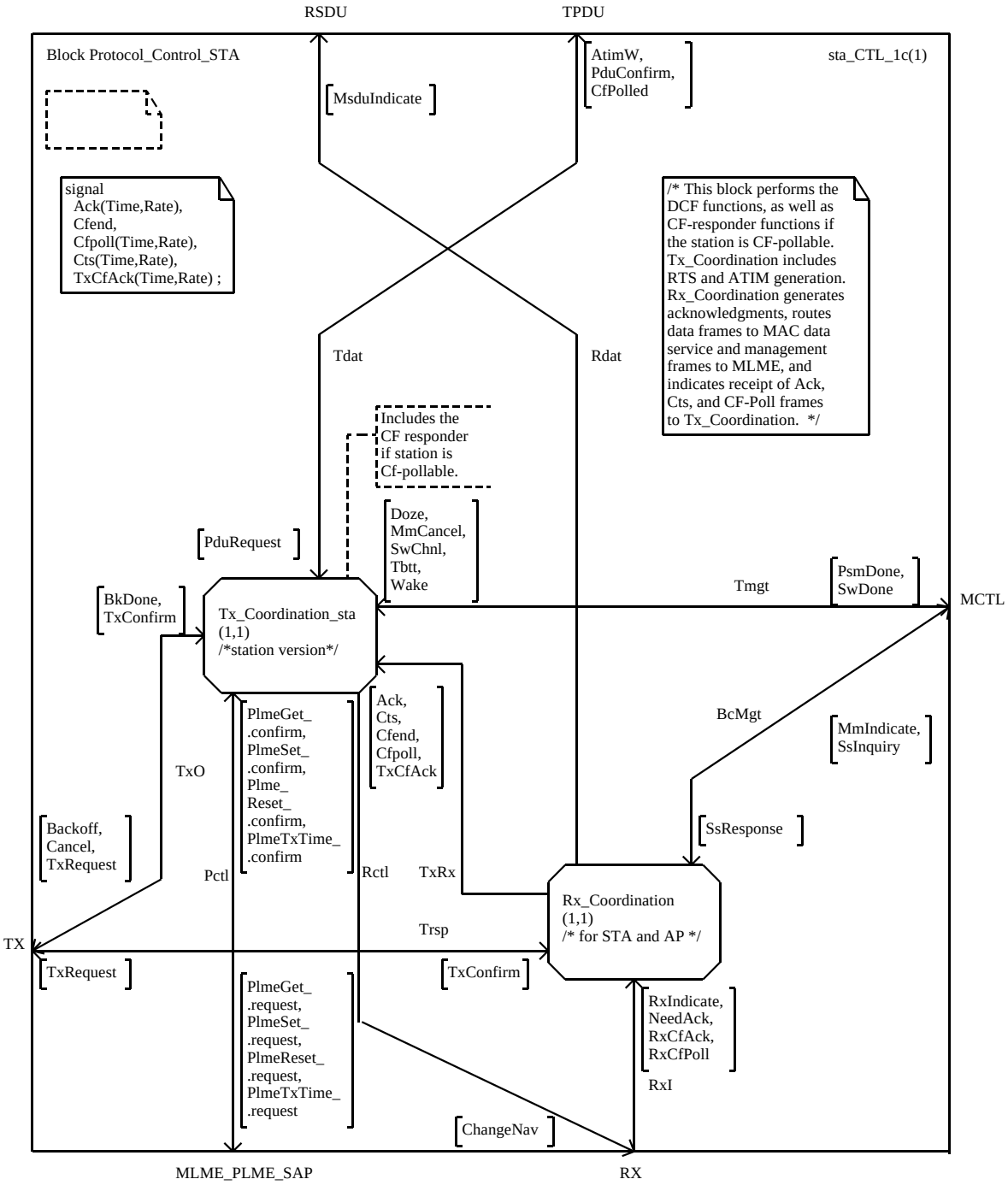



Copyright © 2007 IEEE. All rights reserved.

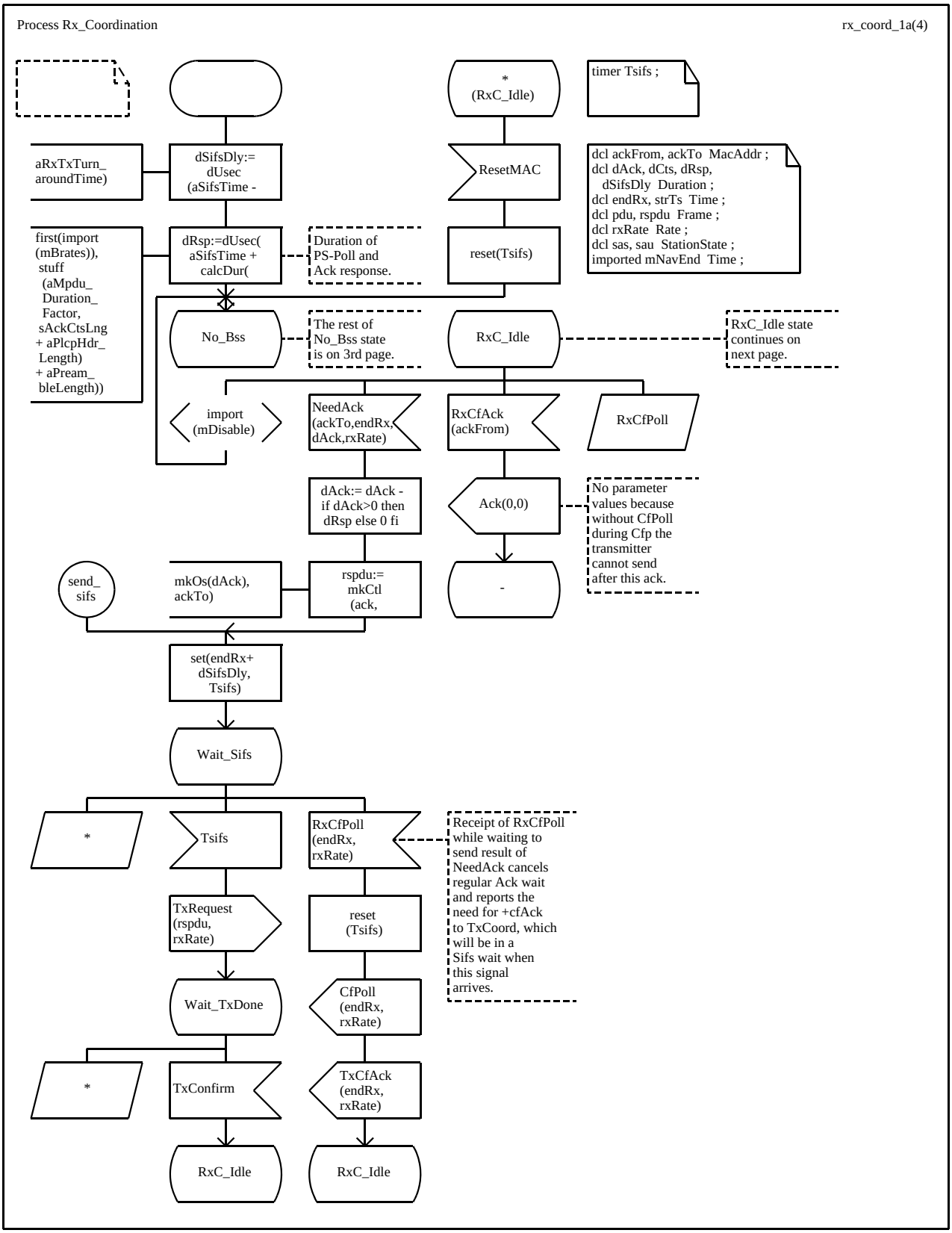




Copyright © 2007 IEEE. All rights reserved.

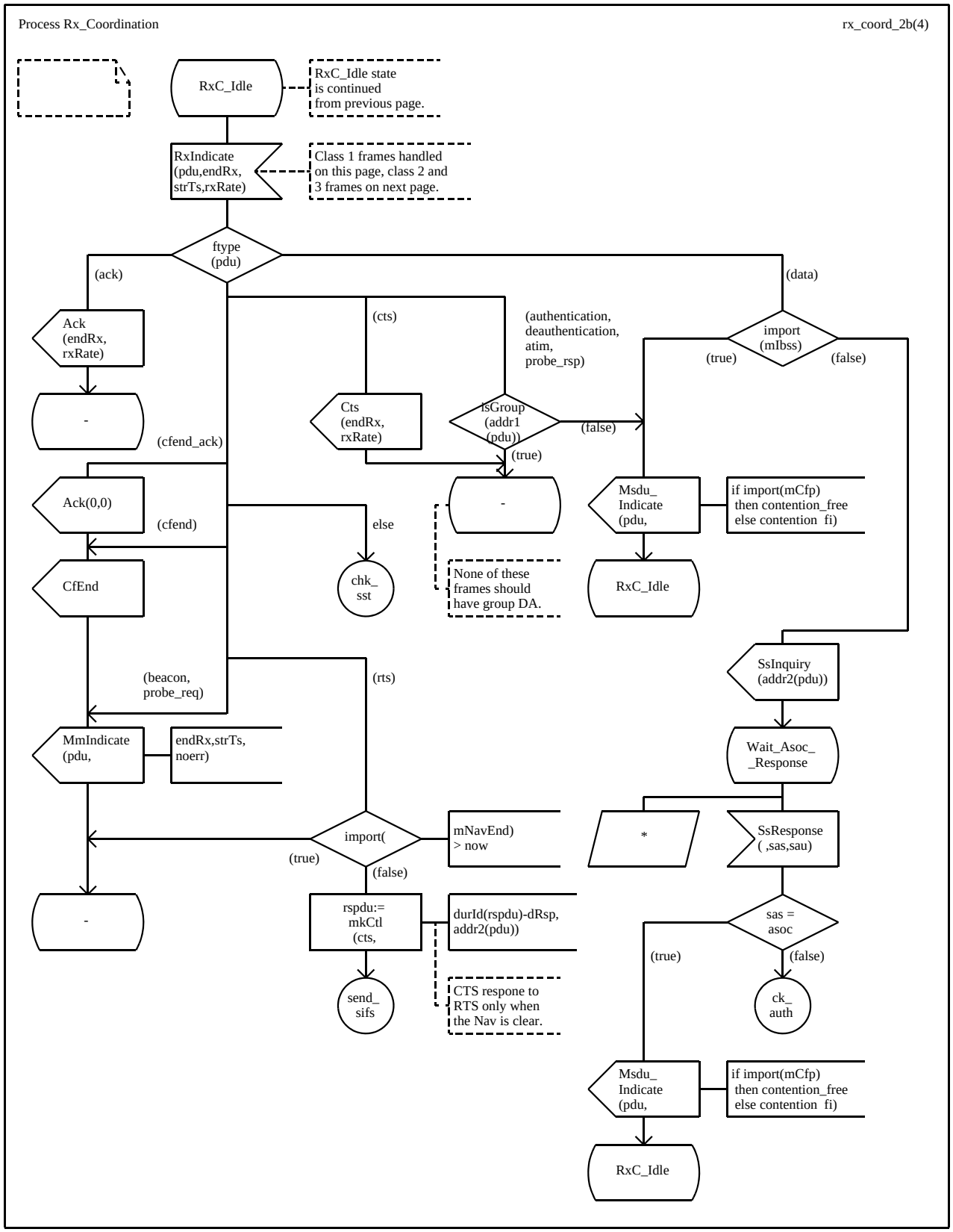



Copyright © 2007 IEEE. All rights reserved.

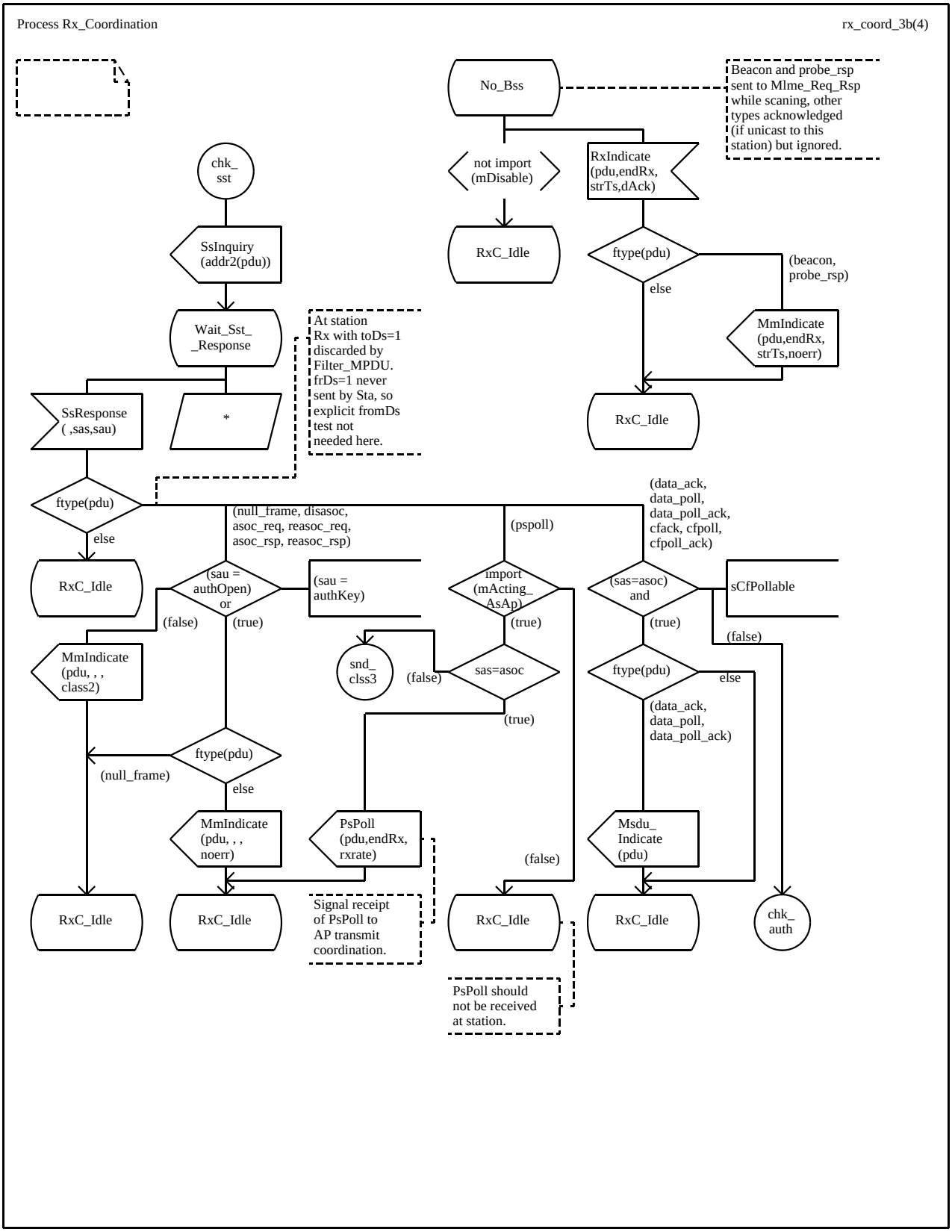




Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

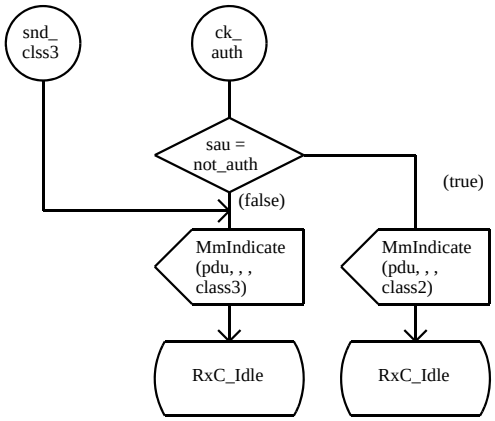




Copyright © 2007 IEEE. All rights reserved.

Process Rx_Coordination

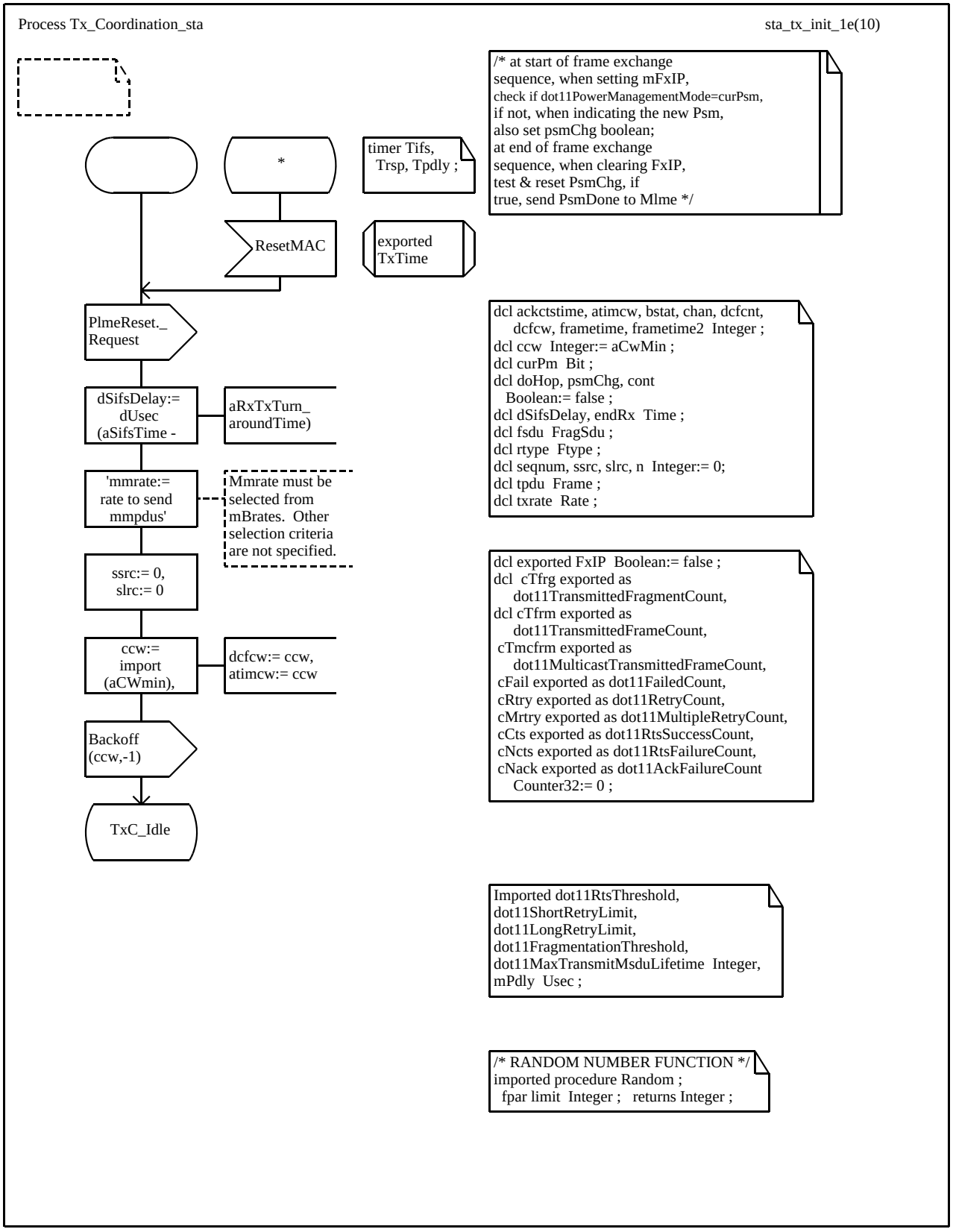
rx_coord_3.1a(4)




Copyright © 2007 IEEE. All rights reserved.

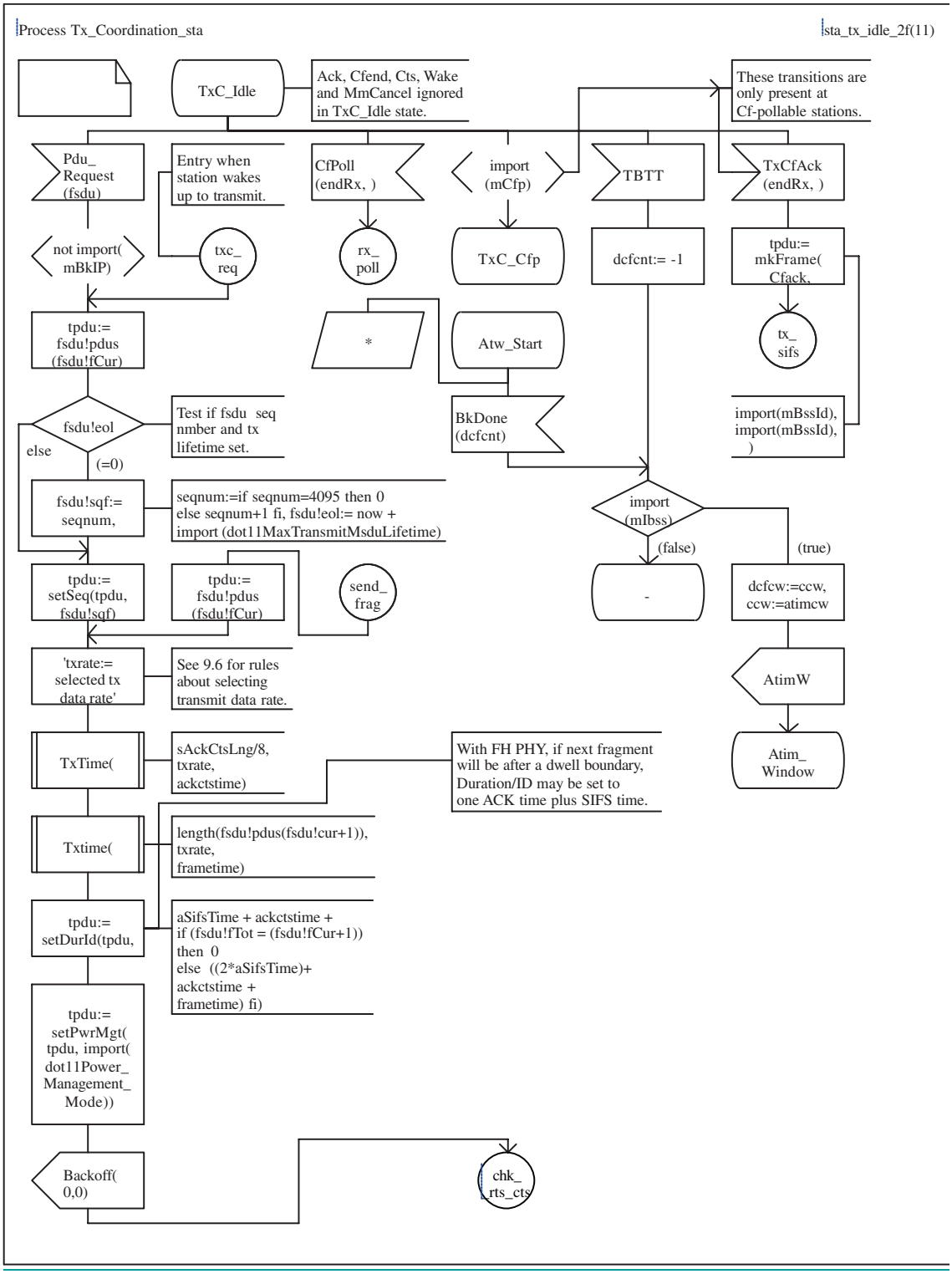


Samsung Ex. 1044, Page 903 of 1232

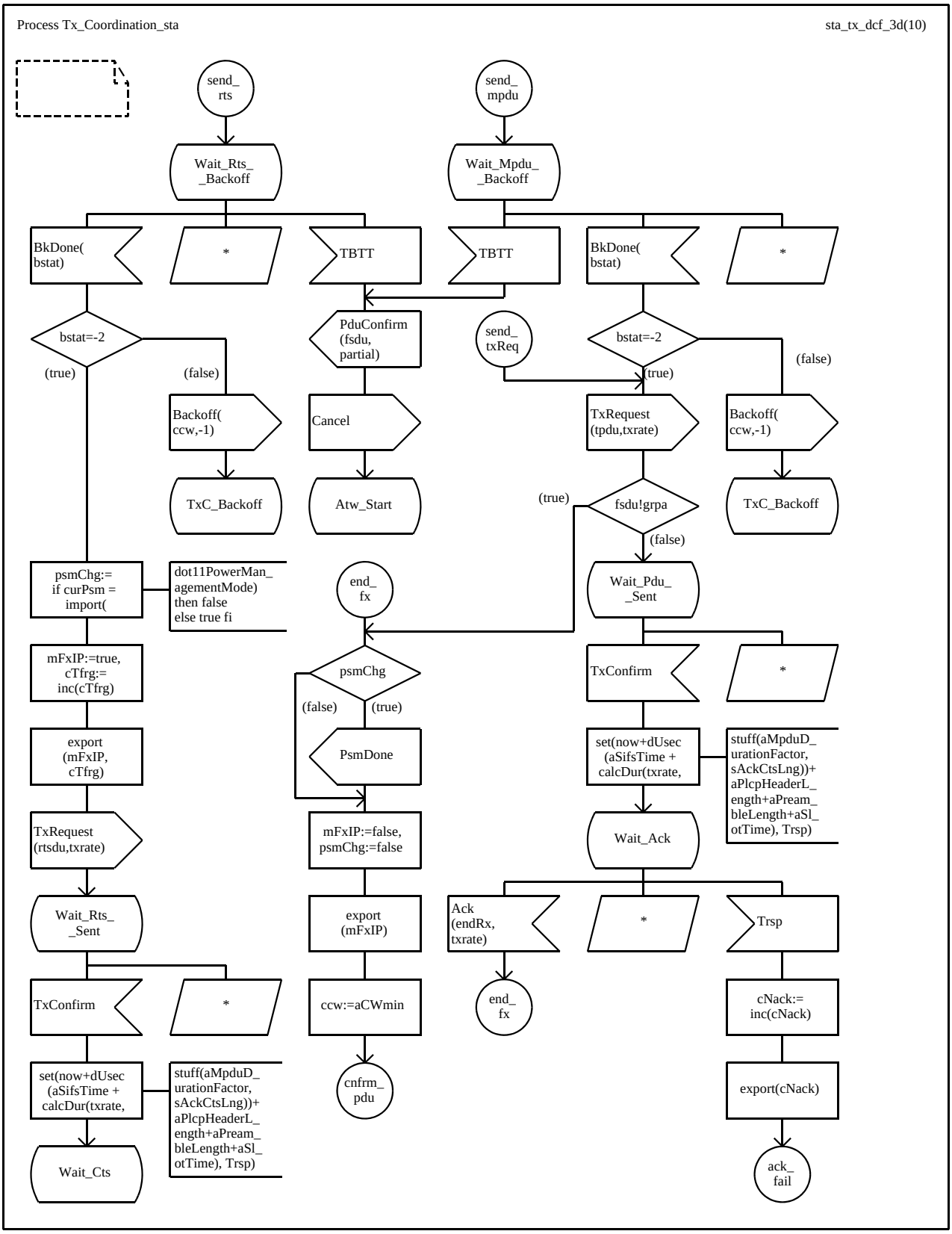




Copyright © 2007 IEEE. All rights reserved.

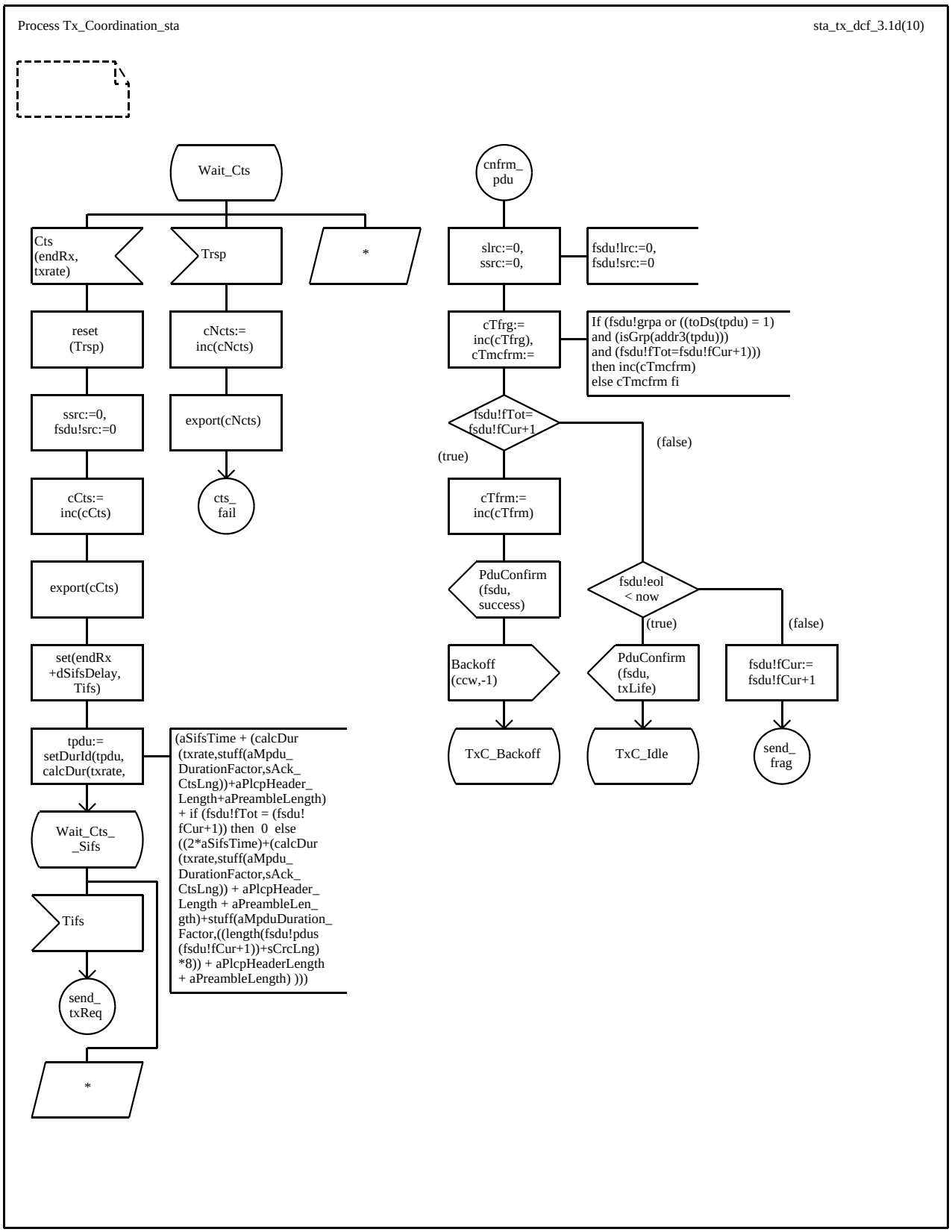



Copyright © 2007 IEEE. All rights reserved.

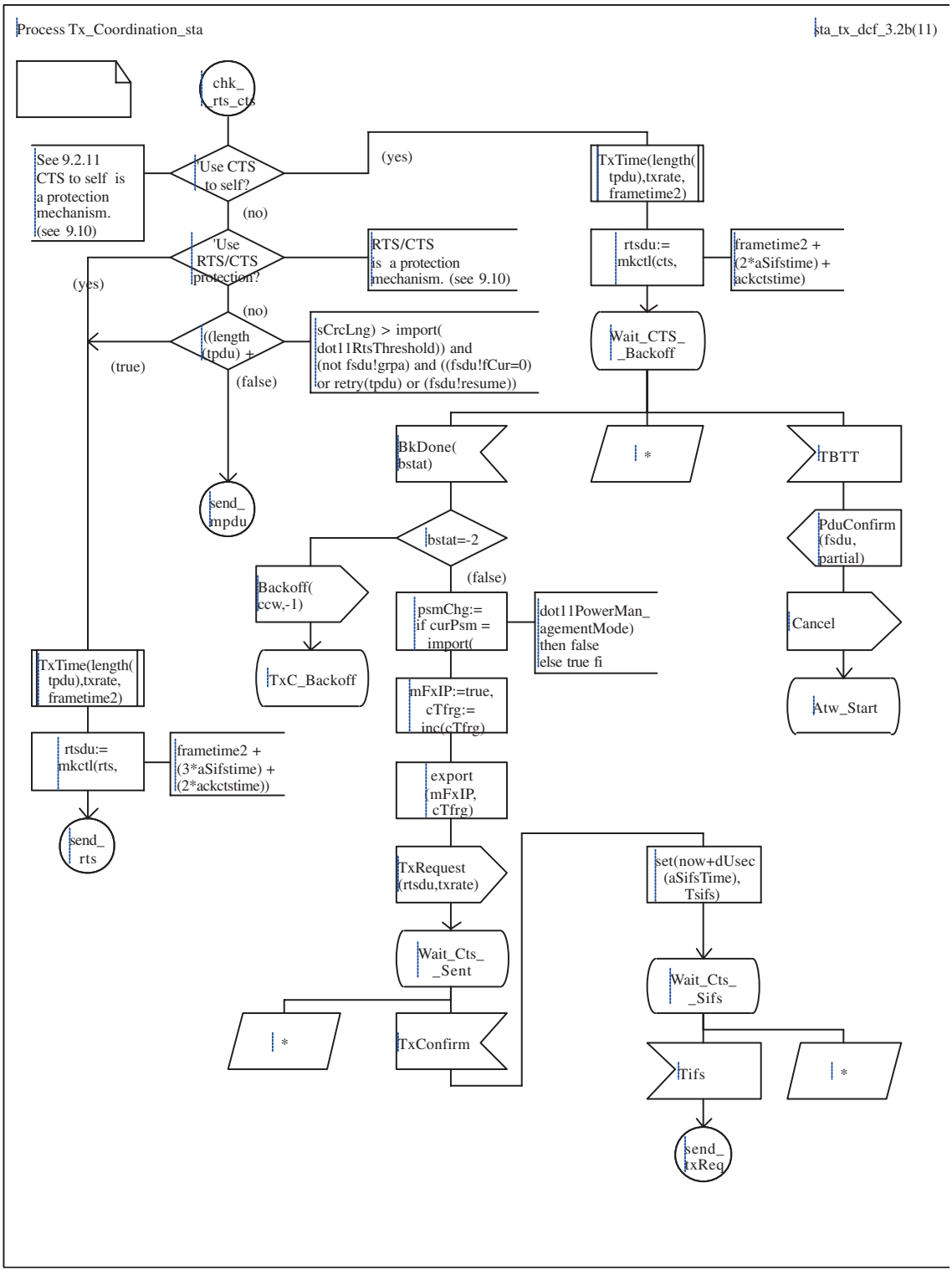




Copyright © 2007 IEEE. All rights reserved.

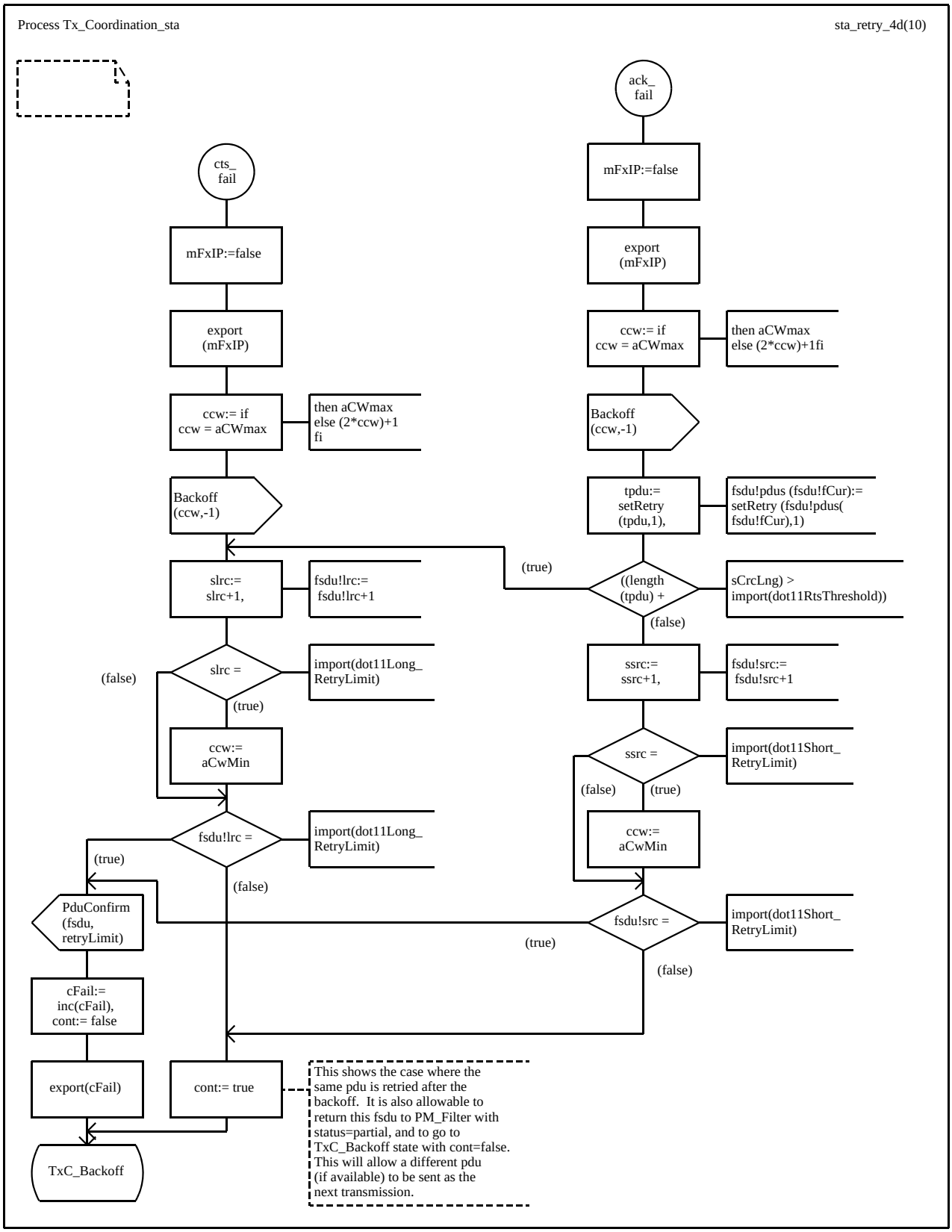



Copyright © 2007 IEEE. All rights reserved.

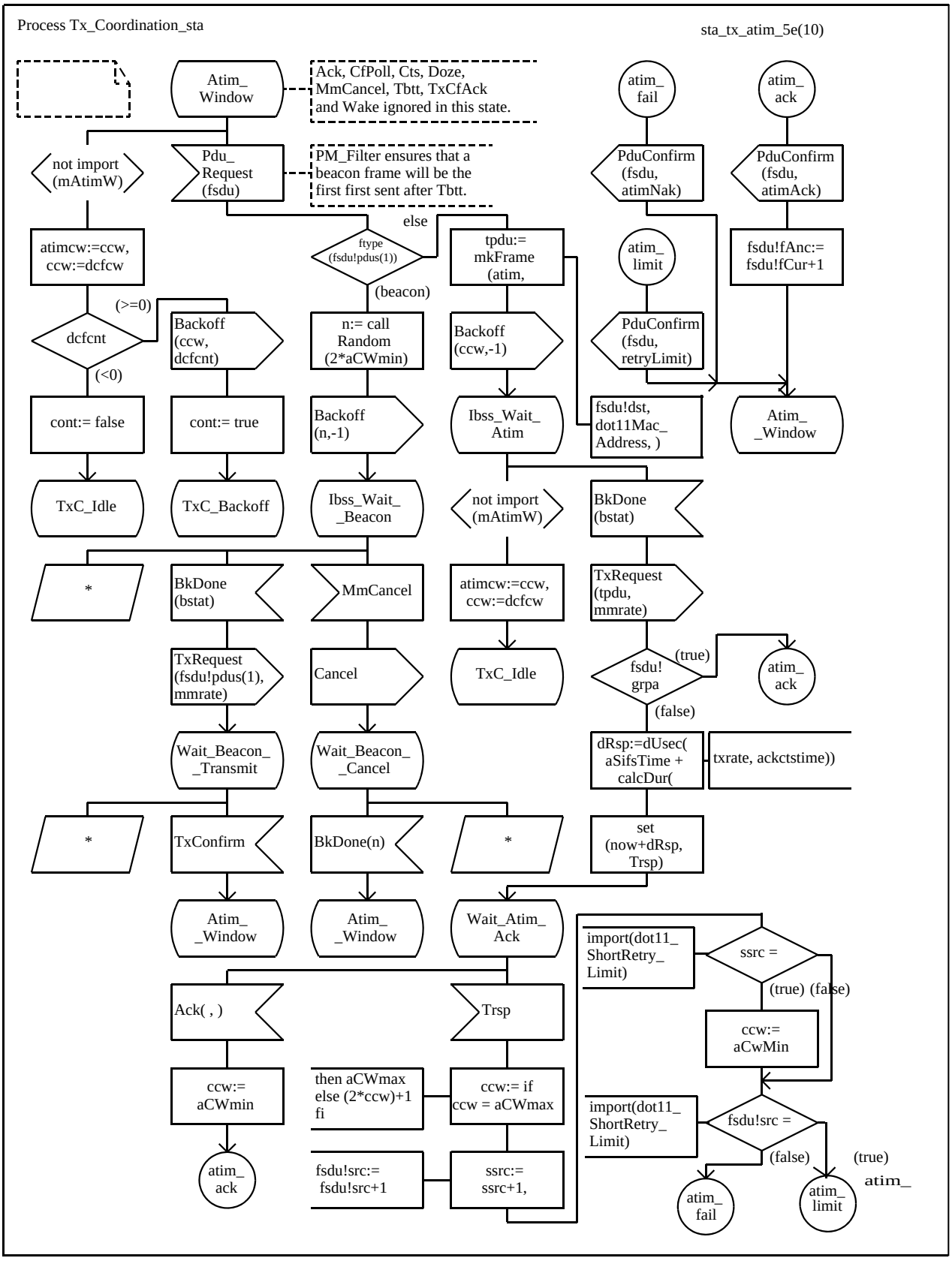




Copyright © 2007 IEEE. All rights reserved.

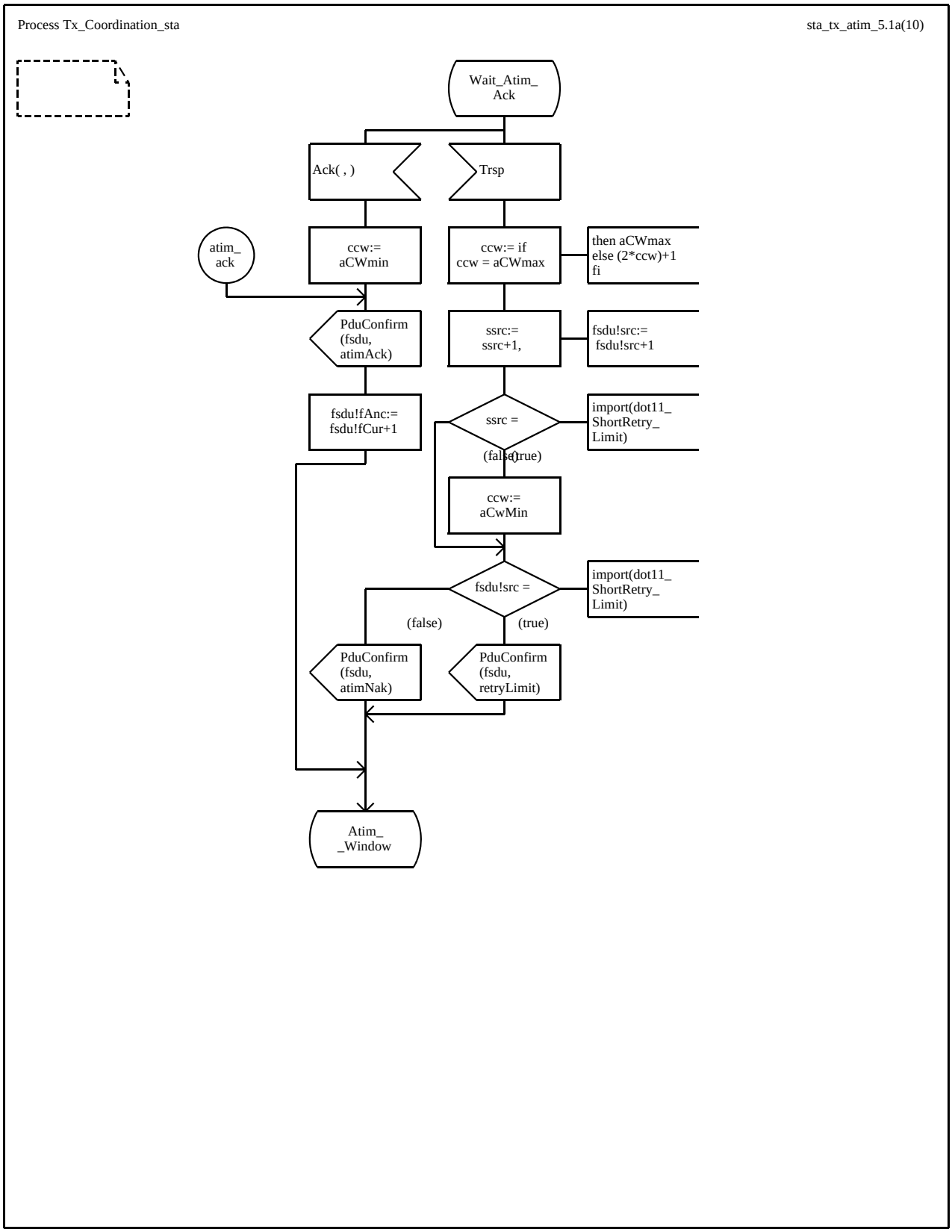



Copyright © 2007 IEEE. All rights reserved.

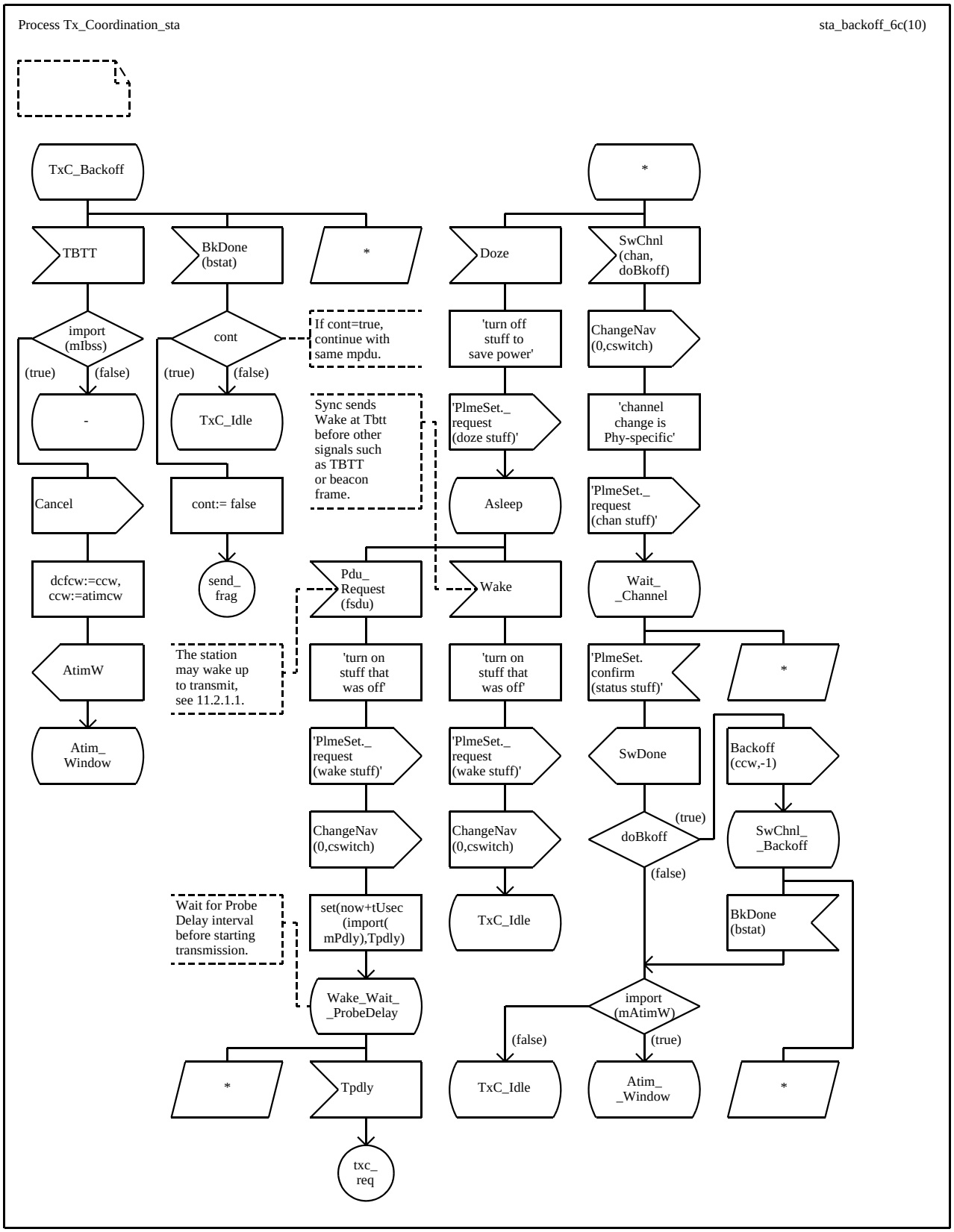




Copyright © 2007 IEEE. All rights reserved.

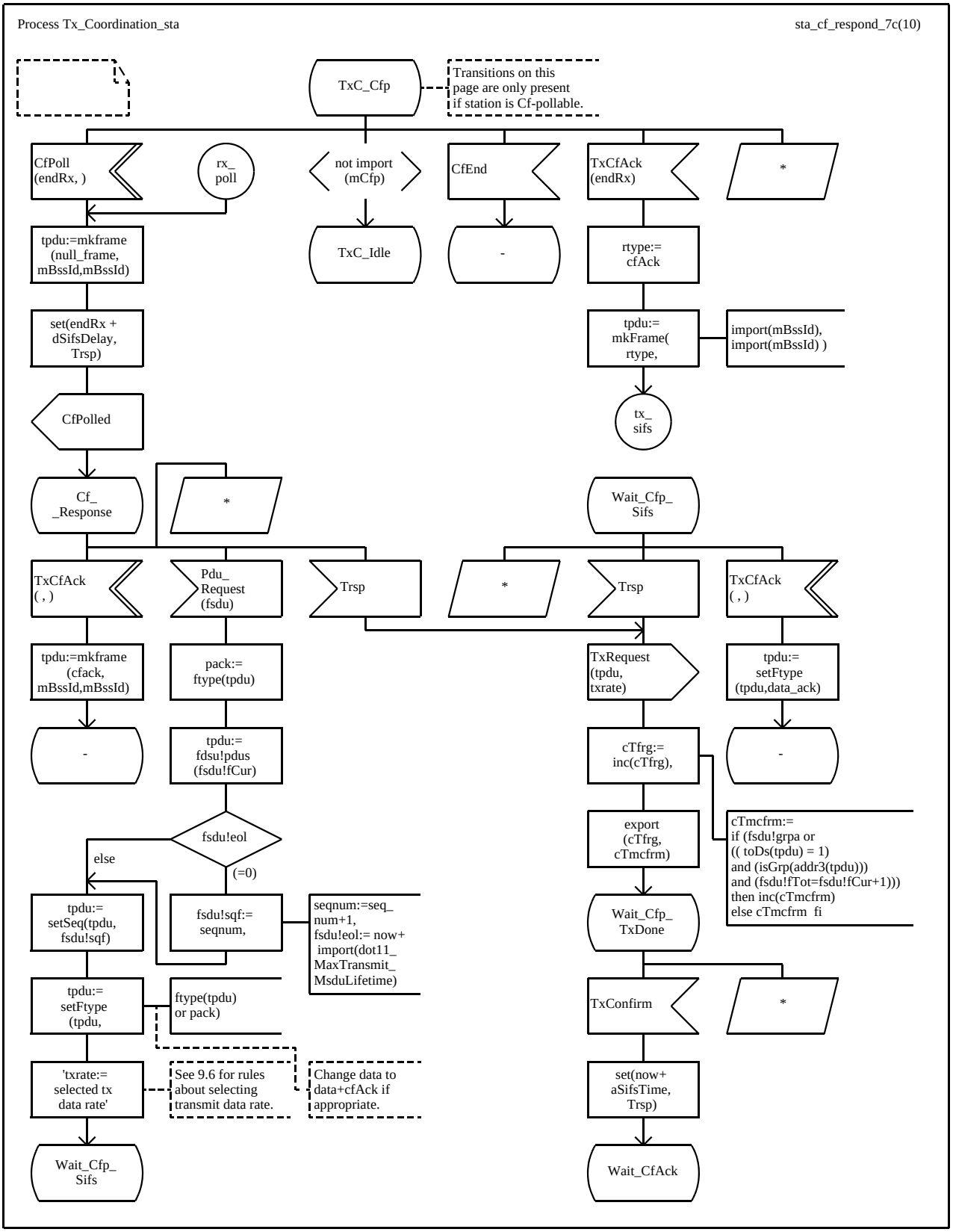



Copyright © 2007 IEEE. All rights reserved.

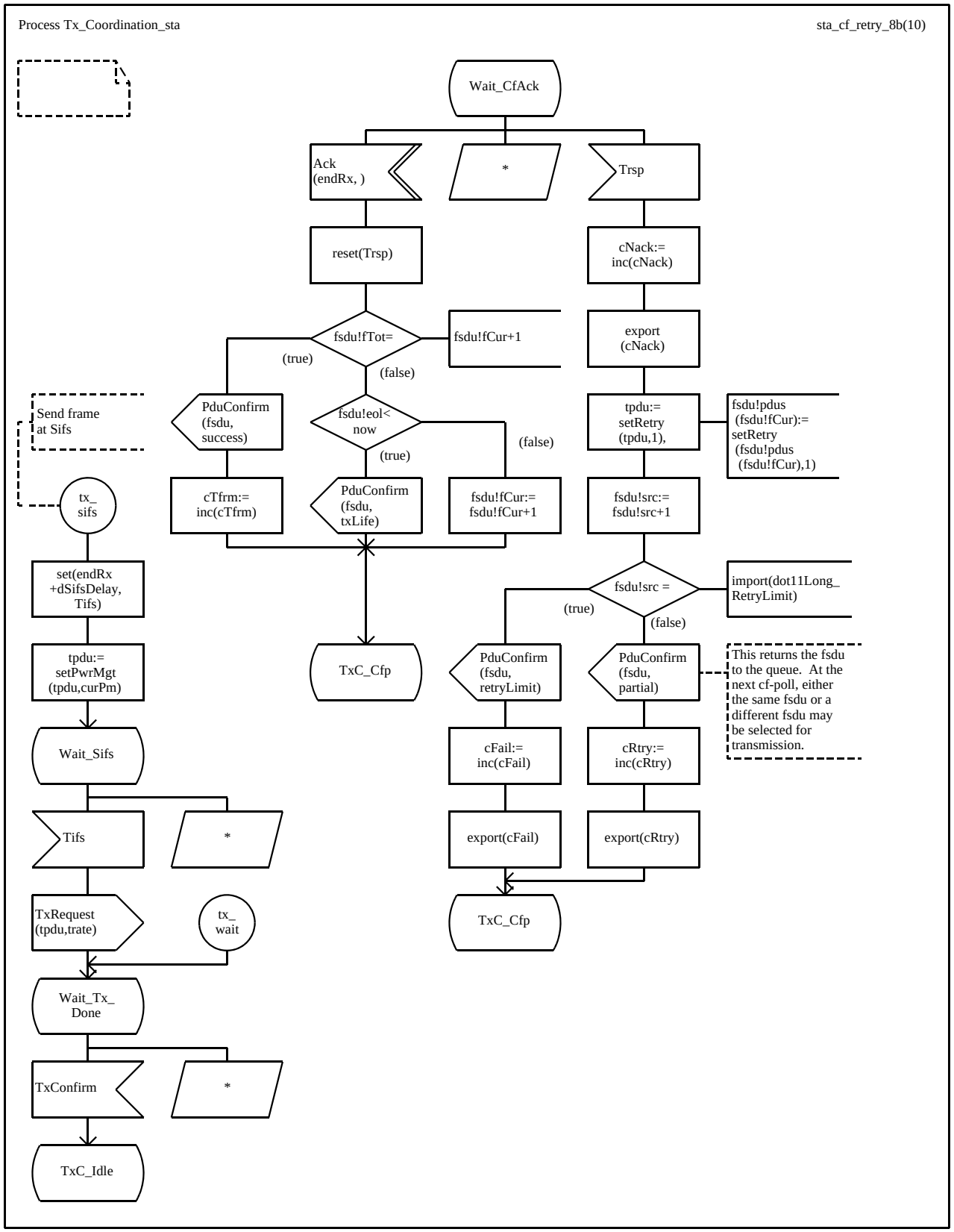




Copyright © 2007 IEEE. All rights reserved.

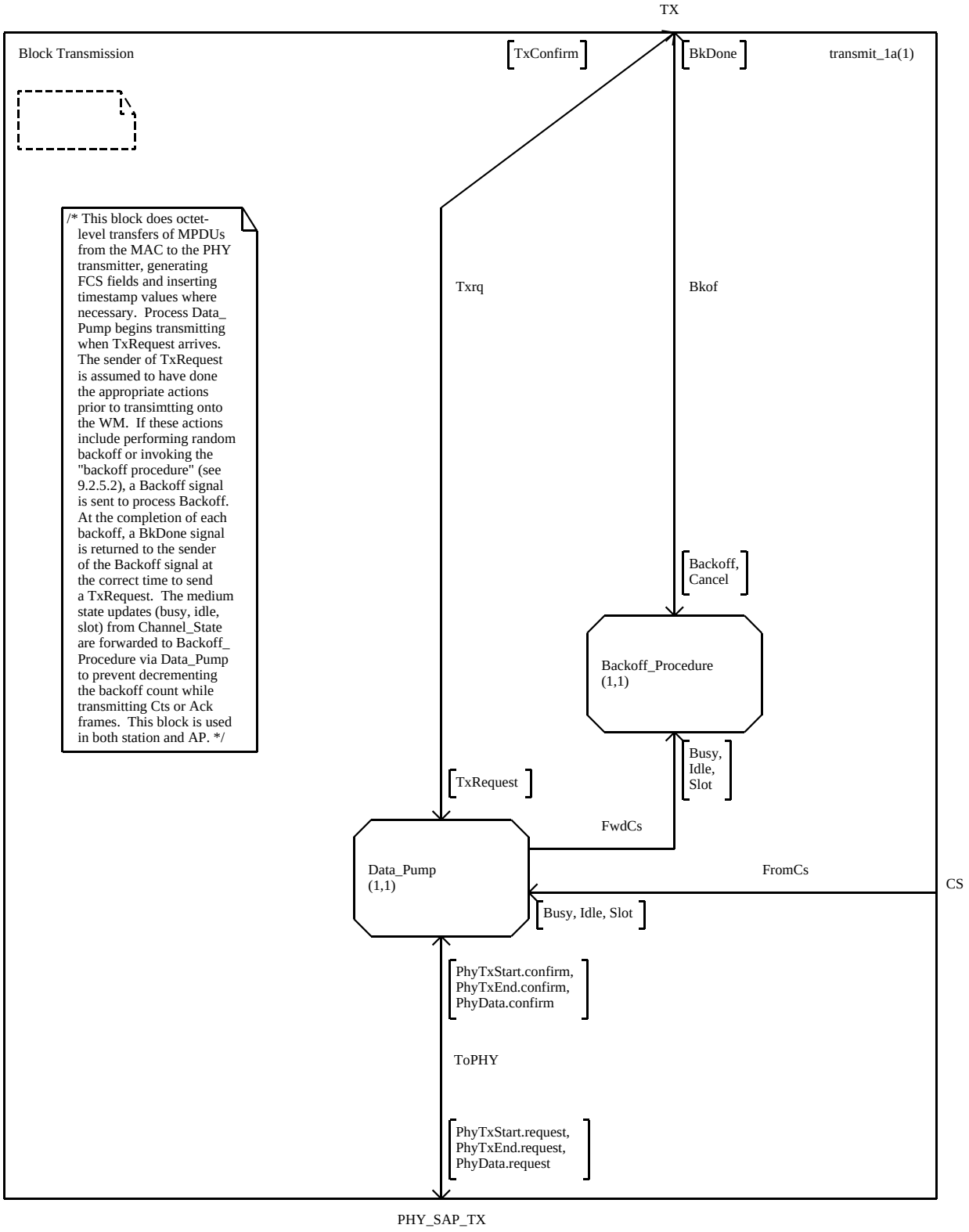



Copyright © 2007 IEEE. All rights reserved.

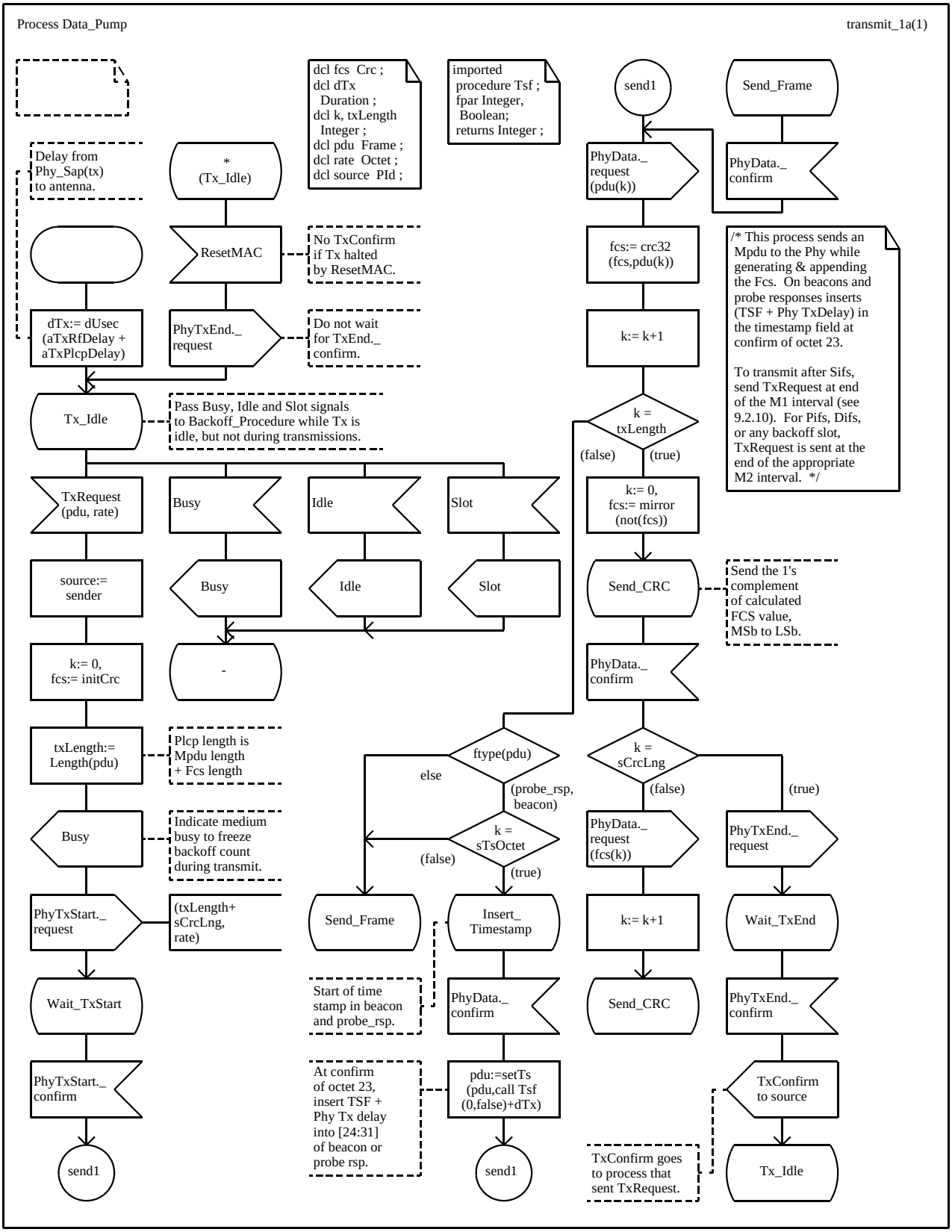




Copyright © 2007 IEEE. All rights reserved.

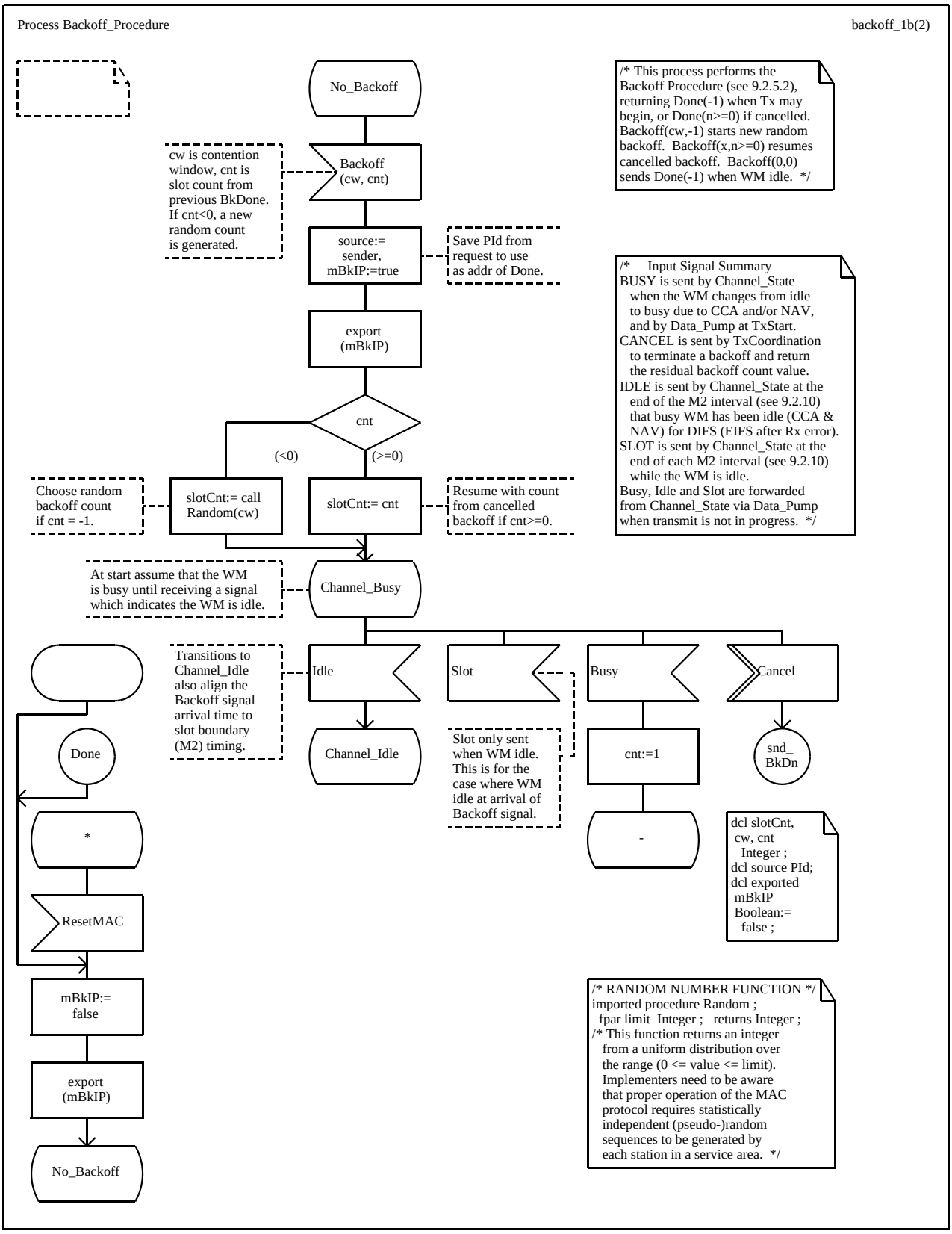



Copyright © 2007 IEEE. All rights reserved.

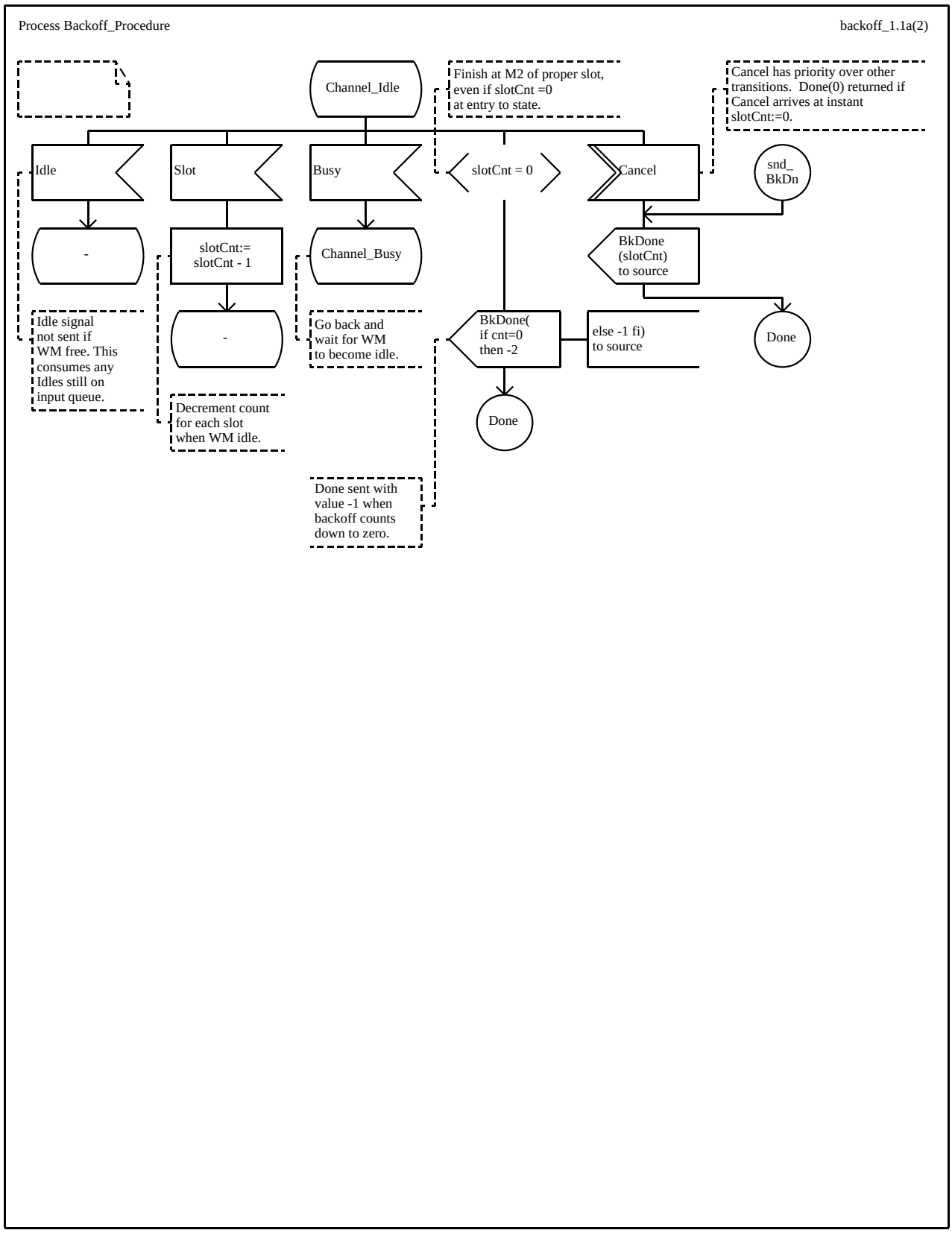




Copyright © 2007 IEEE. All rights reserved.

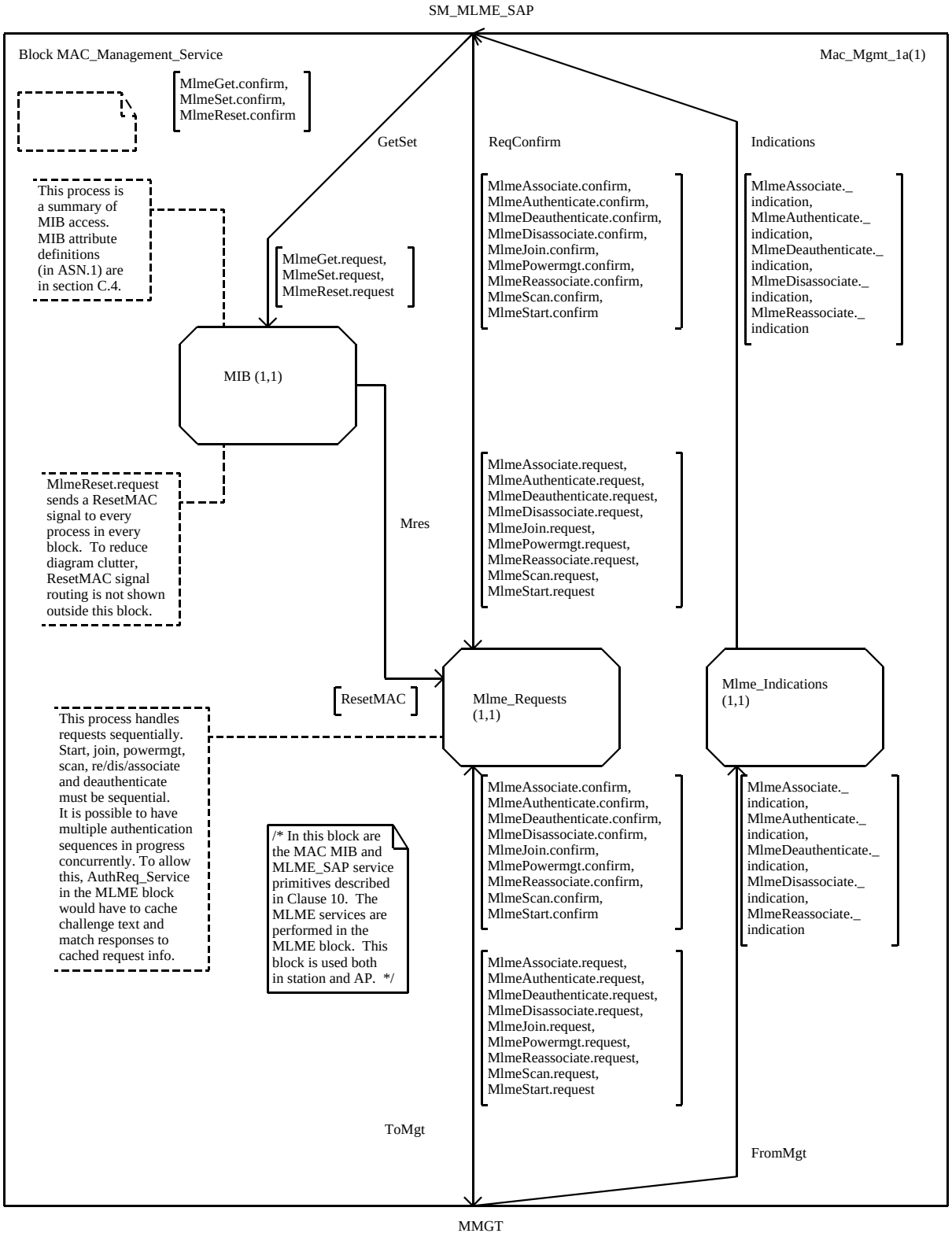



Copyright © 2007 IEEE. All rights reserved.

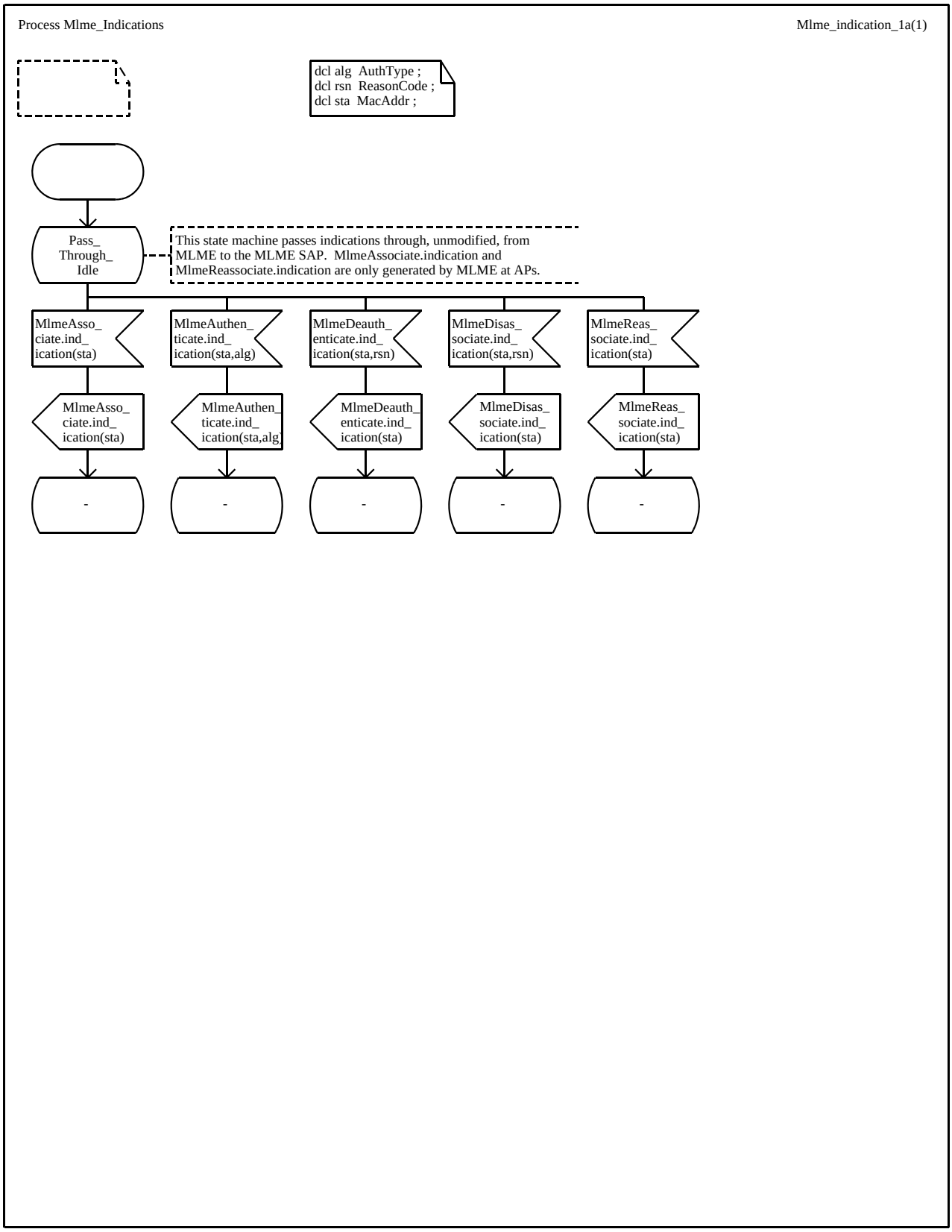




Copyright © 2007 IEEE. All rights reserved.

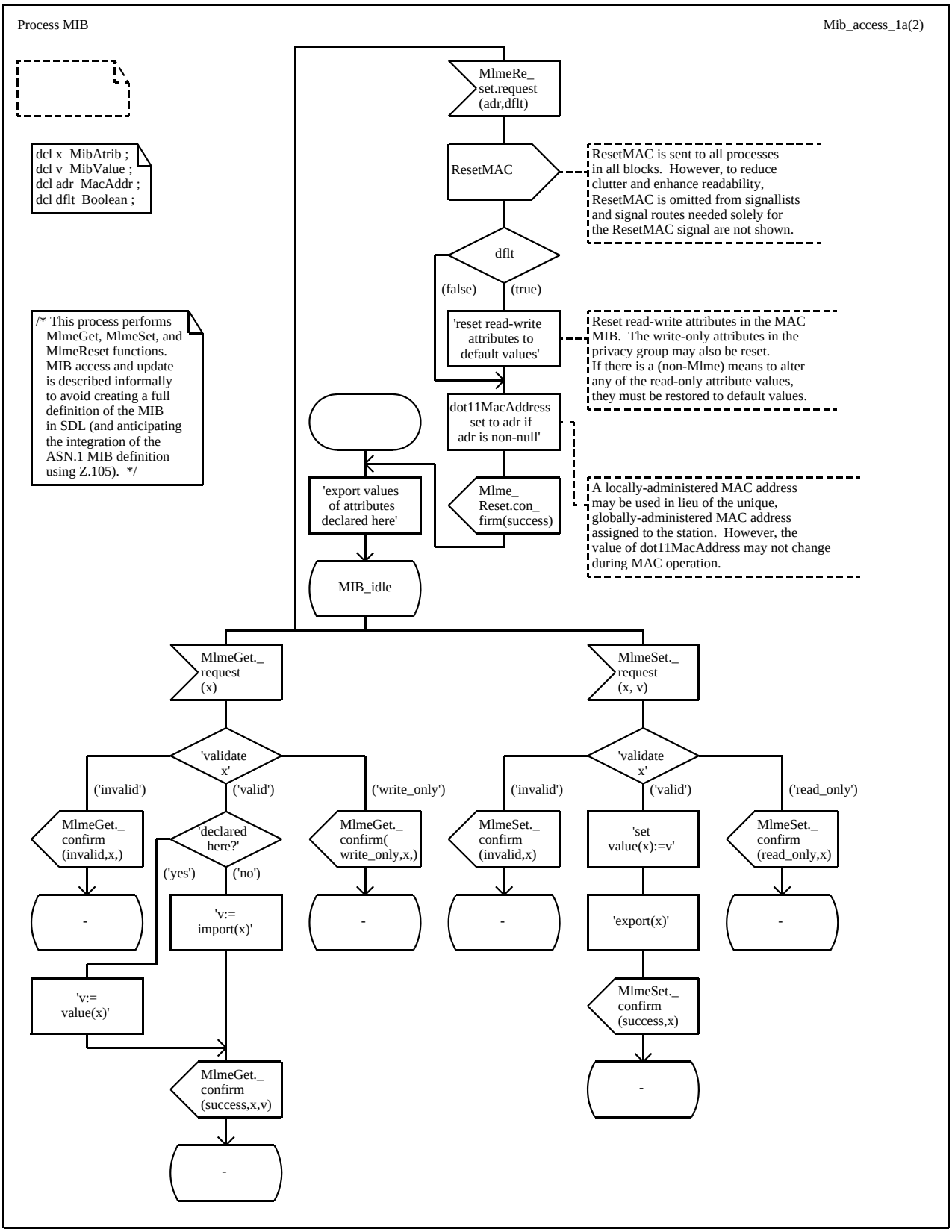



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

Process MIB Mib_import_export_2b(2)



```
/* Import of {Read-Only} MIB counter
   values exported from other processes */
imported
  dot11AckFailureCount,
  dot11FailedCount,
  dot11FcsErrorCount,
  dot11FrameDuplicateCount,
  dot11MulticastReceivedFrameCount,
  dot11MulticastTransmittedFrameCount,
  dot11MultipleRetryCount,
  dot11ReceivedFragmentCount,
  dot11RetryCount,
  dot11RtsFailureCount,
  dot11RtsSuccessCount,
  dot11TransmittedFragmentCount,
  dot11WepExcludedCount,
  dot11WepIcvErrorCount,
  dot11WepUndecryptableCount  Counter32 ;
```

```
/* Declarations of MIB attributes exported from
    this process */

       /* Read-Write attributes */
dcl exported
   dot11AuthenticationAlgorithms  AuthTypeSet:=
     incl(open_system, shared_key),
  dot11ExcludeUnencrypted  Boolean:= false,
  dot11FragmentationThreshold  Integer:= 2346,
  dot11GroupAddresses  MacAddrSet:= empty,
  dot11LongRetryLimit  Integer:= 4,
  dot11MaxReceiveLifetime  Kusec:= 512,
  dot11MaxTransmitMsduLifetime  Kusec:= 512,
  dot11MediumOccupancyLimit  Kusec:= 100,
  dot11PrivacyInvoked  Boolean:= false,
  mReceiveDTIMs  Boolean:= true,
  dot11CfpPeriod  Integer:= 1,
  dot11CfpMaxDuration  Kusec:= 200,
  dot11AuthenticationResponseTimeout  Kusec:= 512,
  dot11RtsThreshold  Integer:= 3000,
  dot11ShortRetryLimit  Integer:= 7,
  dot11WepDefaultKeyId  KeyIndex:= 0,
  dot11CurrentChannelNumber  Integer:= 0,
  dot11CurrentSet  Integer:= 0,
  dot11CurrentPattern  Integer:= 0,
  dot11CurrentIndex  Integer:= 0 ;

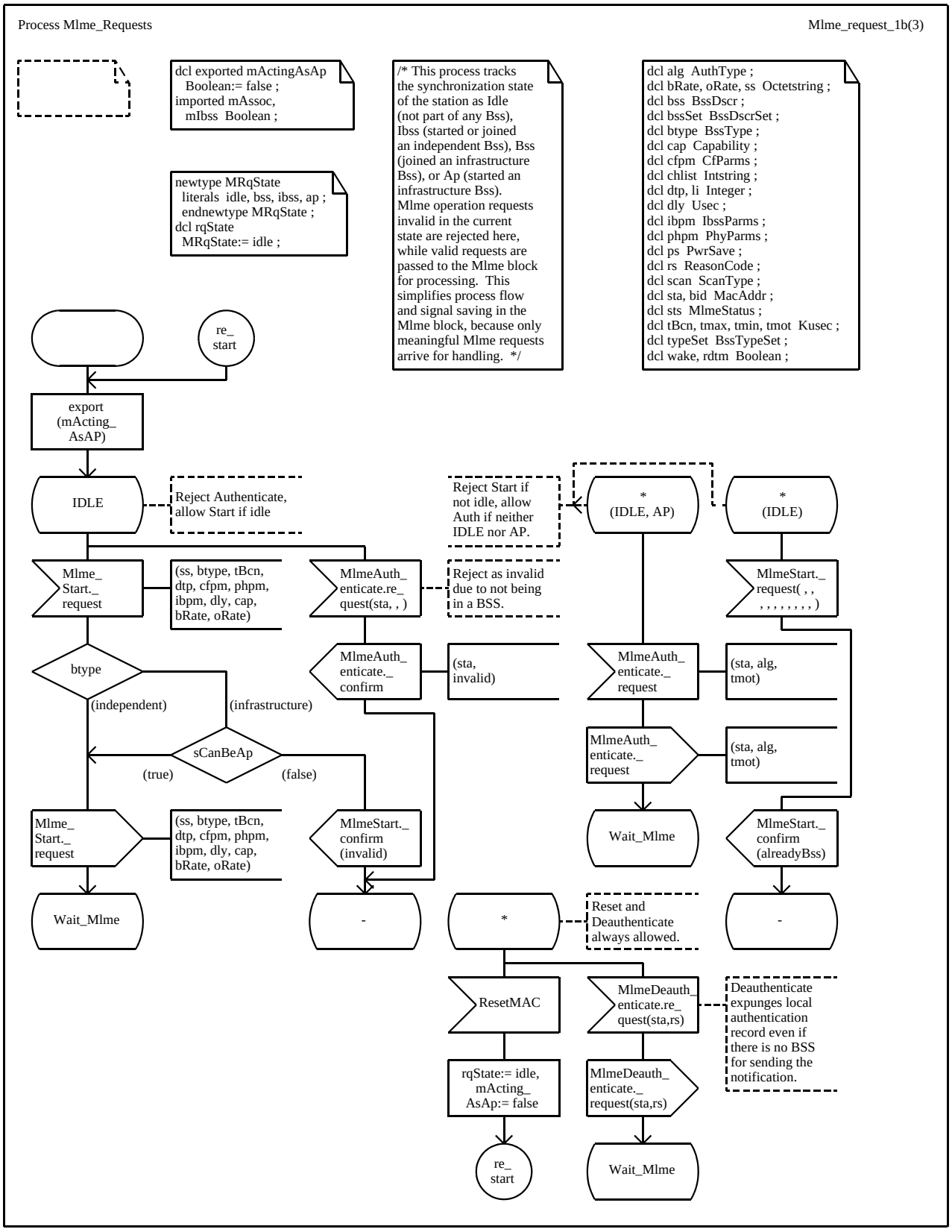
       /* Write-Only attributes */
dcl exported
  dot11WepDefaultKeys  KeyVector:= nullKey,
  dot11WepKeyMappings
    KeyMapArray:= (. nullAddr, false, nullKey .) ;
```

```
/* The following Read-Only attributes in the
   MAC MIB are defined as synonyms (named
   constants) rather than remote variables
   because they describe properties of the
   station which are static, at least during
   any single instance of MAC operation:
     dot11AuthenticationAlgorithms  AuthTypeSet,
     dot11CfPollable  Boolean,
     dot11MacAddress  MacAddr,
     dot11ManufacturerID  Octetstring,
     dot11PrivacyOptionImplemented  Boolean,
     dot11ProductID  Octetstring,
     aStationID  MacAddr,
     dot11WepKeyMappingLength  Integer ;

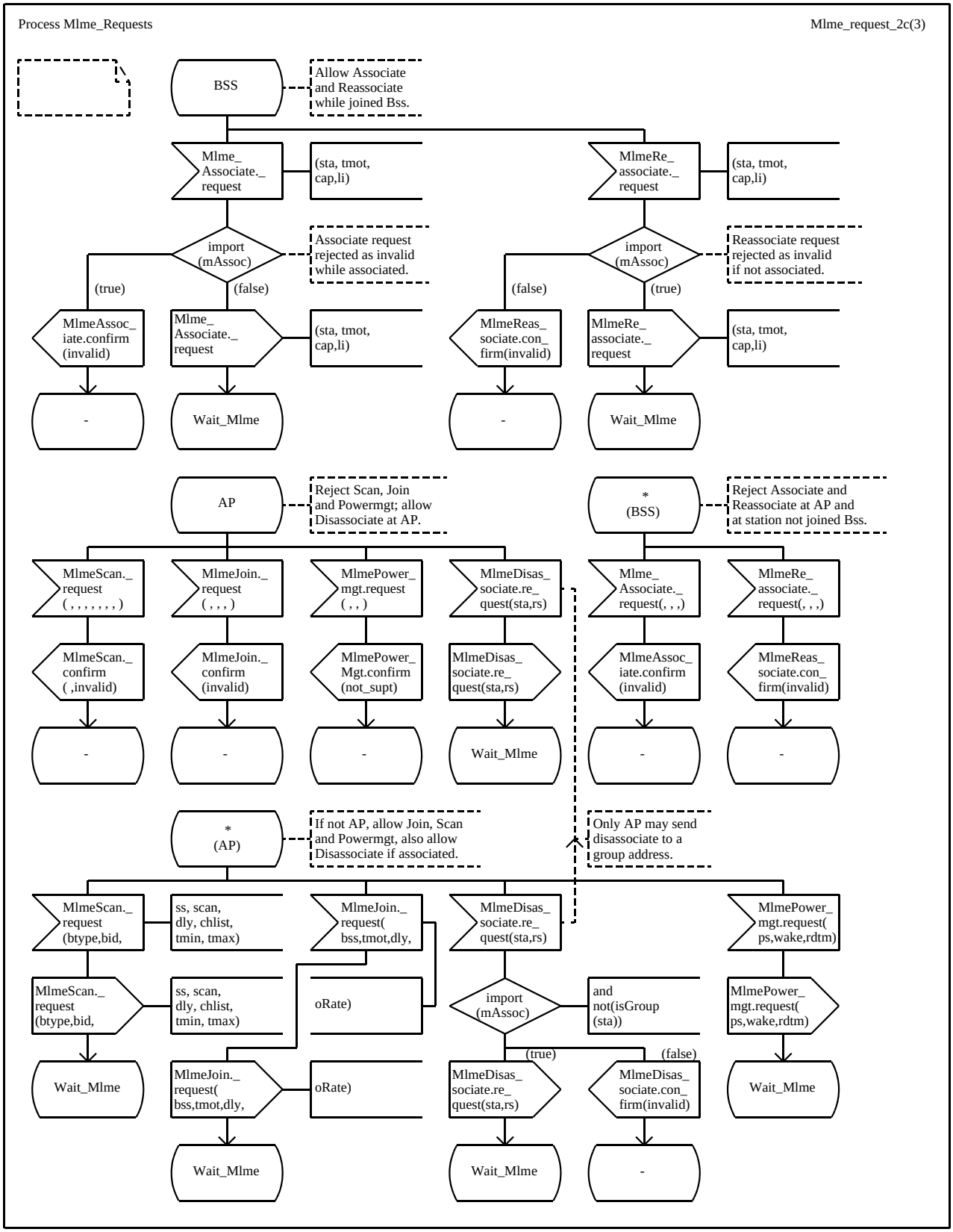
   In addition, all Read-Only attributes in the
   PHY MIB which are accessed by the MAC
   are defined as synonyms.
*/
```

```
/* NOTE:
   The values listed for MAC MIB attributes are the
   specified default values for those attributes.
   The values listed for PHY MIB attributes are either
   the default values for the FH PHY, or arbitrary
   values within the specified range.  The specific
   values for PHY attributes in this SDL description
   of the MAC do not have normative significance.
*/
```


Copyright © 2007 IEEE. All rights reserved.

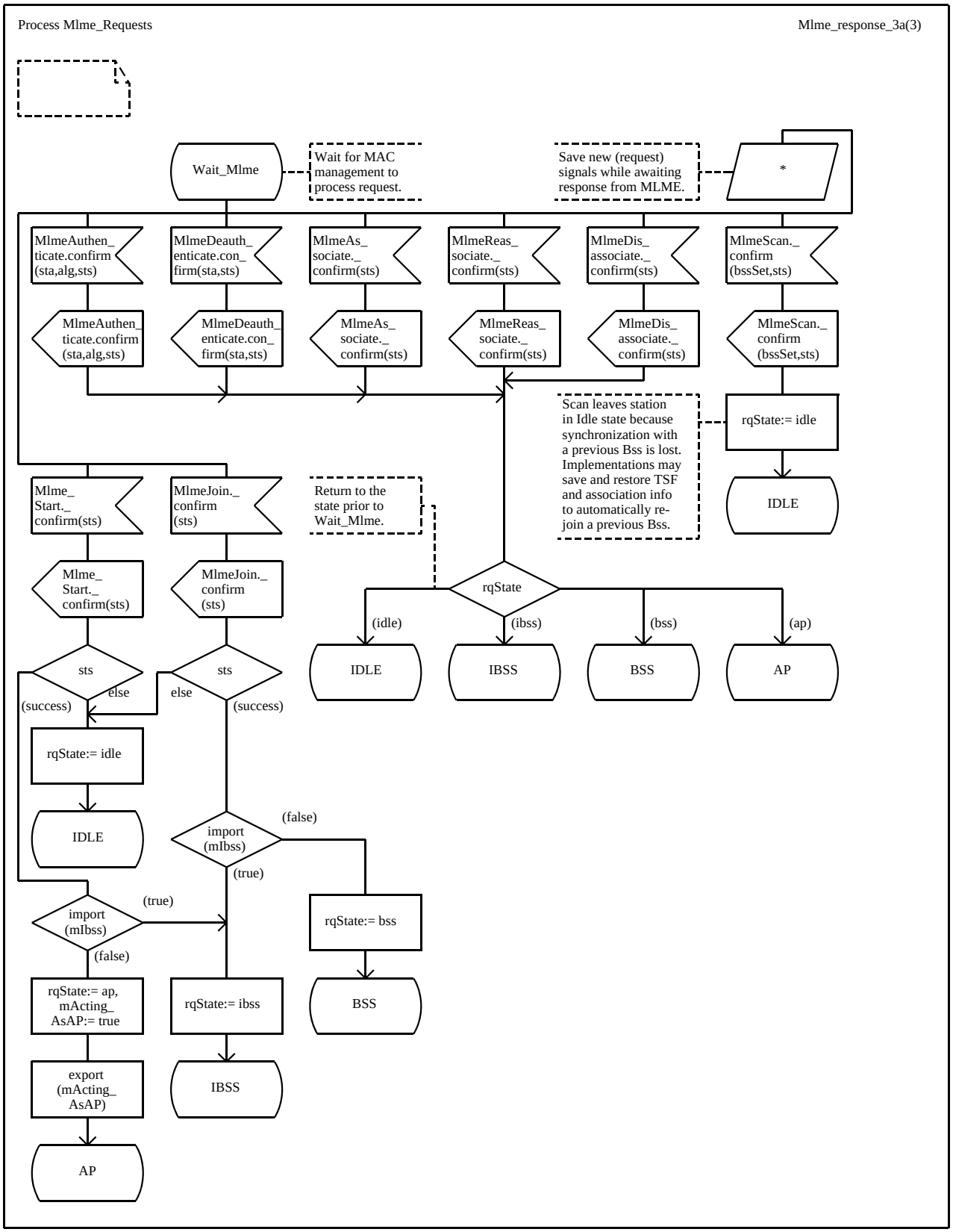



Copyright © 2007 IEEE. All rights reserved.

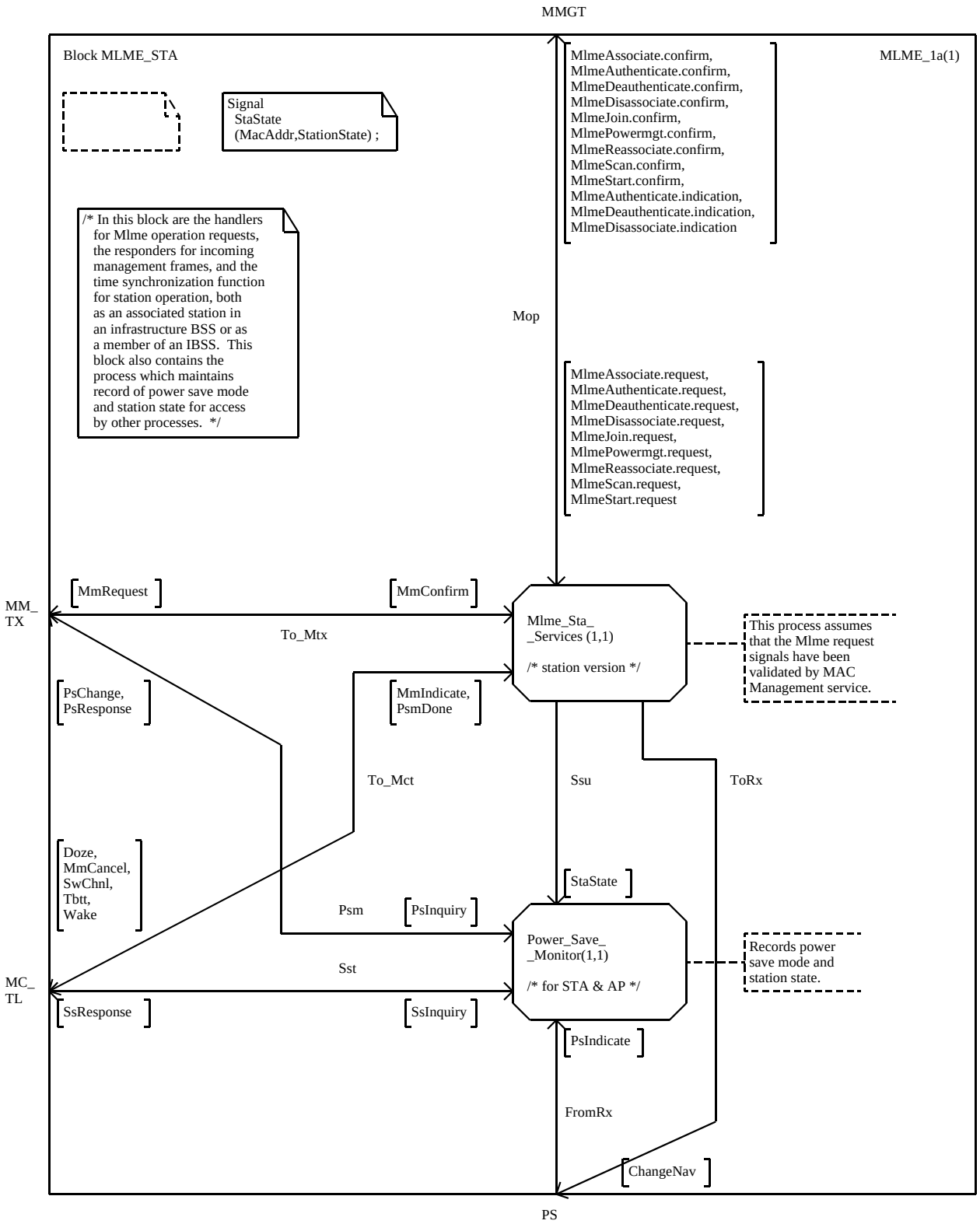




Copyright © 2007 IEEE. All rights reserved.

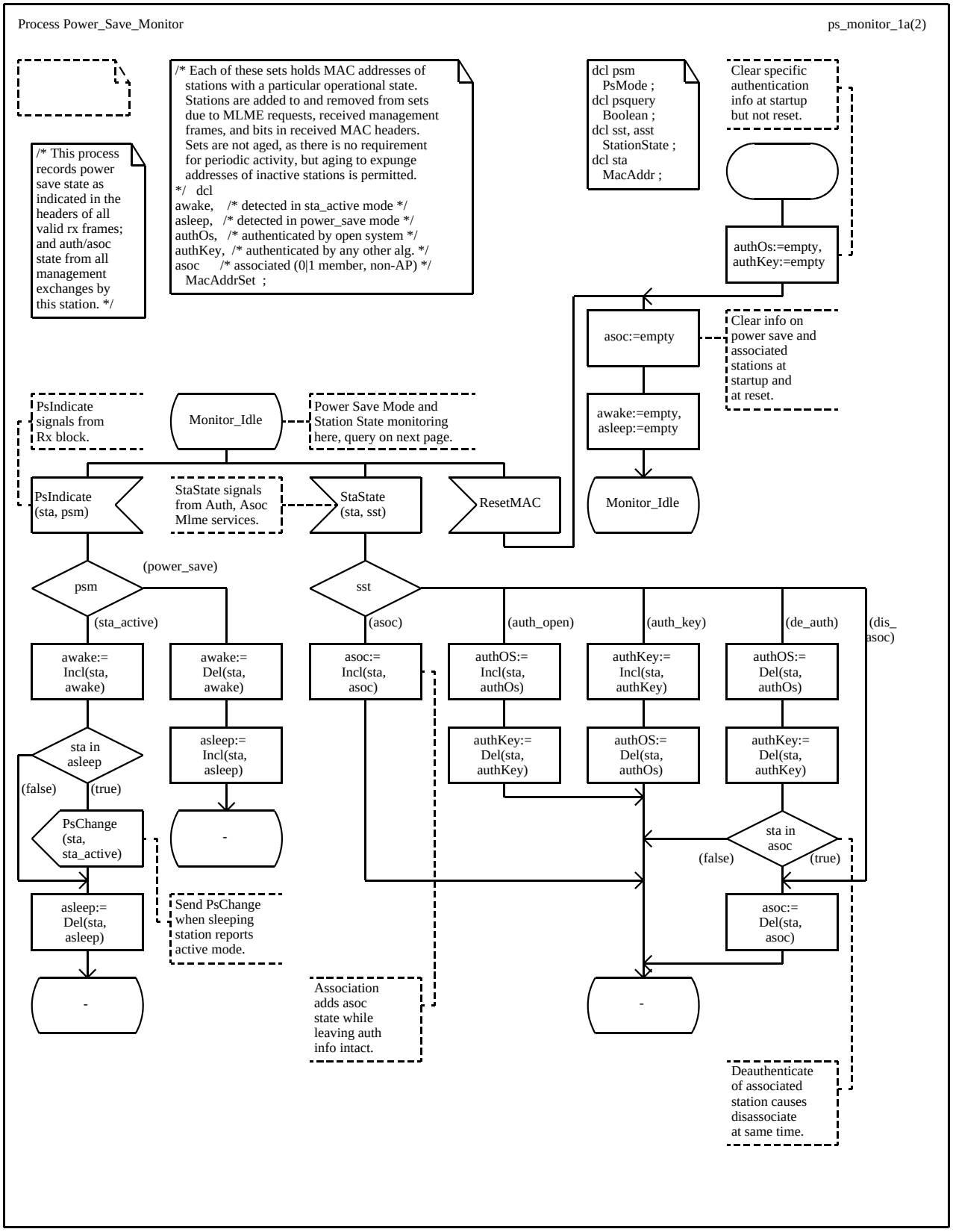



Copyright © 2007 IEEE. All rights reserved.

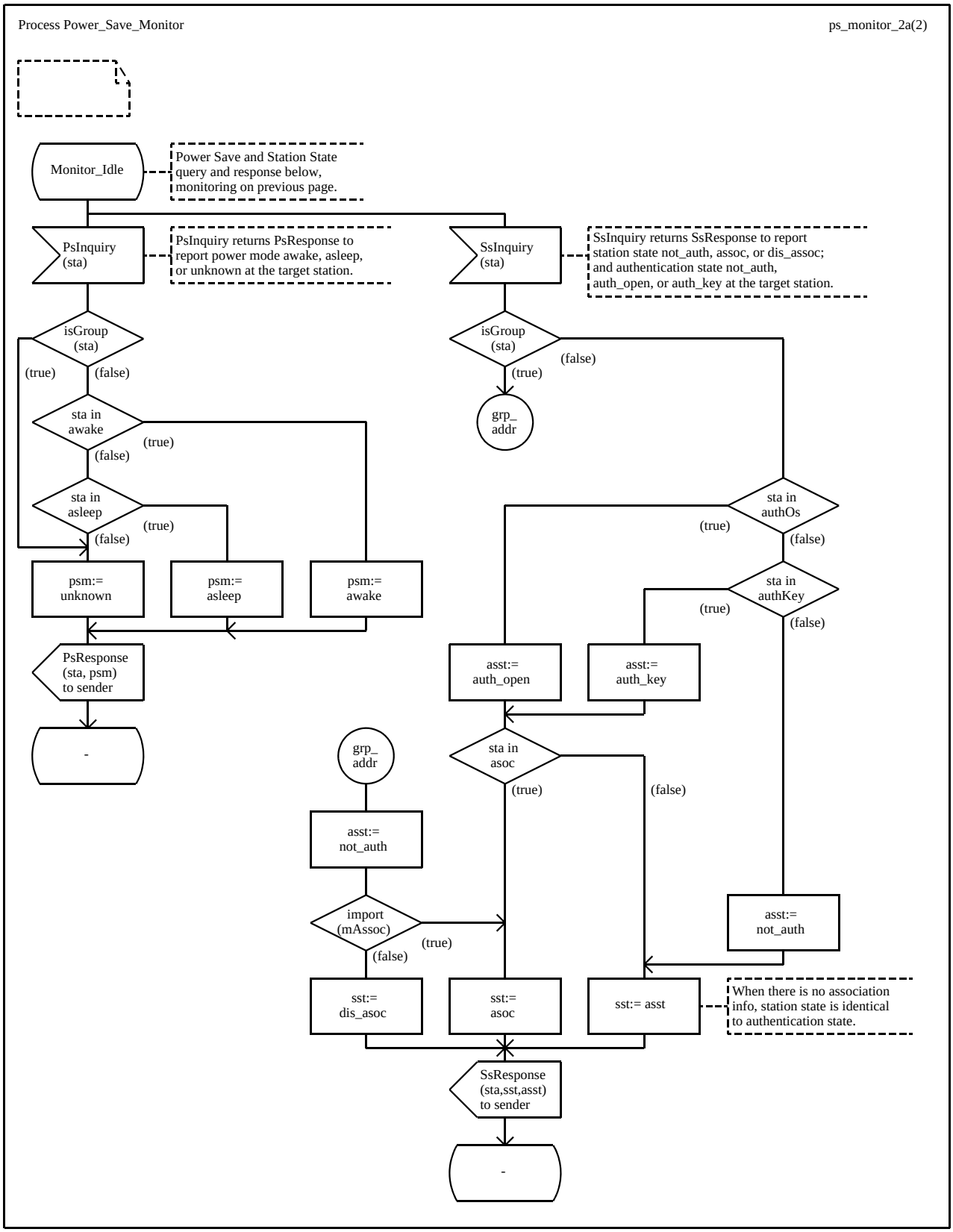




Copyright © 2007 IEEE. All rights reserved.

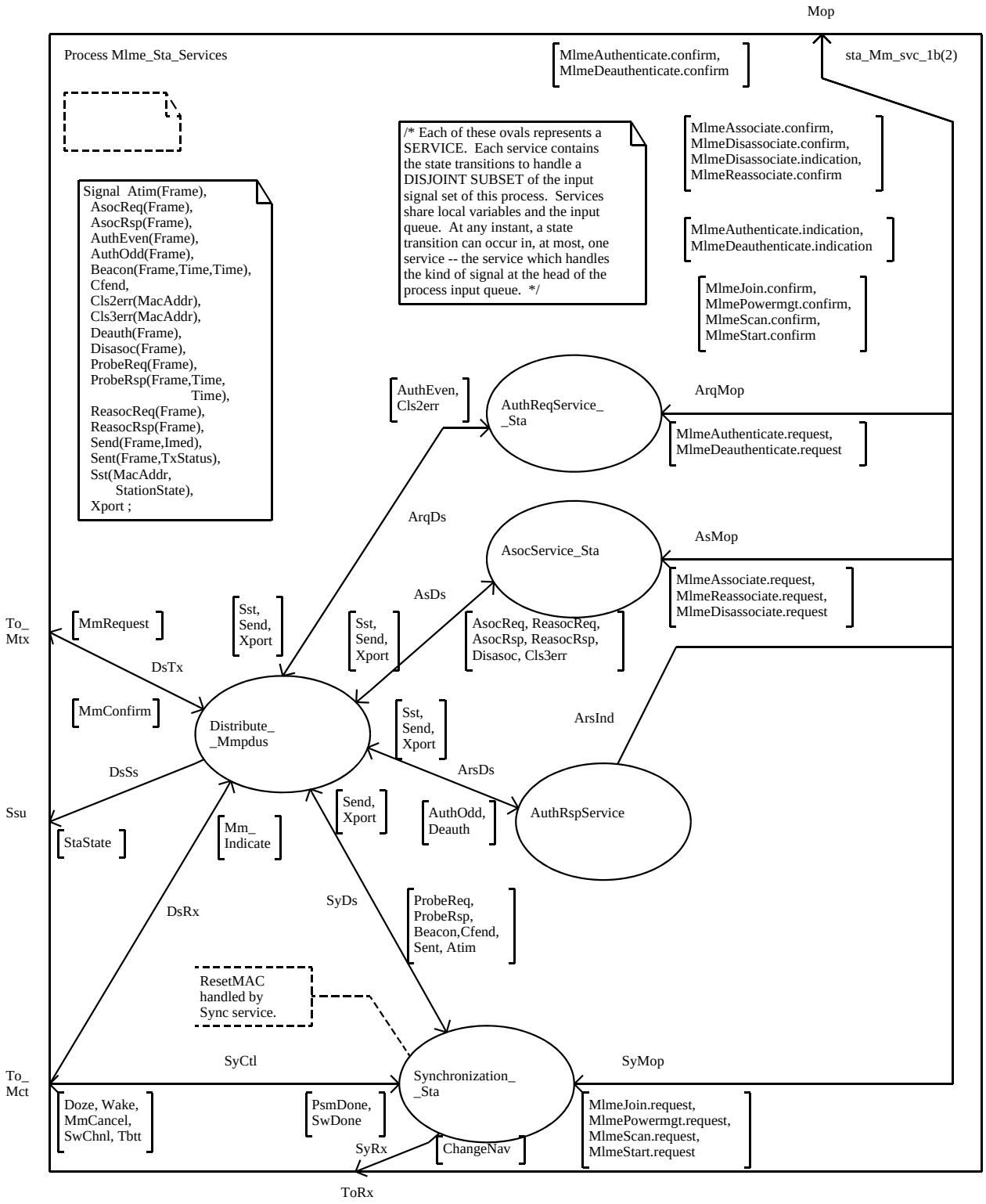



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

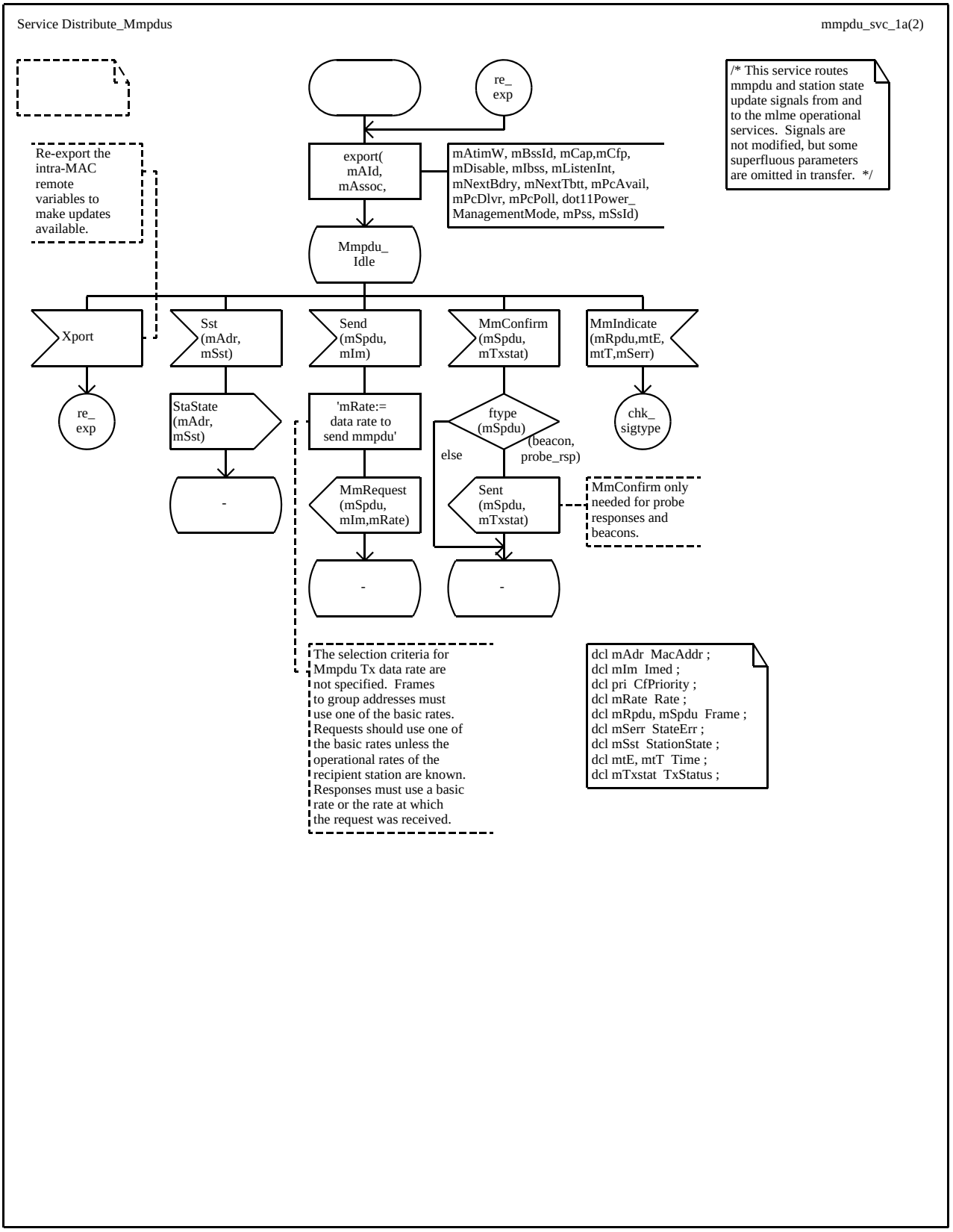
Process Mlme_Sta_Services sta_Mm_svc_1.1b(2)



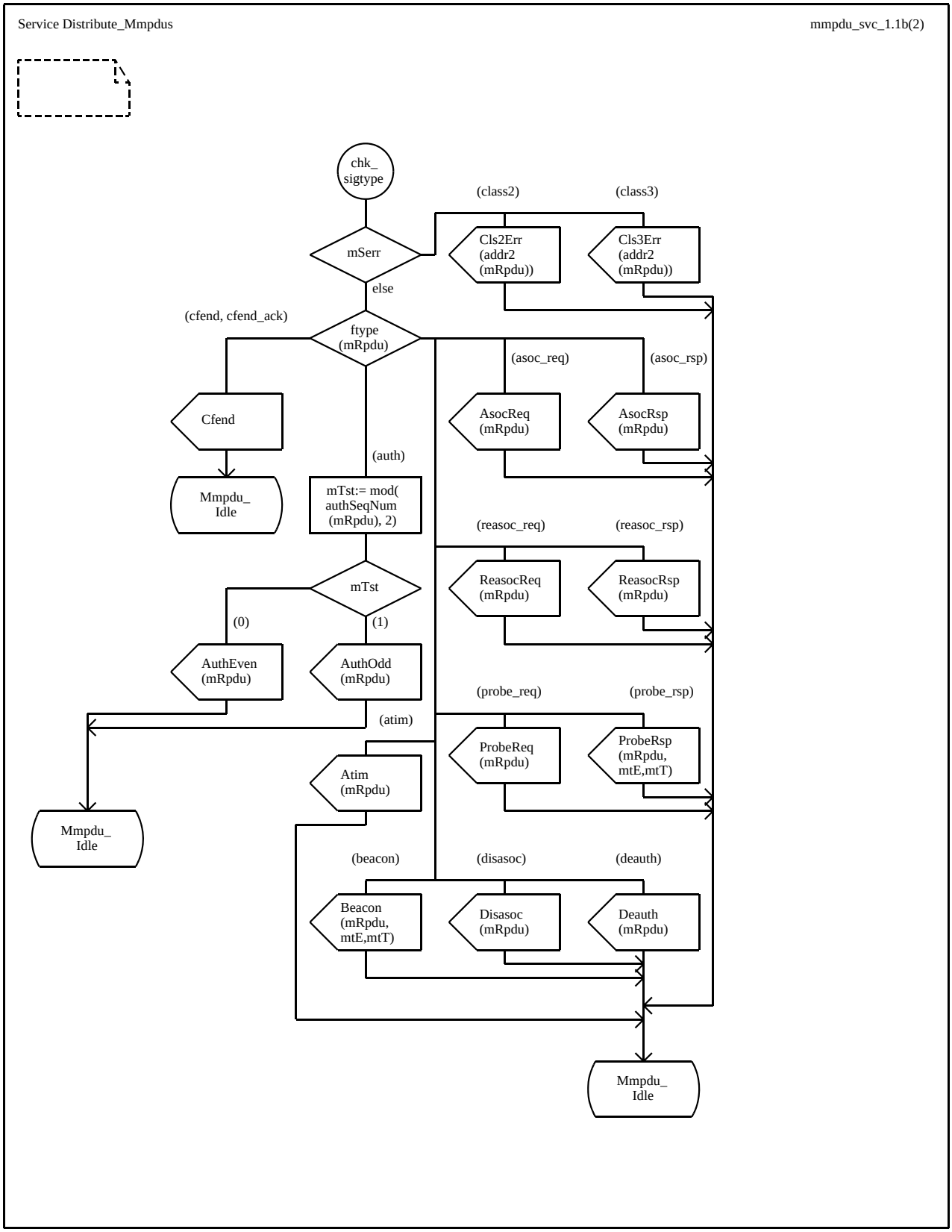
```
Timer Tasoc,
  Tauth, Tchal,
  Tbcn, Tatim ;
```

```
/* Intra-MAC remote variables */
dcl exported
dot11PowerManagementMode  PwrSave:= sta_active,
dot11DesiredSsid  Octetstring,
dot11DesiredBssType,
dot11OperationalRateSet  Ratestring:= mkOS(2,1),
dot11BeaconPeriod  TU,
dot11DtimPeriod  Integer:= 1,
dot11AssociationResponseTimeOut  TU,
dot11MultiDomainCapabilityEnabled  Boolean:= false,
mAId  AsocId:= 0,
mAssoc  Boolean:= false,
mAtimW  Boolean:= false,
mBrates Ratestring:= mkOS(2,1),
mBssId  MacAddr:= nullAddr,
mCap  Octetstring:= O2,
mCfp  Boolean:= false,
mDisable  Boolean:= true,
mDtimCount  Integer:= 1,
mIbss  Boolean:= false,
mNextBdry  Time:= 0,
mNextTbtt  Time:= 0,
mPcAvail  Boolean:= false,
mPcPoll  Boolean:= false,
mPdly  Usec:= 0,
mPss  PsState:= awake,
mSsId  Octetstring:= null;
```


Copyright © 2007 IEEE. All rights reserved.

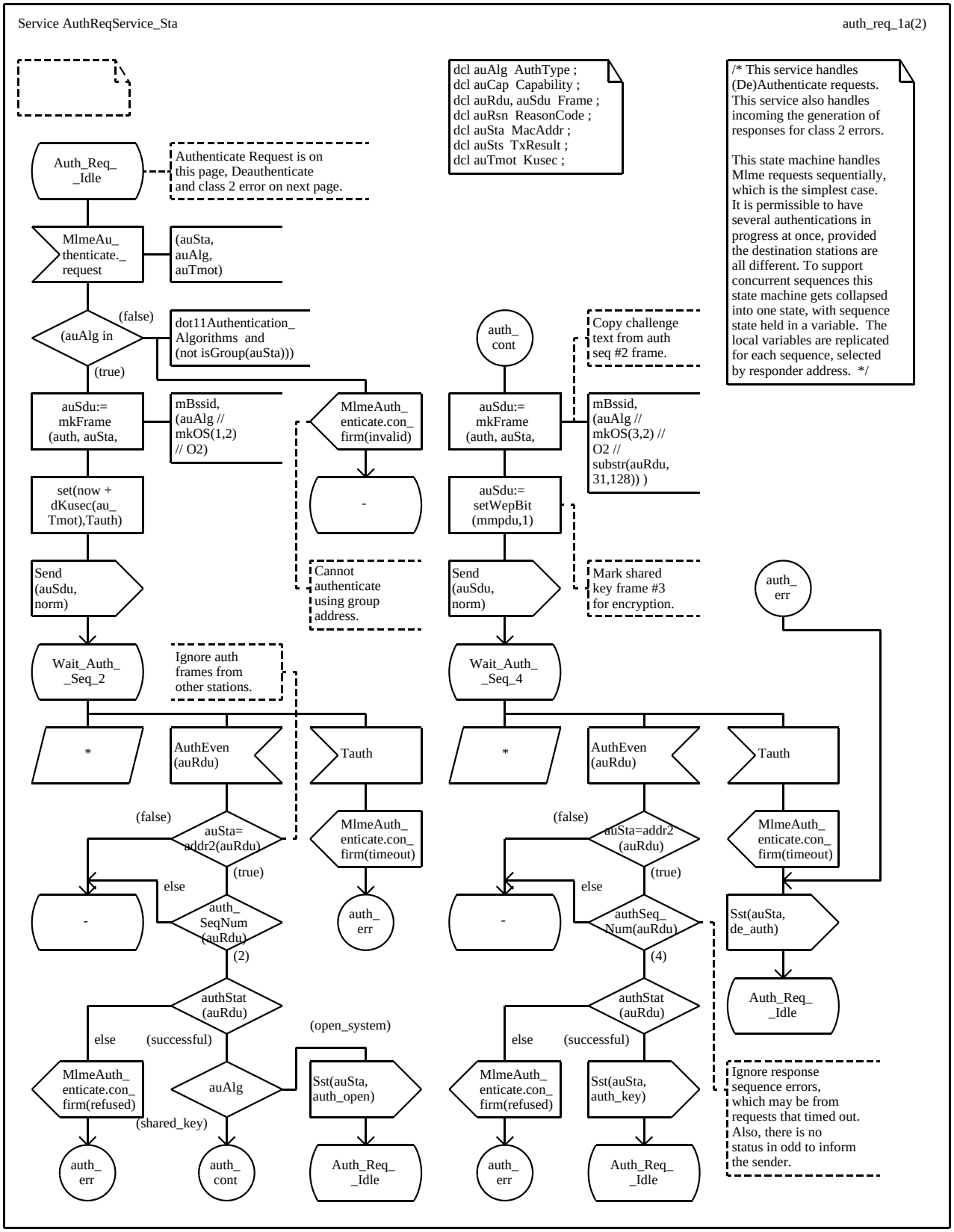



Copyright © 2007 IEEE. All rights reserved.

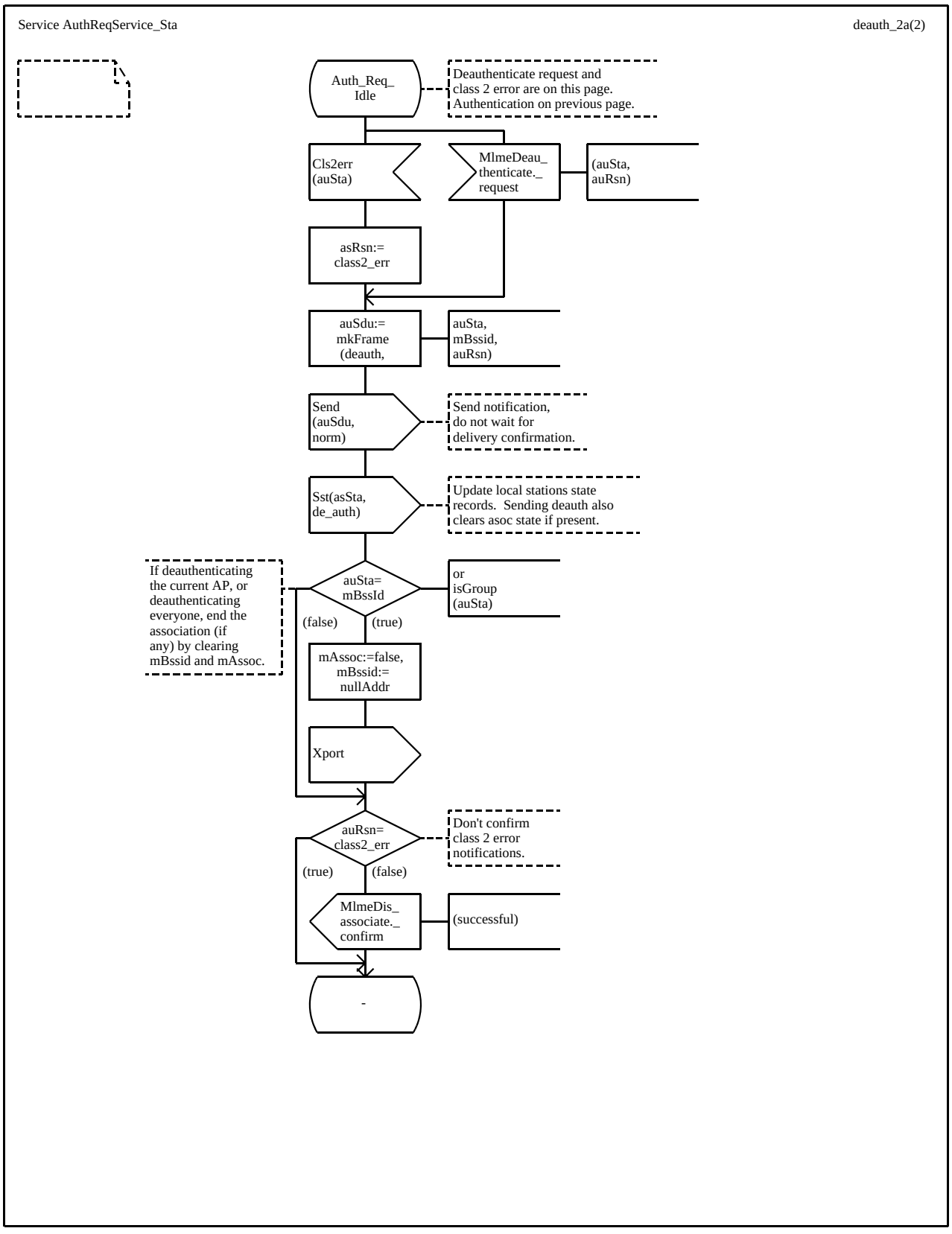




Copyright © 2007 IEEE. All rights reserved.

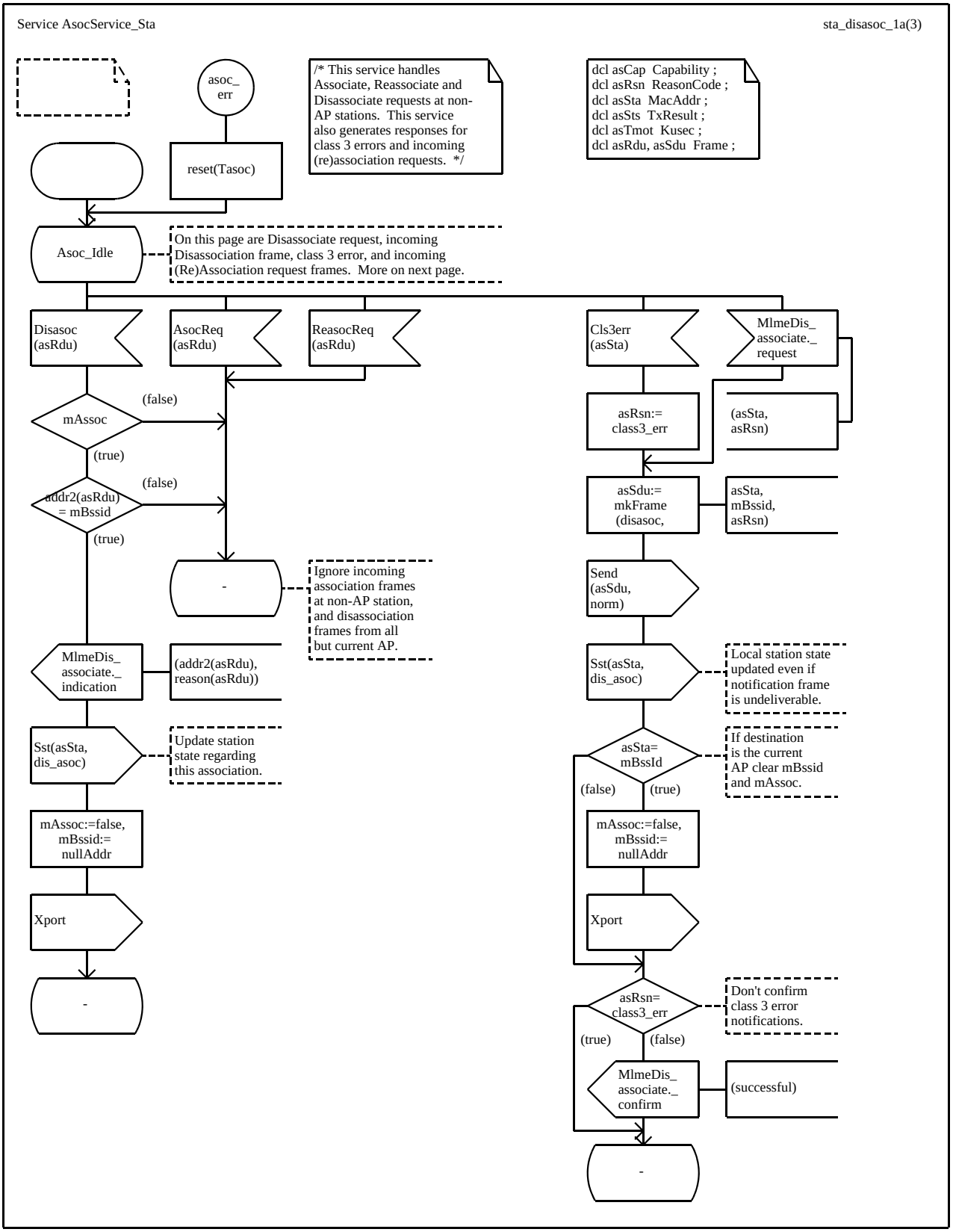



Copyright © 2007 IEEE. All rights reserved.

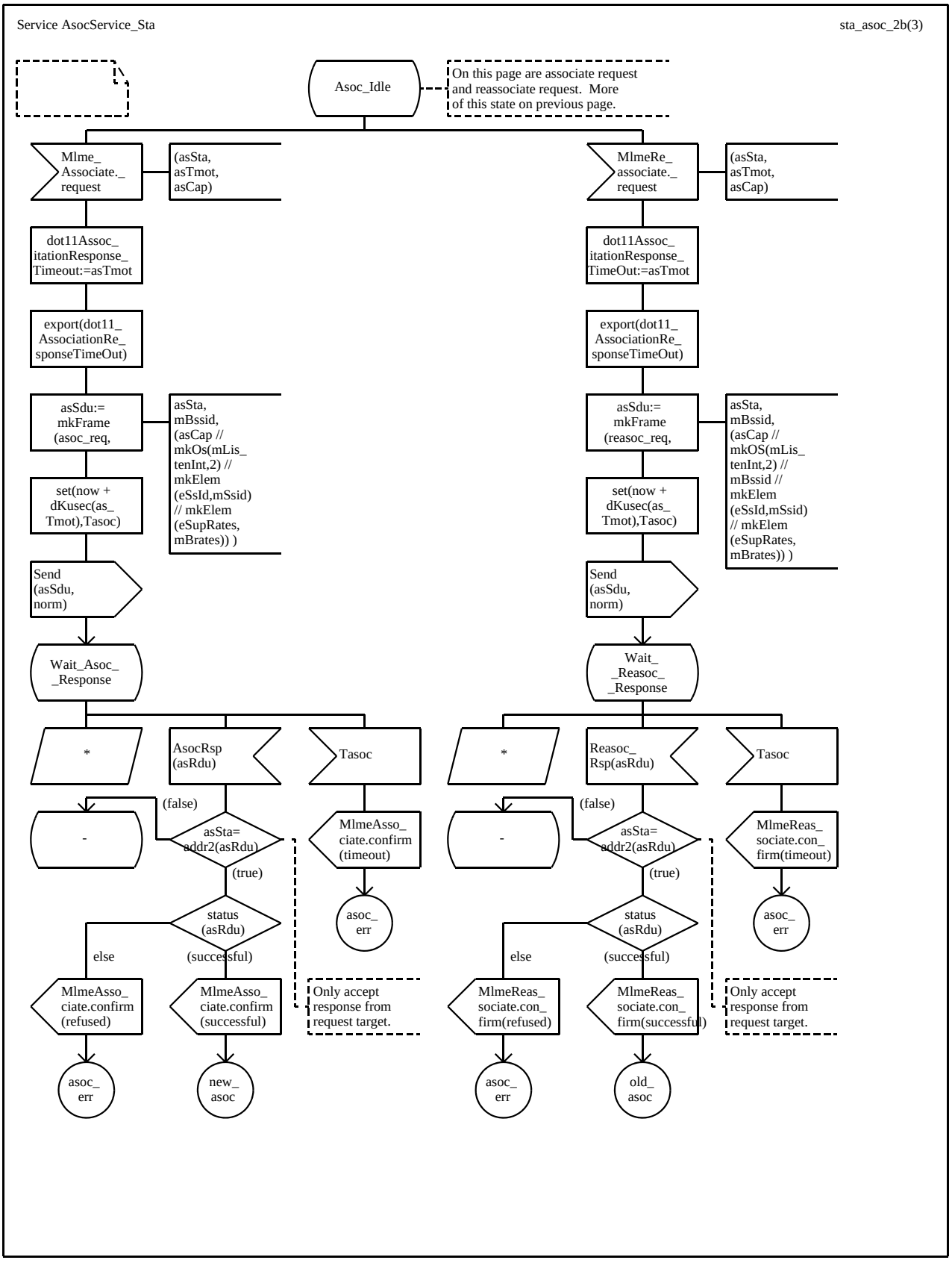




Copyright © 2007 IEEE. All rights reserved.

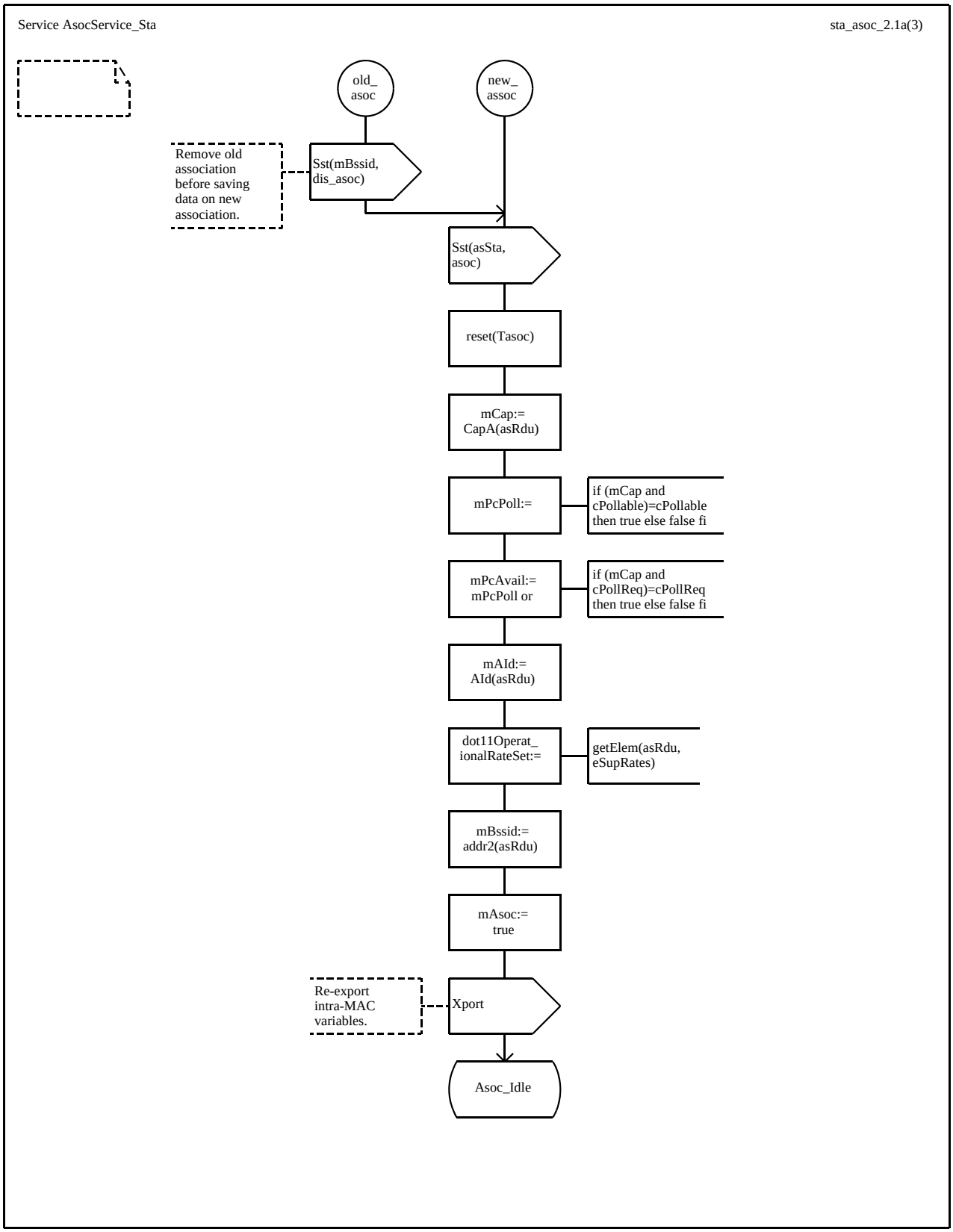



Copyright © 2007 IEEE. All rights reserved.

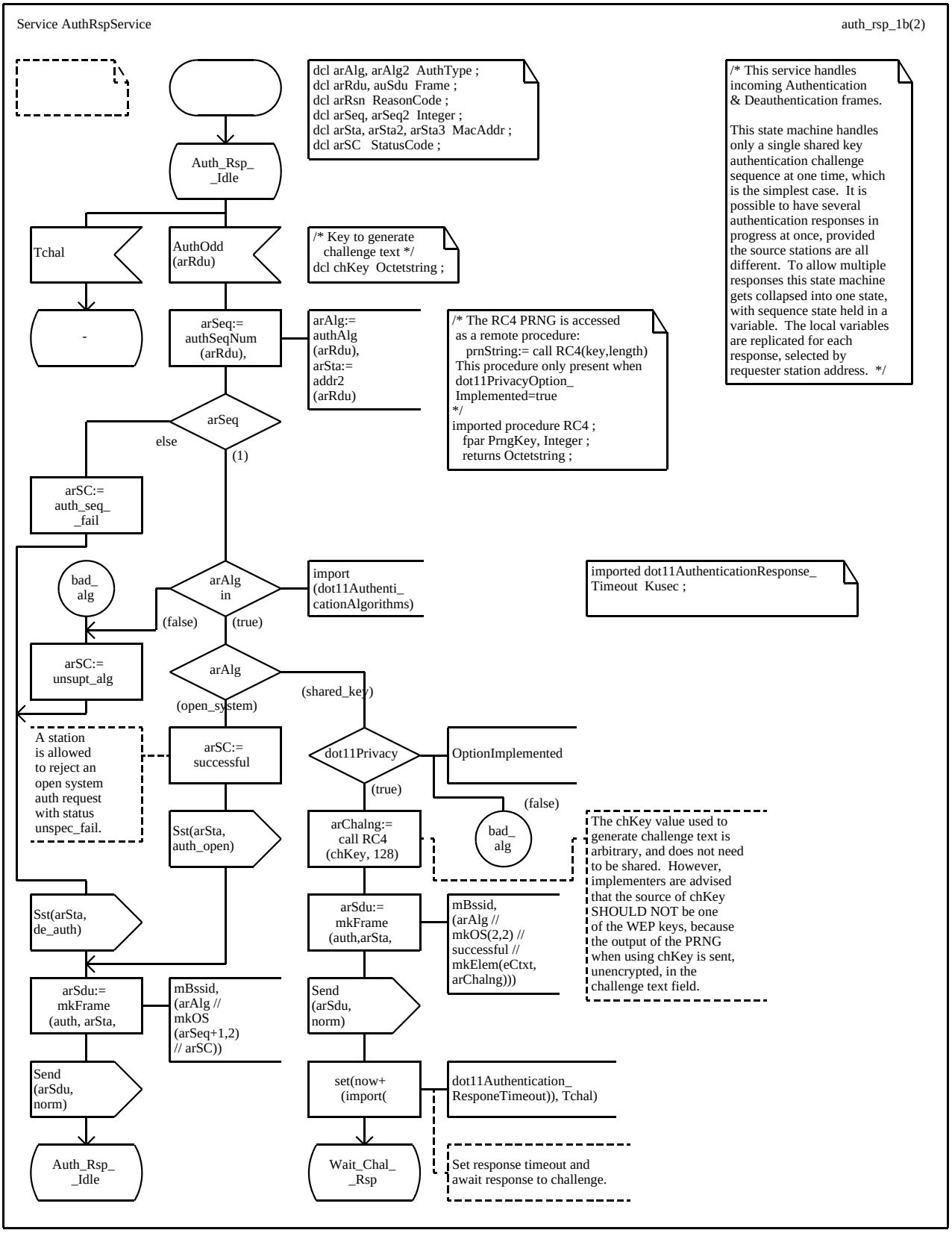




Copyright © 2007 IEEE. All rights reserved.

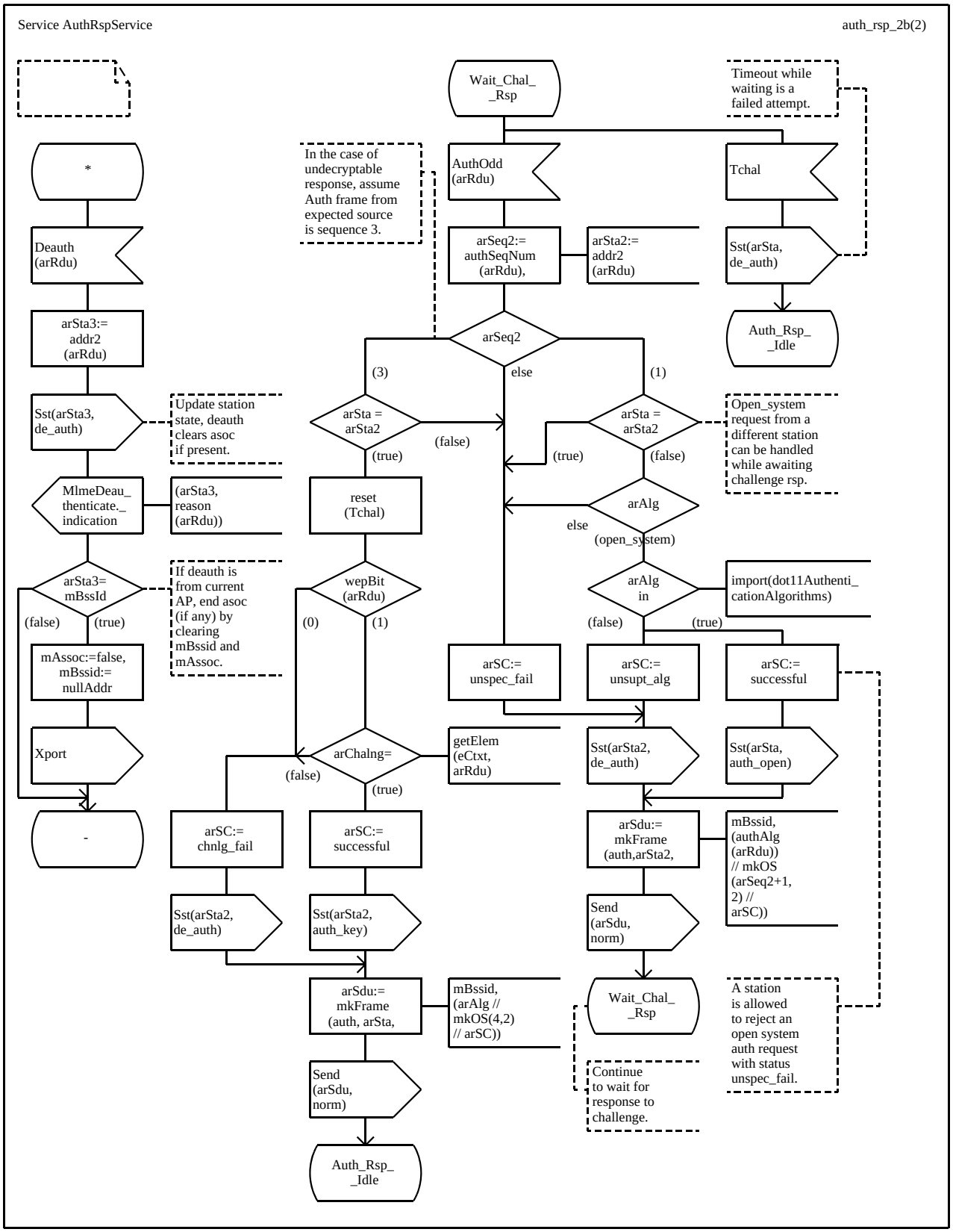



Copyright © 2007 IEEE. All rights reserved.

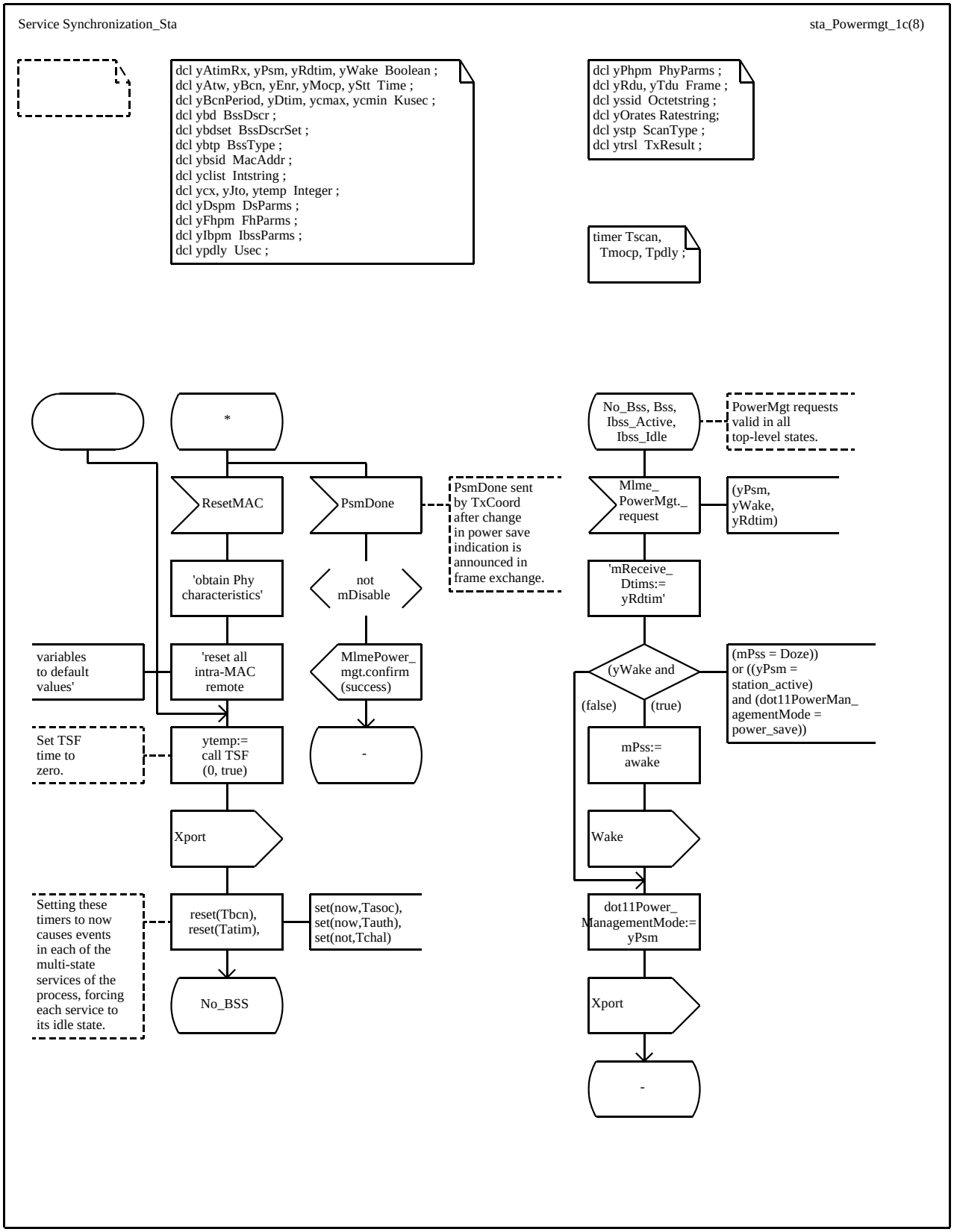




Copyright © 2007 IEEE. All rights reserved.

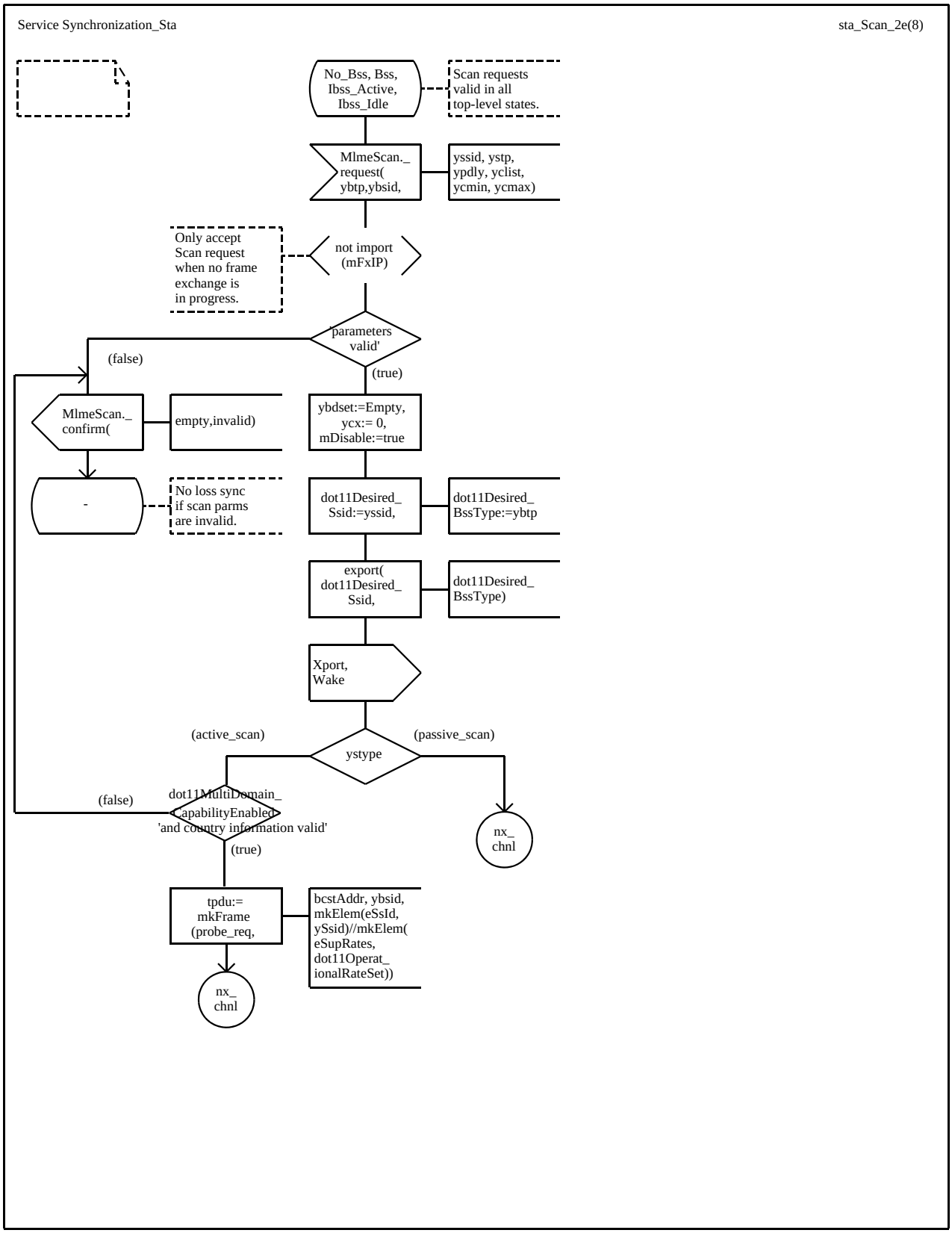



Copyright © 2007 IEEE. All rights reserved.

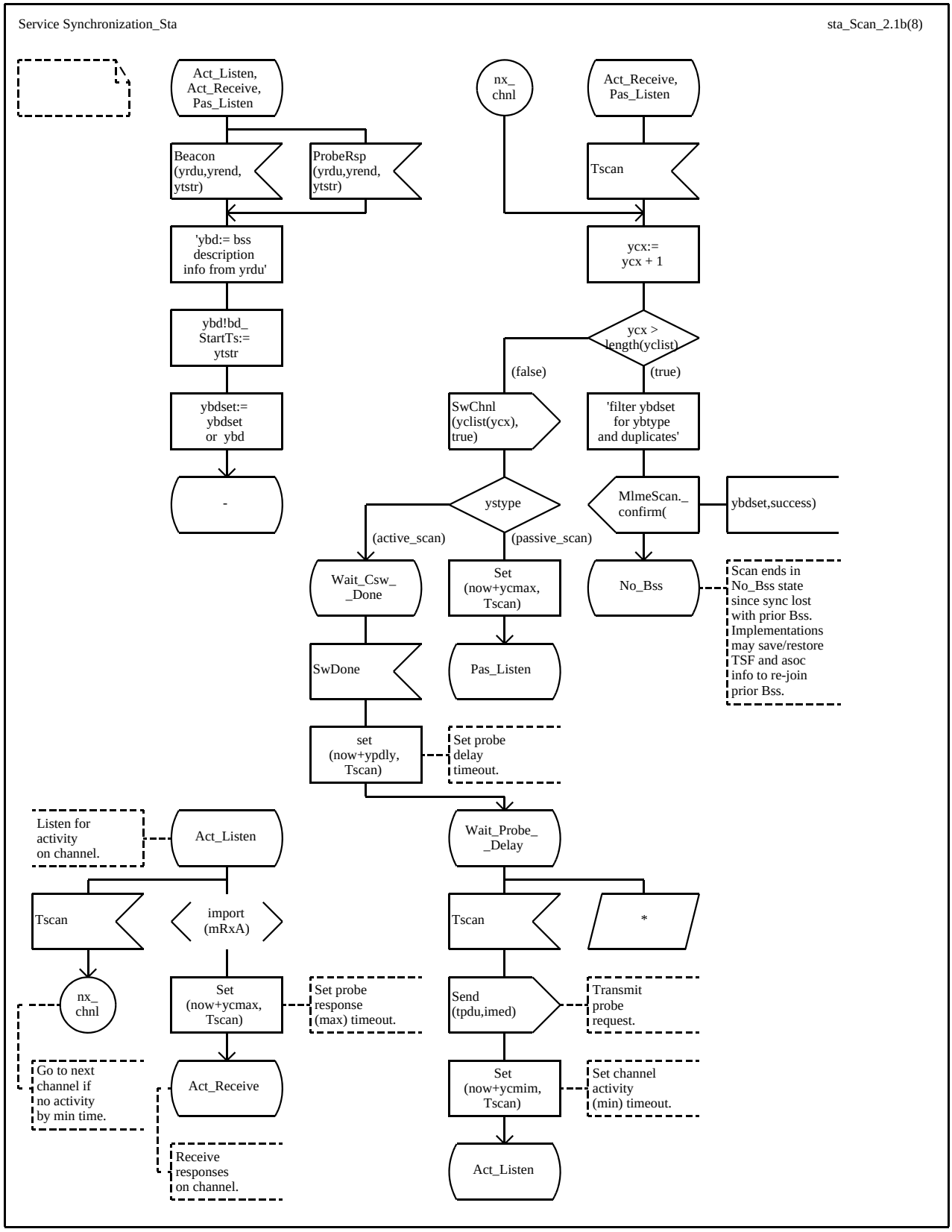




Copyright © 2007 IEEE. All rights reserved.

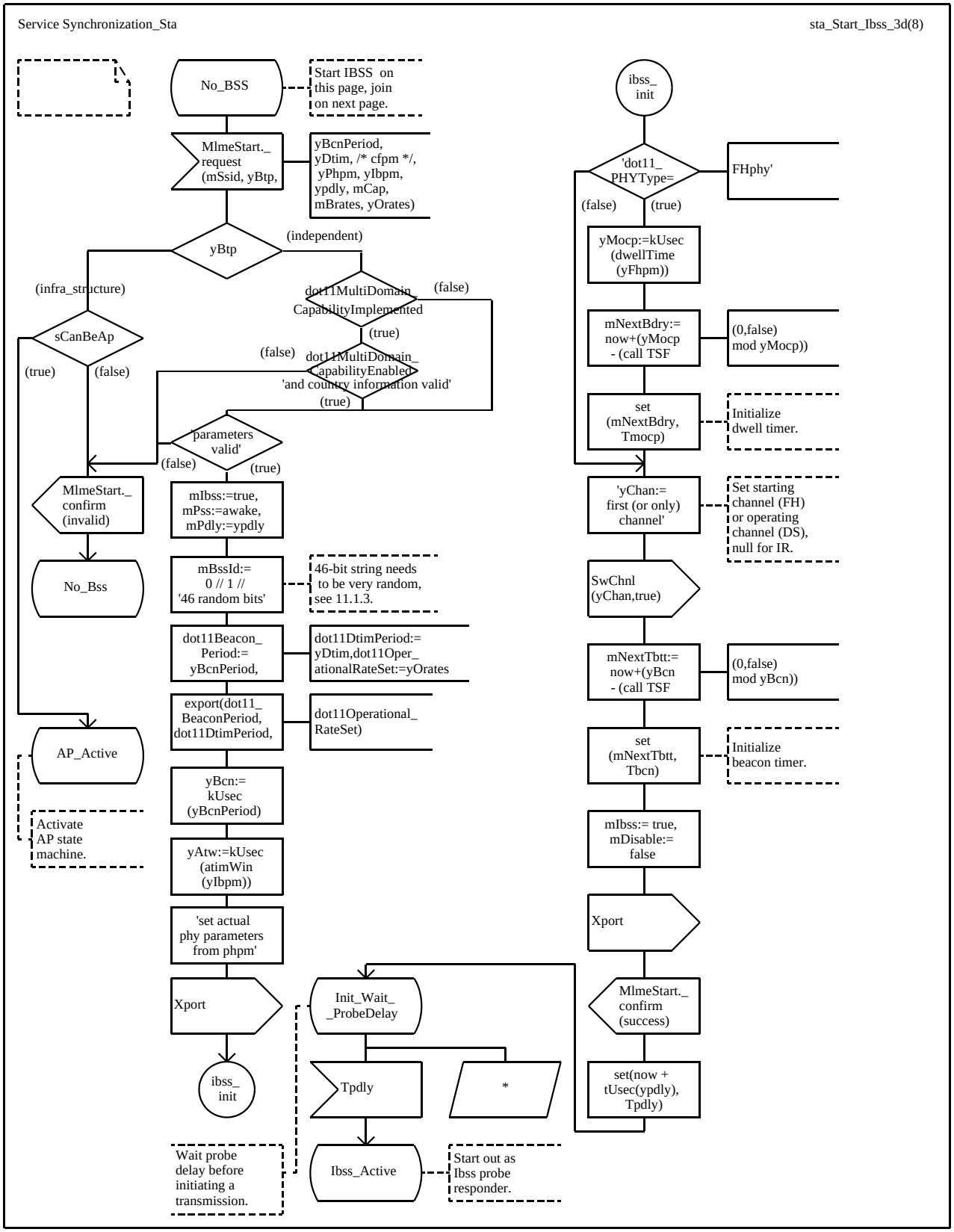



Copyright © 2007 IEEE. All rights reserved.

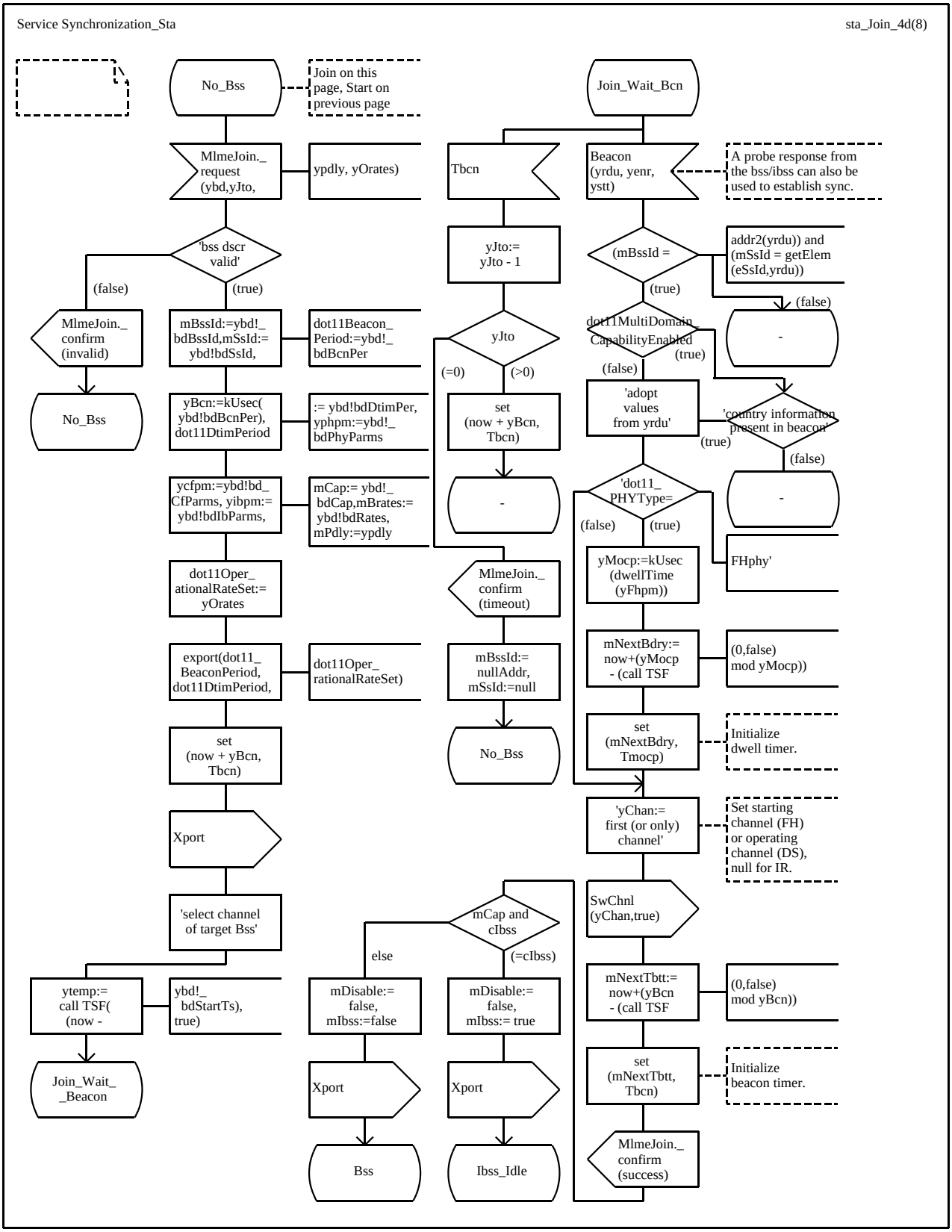




Copyright © 2007 IEEE. All rights reserved.

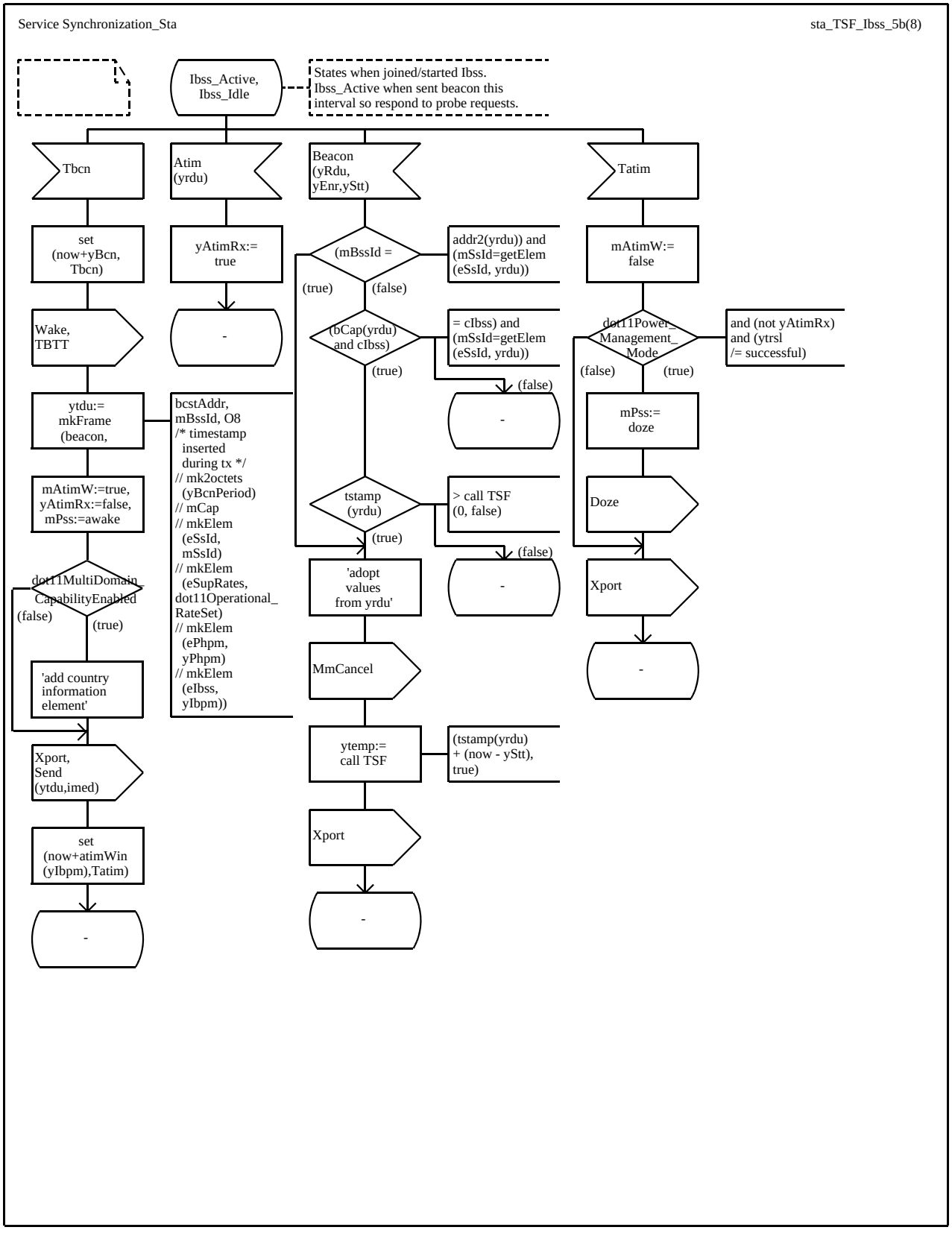



Copyright © 2007 IEEE. All rights reserved.

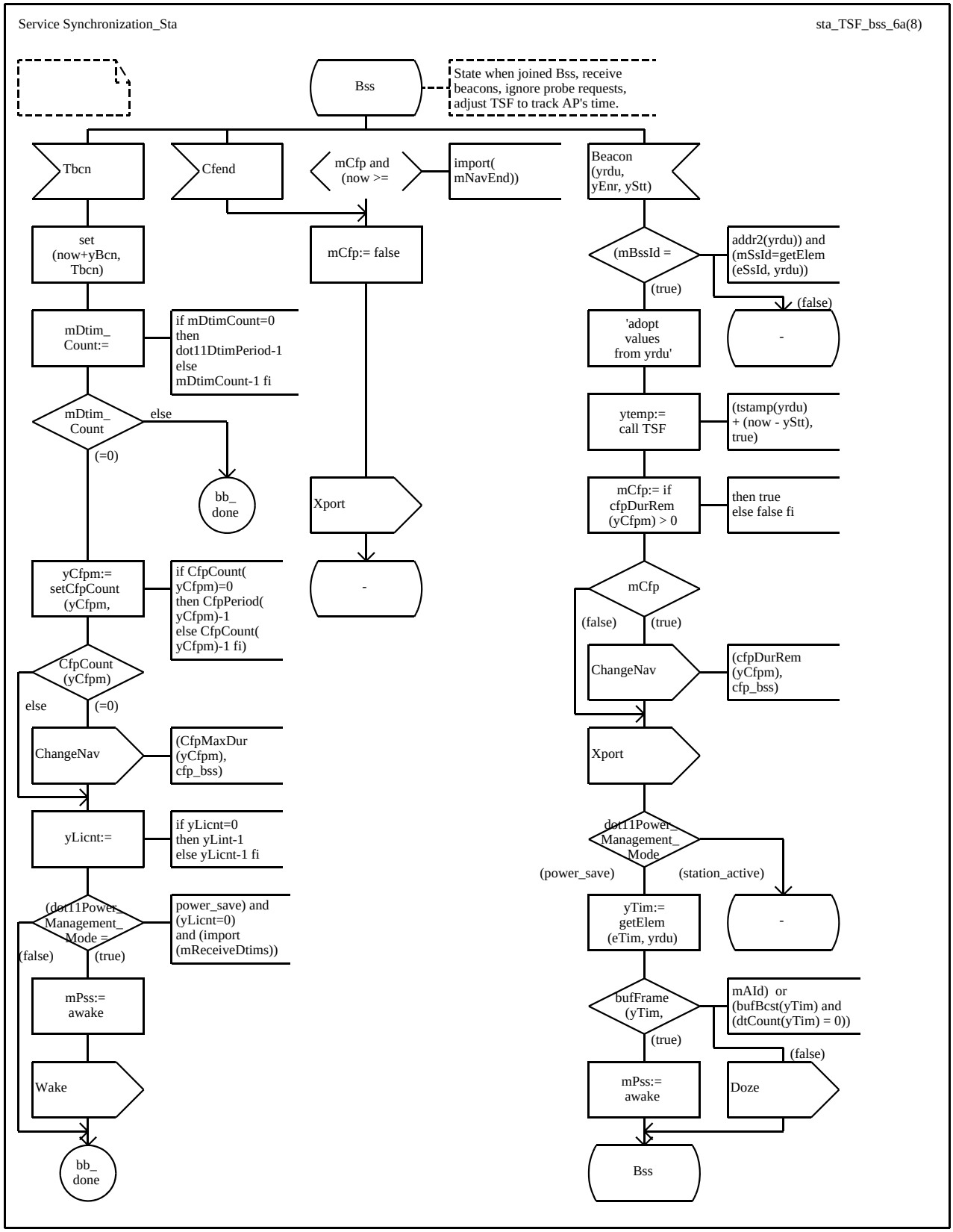




Copyright © 2007 IEEE. All rights reserved.

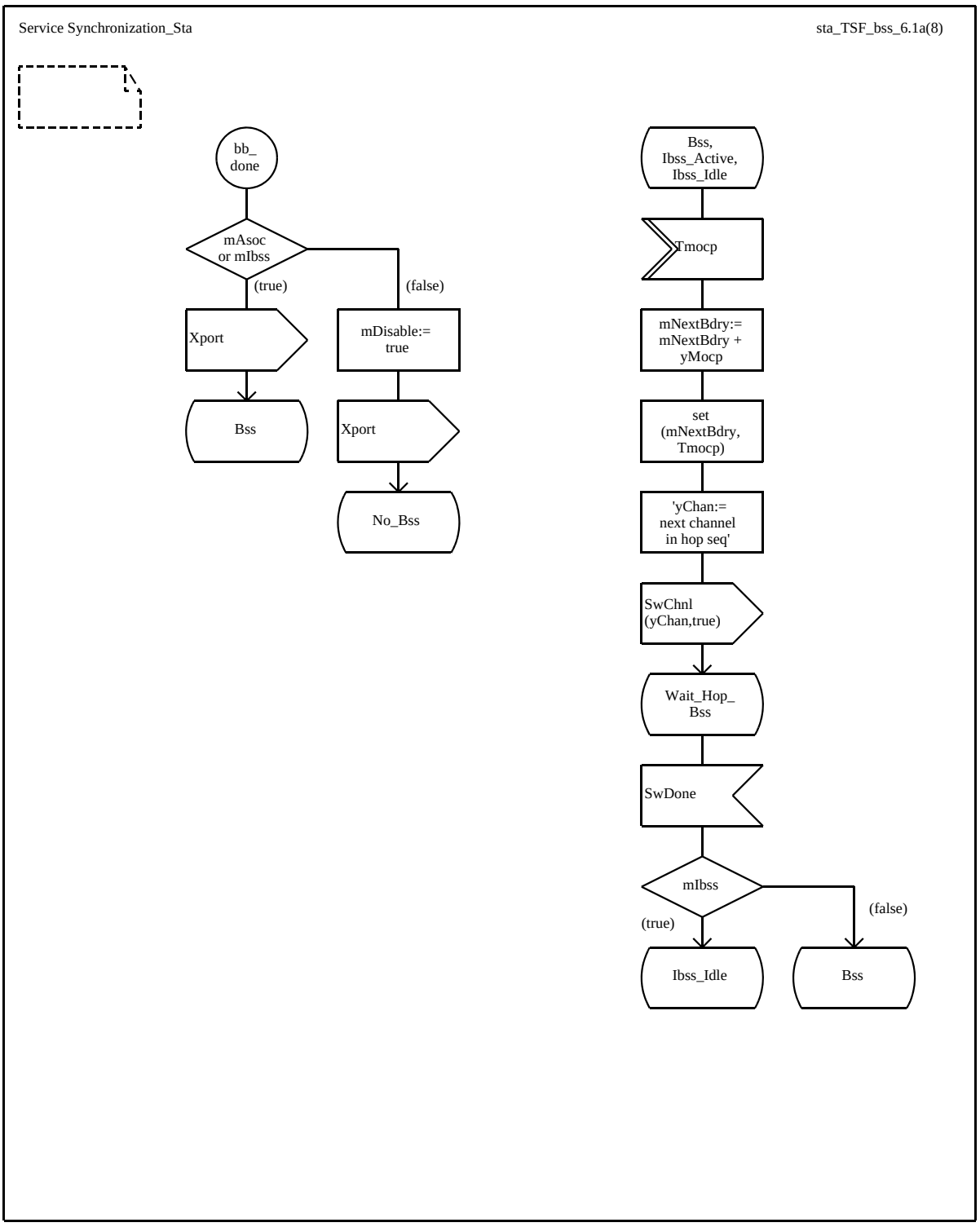



Copyright © 2007 IEEE. All rights reserved.

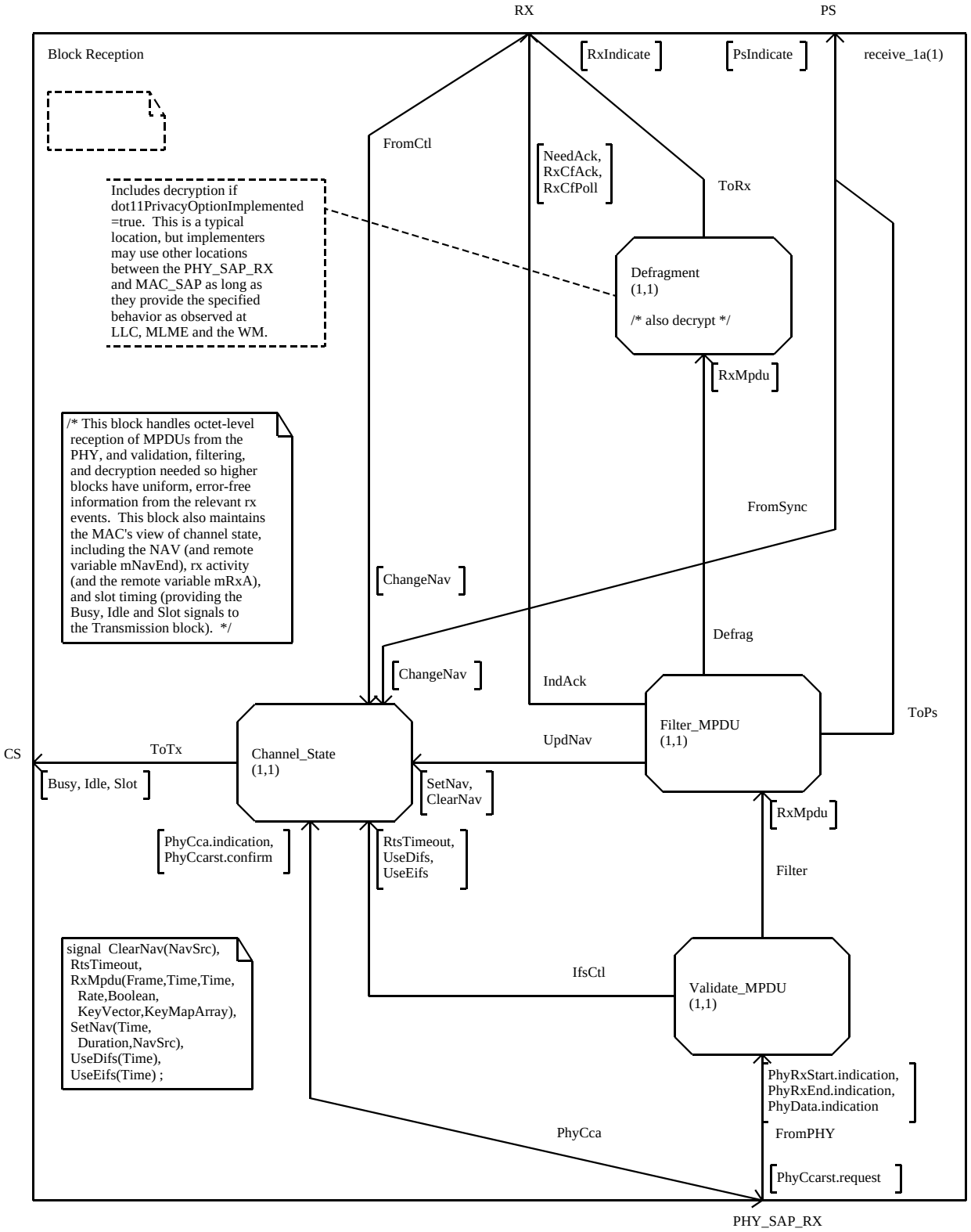




Copyright © 2007 IEEE. All rights reserved.

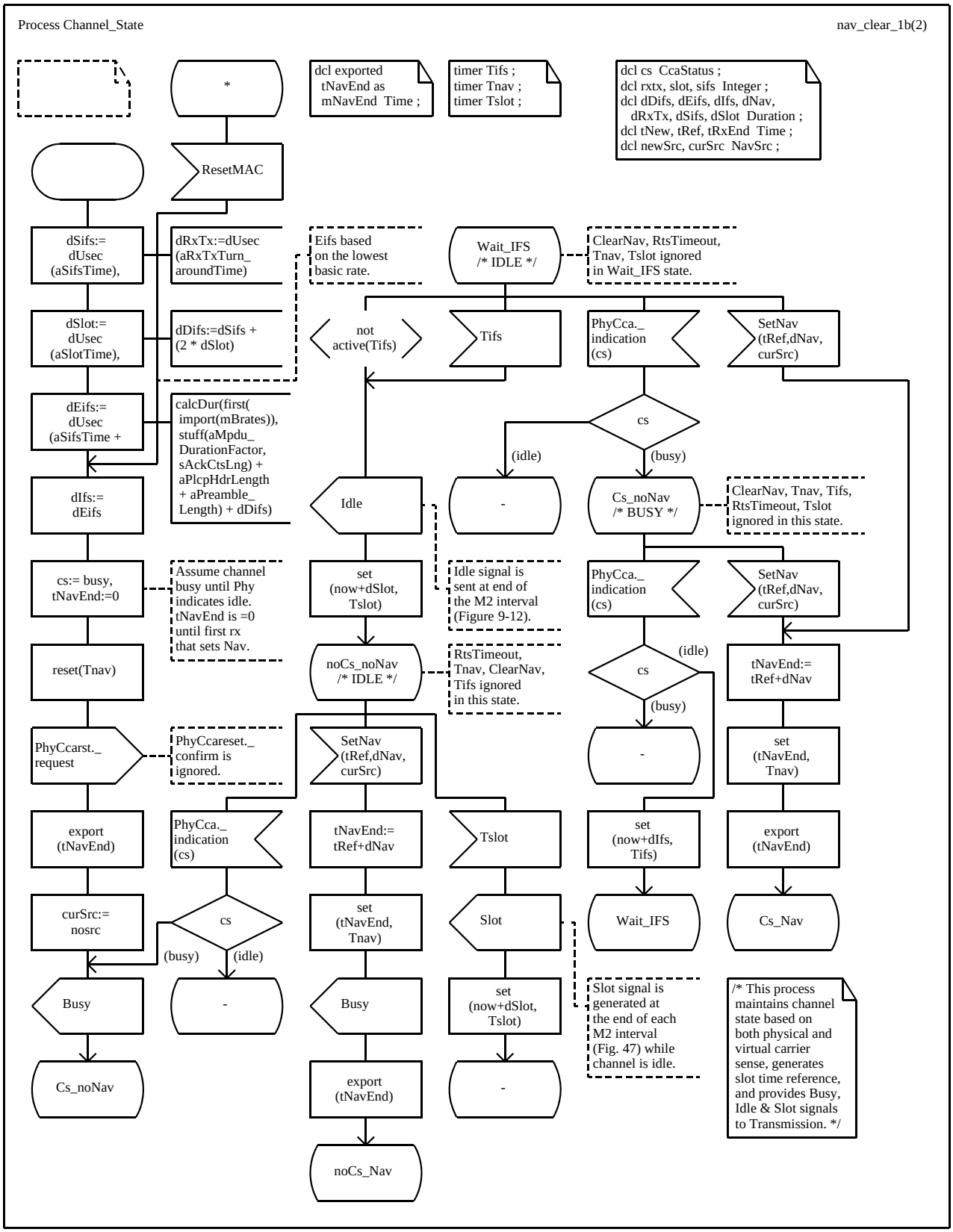



Copyright © 2007 IEEE. All rights reserved.

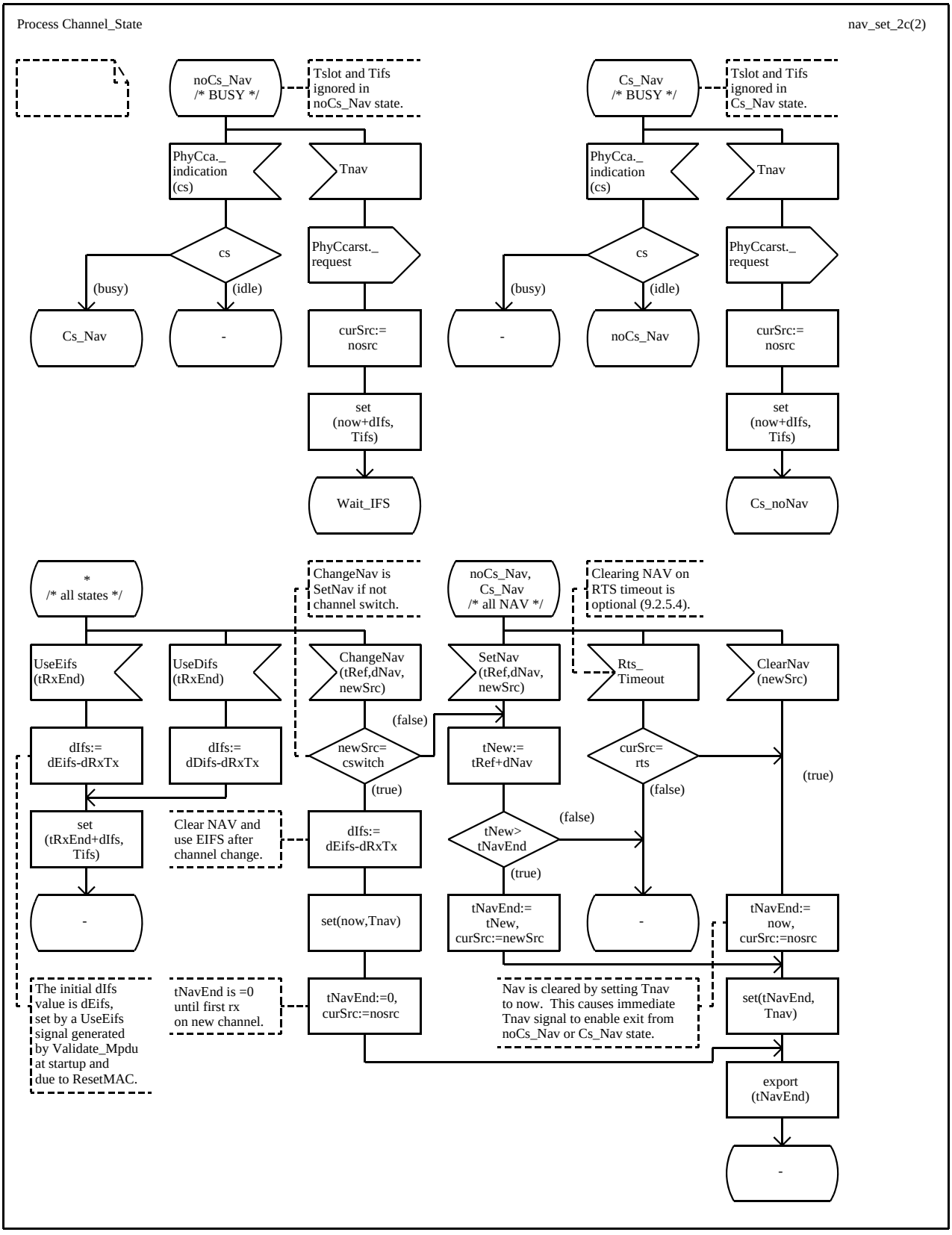




Copyright © 2007 IEEE. All rights reserved.

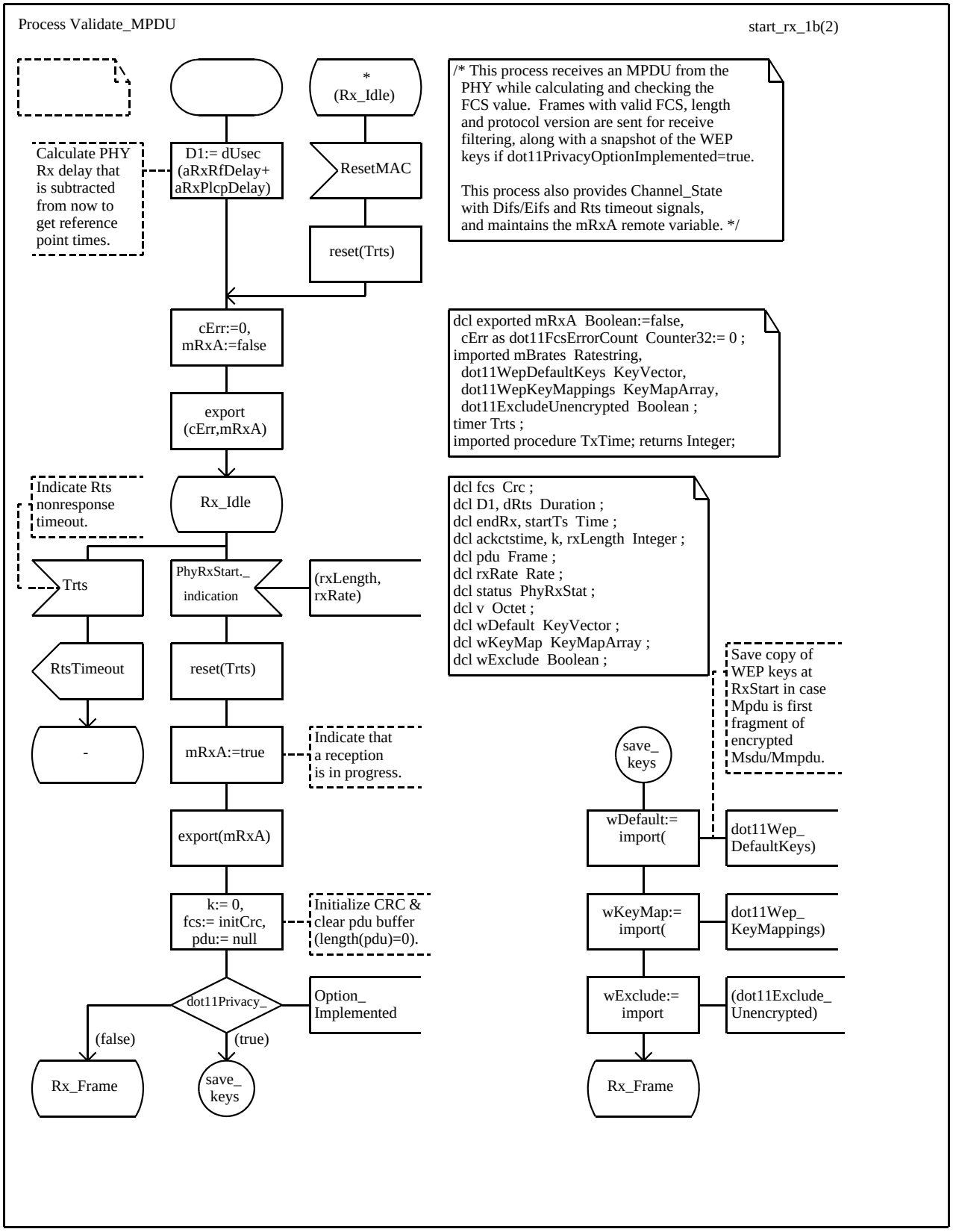



Copyright © 2007 IEEE. All rights reserved.

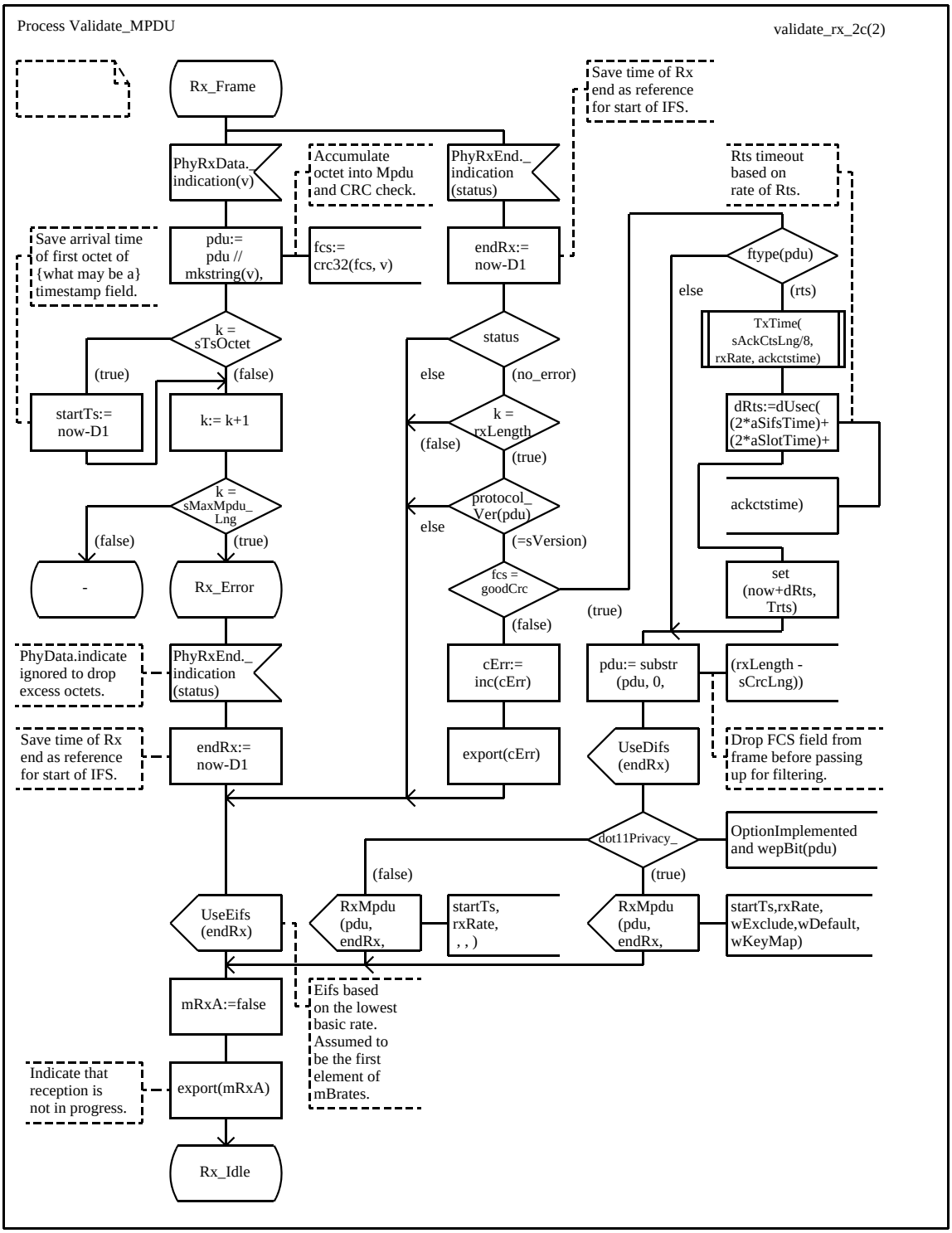




Copyright © 2007 IEEE. All rights reserved.

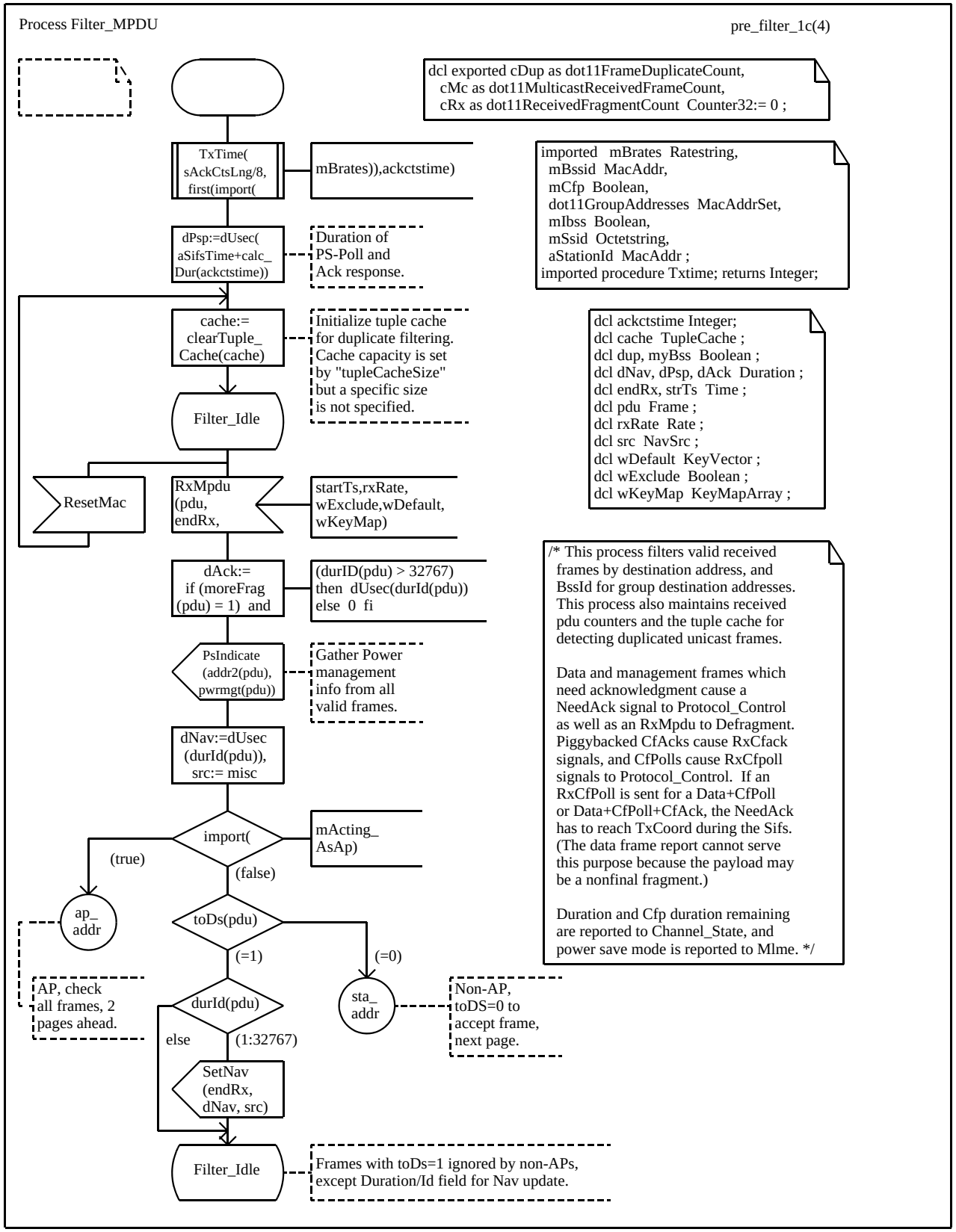



Copyright © 2007 IEEE. All rights reserved.

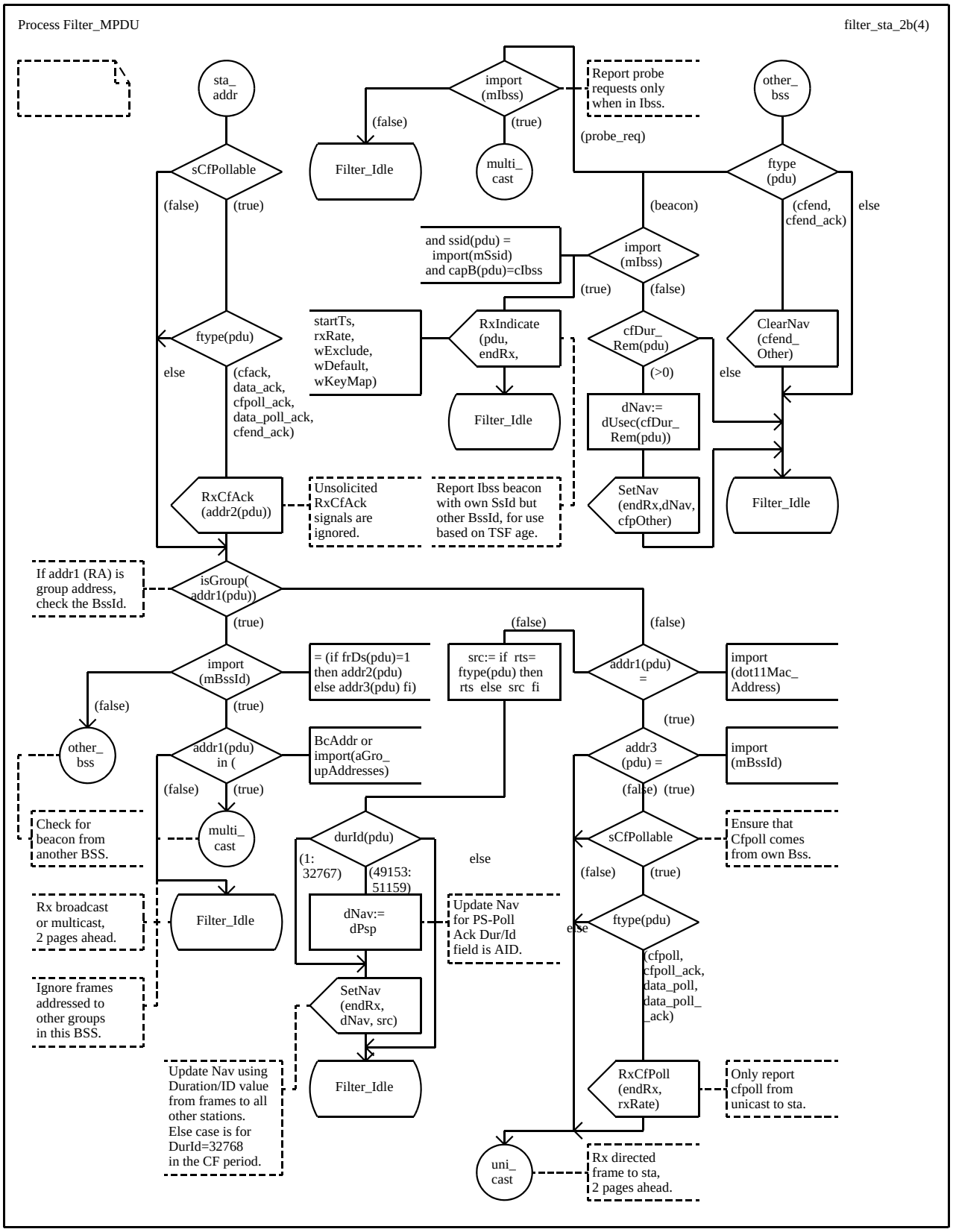




Copyright © 2007 IEEE. All rights reserved.

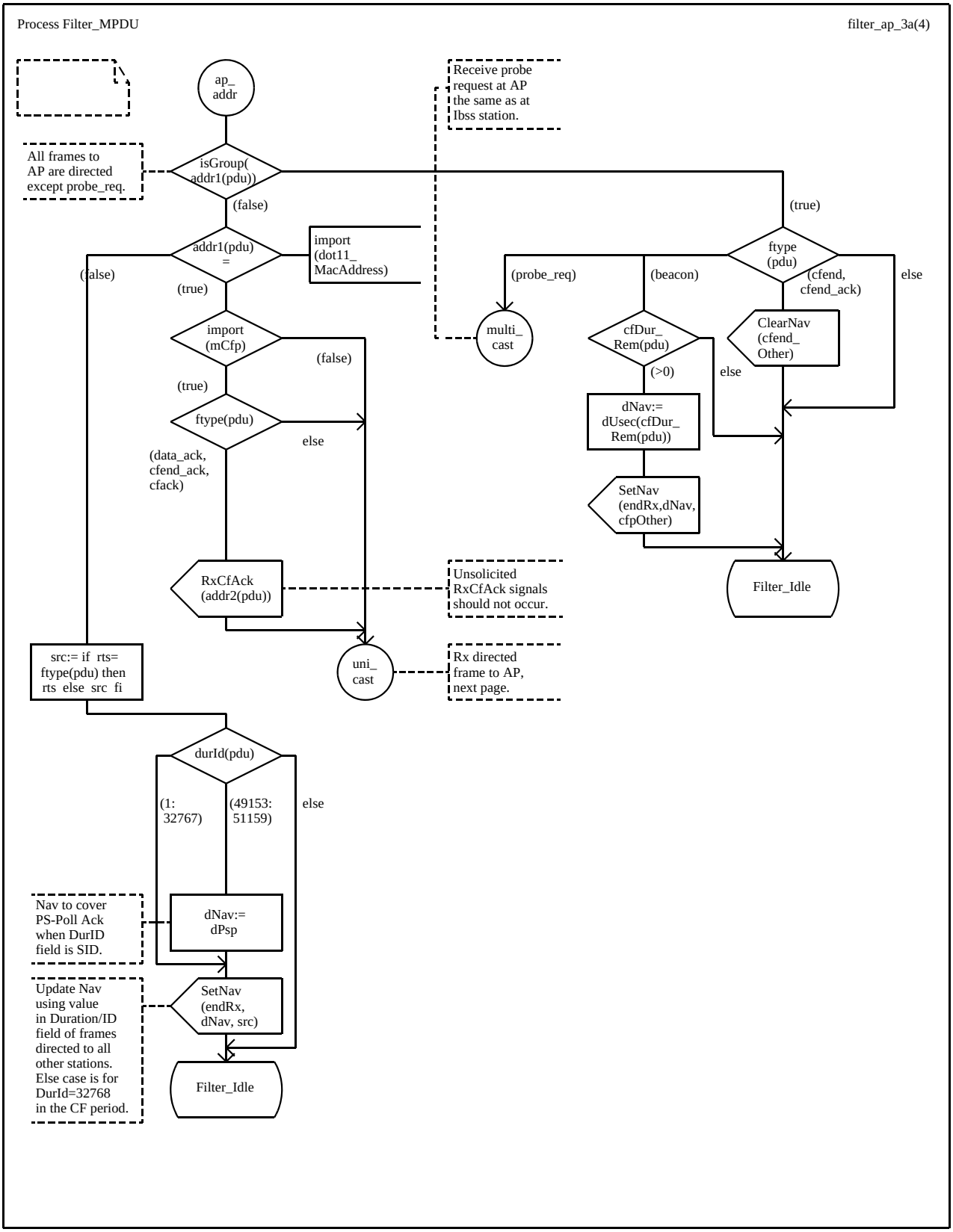



Copyright © 2007 IEEE. All rights reserved.

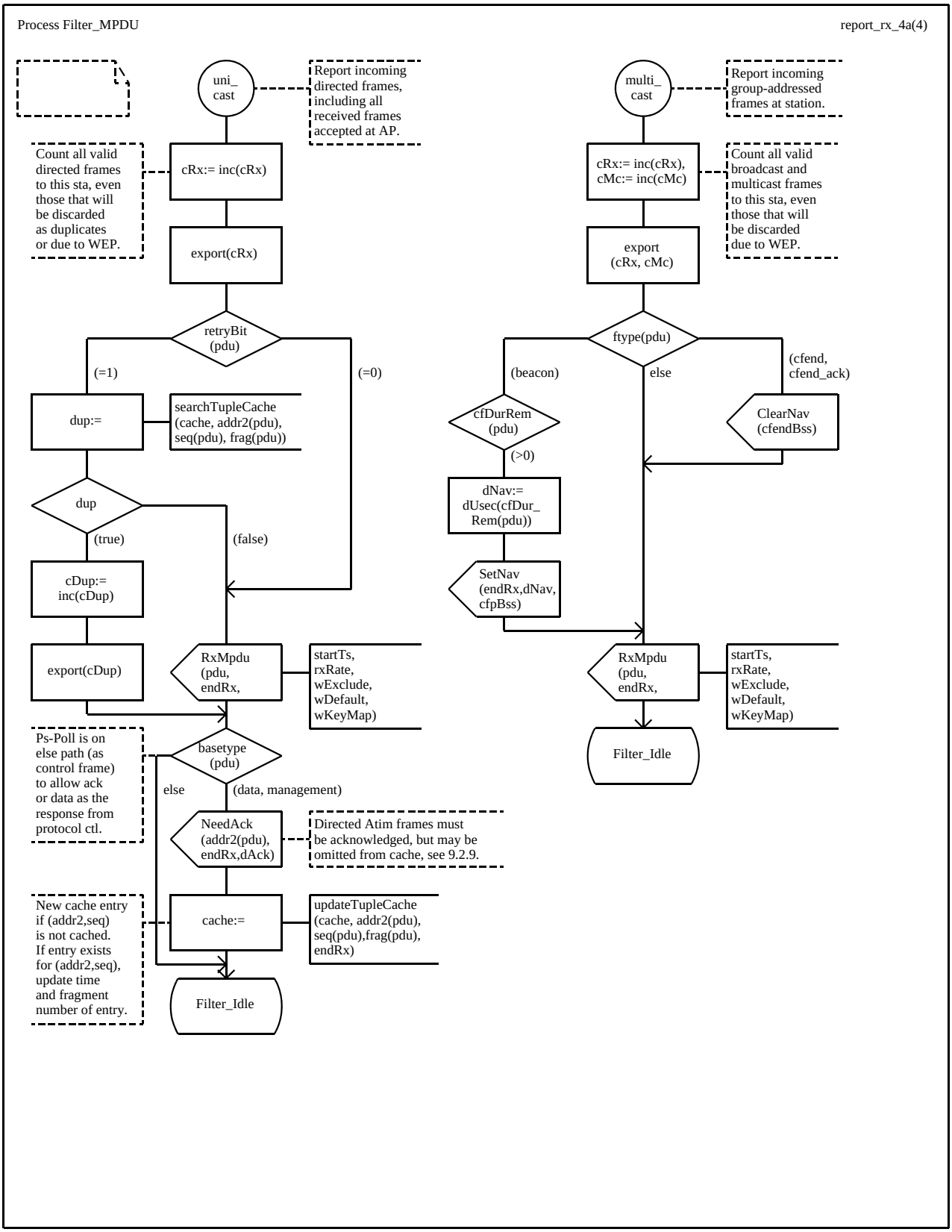




Copyright © 2007 IEEE. All rights reserved.

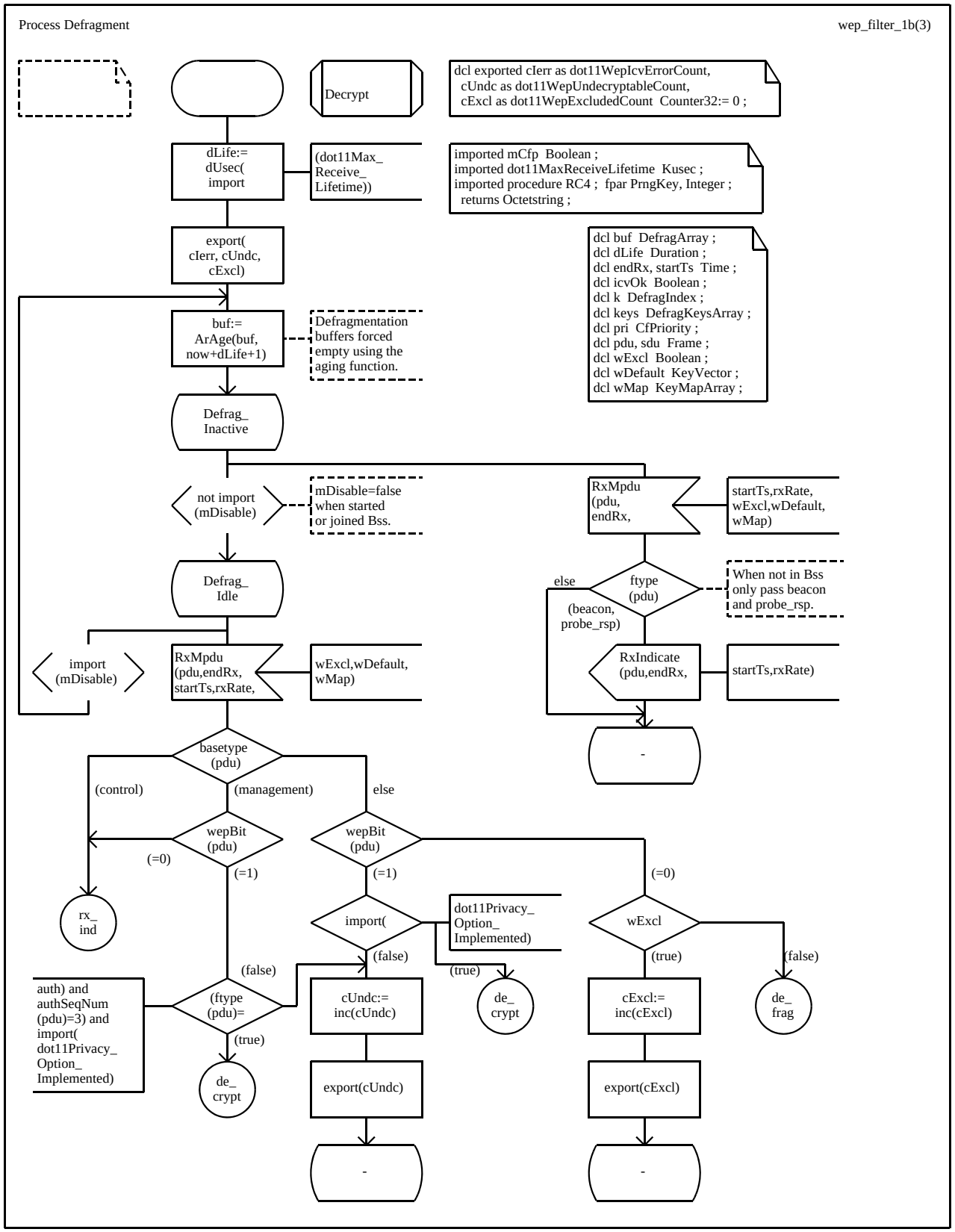



Copyright © 2007 IEEE. All rights reserved.

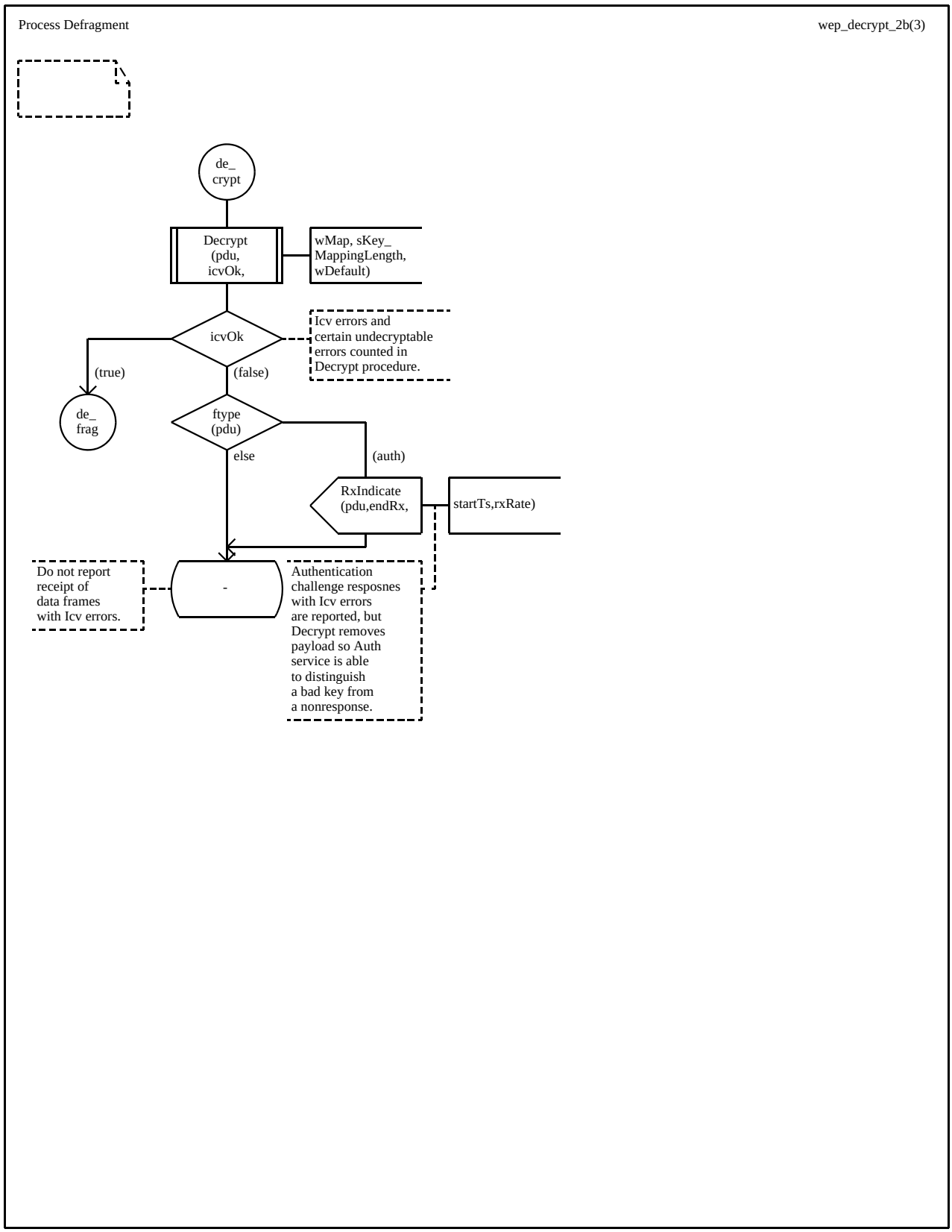




Copyright © 2007 IEEE. All rights reserved.

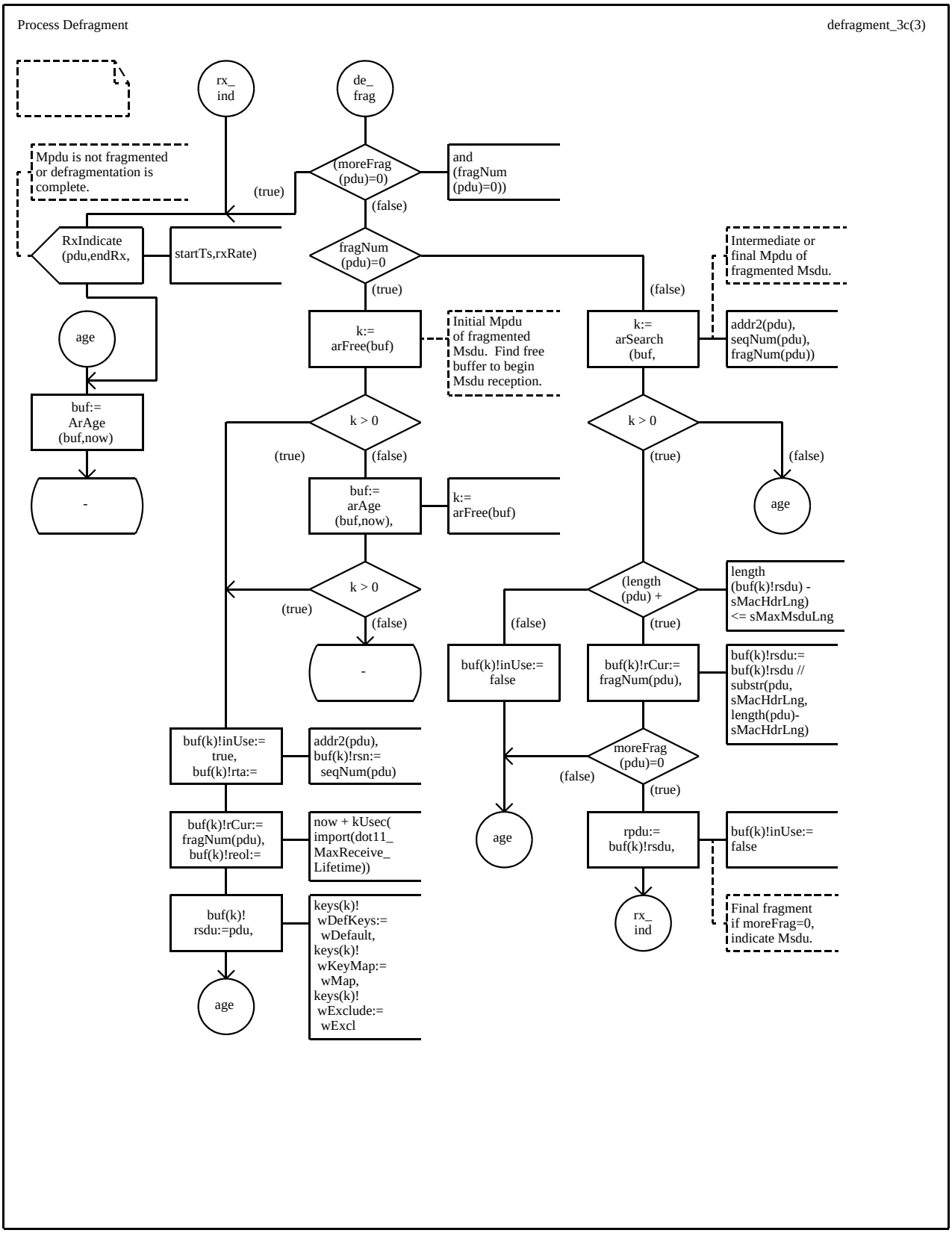



Copyright © 2007 IEEE. All rights reserved.

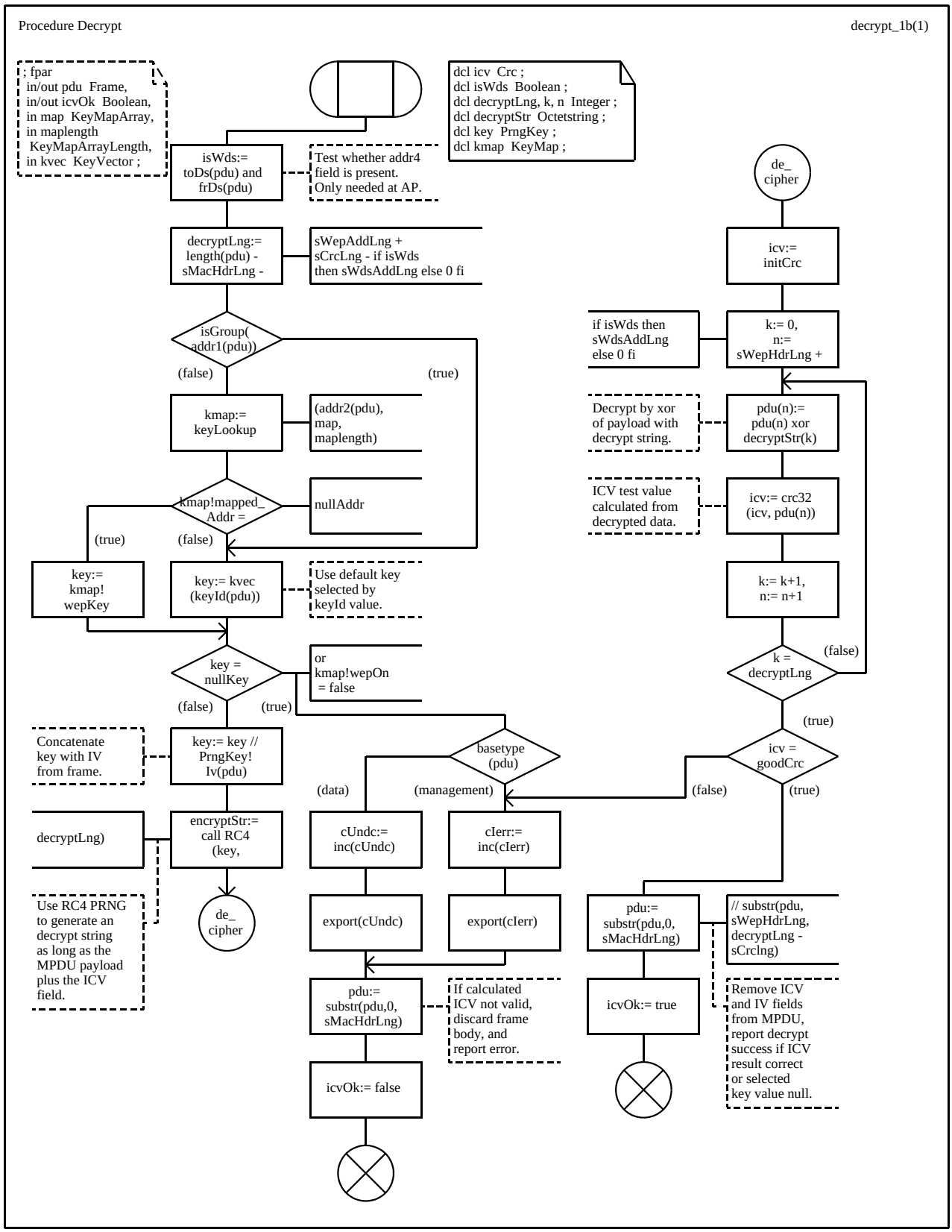




Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.

## C.4 State machines for MAC AP

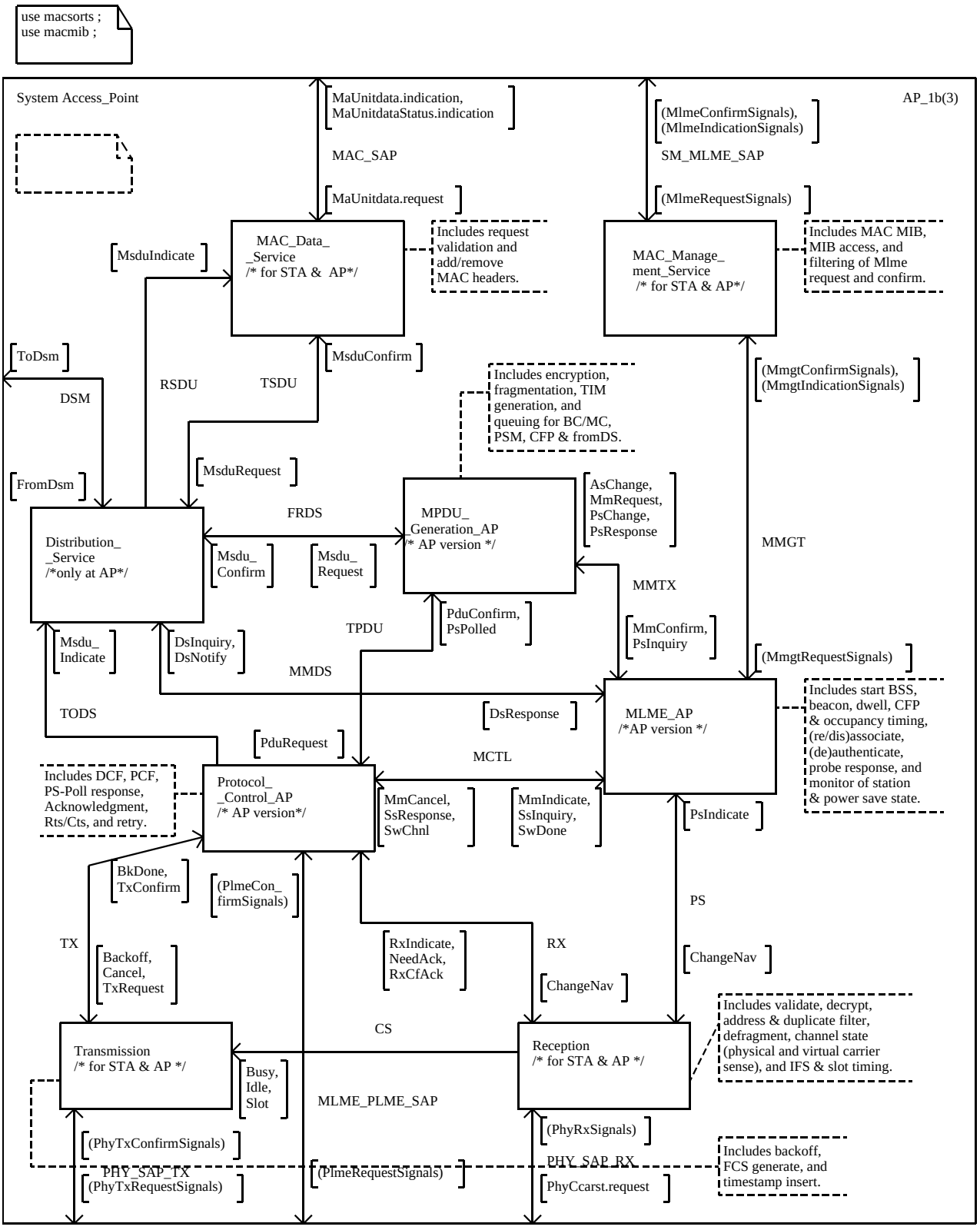
The following SDL-92 system specification defines operation of the MAC protocol at an IEEE 802.11 AP. Many aspects of AP operation are identical to the STA operation. These are defined in blocks and processes referenced from both the STA and AP system specifications. Blocks and processes used in both STA and AP are identifiable by the SDL comment /* for STA & AP */ below the block or process name. Blocks and processes specific to AP operation are identifiable by the SDL comment /* AP version */ below the block or process name. Definitions for the /* AP version */ and the /* STA & AP */ blocks and processes appear in this subclause.

The remainder of this subclause is the formal description, in SDL/GR, of an IEEE 802.11 AP.

This subclause describes the security behavior of only 8.2.1 and 8.2.2.

This subclause does not describe the behavior of a STA with QoS facility.

Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.

```
use macsorts ;
use macmib ;
```

System Access_Point

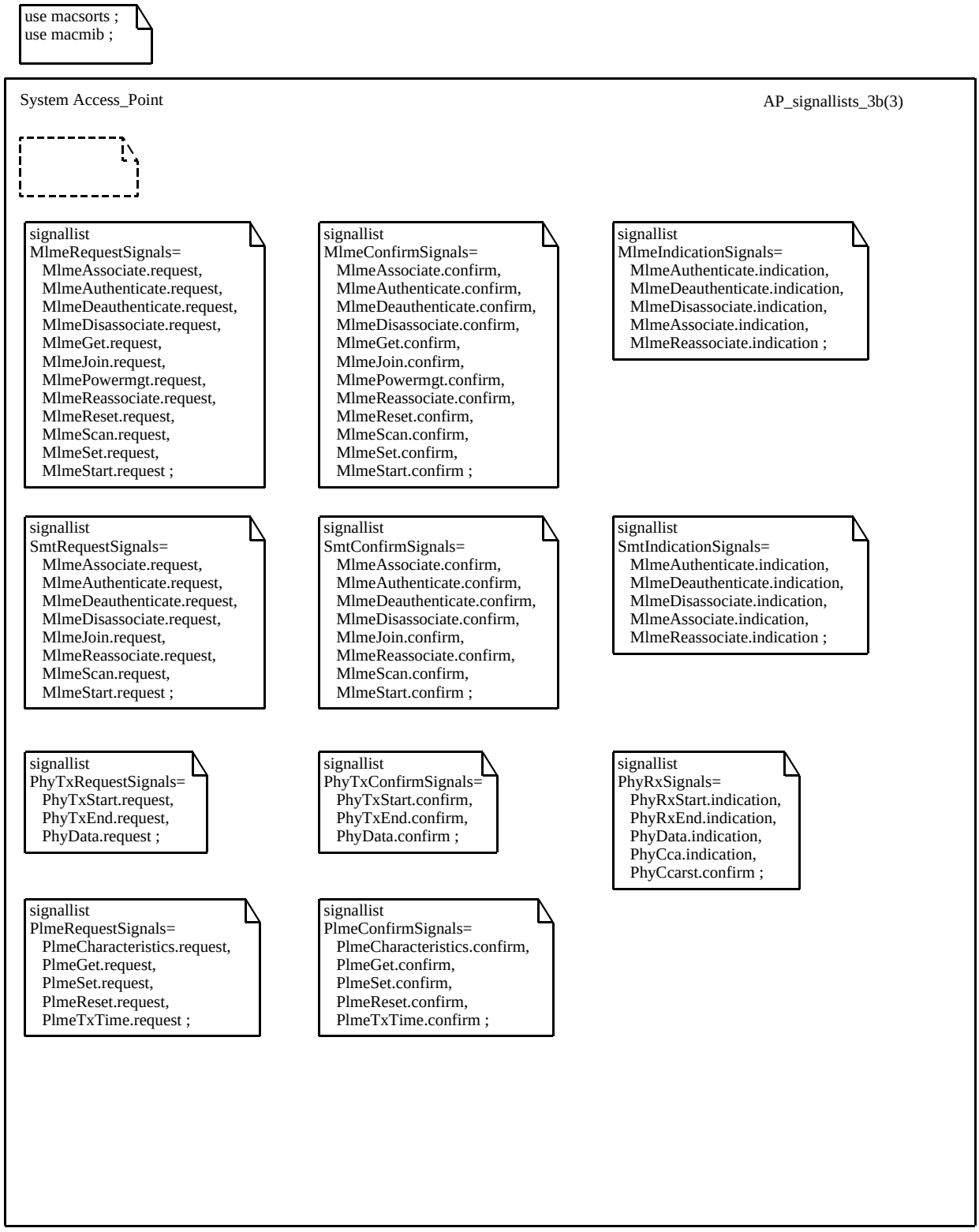
AP_signals_2d(3)

```
newtype DsStatus  literals
  assoc, disassoc, reassoc, unknown
endnewtype DsStatus ;
```

```
signal
  AsChange(Frame,DsStatus),
  Backoff(Integer,Integer),
  BkDone(Integer),
  Busy,
  Cancel,
  ChangeNav(Time,Duration,NavSrc),
  DsInquiry(MacAddr,MacAddr),
  DsNotify(MacAddr,DsStatus),
  DsResponse(MacAddr,MacAddr,DsStatus),
  FromDsm(MacAddr,MacAddr,Octetstring),
  Idle,
  MaUnitdata.indication(MacAddr,MacAddr,
    Routing,Octetstring,RxStatus,
    CfPriority,ServiceClass),
  MaUnitdata.request(MacAddr,MacAddr,
    Routing,Octetstring,CfPriority,ServiceClass),
  MaUnitdataStatus.indication(MacAddr,
    MacAddr,TxStatus,CfPriority,ServiceClass),
  MlmeAssociate.confirm(MlmeStatus),
  MlmeAssociate.indication(MacAddr),
  MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeAuthenticate.confirm
    (MacAddr,AuthType,MlmeStatus),
  MlmeAuthenticate.indication(MacAddr,AuthType),
  MlmeAuthenticate.request(MacAddr,AuthType,Kusec),
  MlmeDeauthenticate.confirm(MacAddr,MlmeStatus),
  MlmeDeauthenticate.indication(MacAddr,ReasonCode),
  MlmeDeauthenticate.request(MacAddr,ReasonCode),
  MlmeDisassociate.confirm(MlmeStatus),
  MlmeDisassociate.indication(MacAddr,ReasonCode),
  MlmeDisassociate.request(MacAddr,ReasonCode),
  MlmeGet.confirm(MibStatus,MibAtrib,MibValue),
  MlmeGet.request(MibAtrib),
  MlmeJoin.confirm(MlmeStatus),
  MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
  MlmePowermgt.confirm(MlmeStatus),
  MlmePowermgt.request(PwrSave,Boolean,Boolean),
  MlmeReassociate.confirm(MlmeStatus),
  MlmeReassociate.indication(MacAddr),
  MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
  MlmeReset.confirm(MlmeStatus),
  MlmeReset.request,
  MlmeScan.confirm(BssDscrSet,MlmeStatus),
  MlmeScan.request(BssTypeSet,MacAddr,Octetstring,
    ScanType,Usec,Intstring,Kusec,Kusec),
  MlmeSet.confirm(MibStatus,MibAtrib),
  MlmeSet.request(MibAtrib,MibValue),
  MlmeStart.confirm(MlmeStatus),
  MlmeStart.request(Octetstring,BssType,Kusec,
    Integer,CfParms,PhyParms,IbssParms,Usec,
    Capability,Ratestring,Ratestring) ;
```

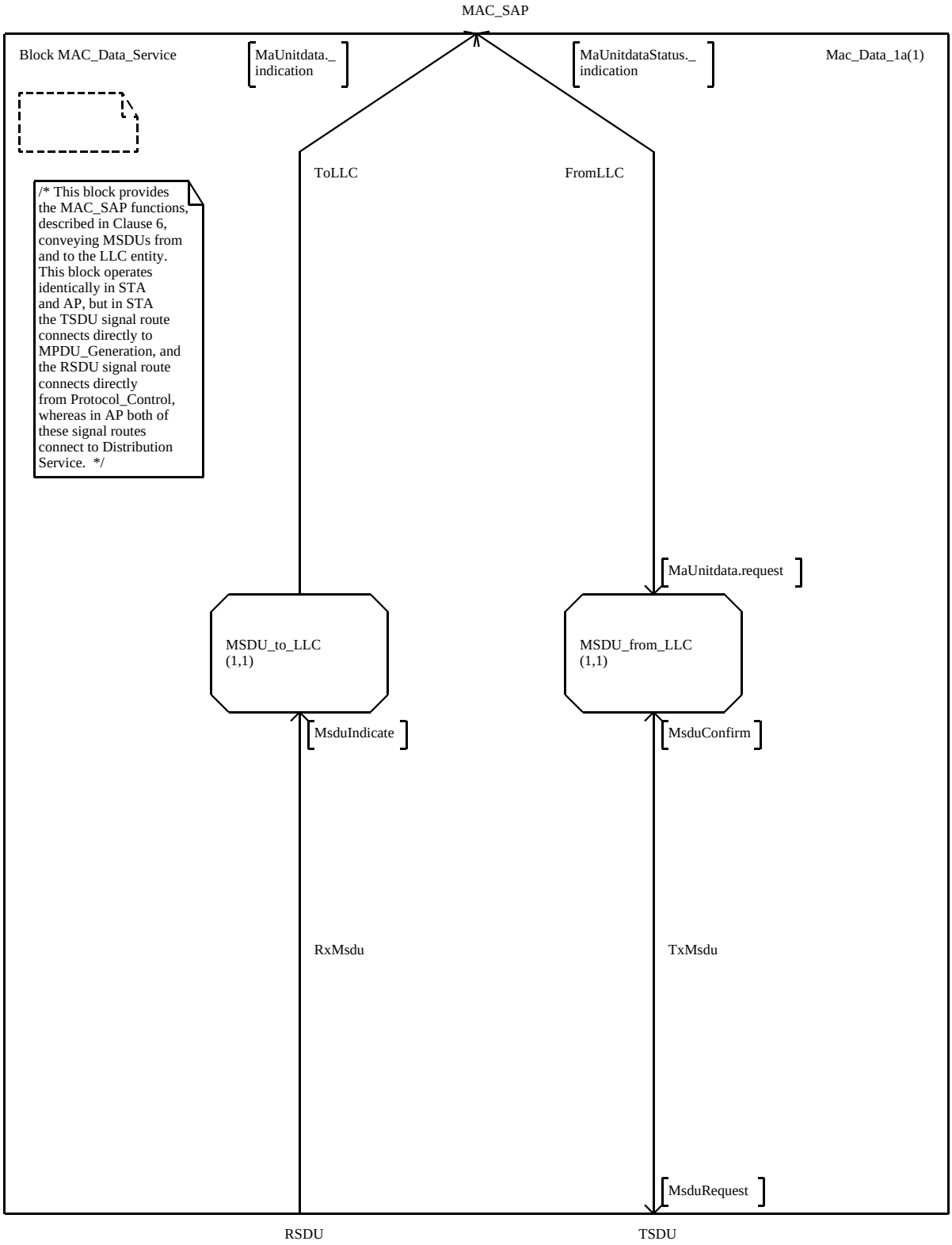
```
signal
  MmCancel,
  MmConfirm(Frame,TxStatus),
  MmIndicate(Frame,Time,Time,StateErr),
  MmRequest(Frame,Imed,Rate),
  MsduConfirm(Frame,CfPriority,TxStatus),
  MsduIndicate(Frame,CfPriority),
  MsduRequest(Frame,CfPriority),
  NeedAck(MacAddr,Time,Duration,Rate),
  PduConfirm(FragSdu,TxResult),
  PduRequest(FragSdu),
  PhyCca.indication(Ccastatus),
  PhyCcarst.confirm,
  PhyCcarst.request,
  PhyData.confirm,
  PhyData.indication(Octet),
  PhyData.request(Octet),
  PhyRxEnd.indication(PhyRxStat),
  PhyRxStart.indication(Integer,Rate),
  PhyTxEnd.confirm,
  PhyTxEnd.request,
  PhyTxStart.confirm,
  PhyTxStart.request(Integer,Rate),
  PlmeCharacteristics.confirm(PhyChrstcs),
  PlmeCharacteristics.request,
  PlmeGet.confirm(MibStatus,
    MibAtrib,MibValue),
  PlmeGet.request(MibAtrib),
  PlmeReset.confirm(Boolean),
  PlmeReset.request,
  PlmeSet.confirm(MibStatus,MibAtrib),
  PlmeSet.request(MibAtrib,MibValue),
  PlmeTxTime.confirm(Integer),
  PlmeTxTime.request(Integer, Rate),
  PsmDone,
  PsPolled(MacAddr,AsocId),
  PsChange(MacAddr,PsMode),
  PsIndicate(MacAddr,PsMode),
  PsInquiry(MacAddr),
  PsResponse(MacAddr,PsMode),
  ResetMAC,
  RxCfAck(MacAddr),
  RxIndicate(Frame,Time,Time,Rate),
  Slot,
  SsInquiry(MacAddr),
  SsResponse(MacAddr,
    StationState,StationState),
  SwChnl(Integer,Boolean),
  SwDone,
  ToDsm(MacAddr,MacAddr,Octetstring),
  TxConfirm,
  TxRequest(Frame,Rate) ;
```

Copyright © 2007 IEEE. All rights reserved.

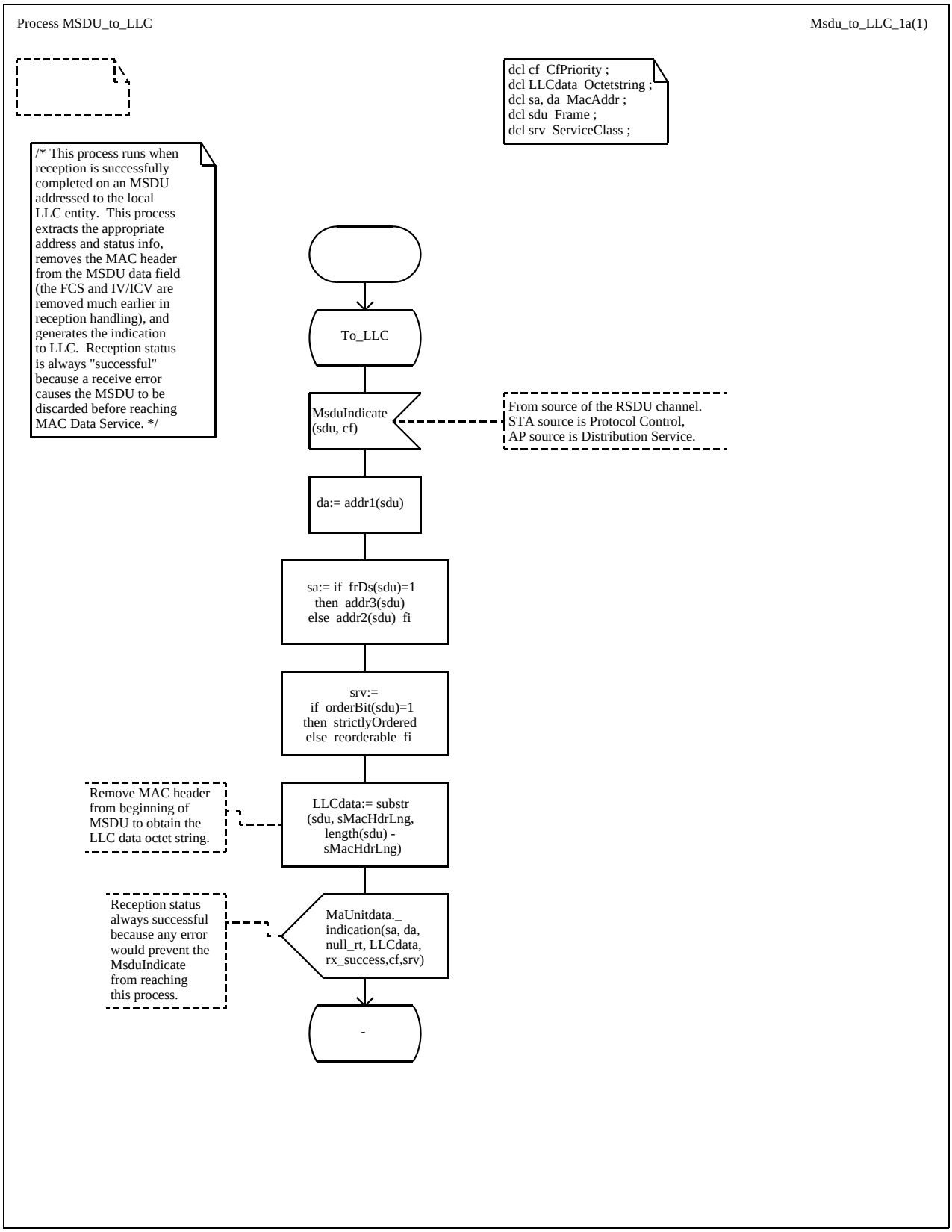




Copyright © 2007 IEEE. All rights reserved.

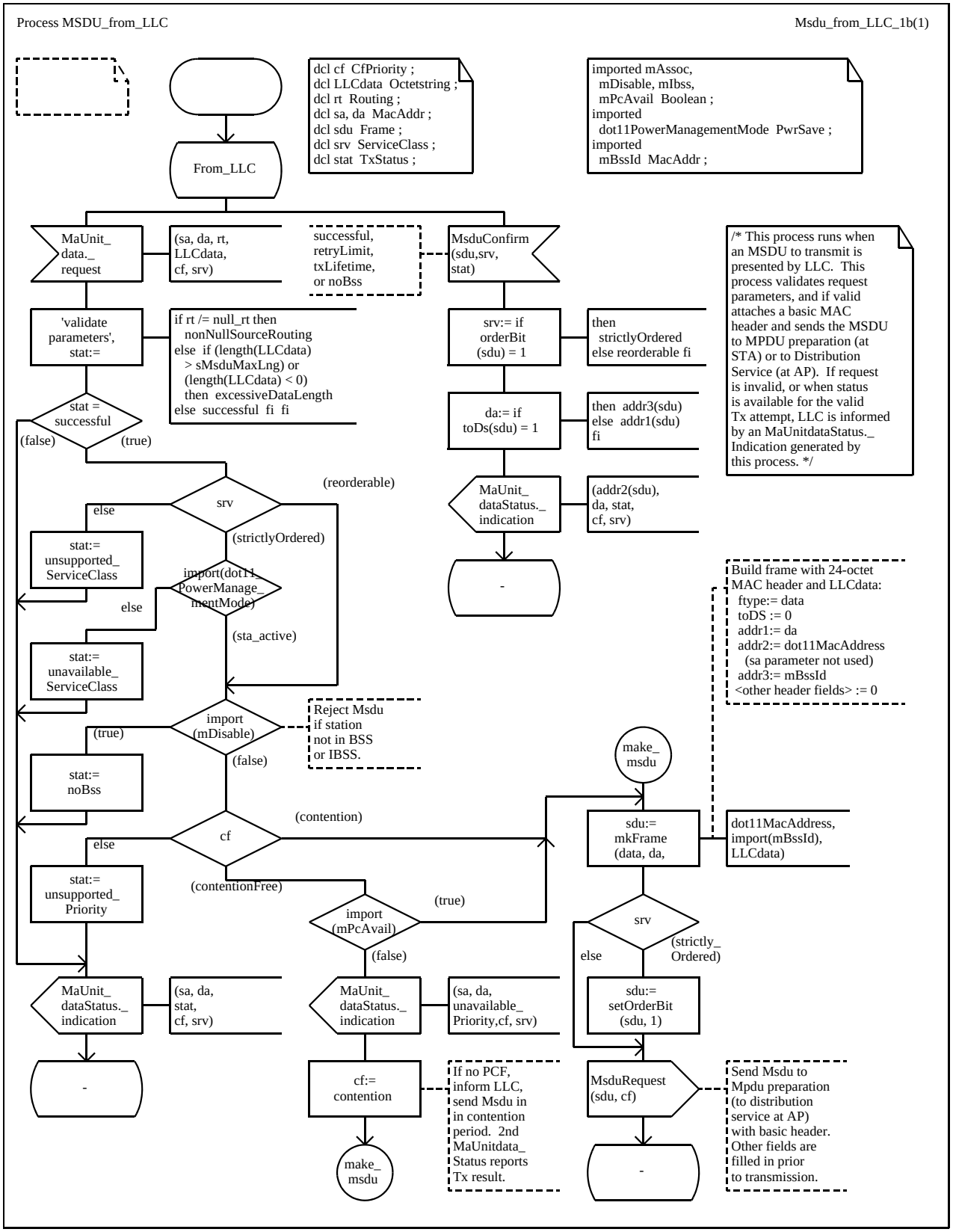



Copyright © 2007 IEEE. All rights reserved.

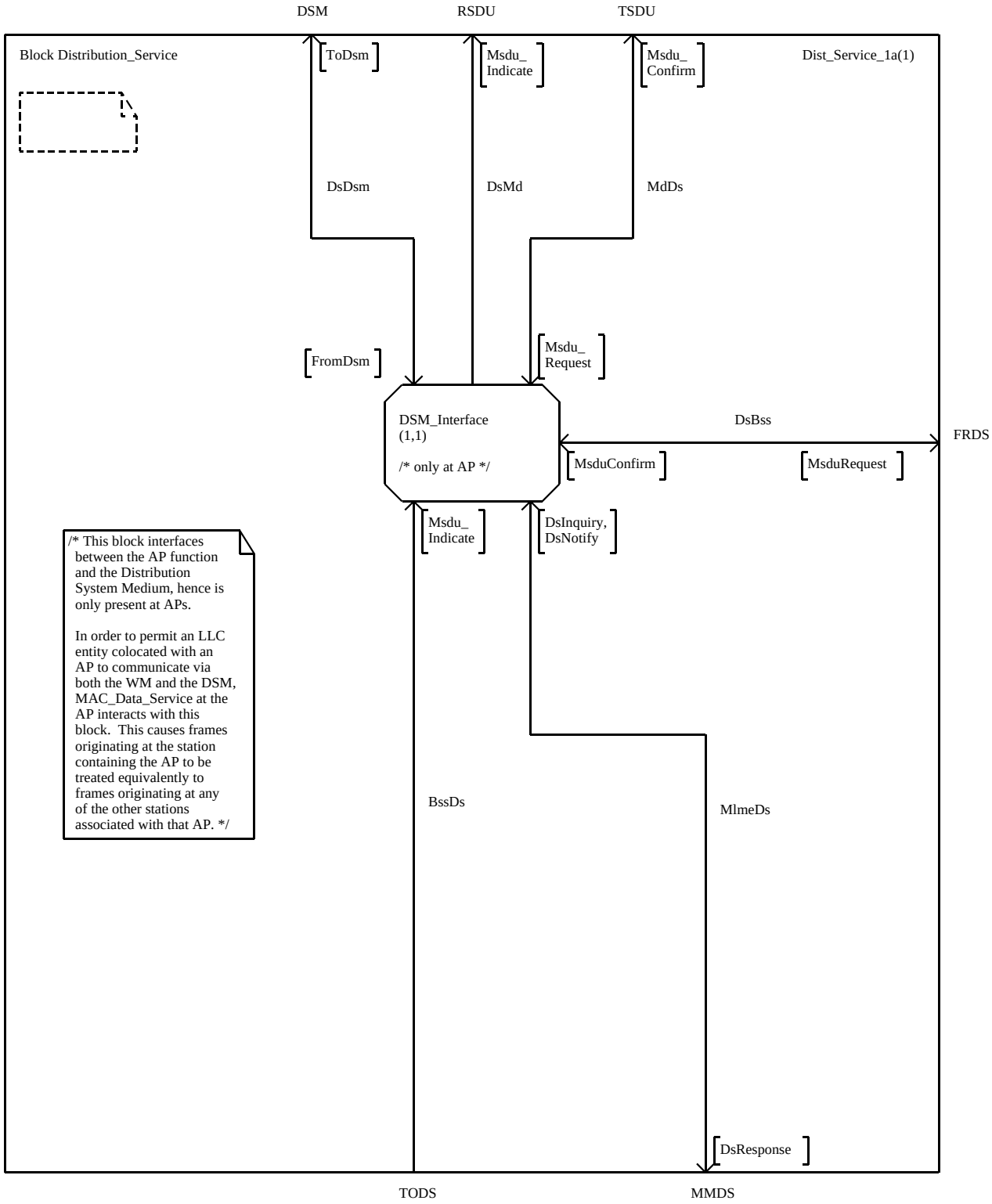




Copyright © 2007 IEEE. All rights reserved.

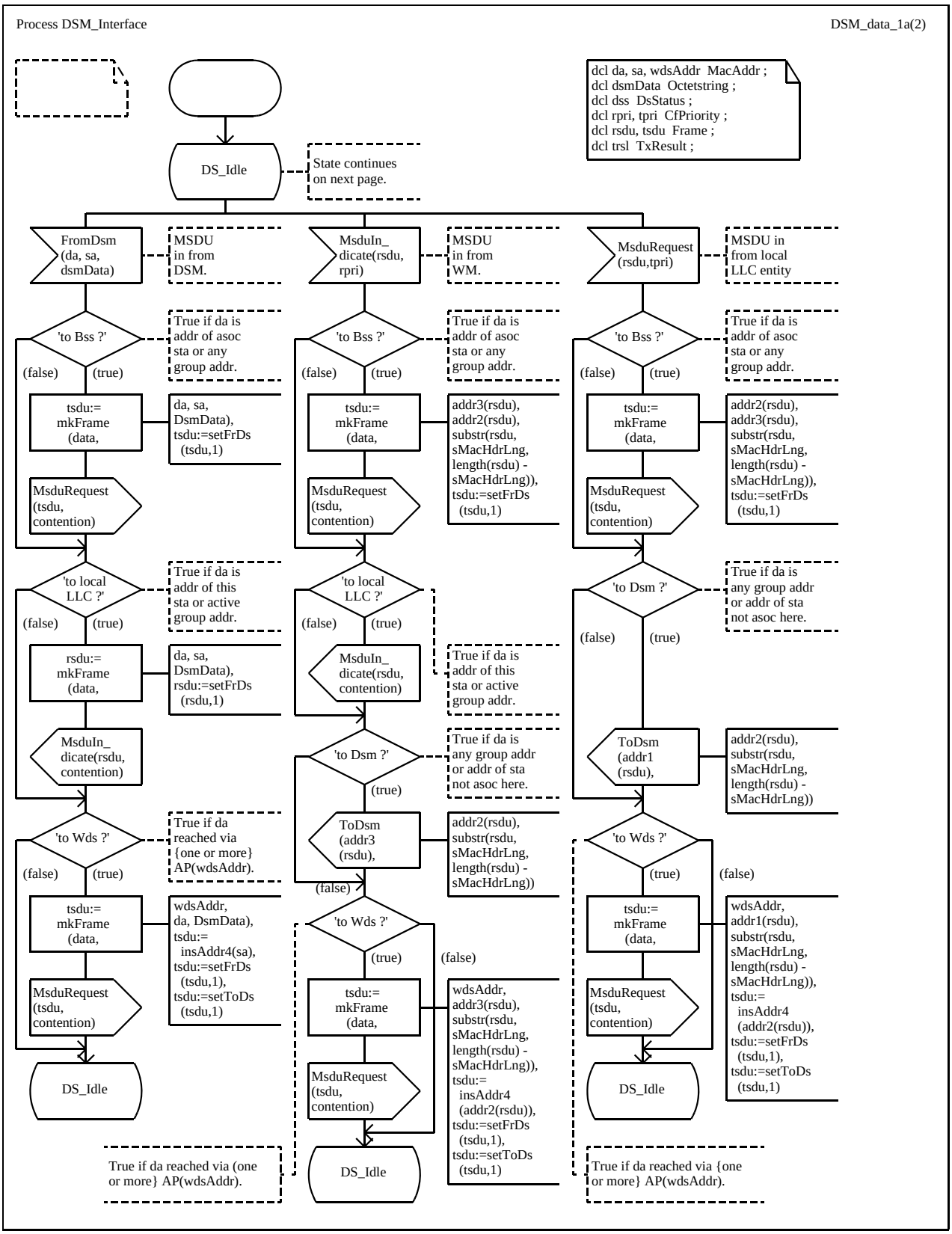



Copyright © 2007 IEEE. All rights reserved.

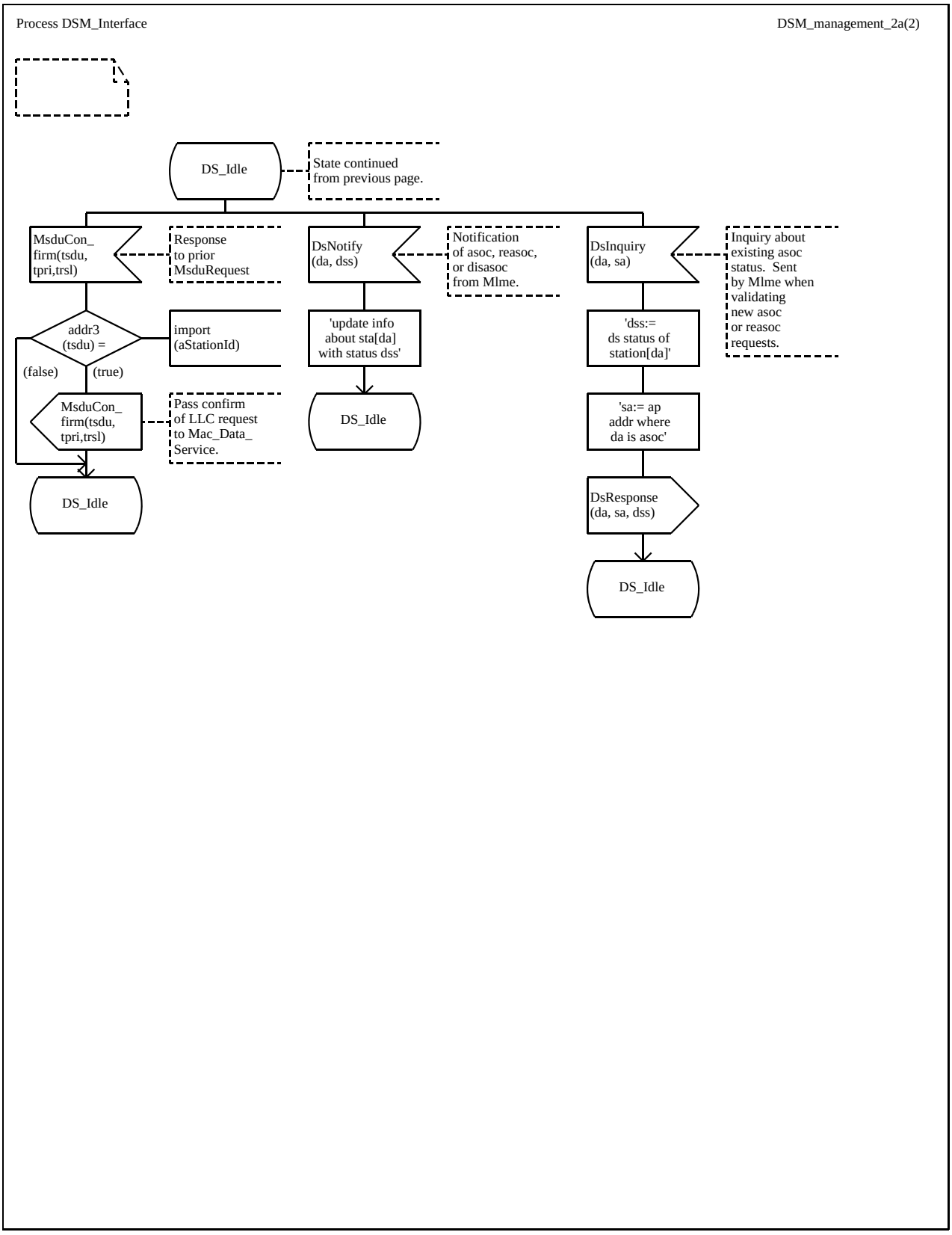




Copyright © 2007 IEEE. All rights reserved.

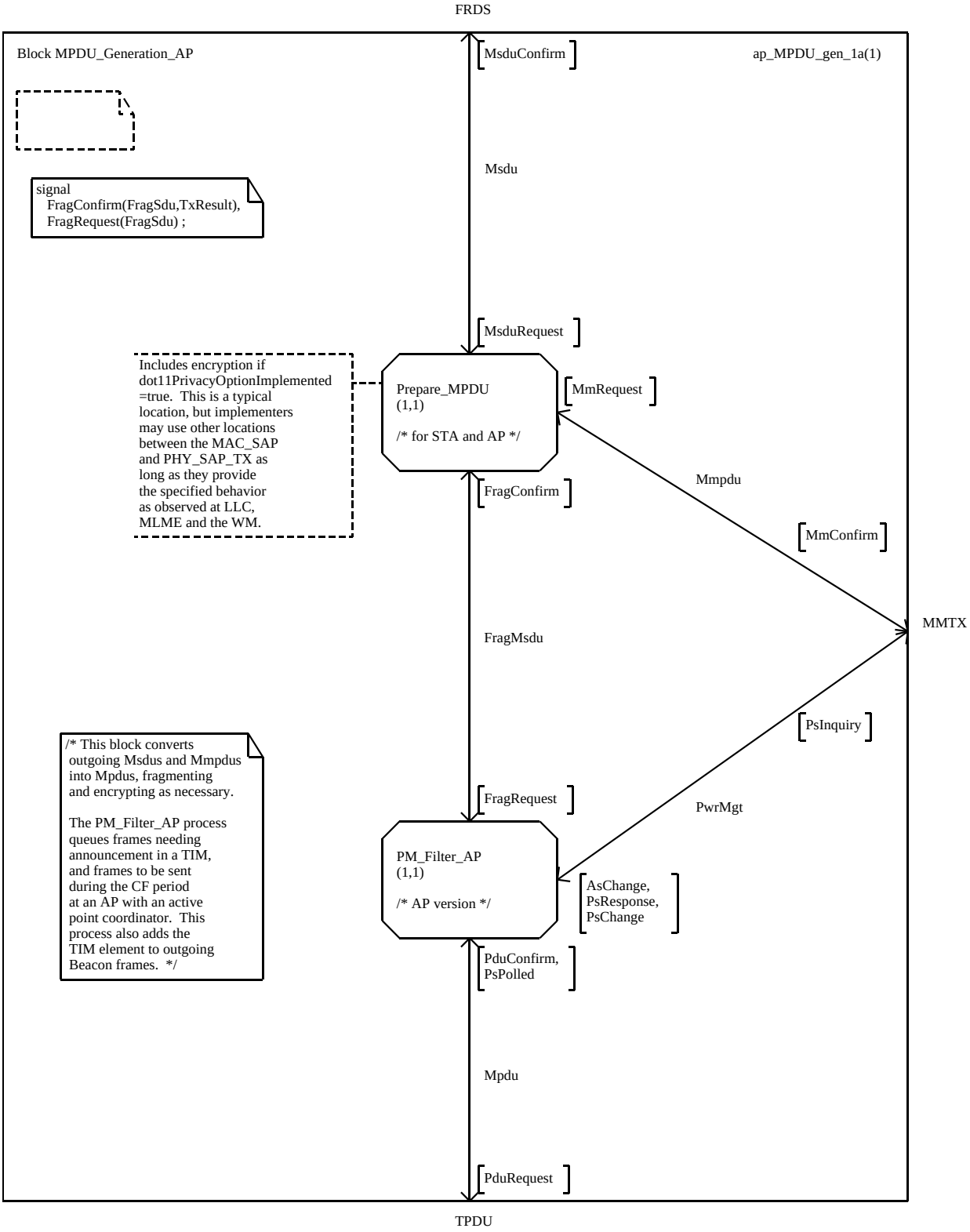



Copyright © 2007 IEEE. All rights reserved.

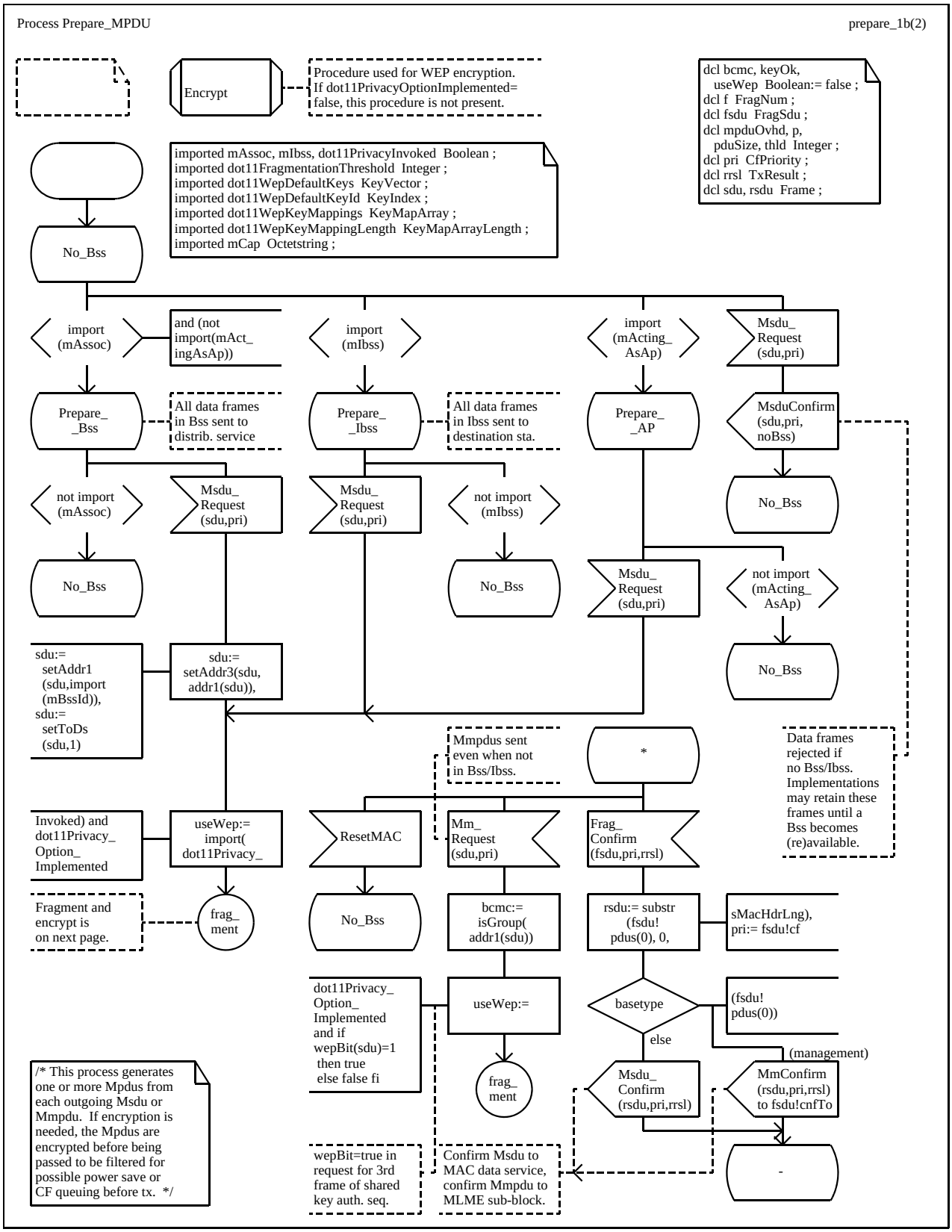




Copyright © 2007 IEEE. All rights reserved.

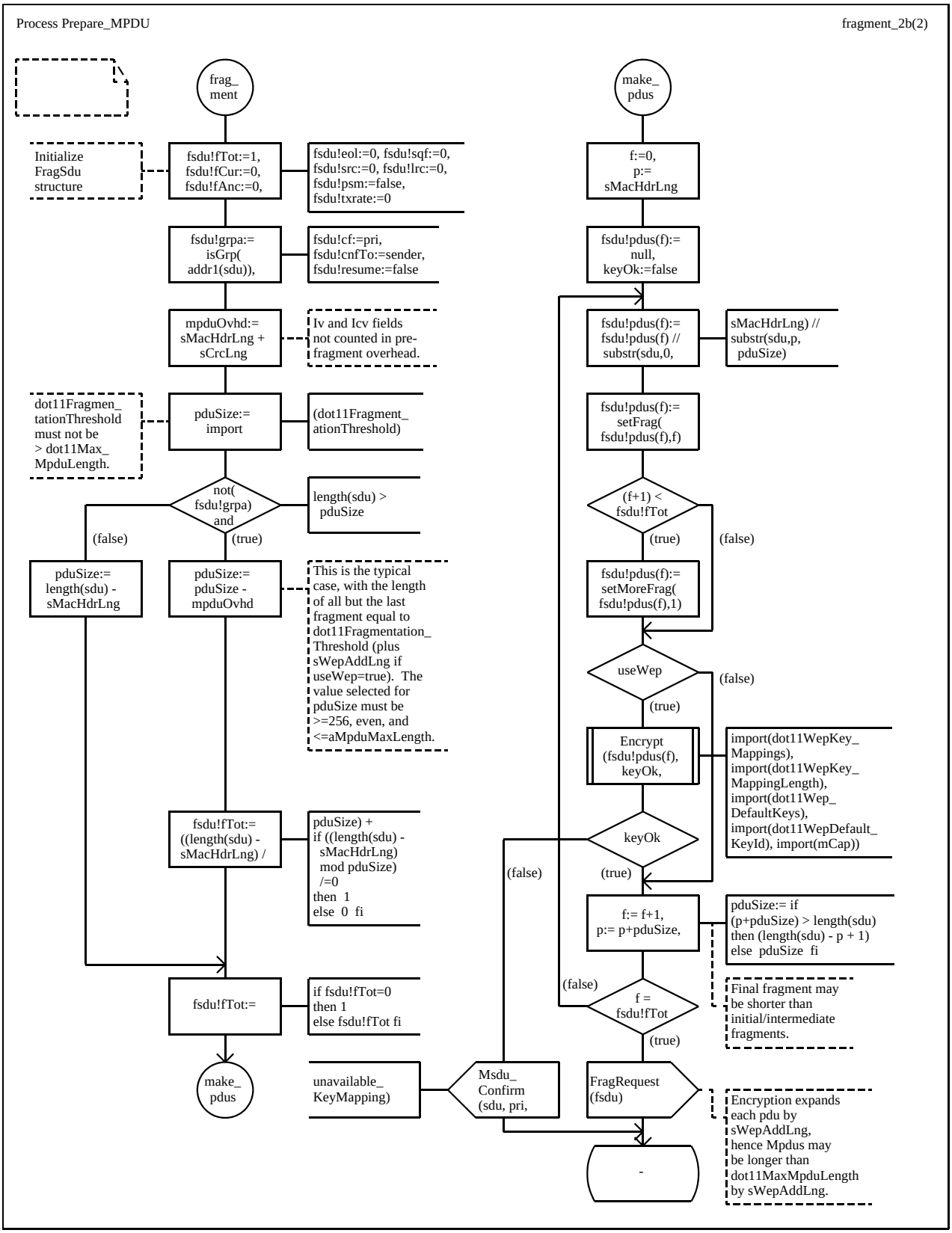



Copyright © 2007 IEEE. All rights reserved.

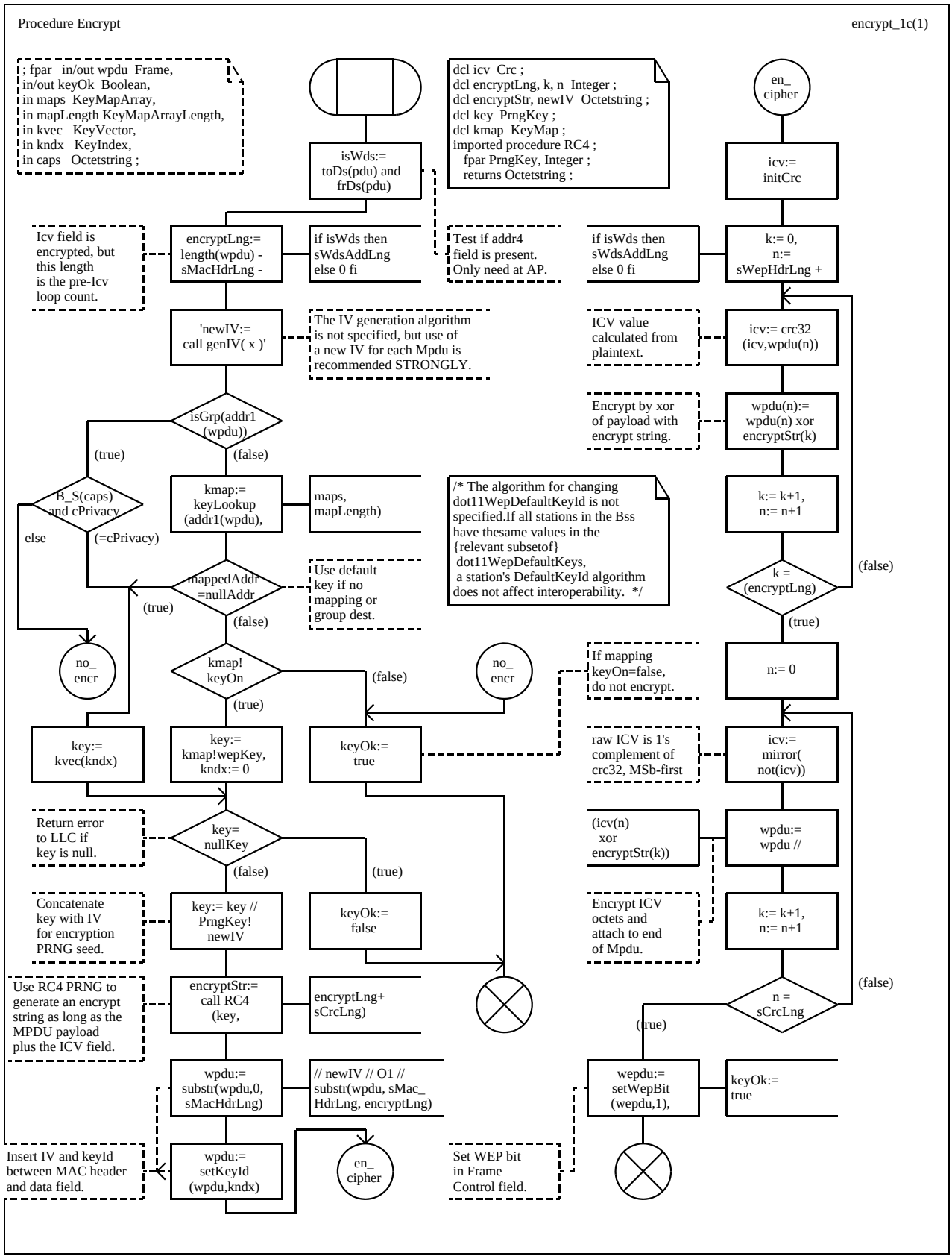




Copyright © 2007 IEEE. All rights reserved.

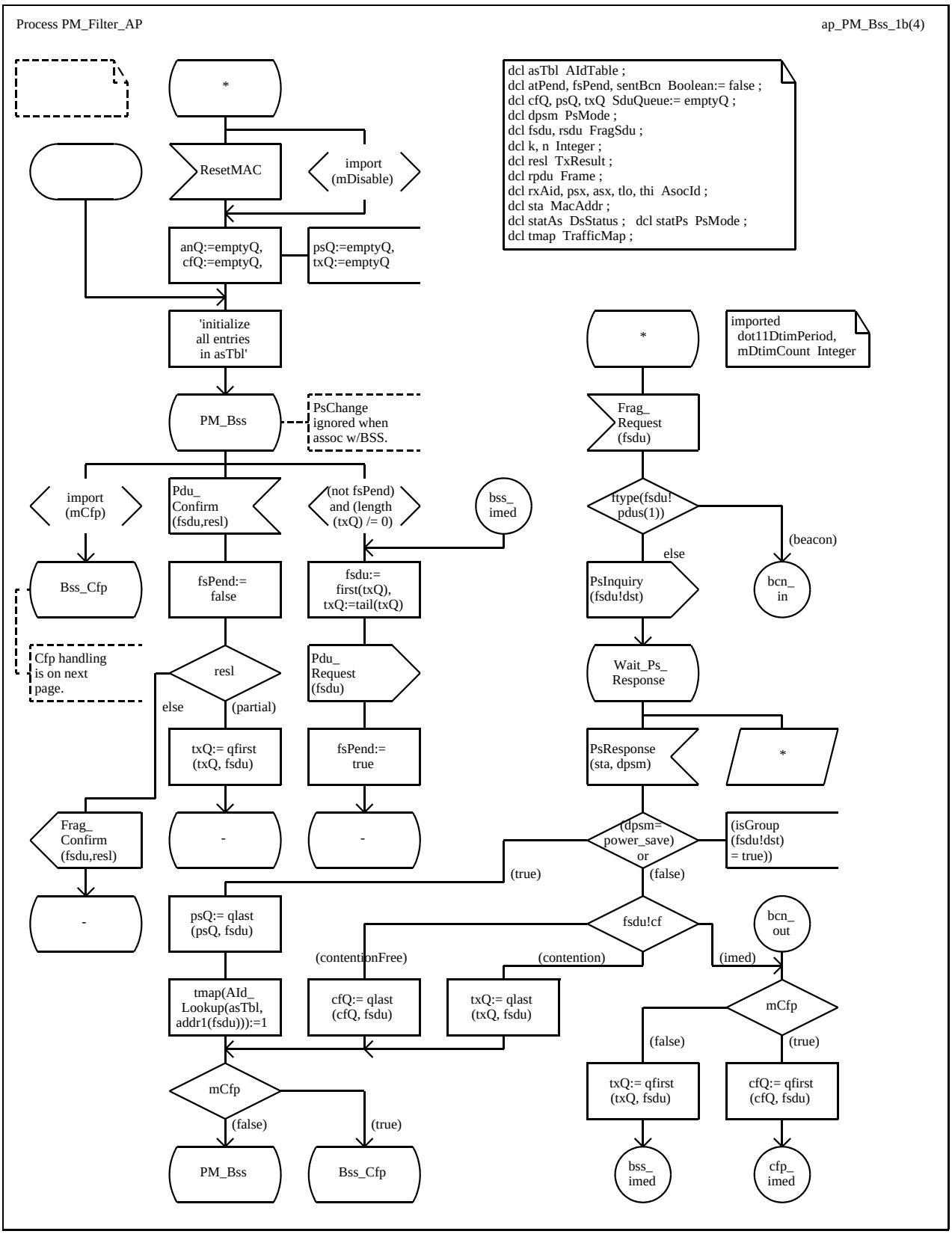



Copyright © 2007 IEEE. All rights reserved.

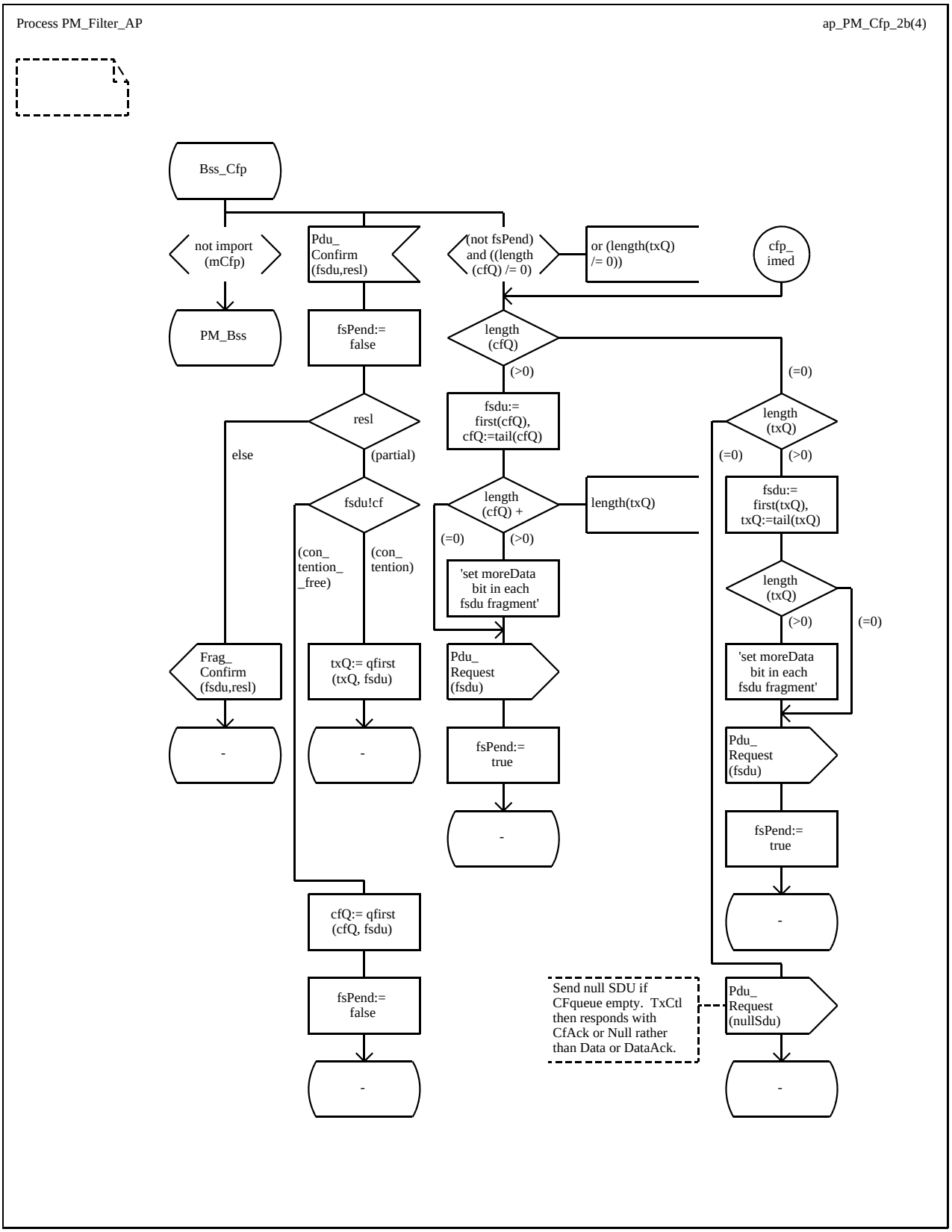




Copyright © 2007 IEEE. All rights reserved.

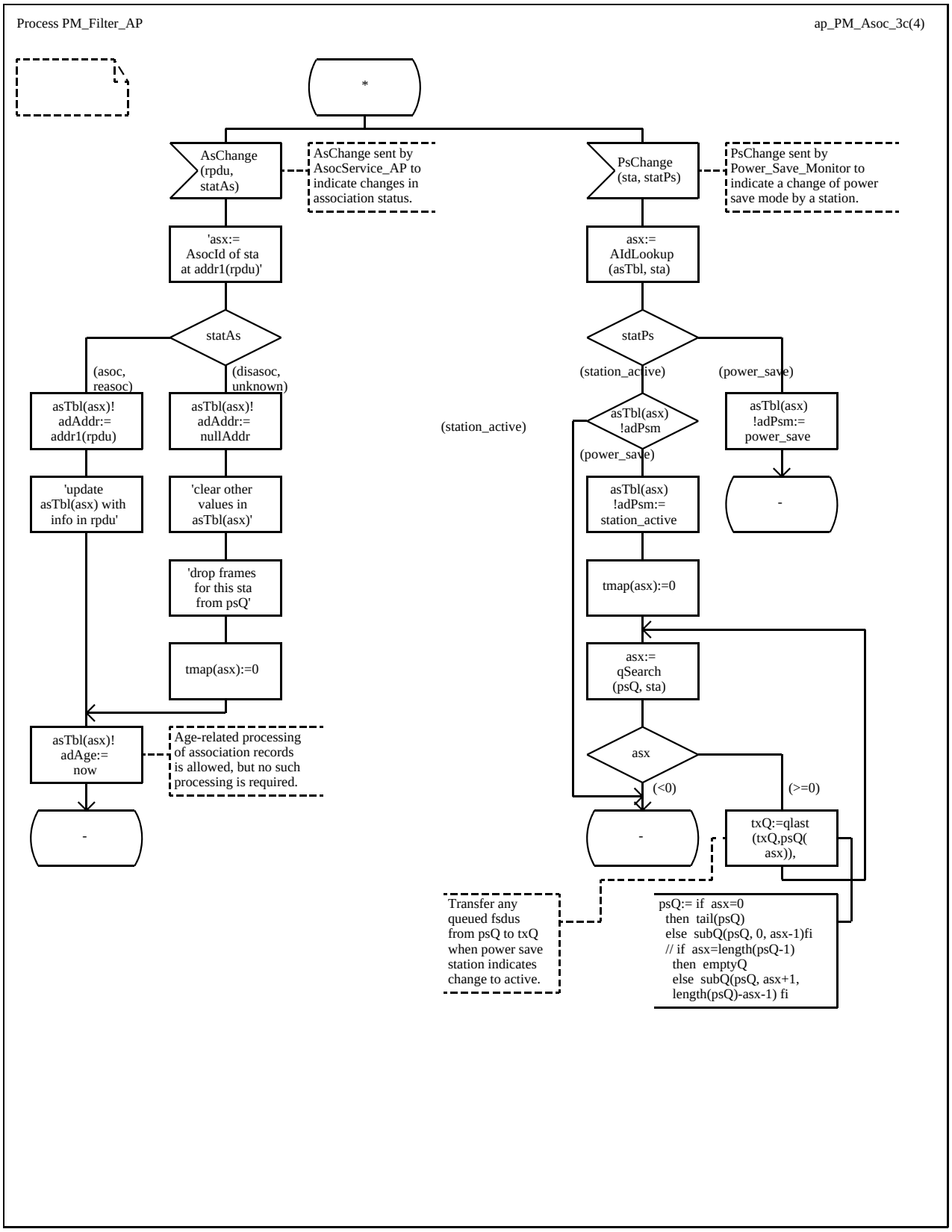



Copyright © 2007 IEEE. All rights reserved.

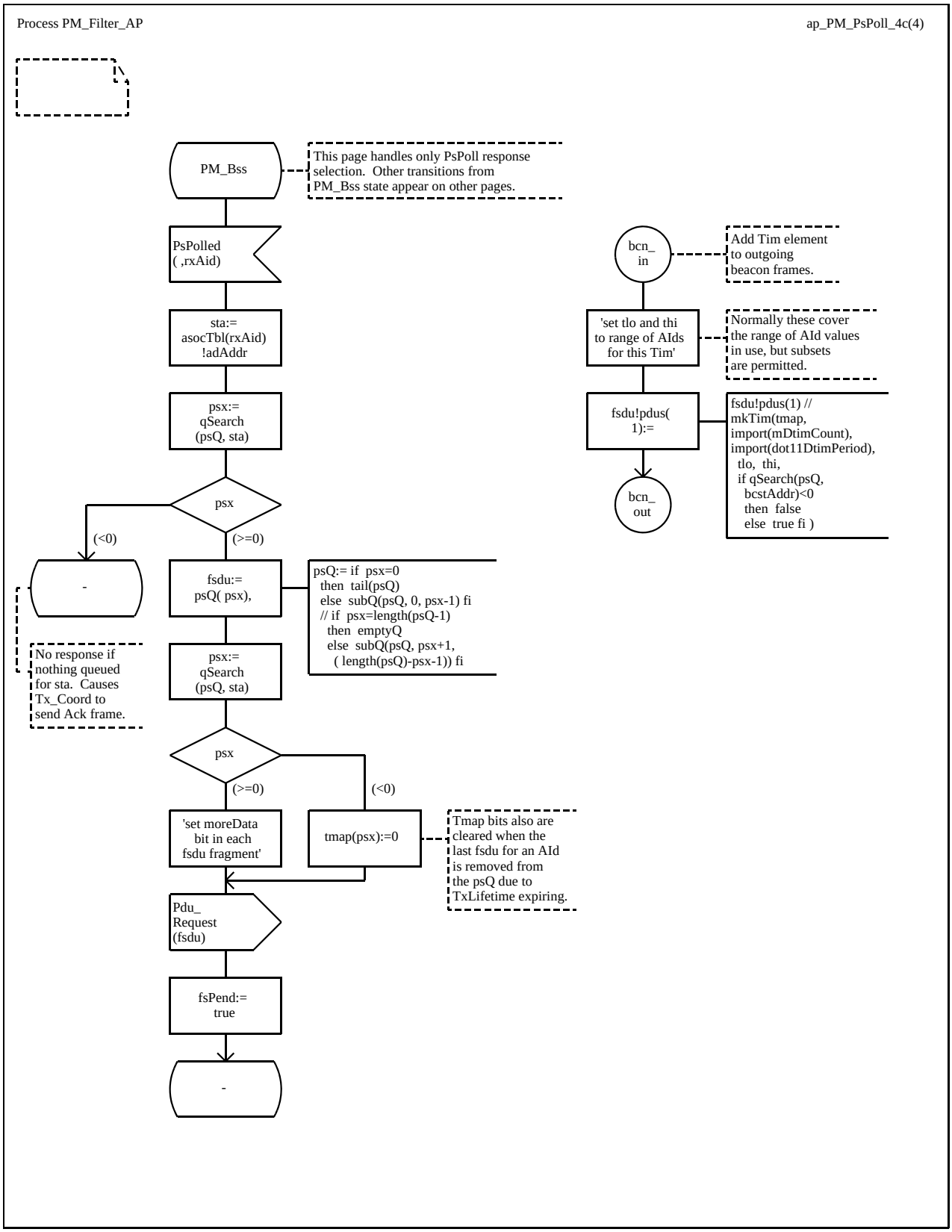




Copyright © 2007 IEEE. All rights reserved.

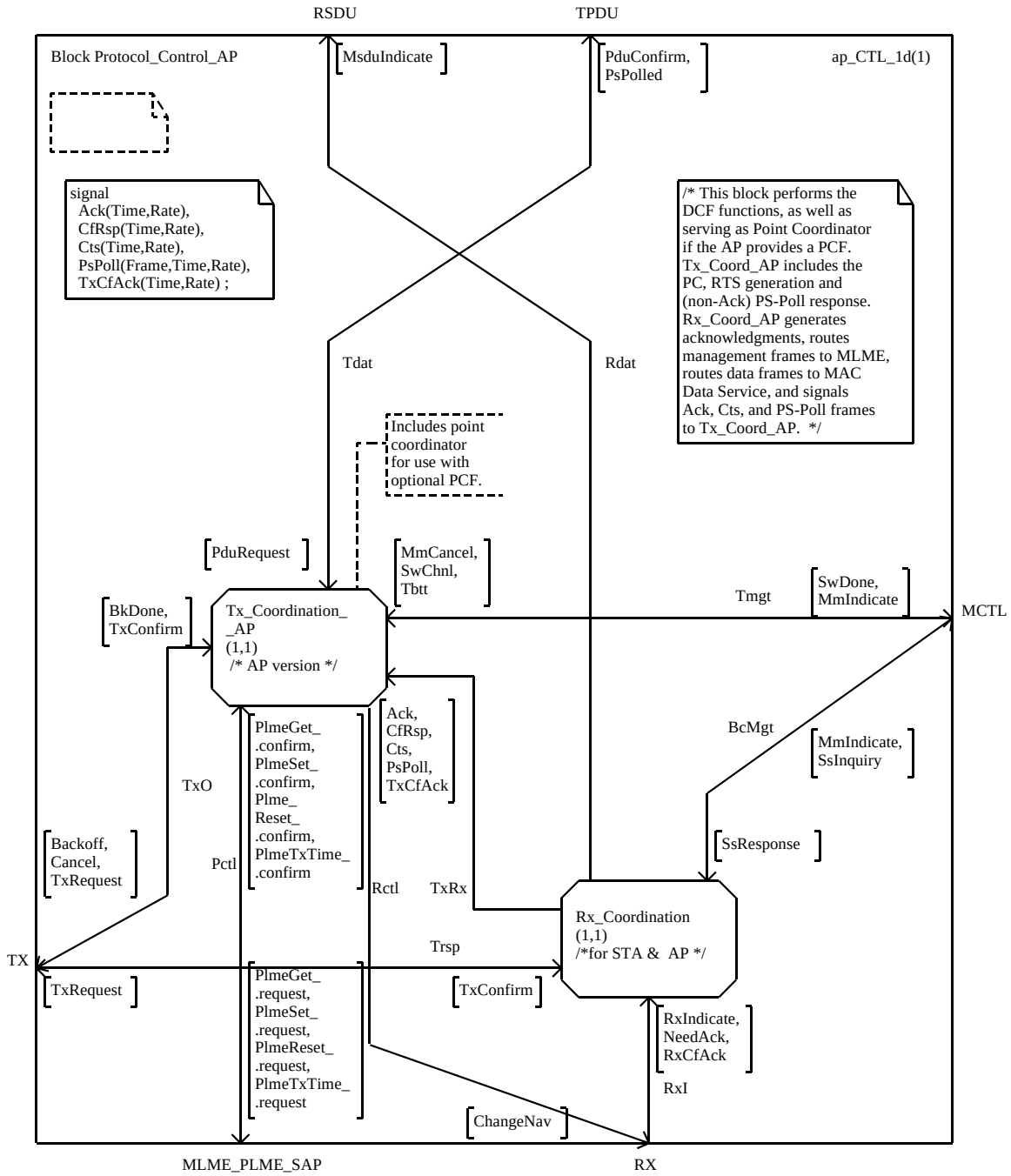



Copyright © 2007 IEEE. All rights reserved.

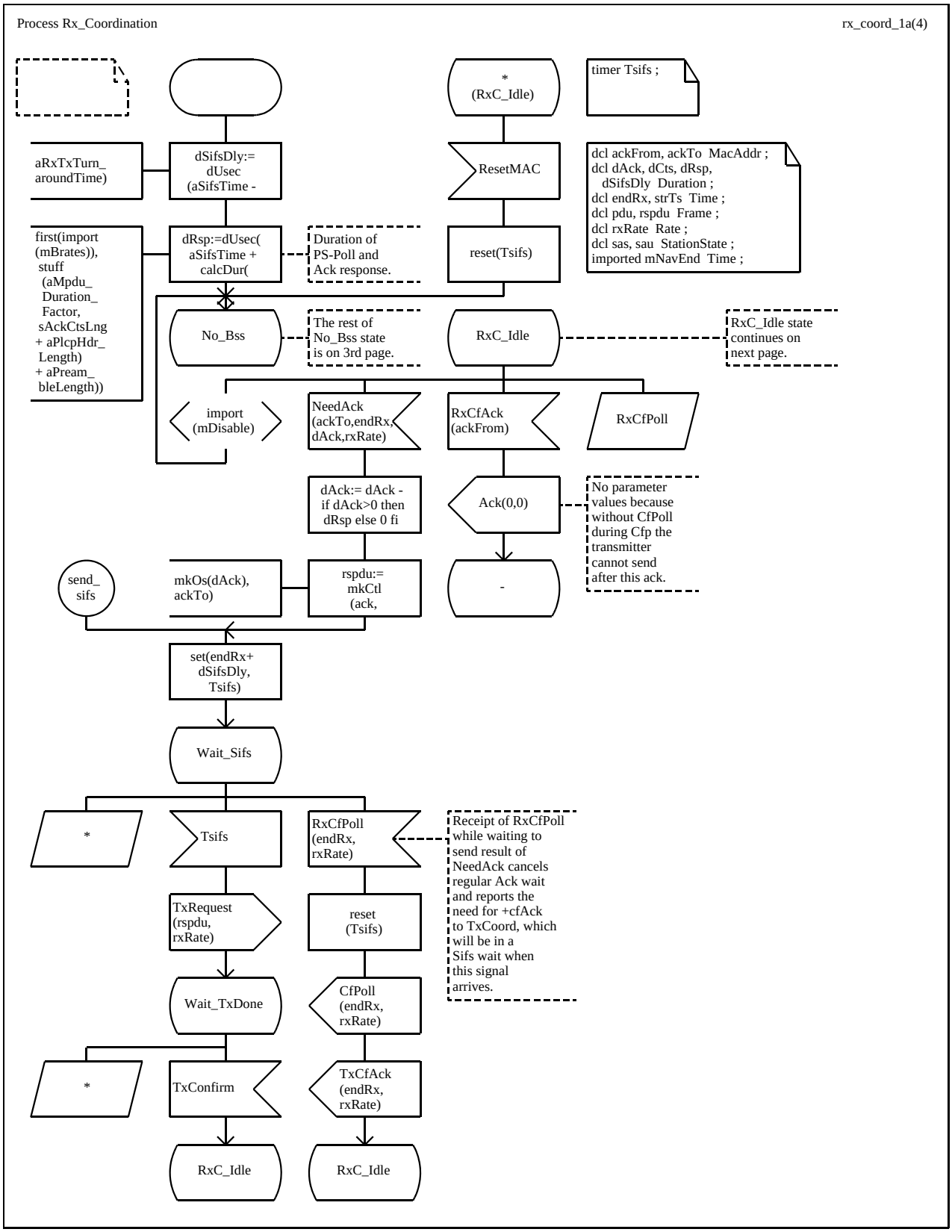




Copyright © 2007 IEEE. All rights reserved.

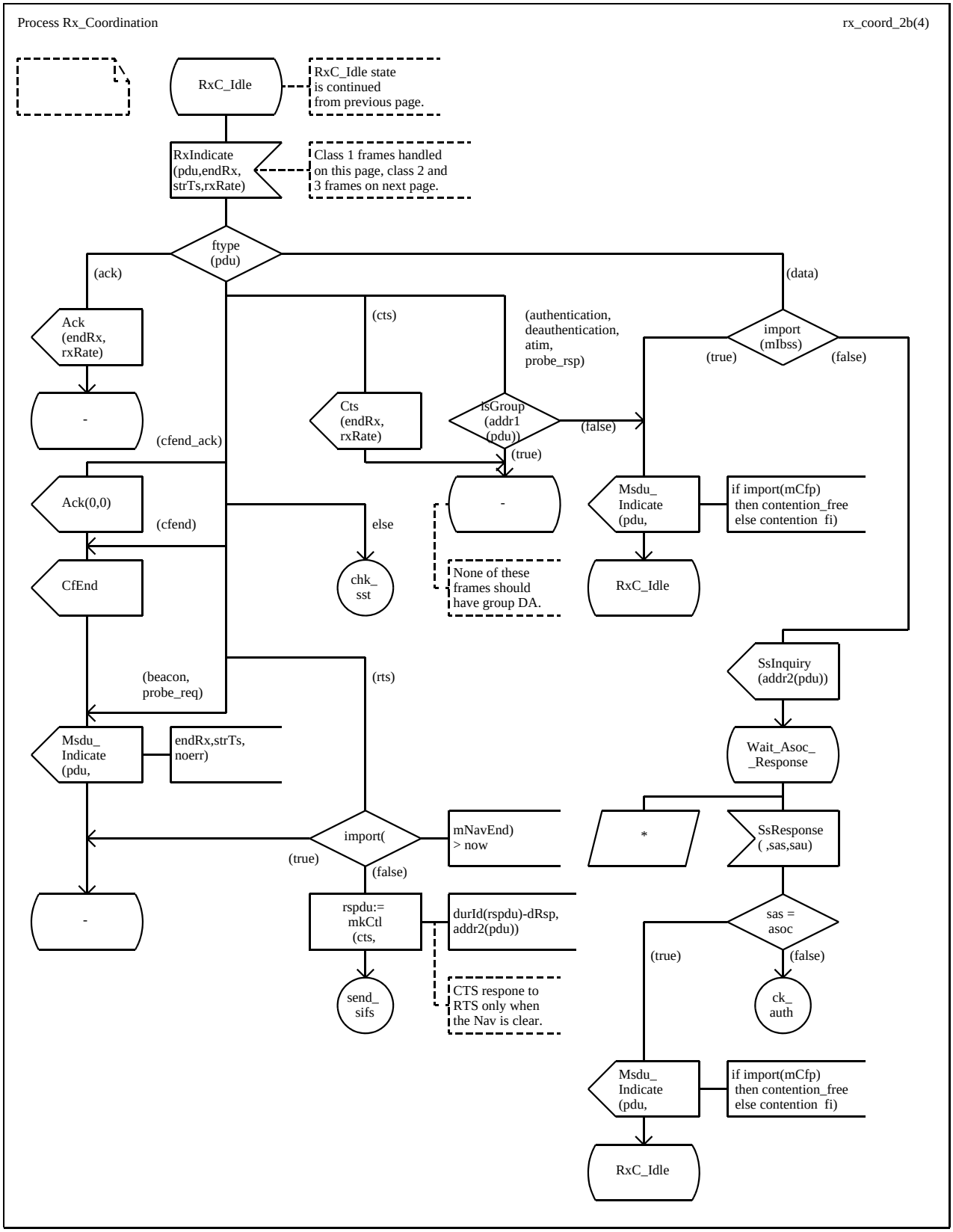



Copyright © 2007 IEEE. All rights reserved.

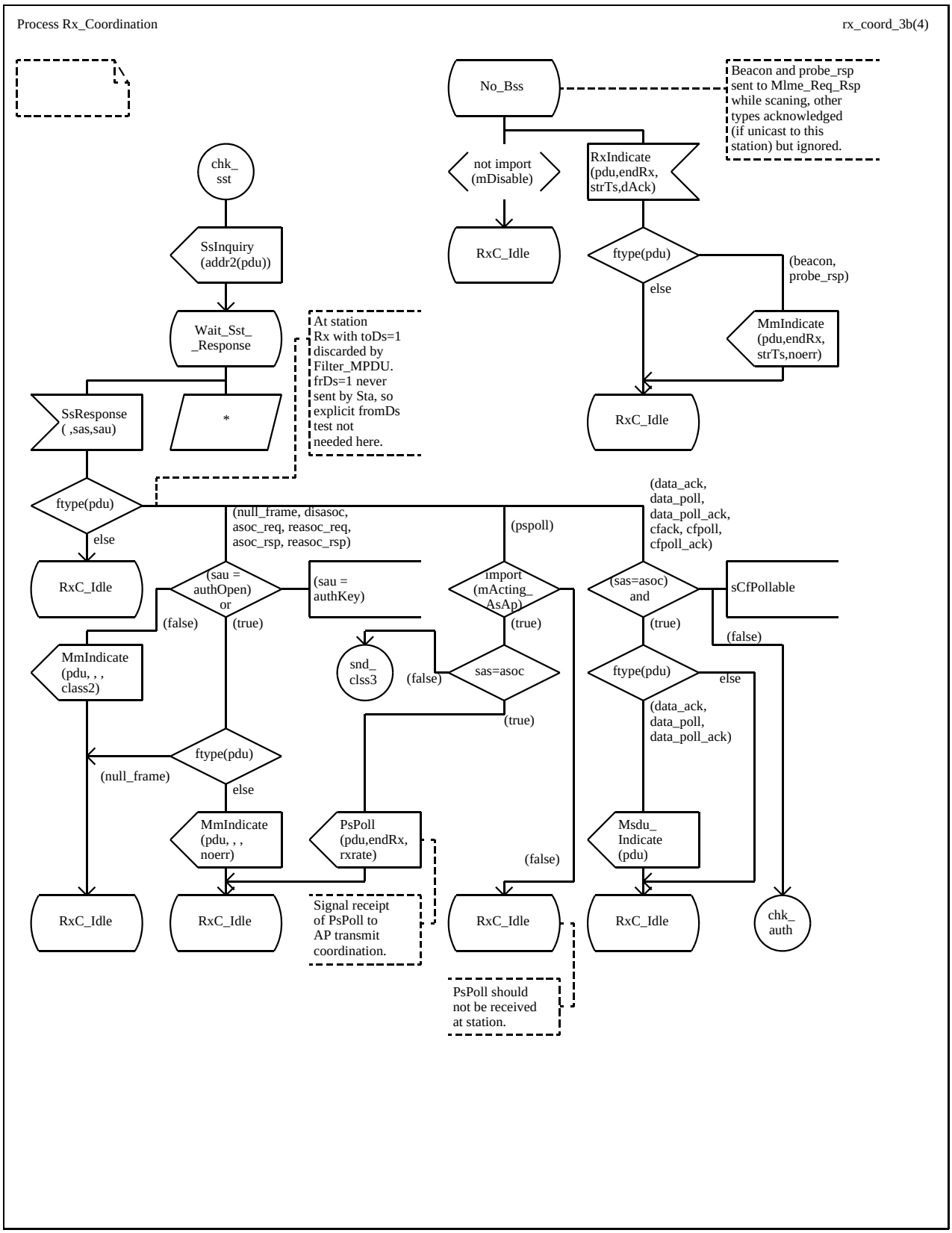




Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.



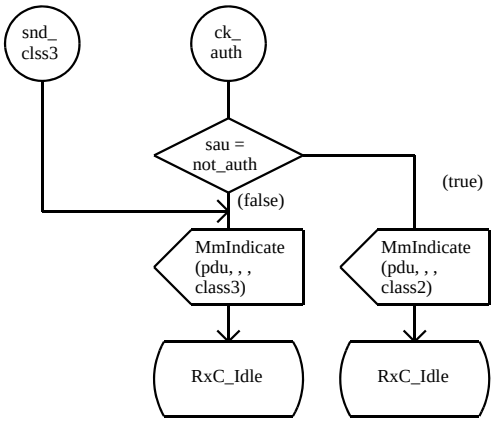


Copyright © 2007 IEEE. All rights reserved.

Process Rx_Coordination

rx_coord_3.1a(4)

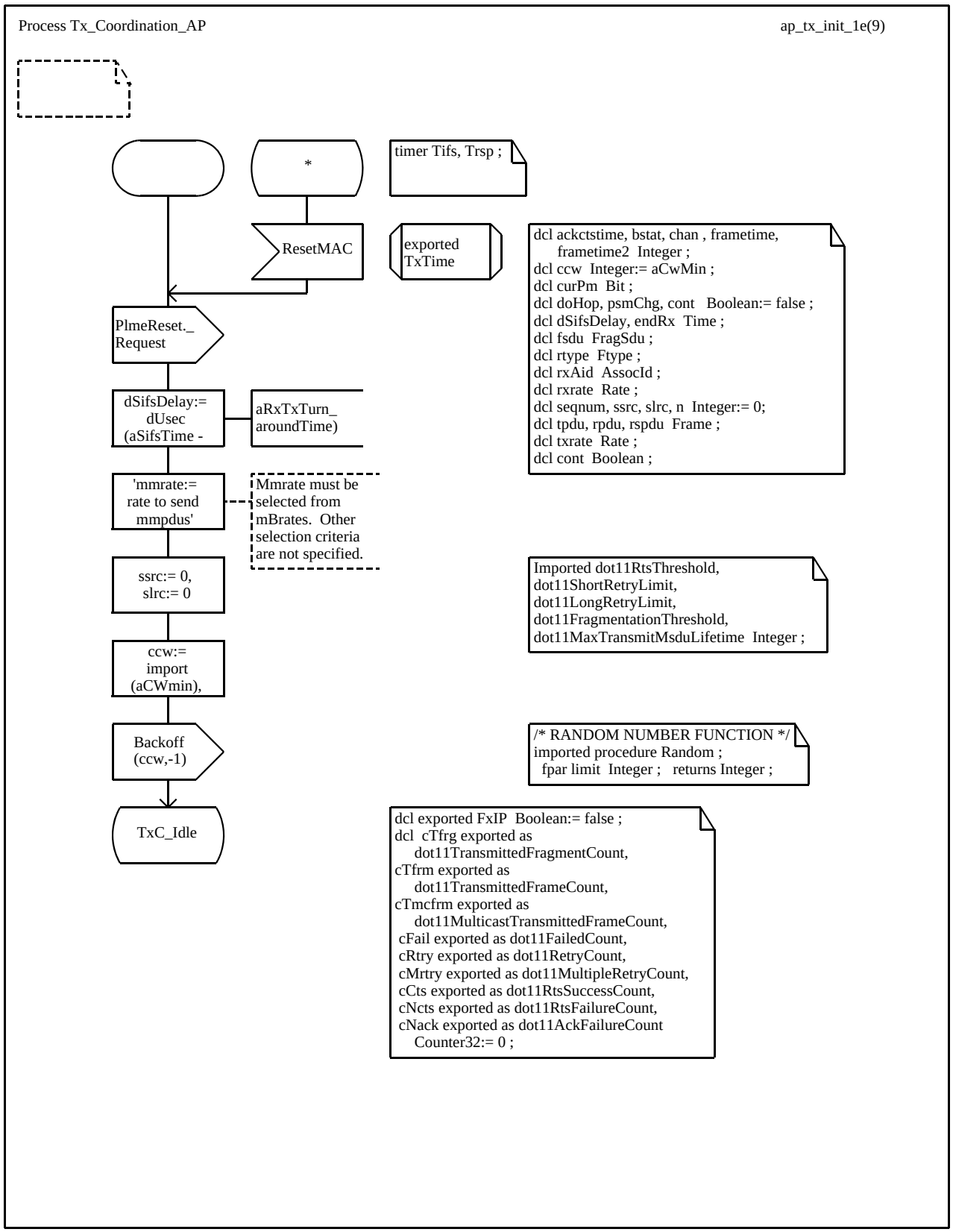





Copyright © 2007 IEEE. All rights reserved.

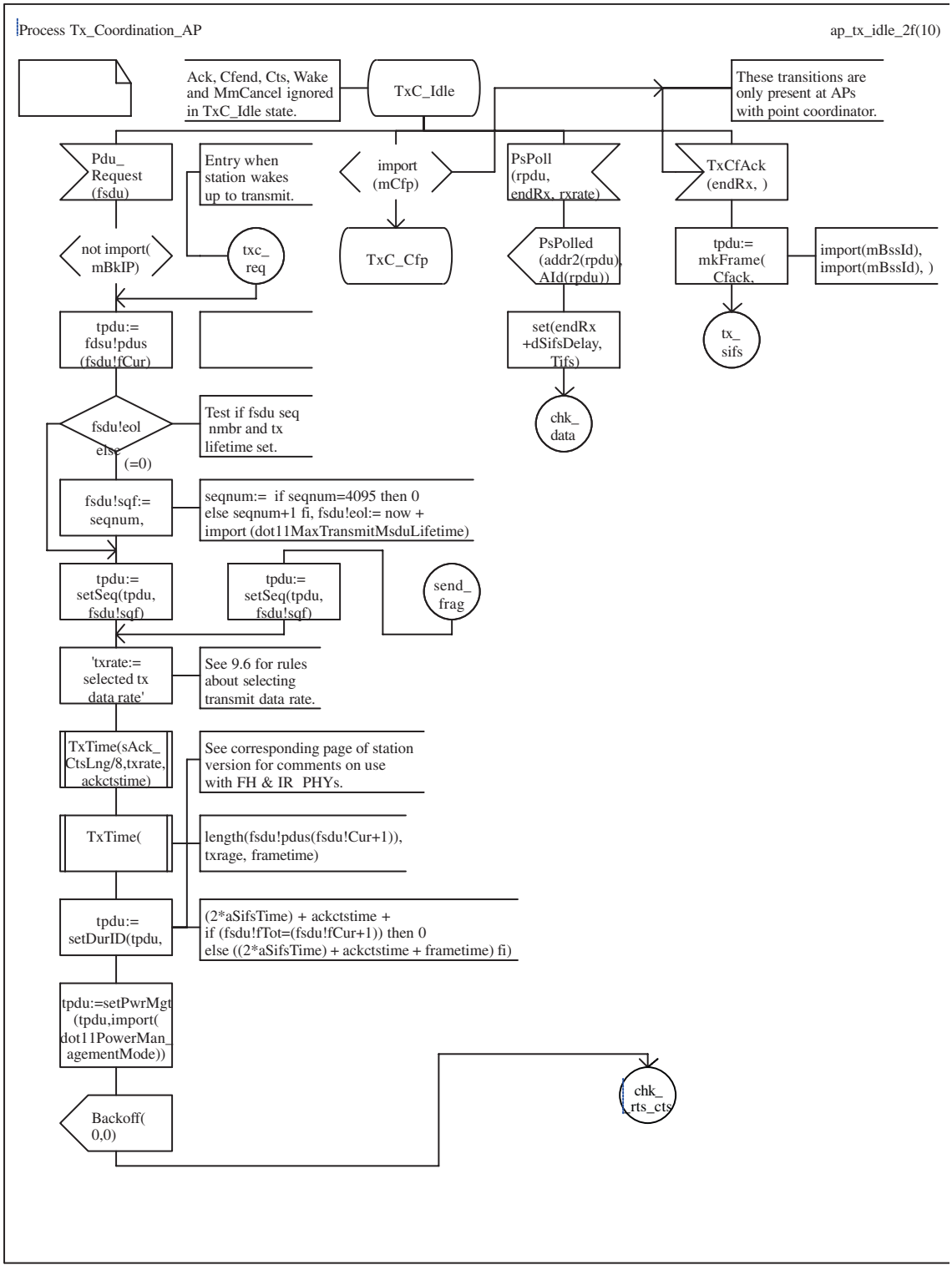


Samsung Ex. 1044, Page 983 of 1232

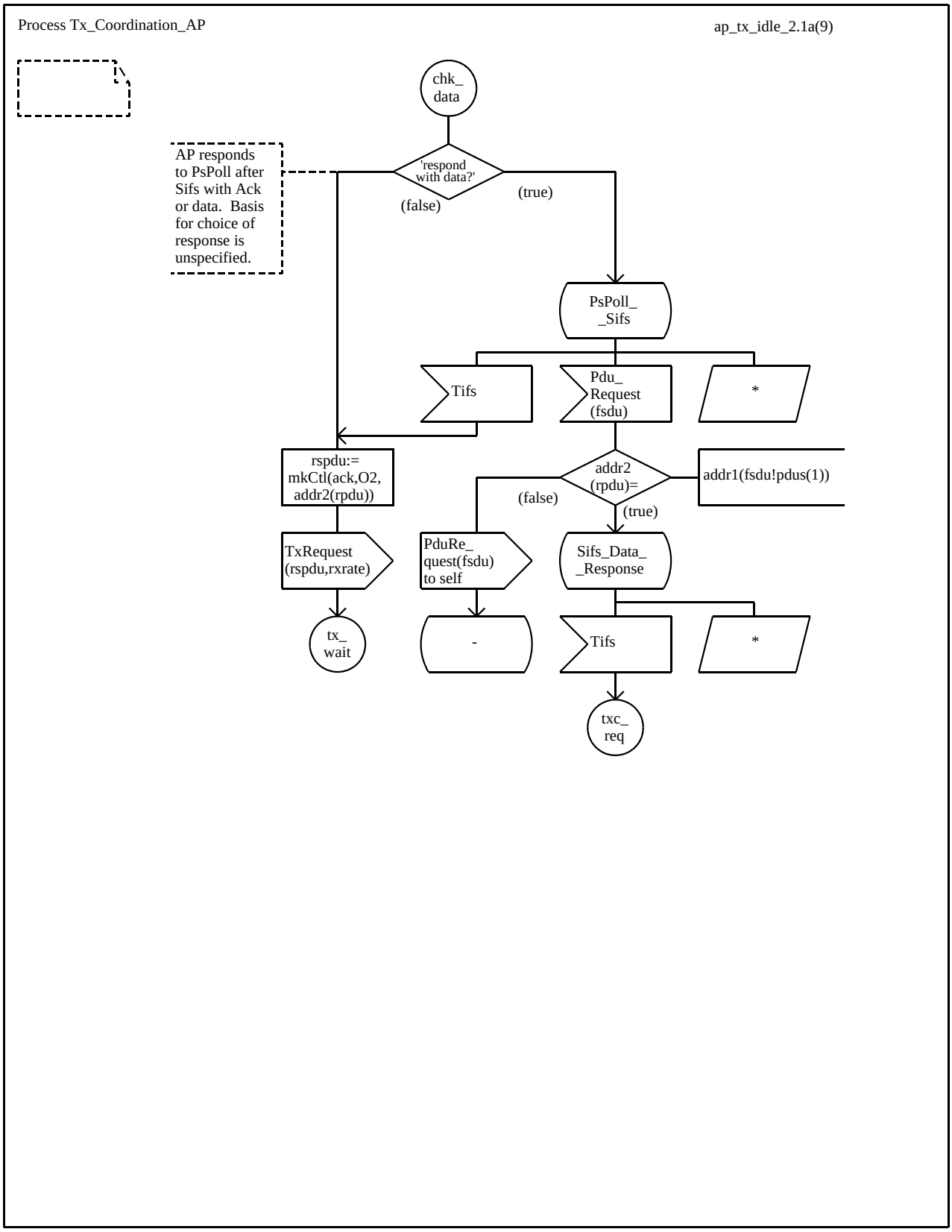




Copyright © 2007 IEEE. All rights reserved.

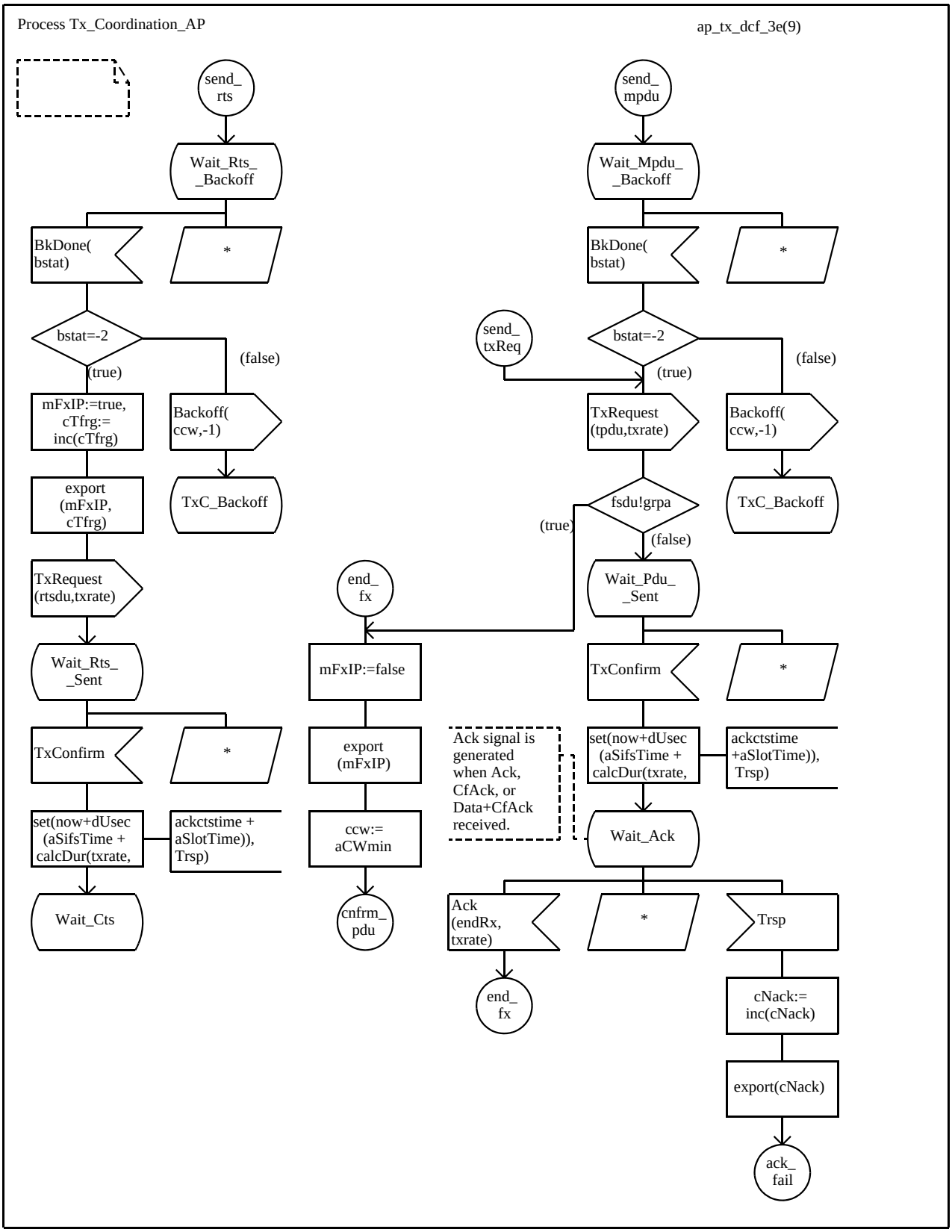



Copyright © 2007 IEEE. All rights reserved.

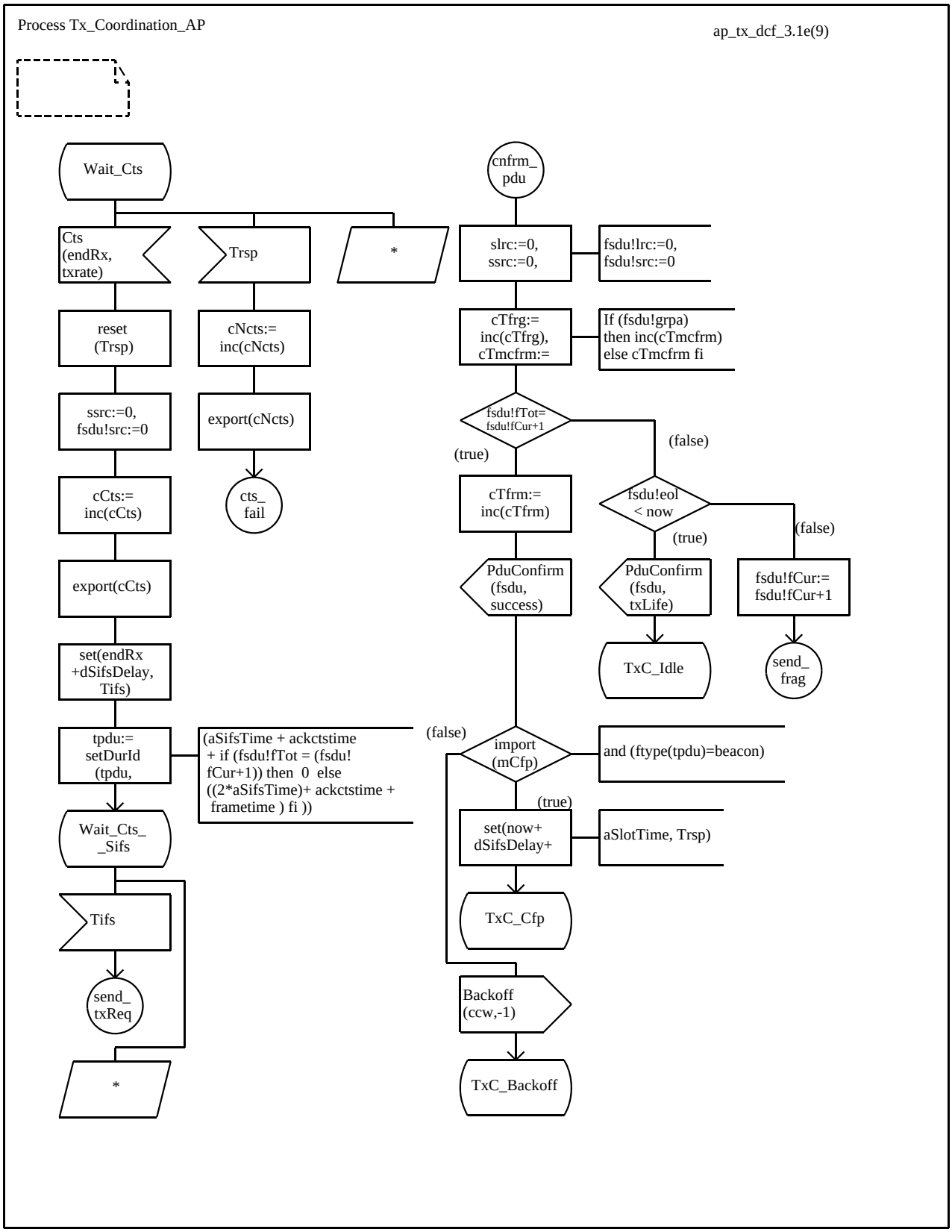




Copyright © 2007 IEEE. All rights reserved.

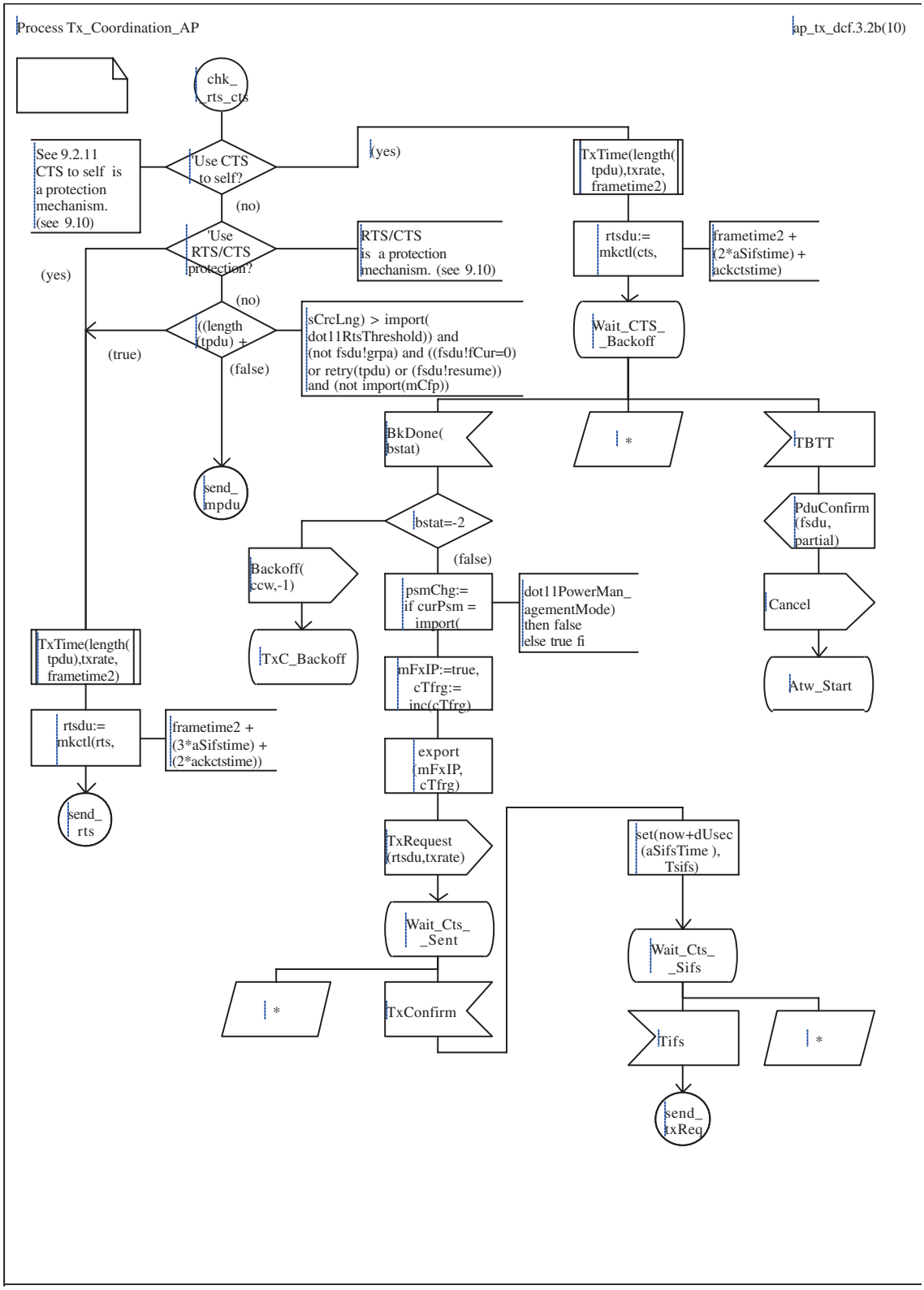



Copyright © 2007 IEEE. All rights reserved.

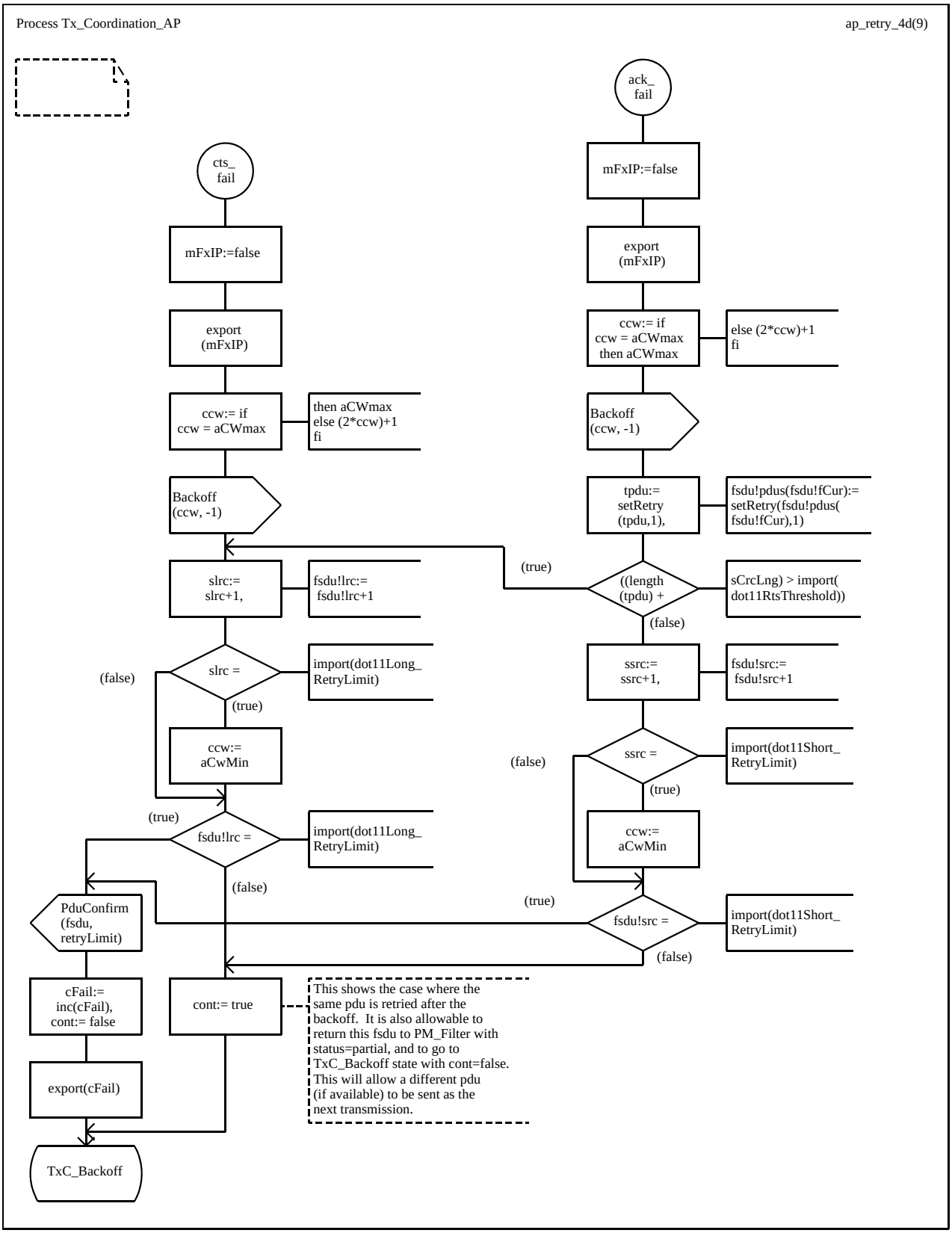




Copyright © 2007 IEEE. All rights reserved.

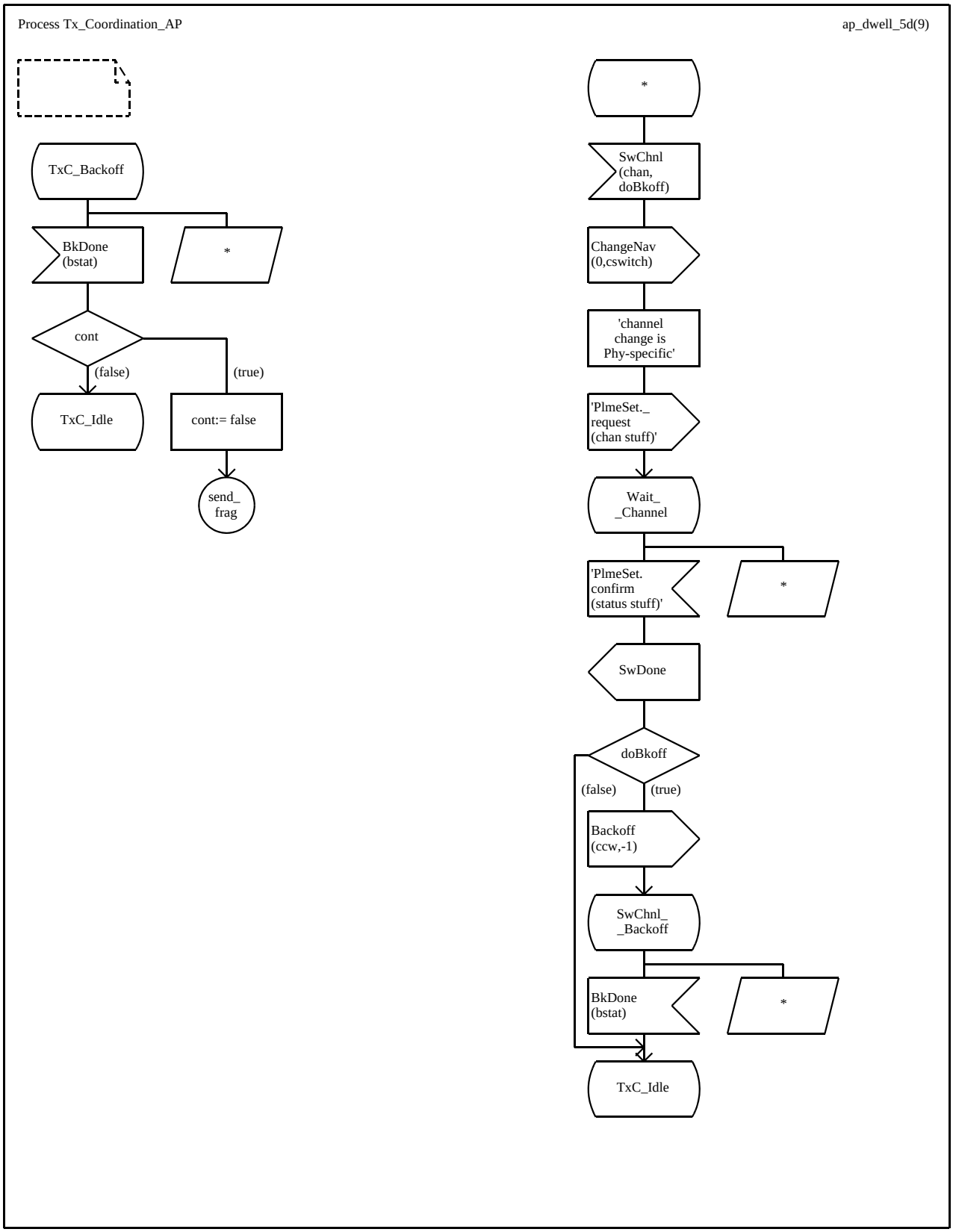



Copyright © 2007 IEEE. All rights reserved.

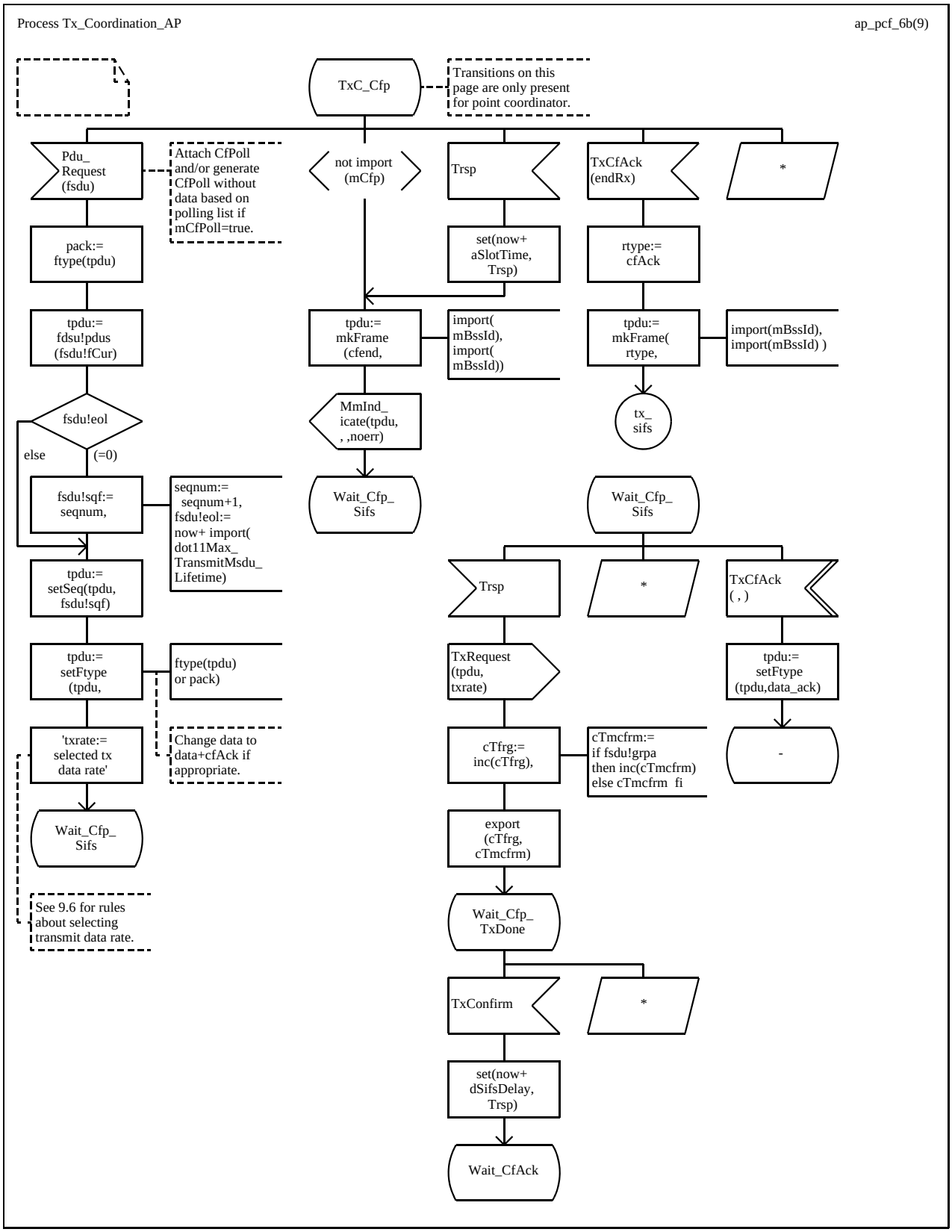




Copyright © 2007 IEEE. All rights reserved.

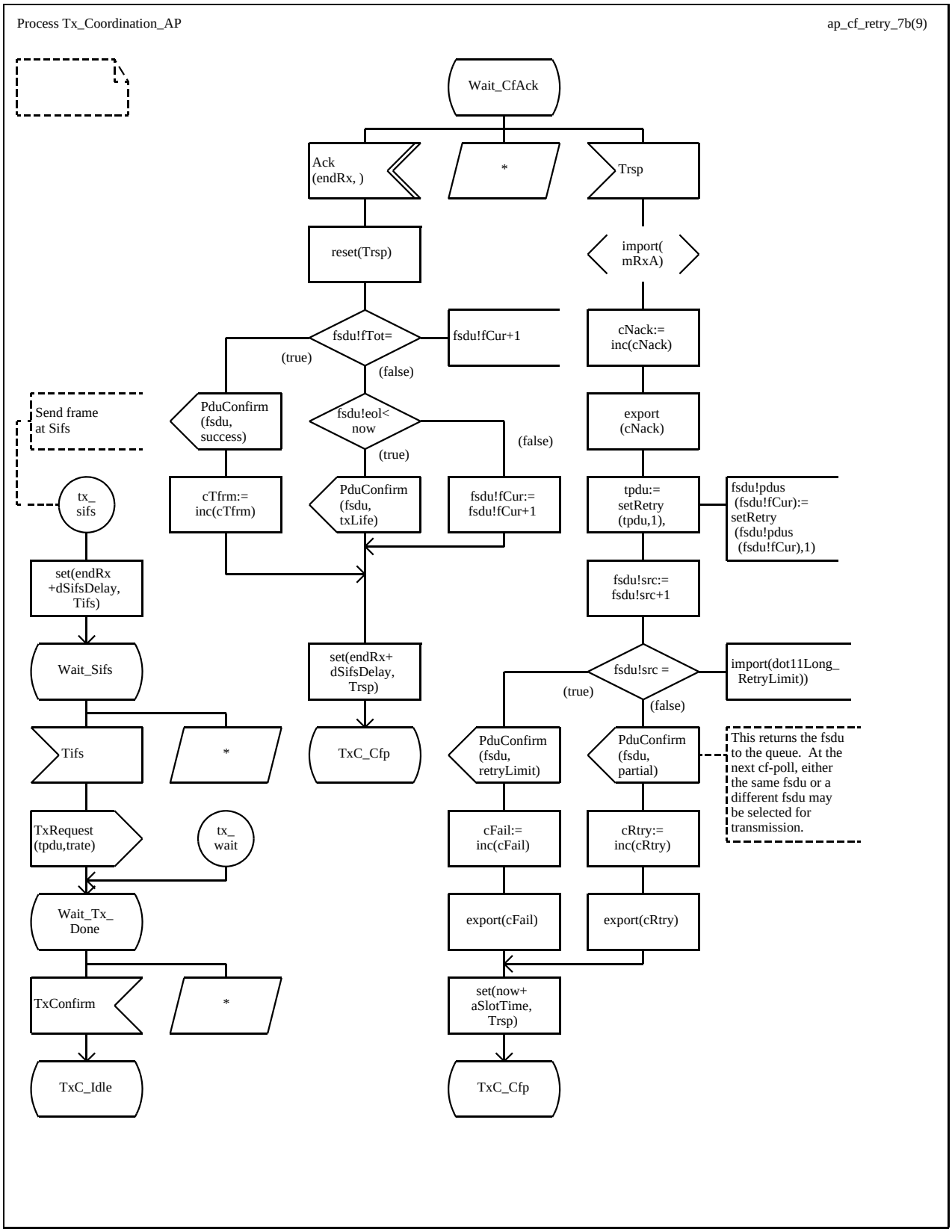



Copyright © 2007 IEEE. All rights reserved.

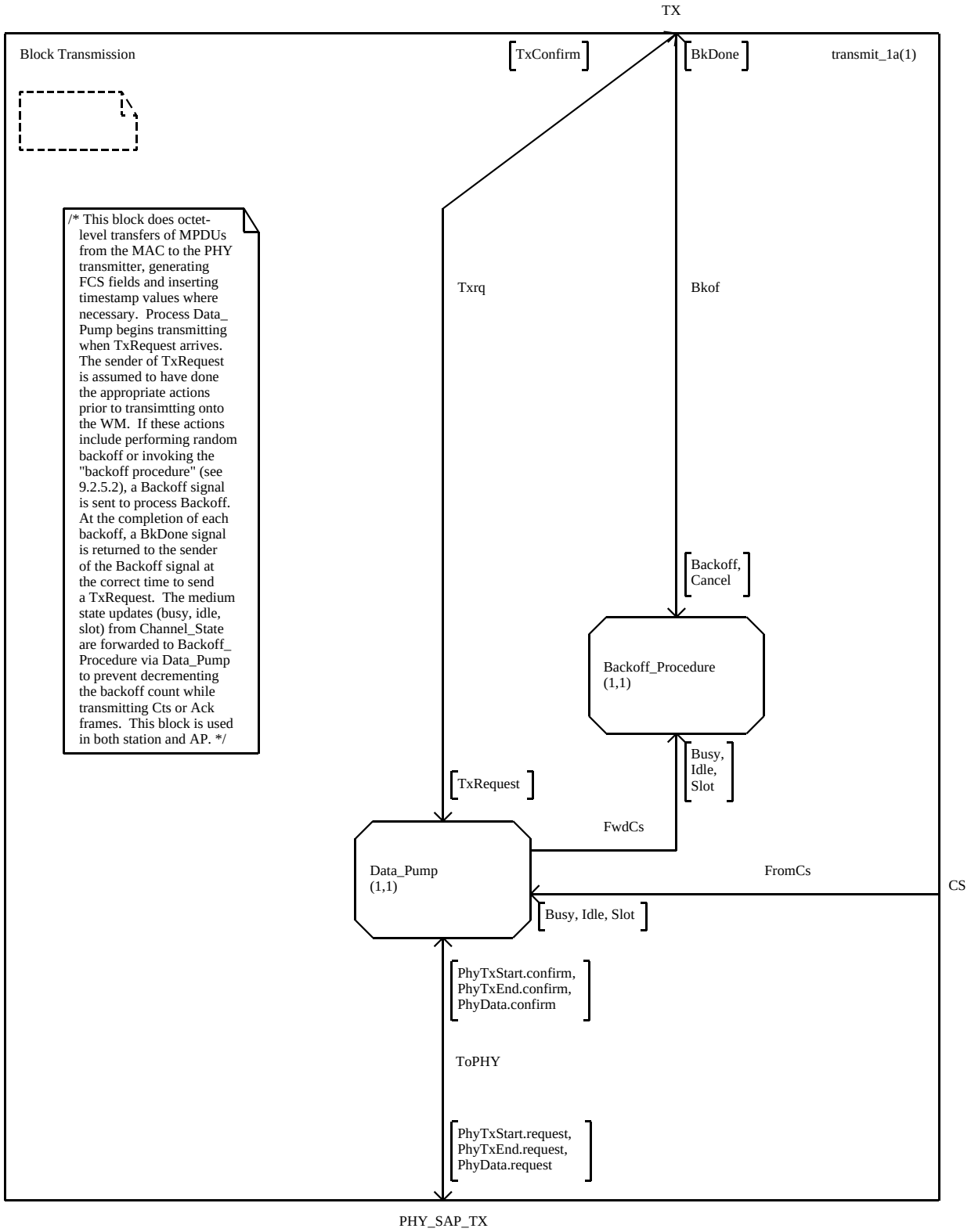




Copyright © 2007 IEEE. All rights reserved.

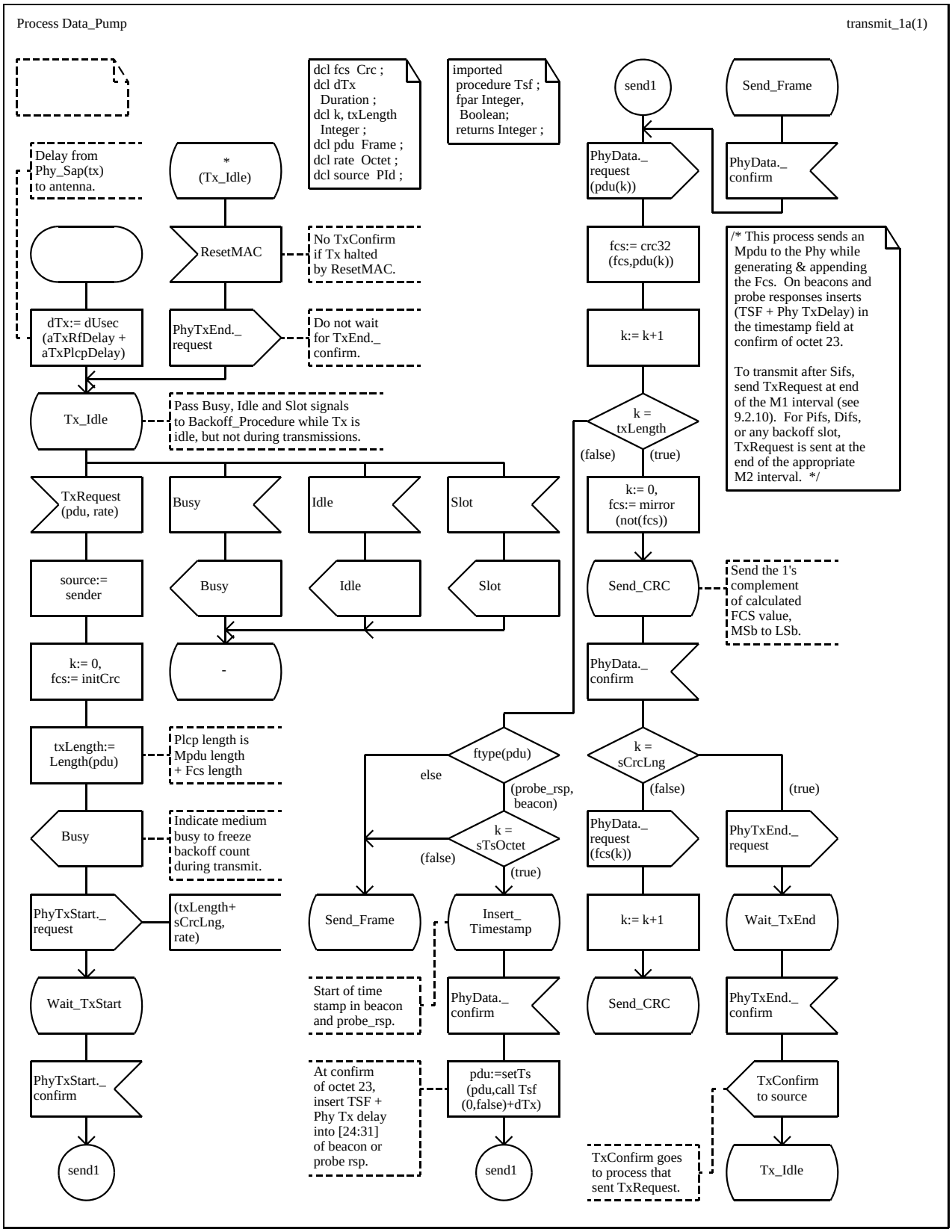



Copyright © 2007 IEEE. All rights reserved.

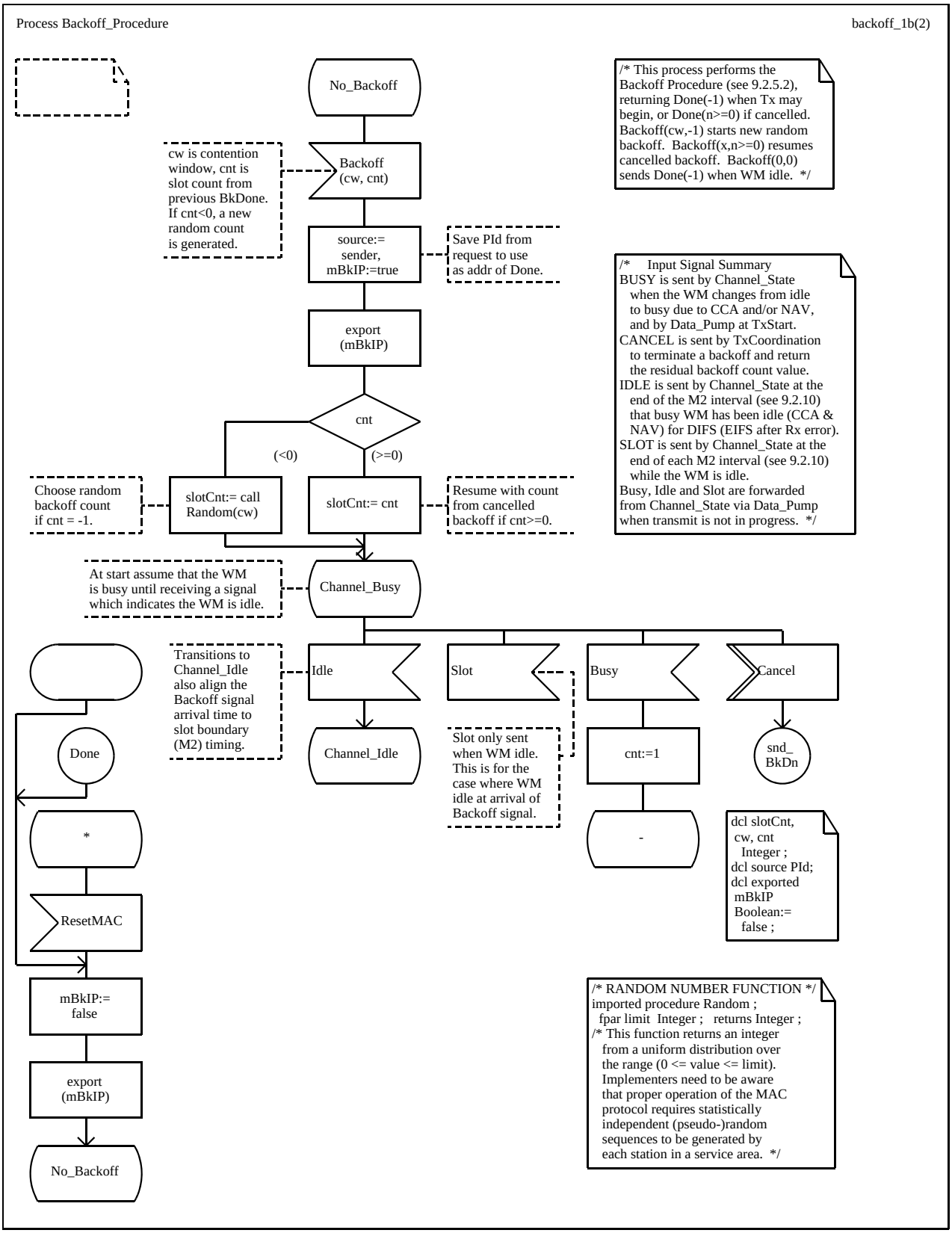




Copyright © 2007 IEEE. All rights reserved.

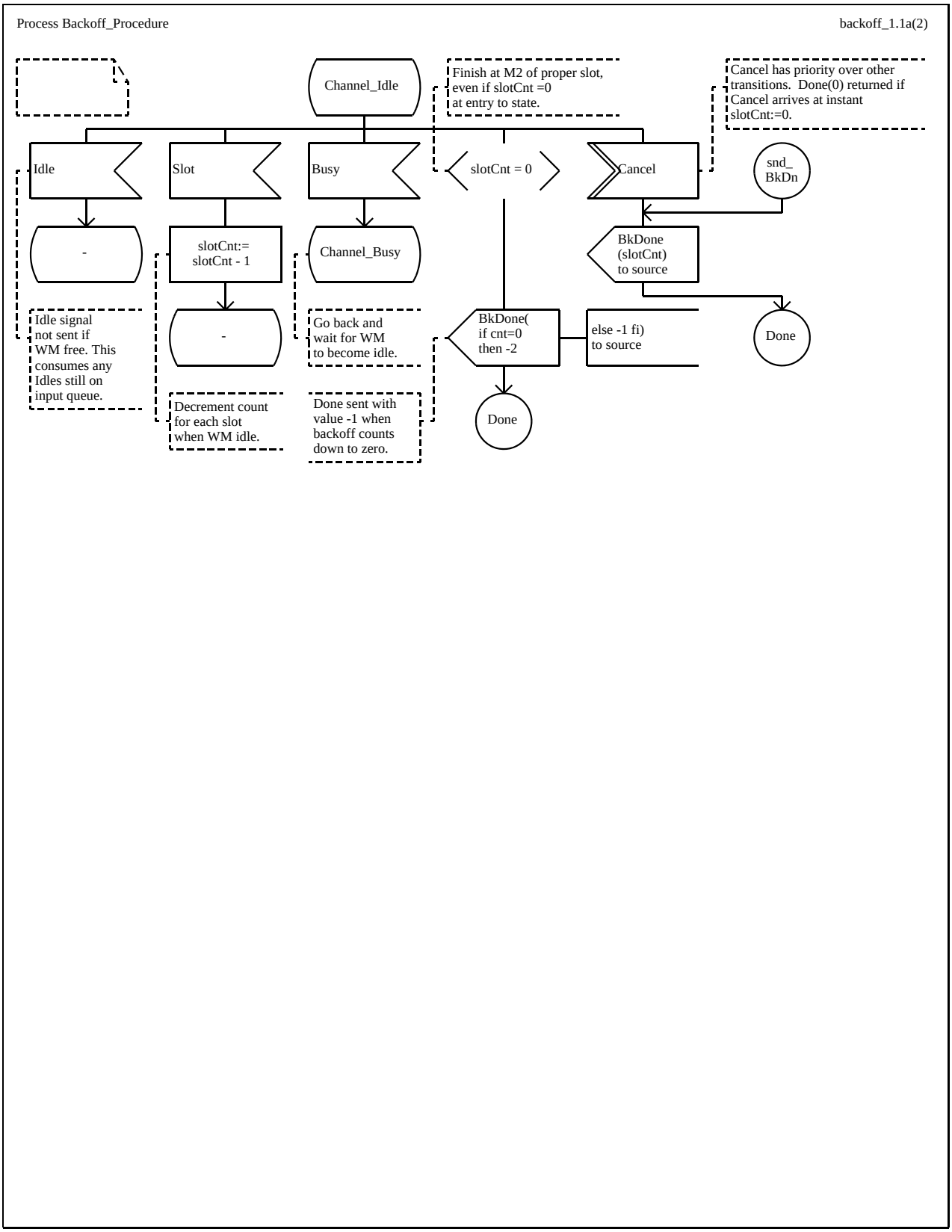



Copyright © 2007 IEEE. All rights reserved.

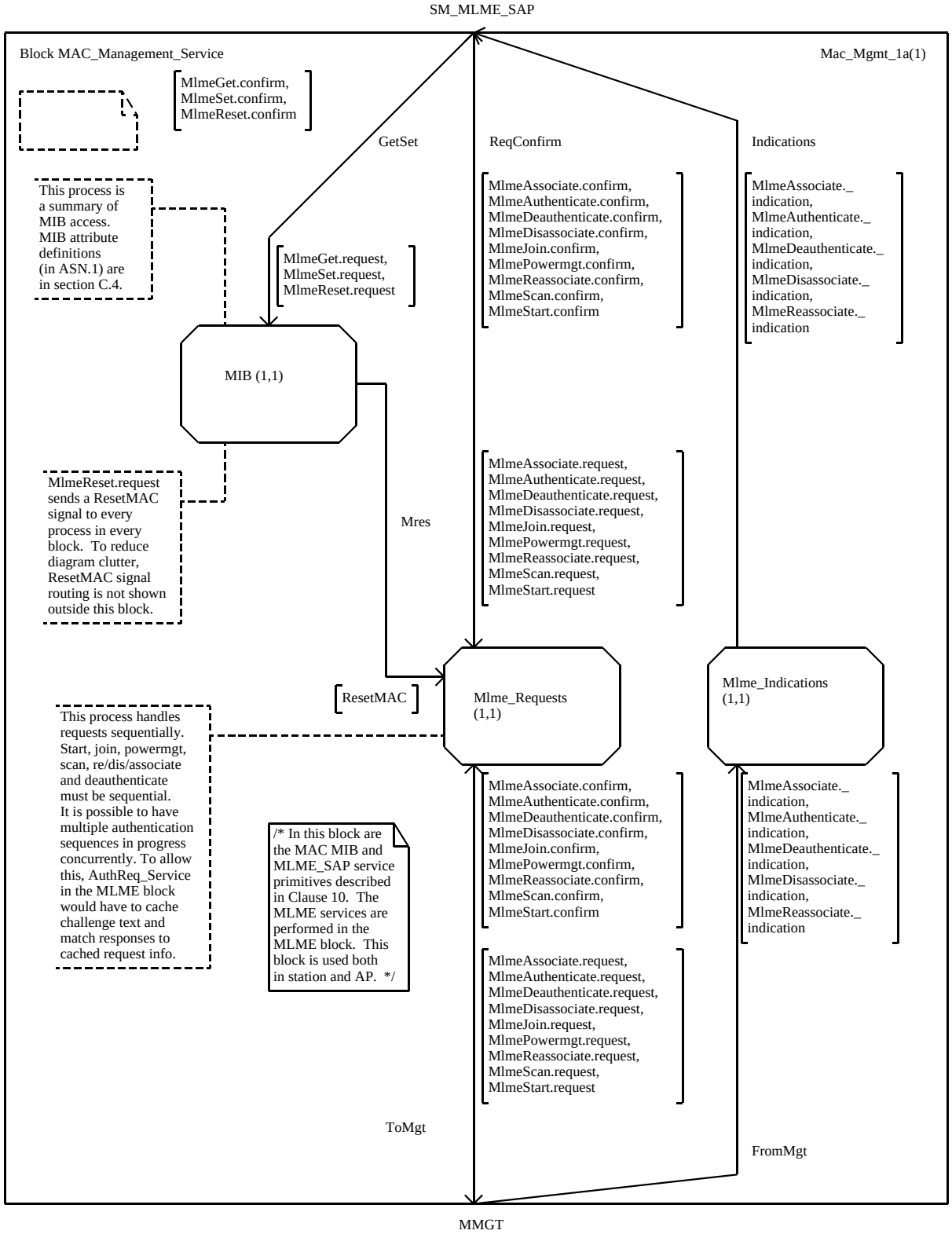




Copyright © 2007 IEEE. All rights reserved.

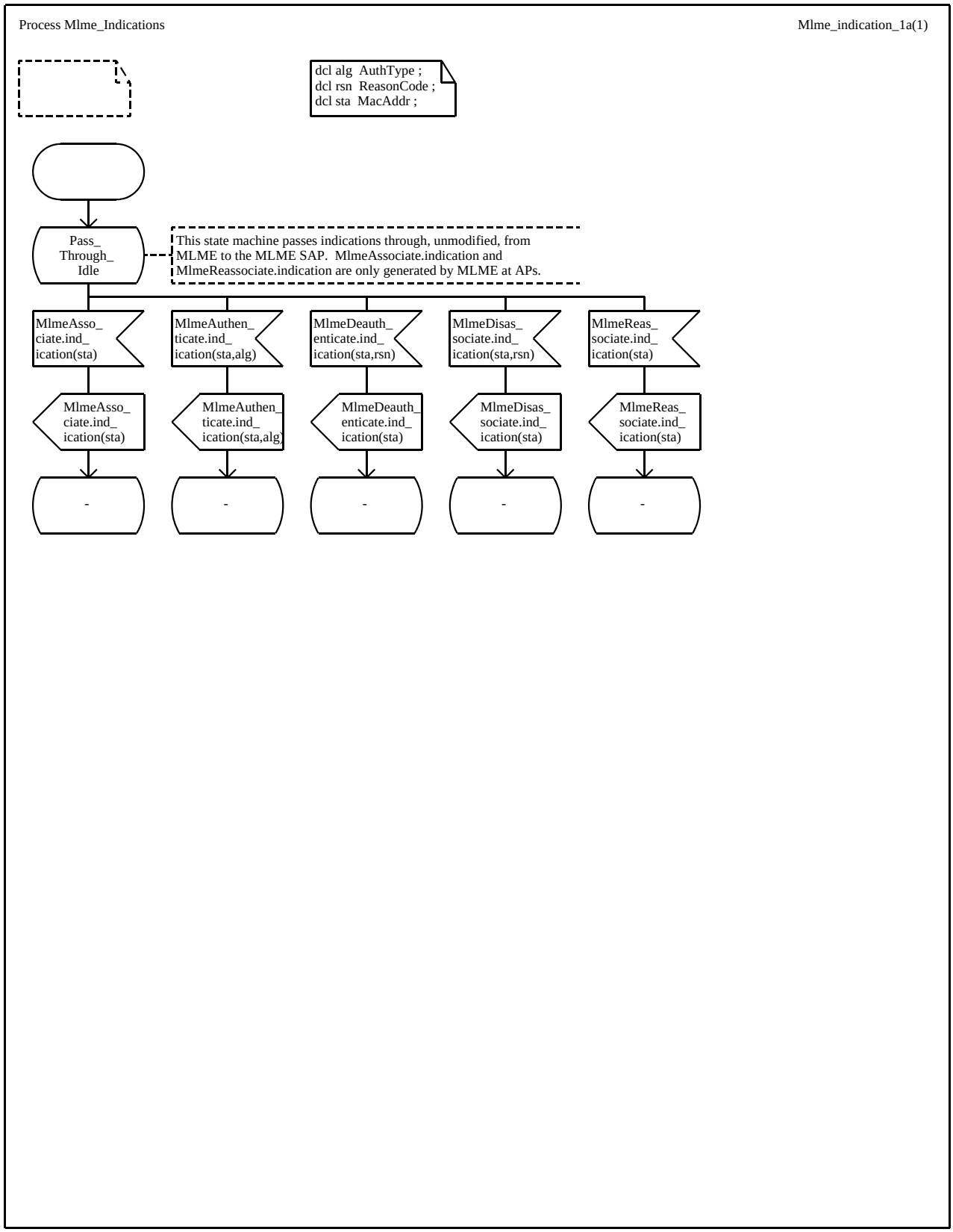



Copyright © 2007 IEEE. All rights reserved.

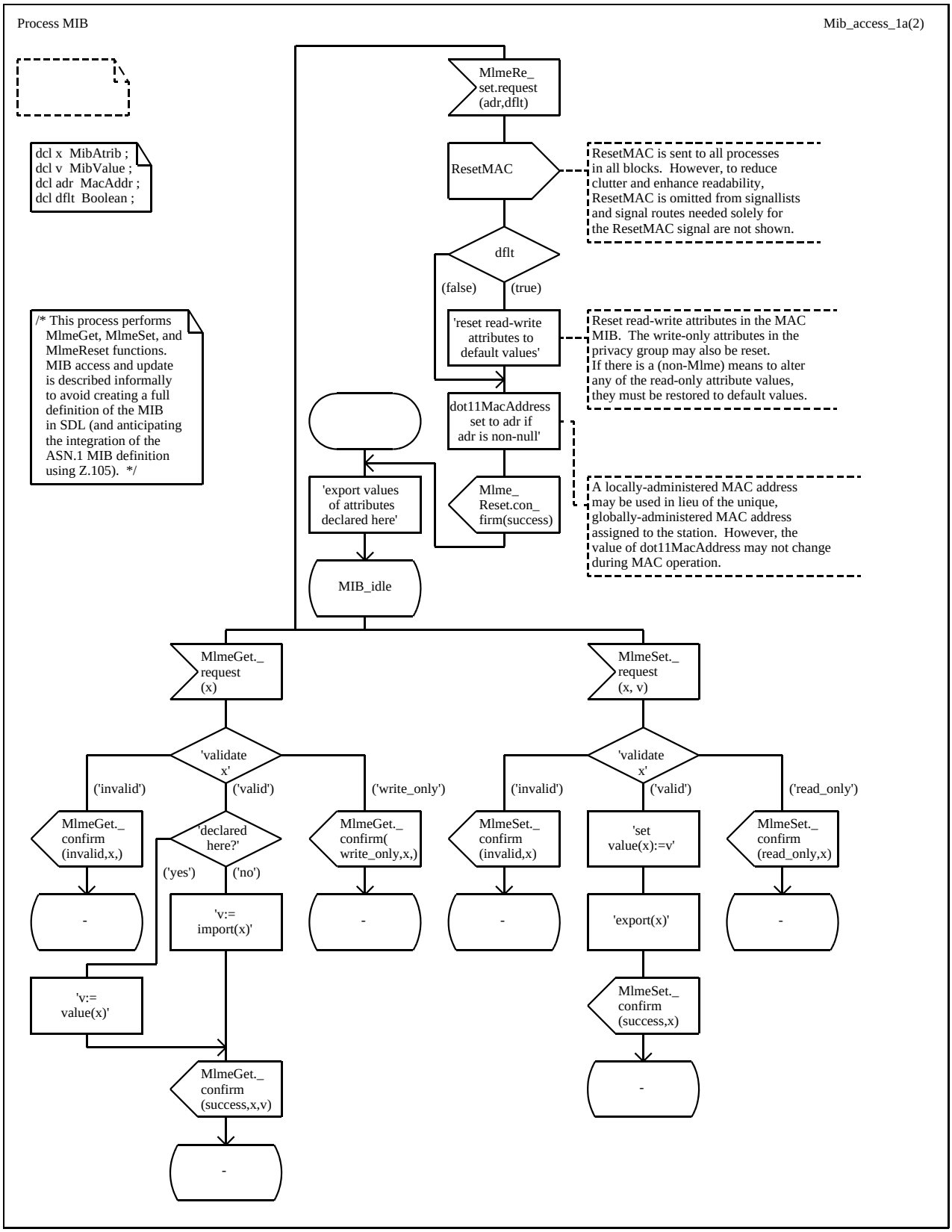




Copyright © 2007 IEEE. All rights reserved.

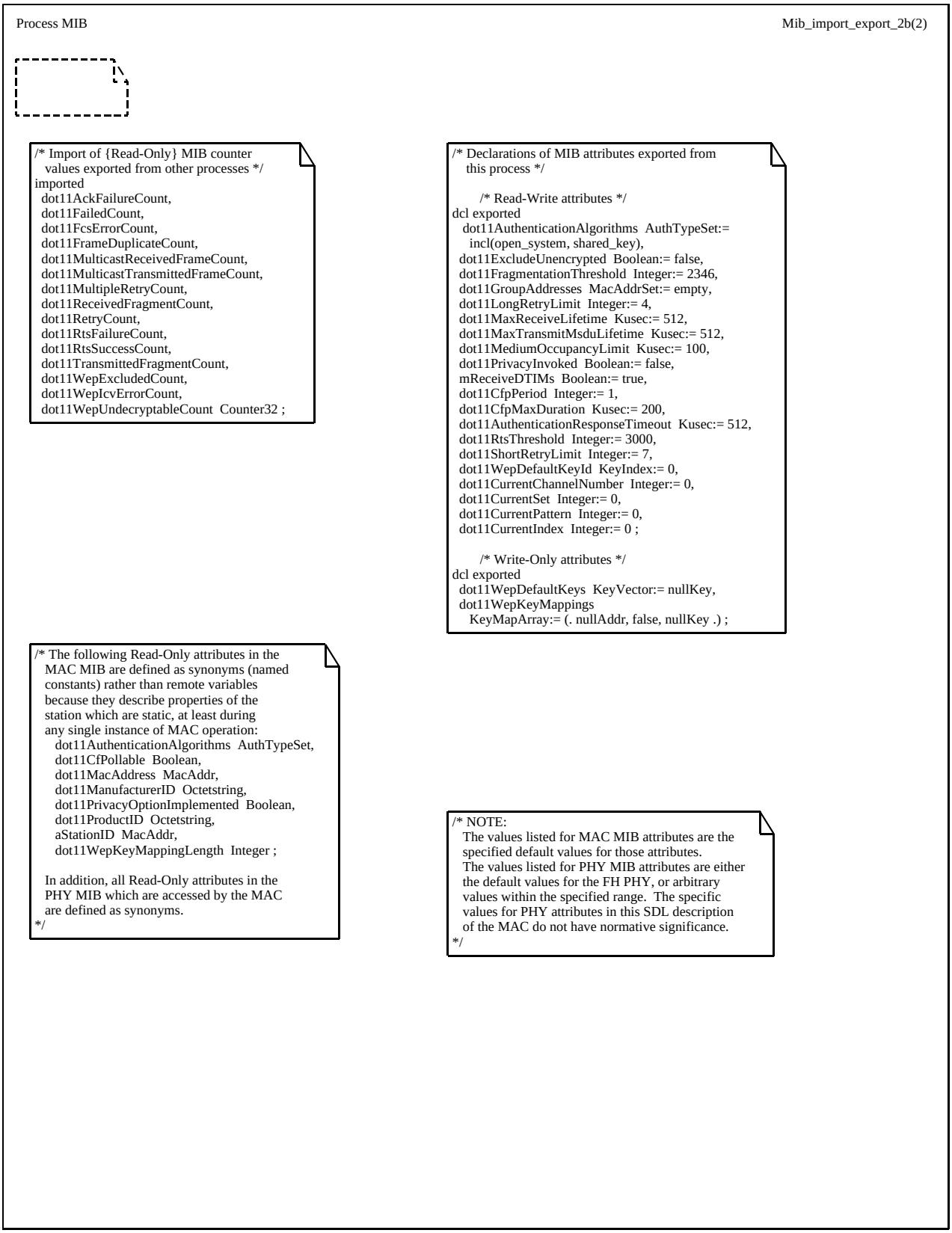



Copyright © 2007 IEEE. All rights reserved.

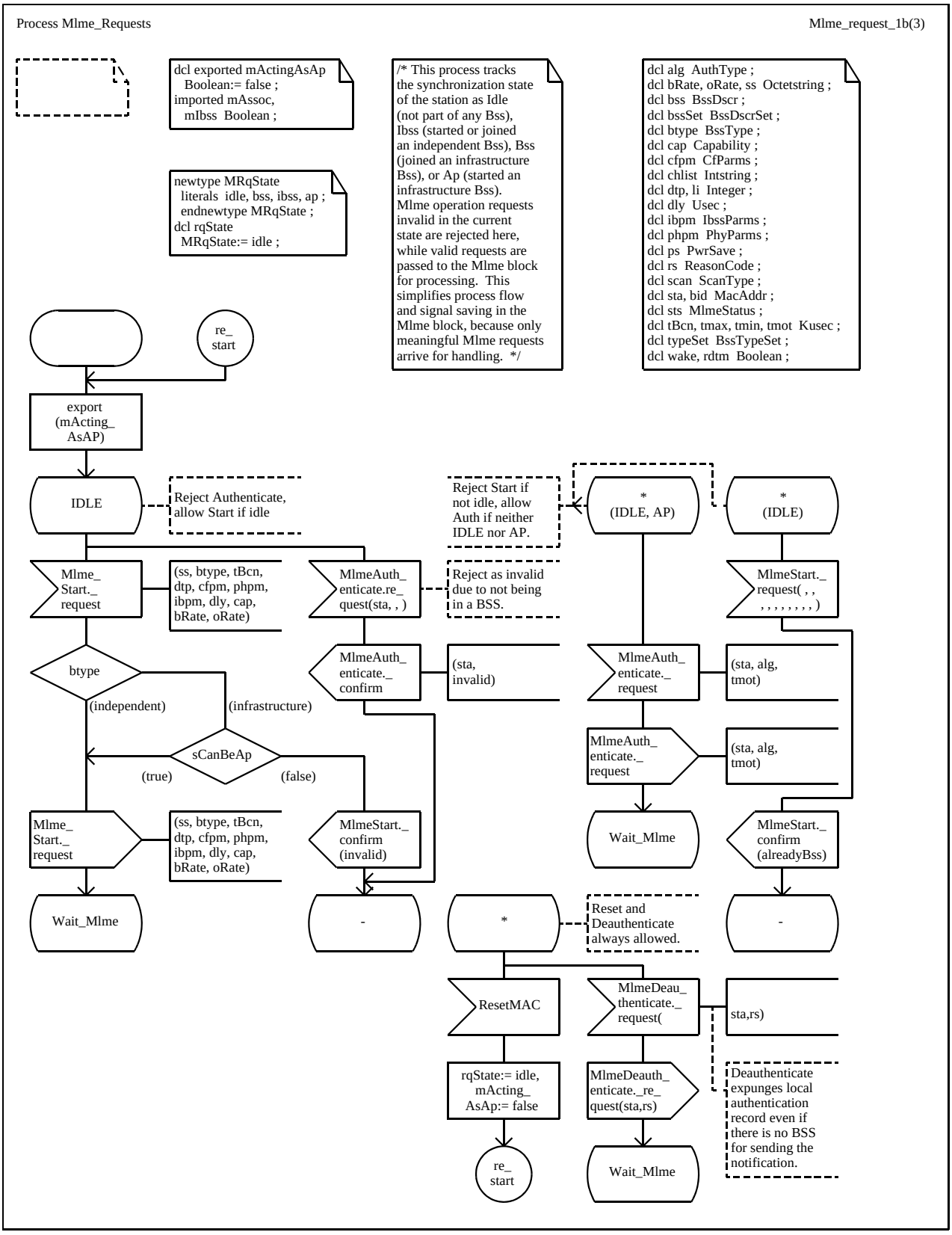




Copyright © 2007 IEEE. All rights reserved.

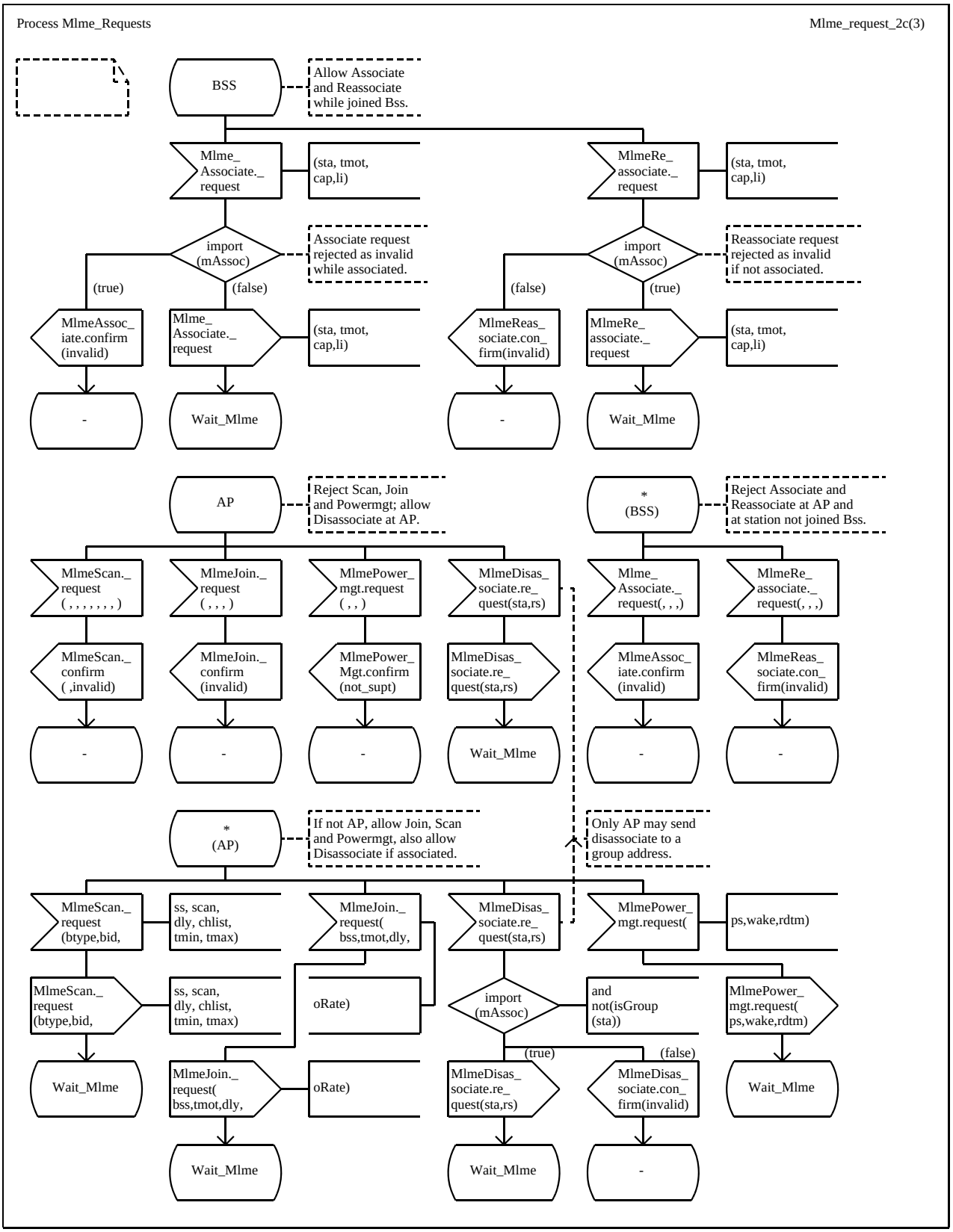



Copyright © 2007 IEEE. All rights reserved.

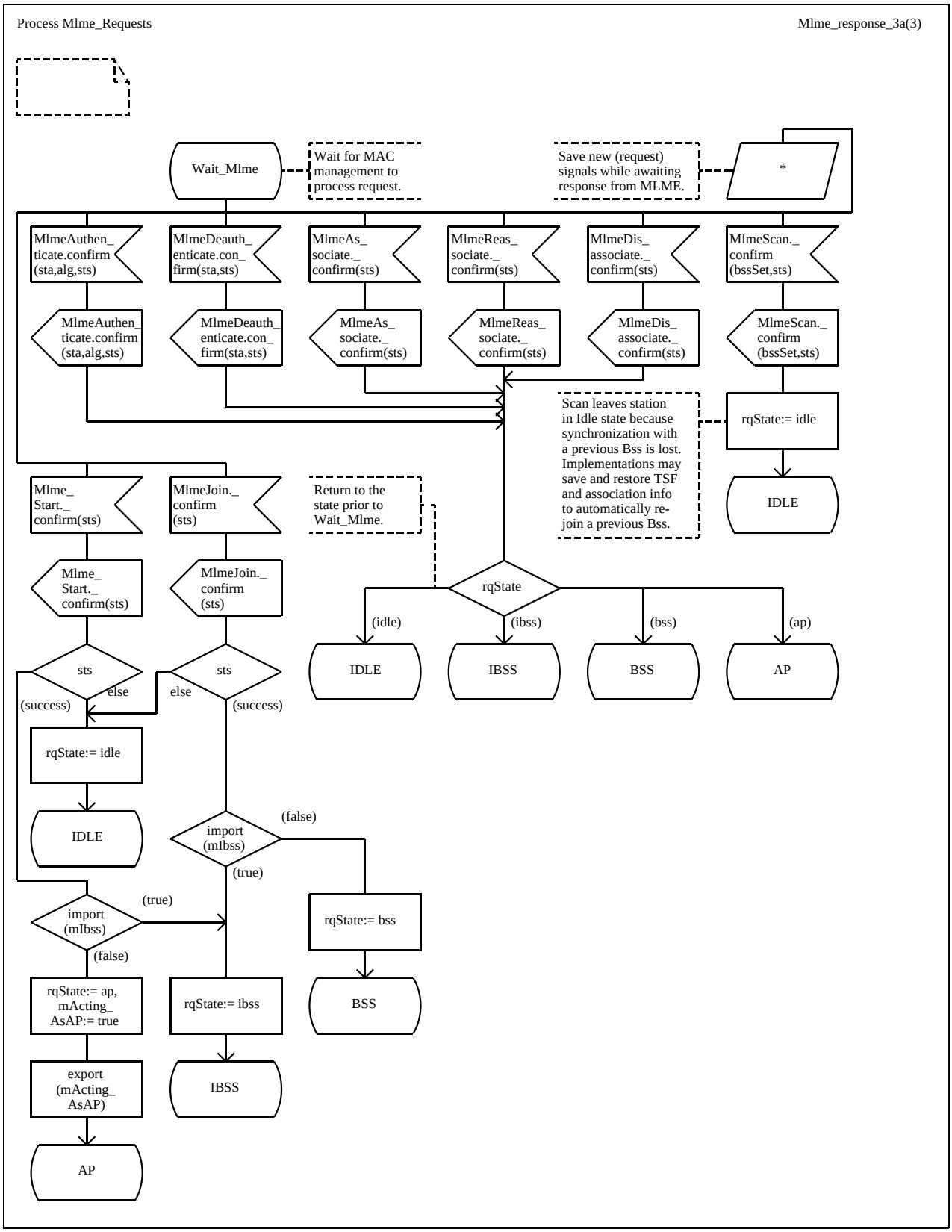




Copyright © 2007 IEEE. All rights reserved.

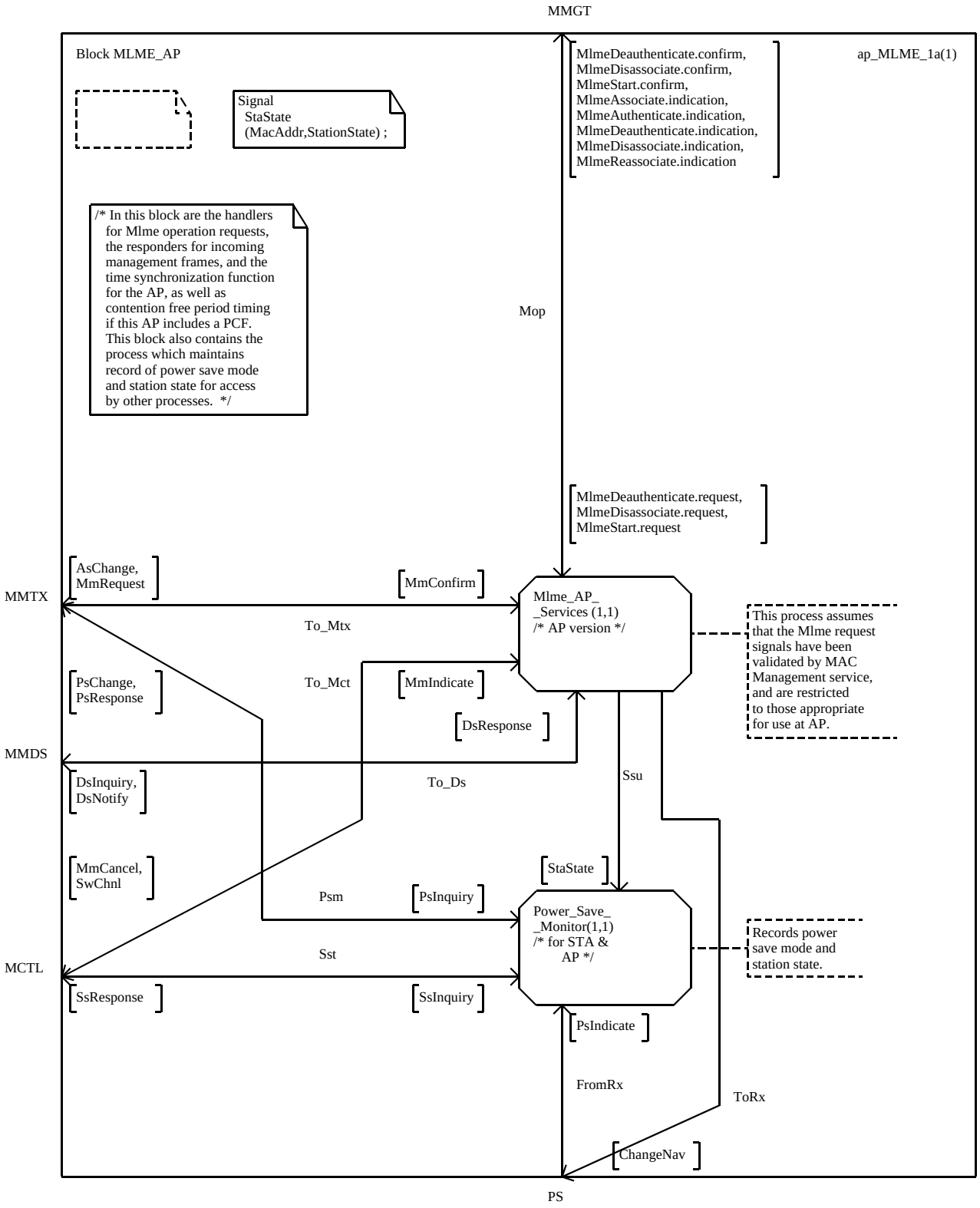



Copyright © 2007 IEEE. All rights reserved.

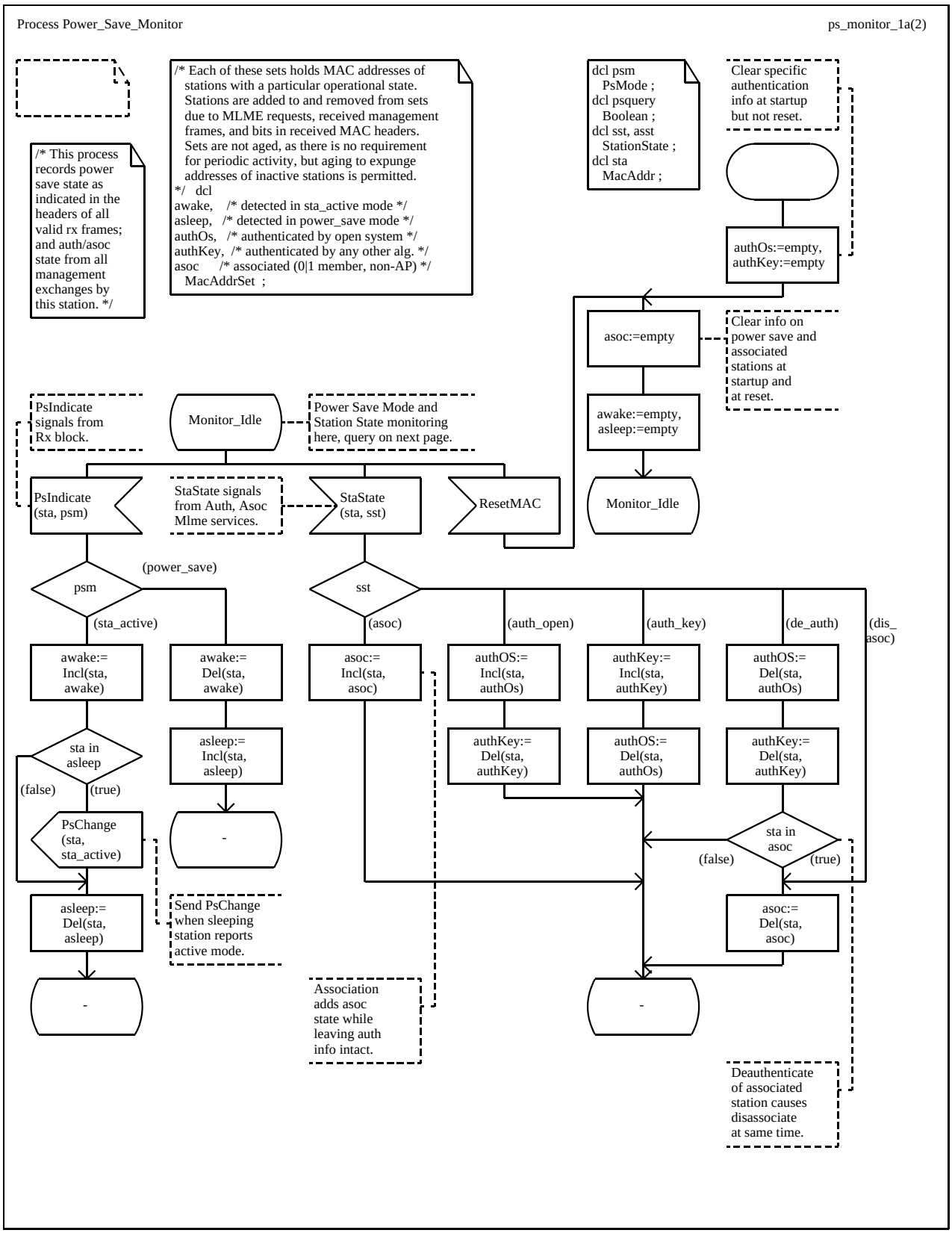




Copyright © 2007 IEEE. All rights reserved.

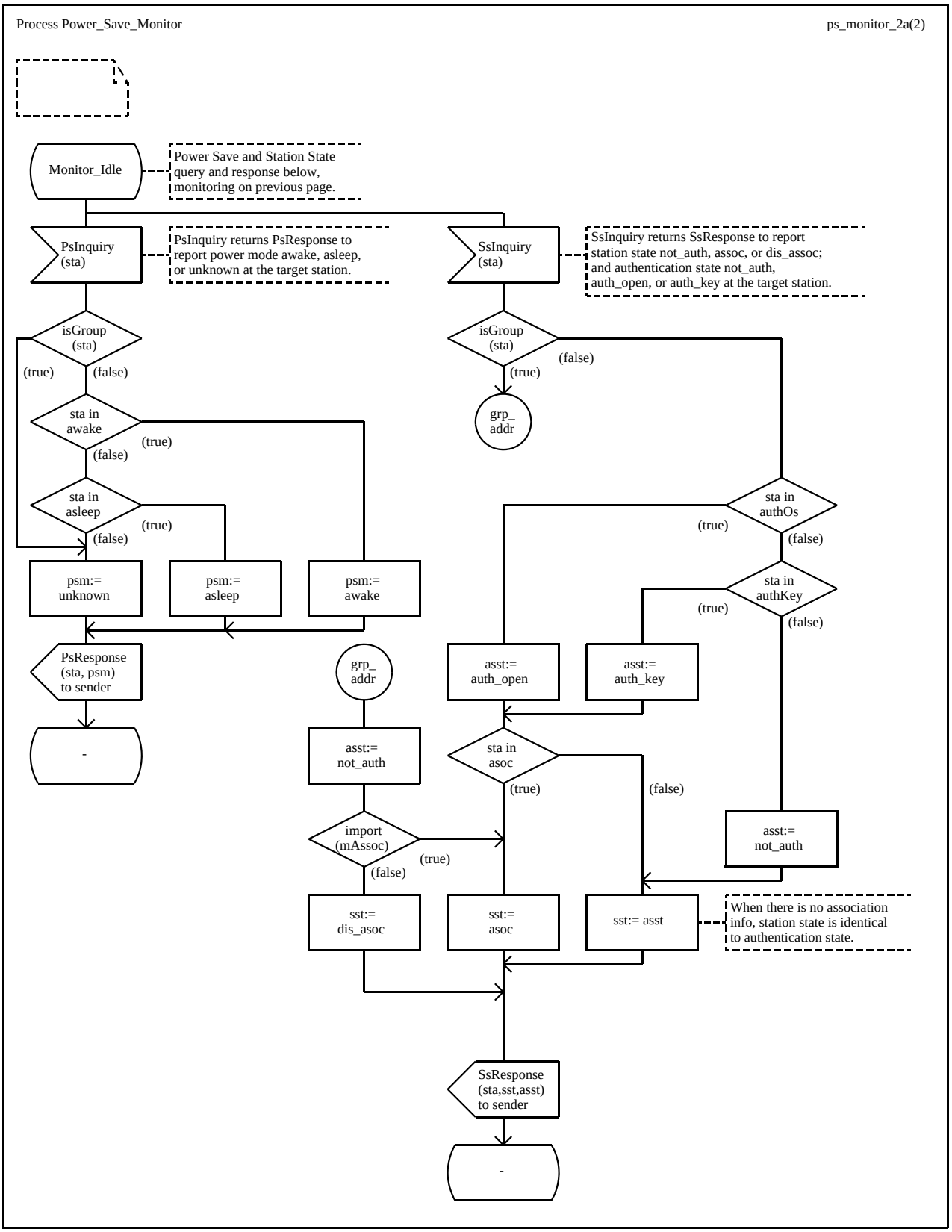



Copyright © 2007 IEEE. All rights reserved.

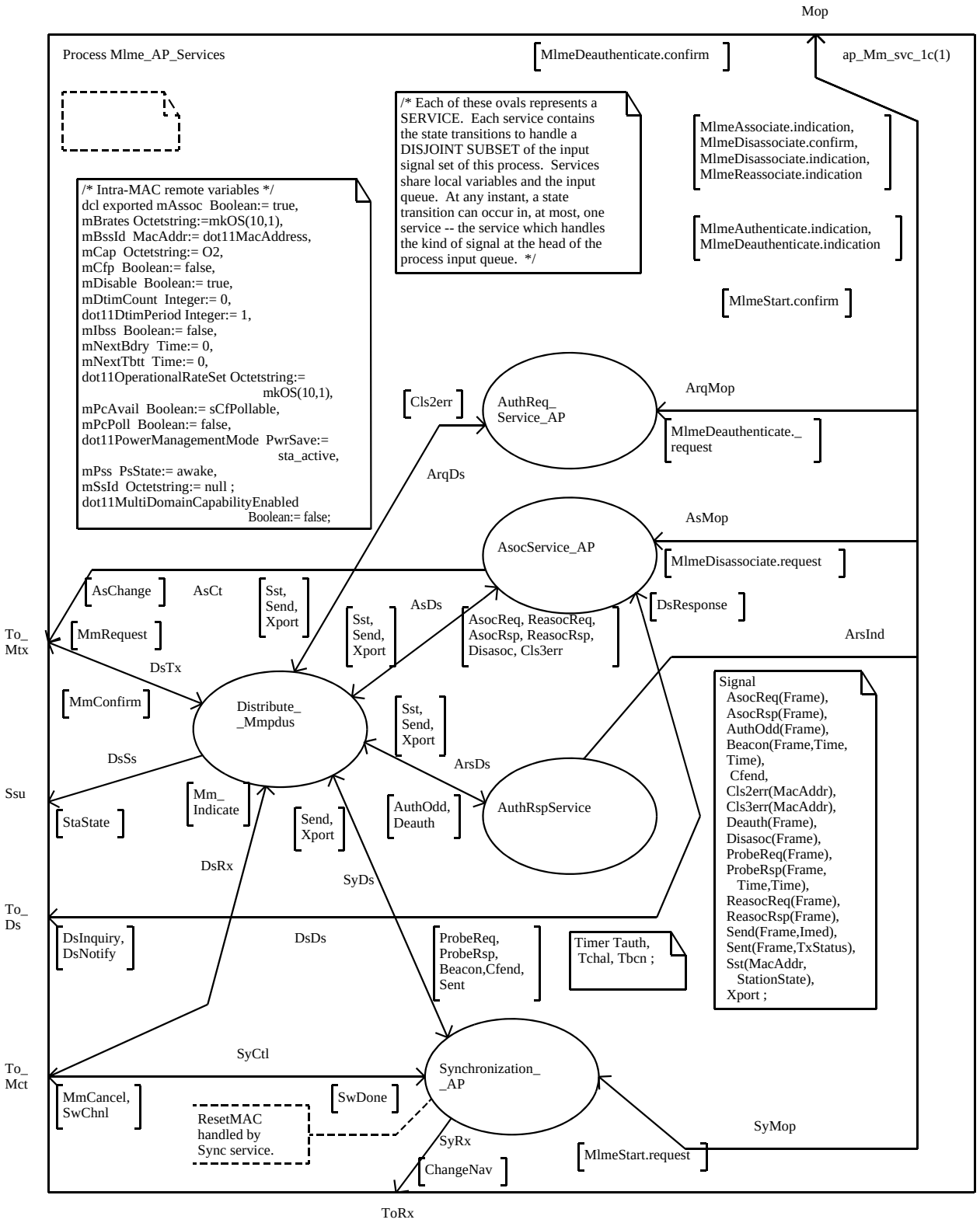




Copyright © 2007 IEEE. All rights reserved.

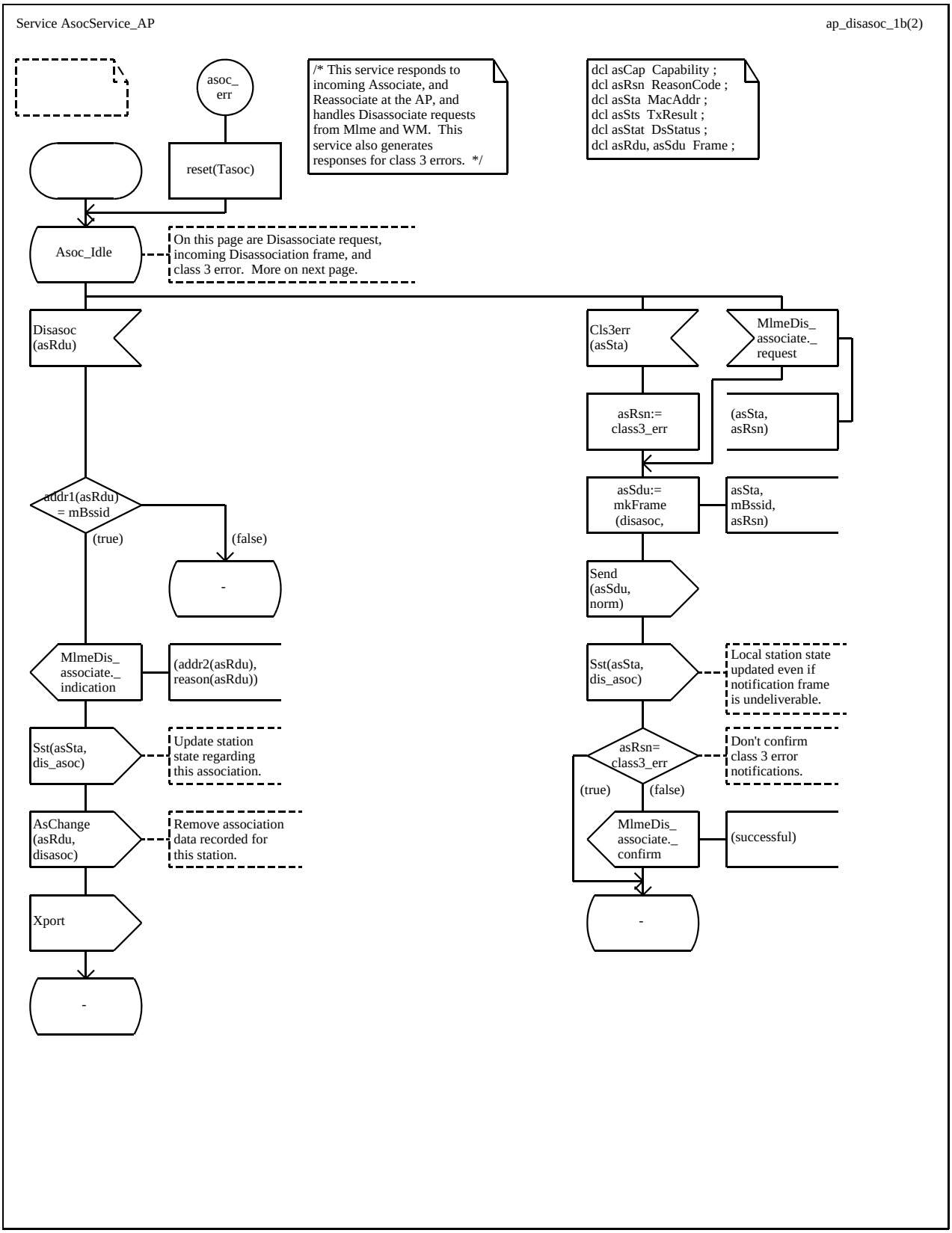



Copyright © 2007 IEEE. All rights reserved.

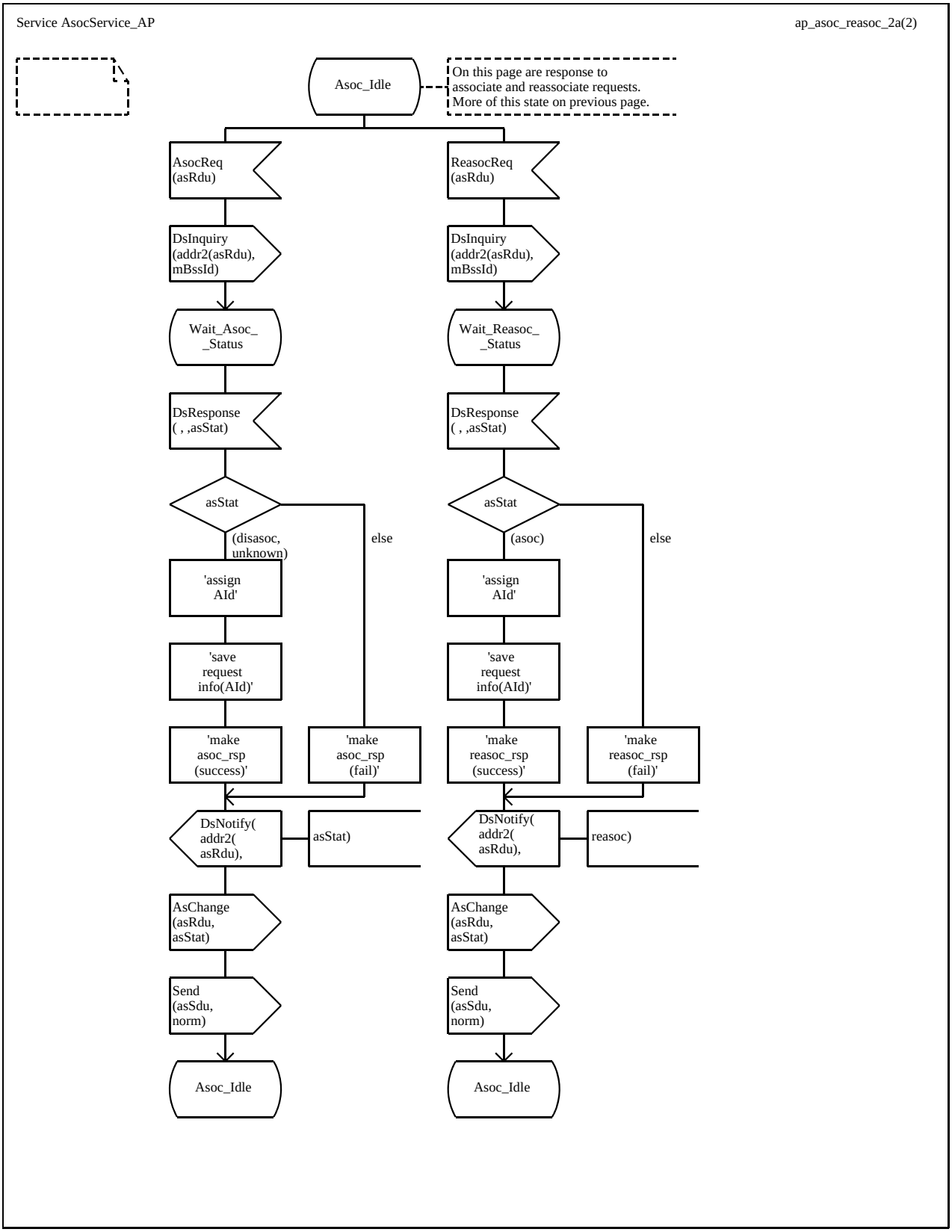




Copyright © 2007 IEEE. All rights reserved.

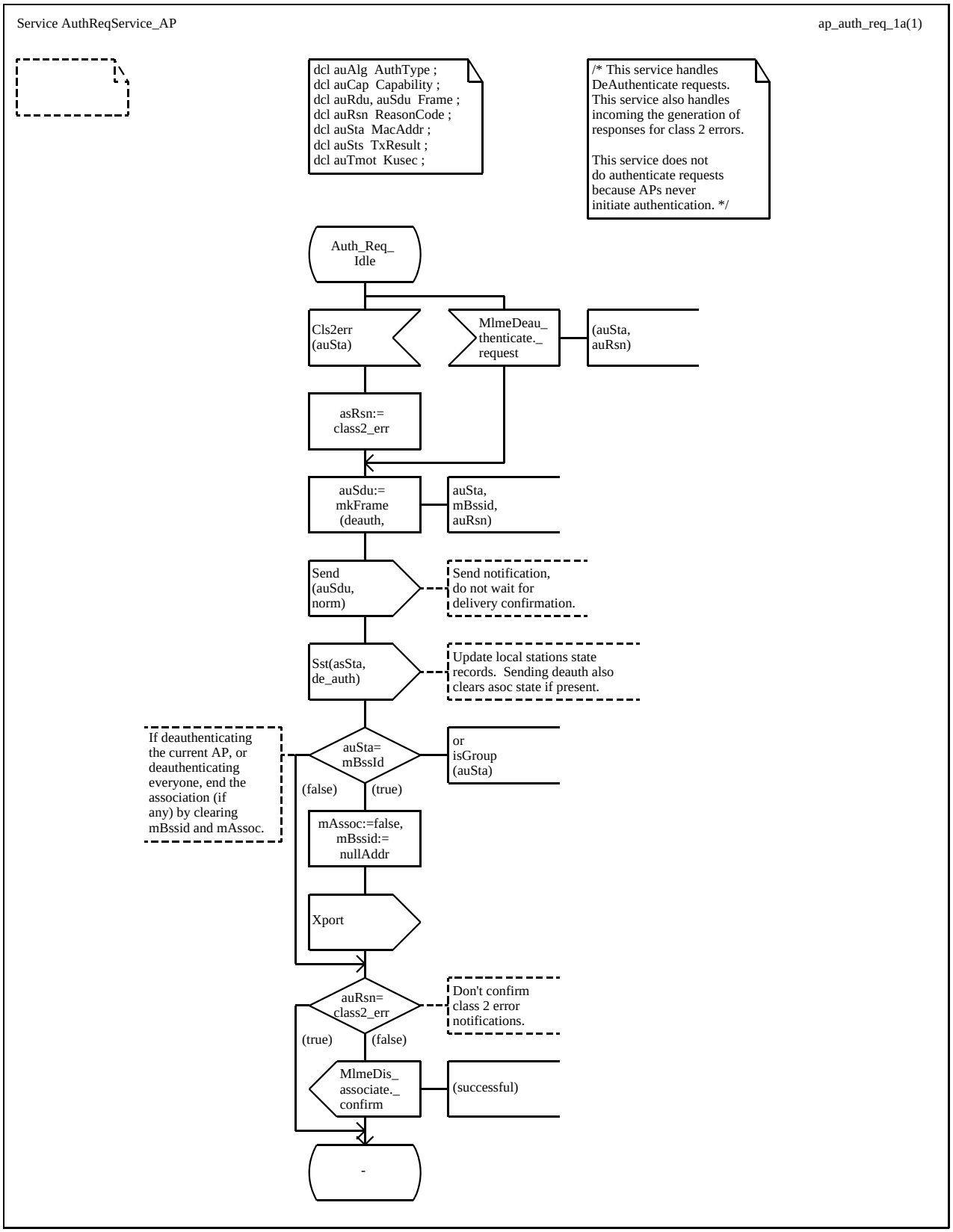



Copyright © 2007 IEEE. All rights reserved.

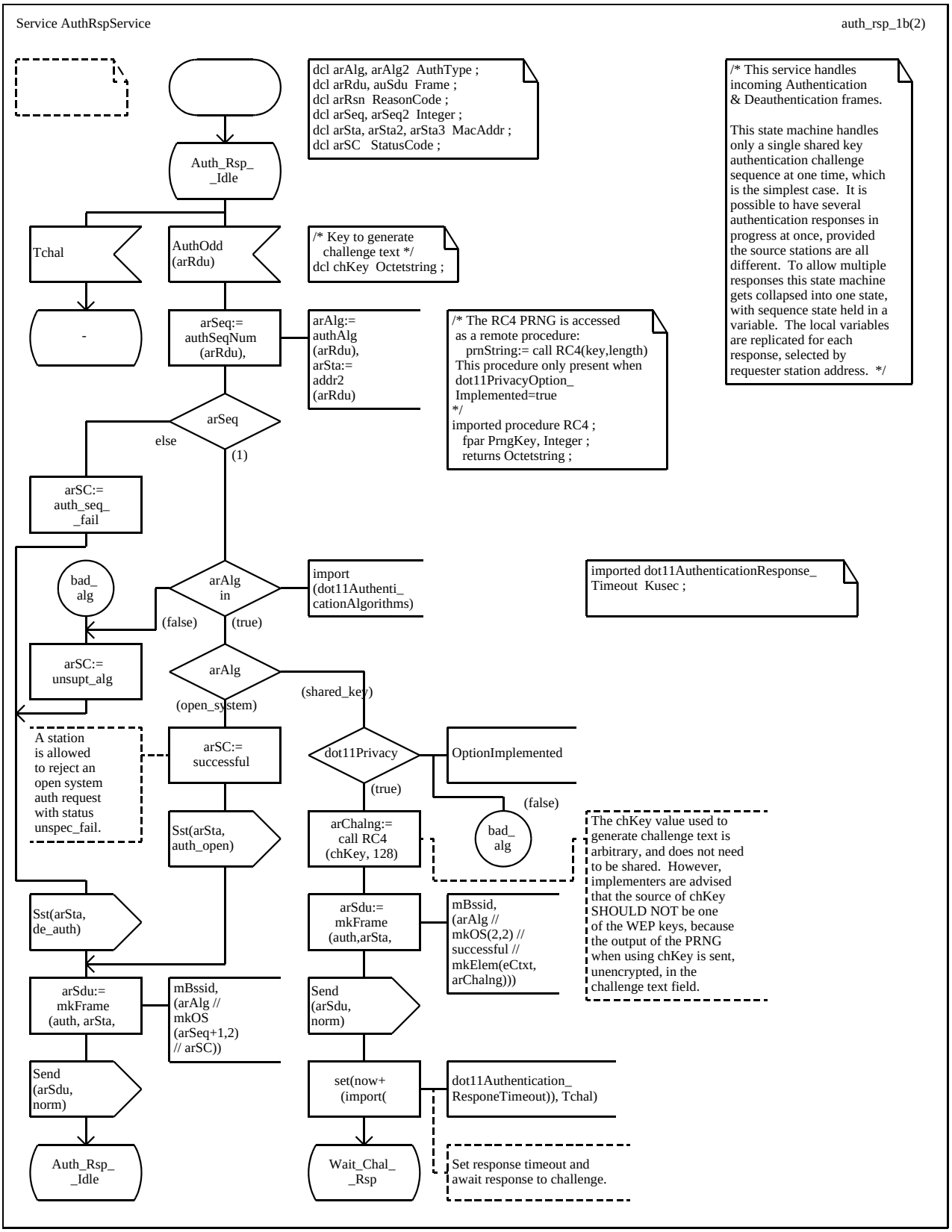




Copyright © 2007 IEEE. All rights reserved.

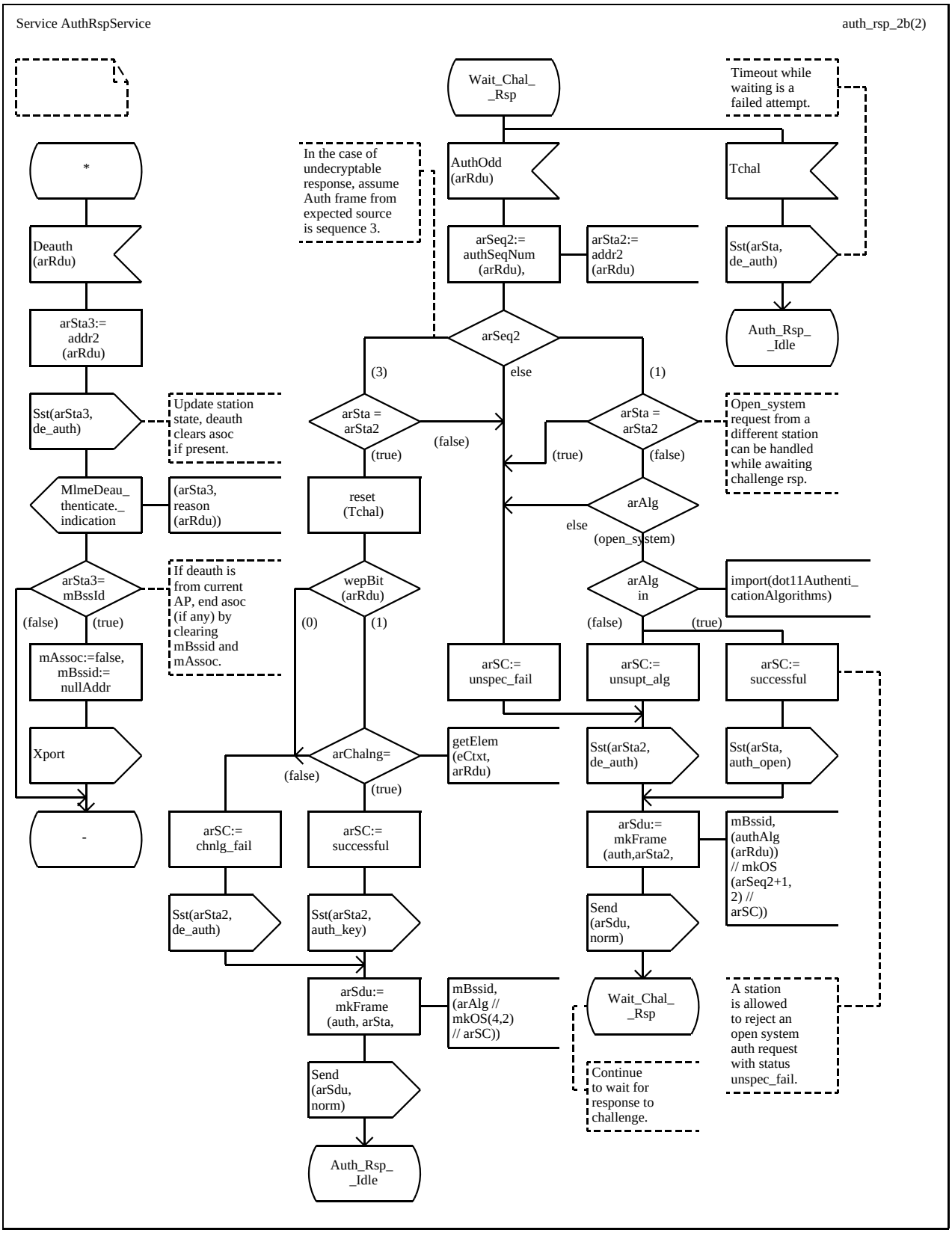



Copyright © 2007 IEEE. All rights reserved.

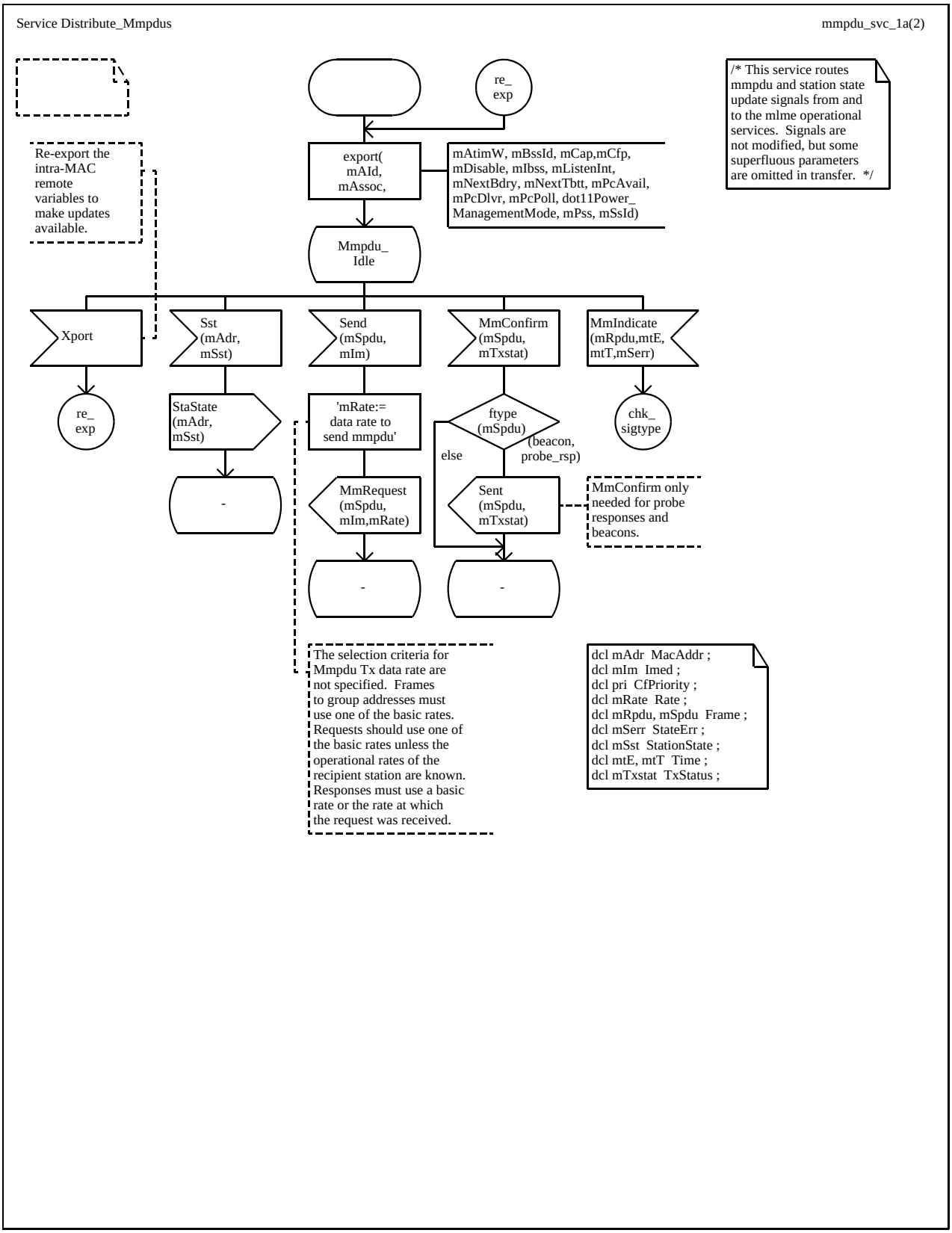




Copyright © 2007 IEEE. All rights reserved.

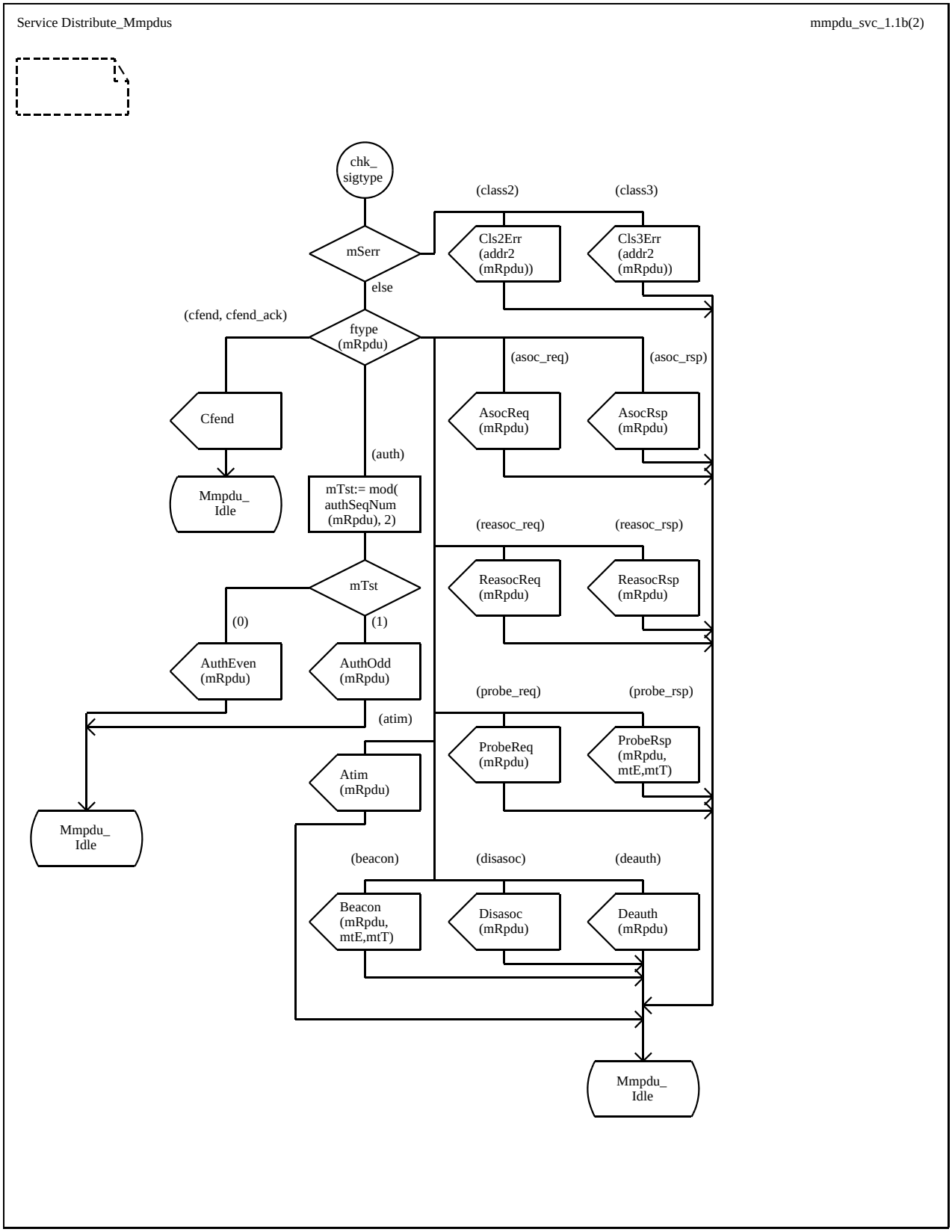



Copyright © 2007 IEEE. All rights reserved.

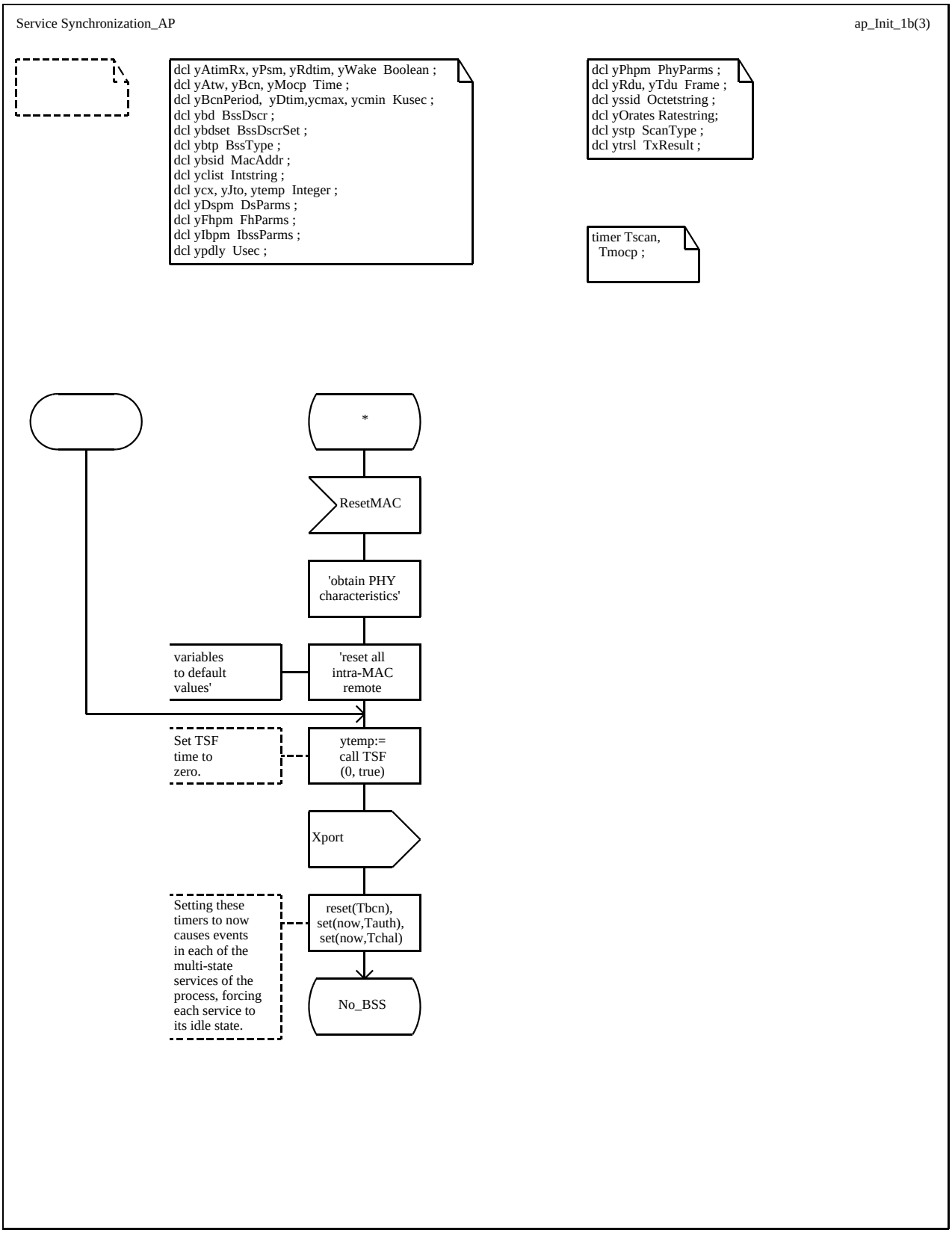




Copyright © 2007 IEEE. All rights reserved.

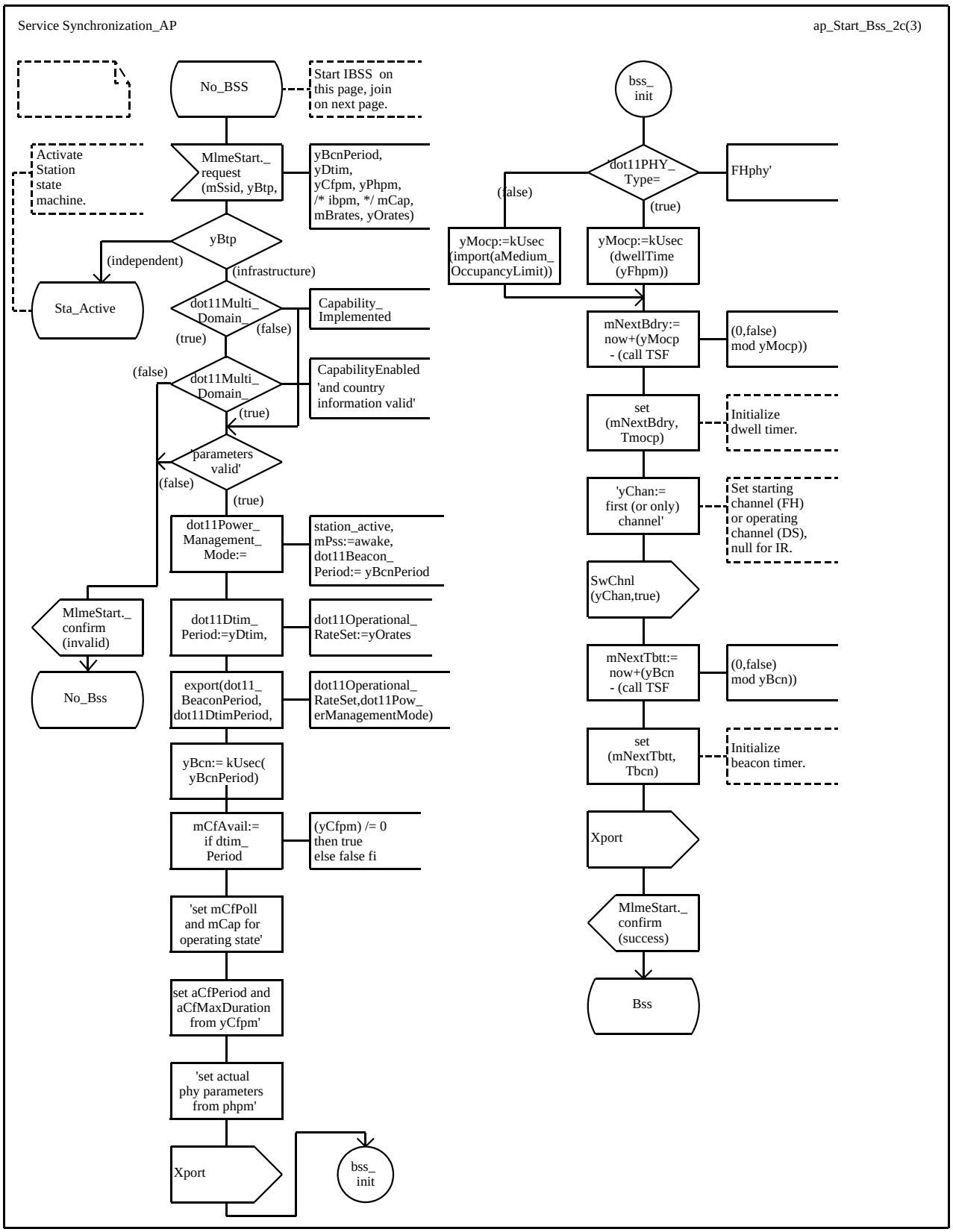



Copyright © 2007 IEEE. All rights reserved.

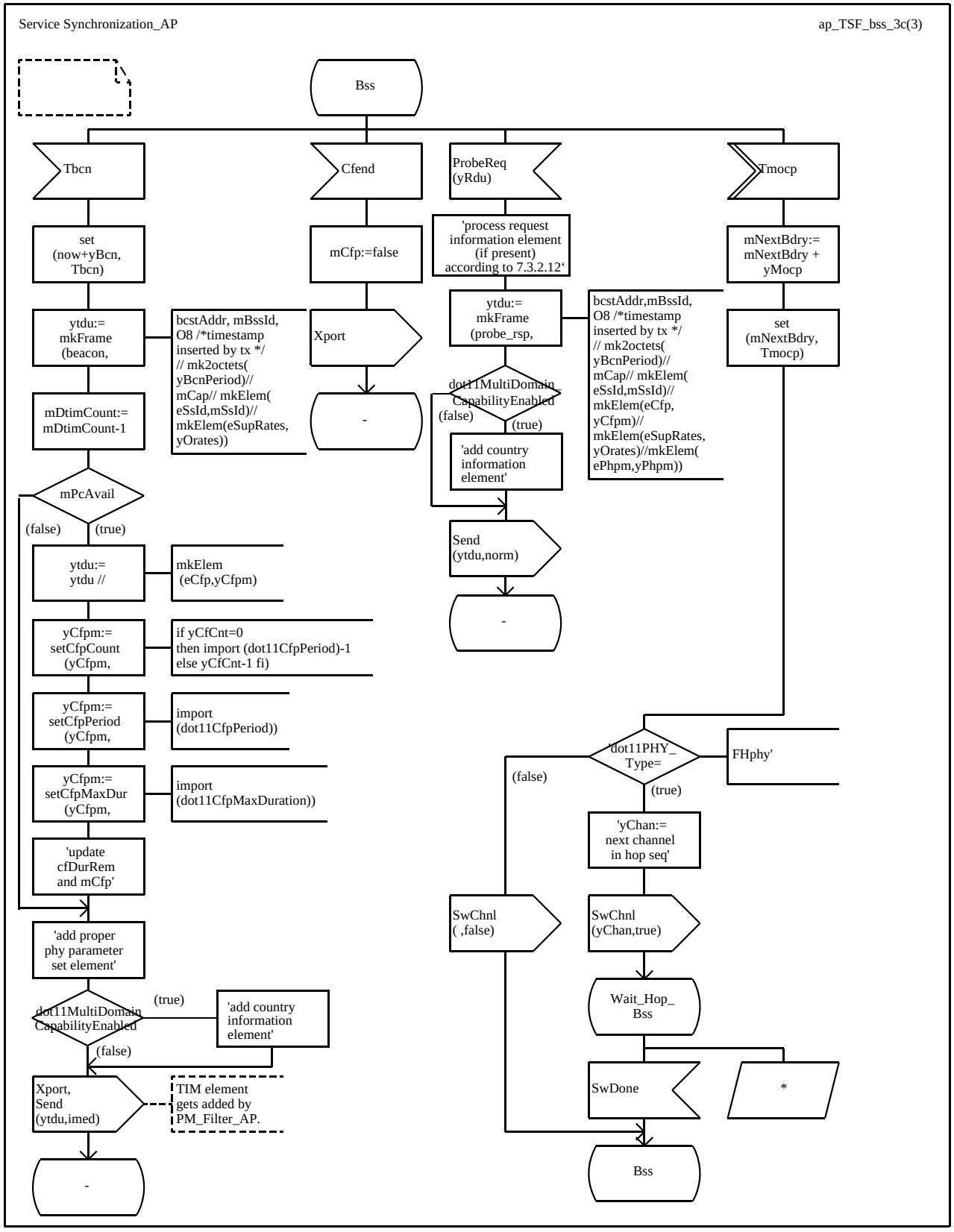




Copyright © 2007 IEEE. All rights reserved.

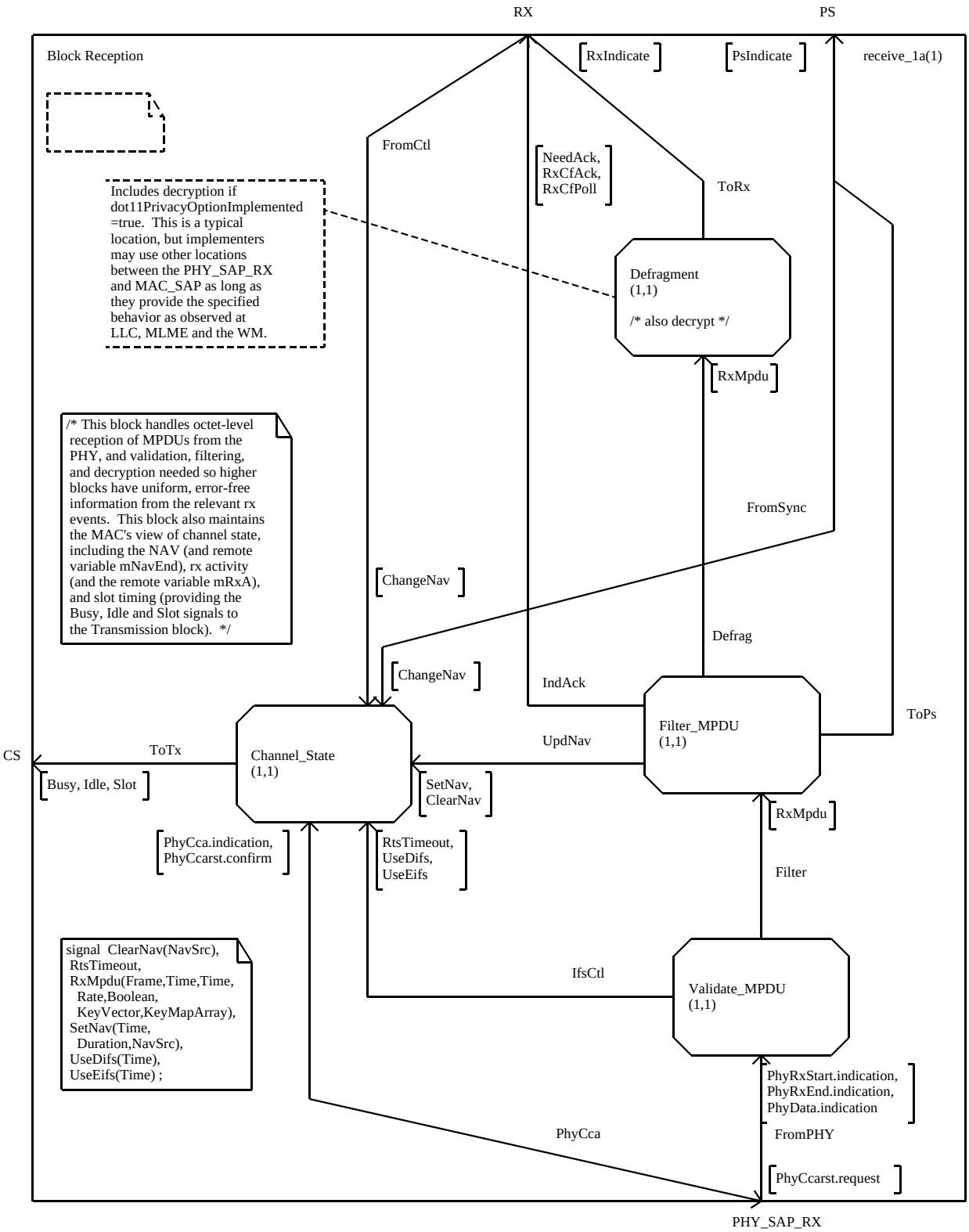



Copyright © 2007 IEEE. All rights reserved.

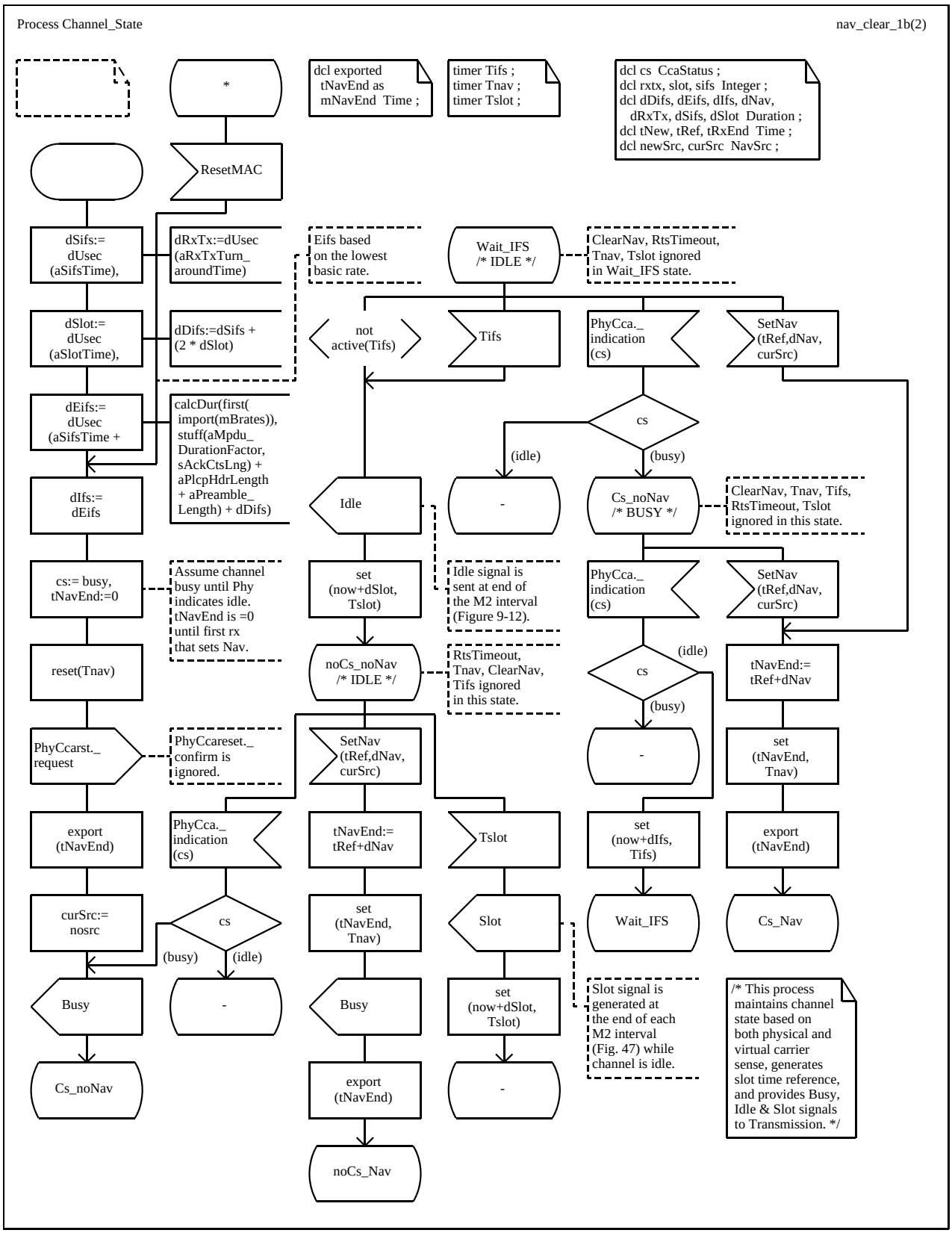




Copyright © 2007 IEEE. All rights reserved.

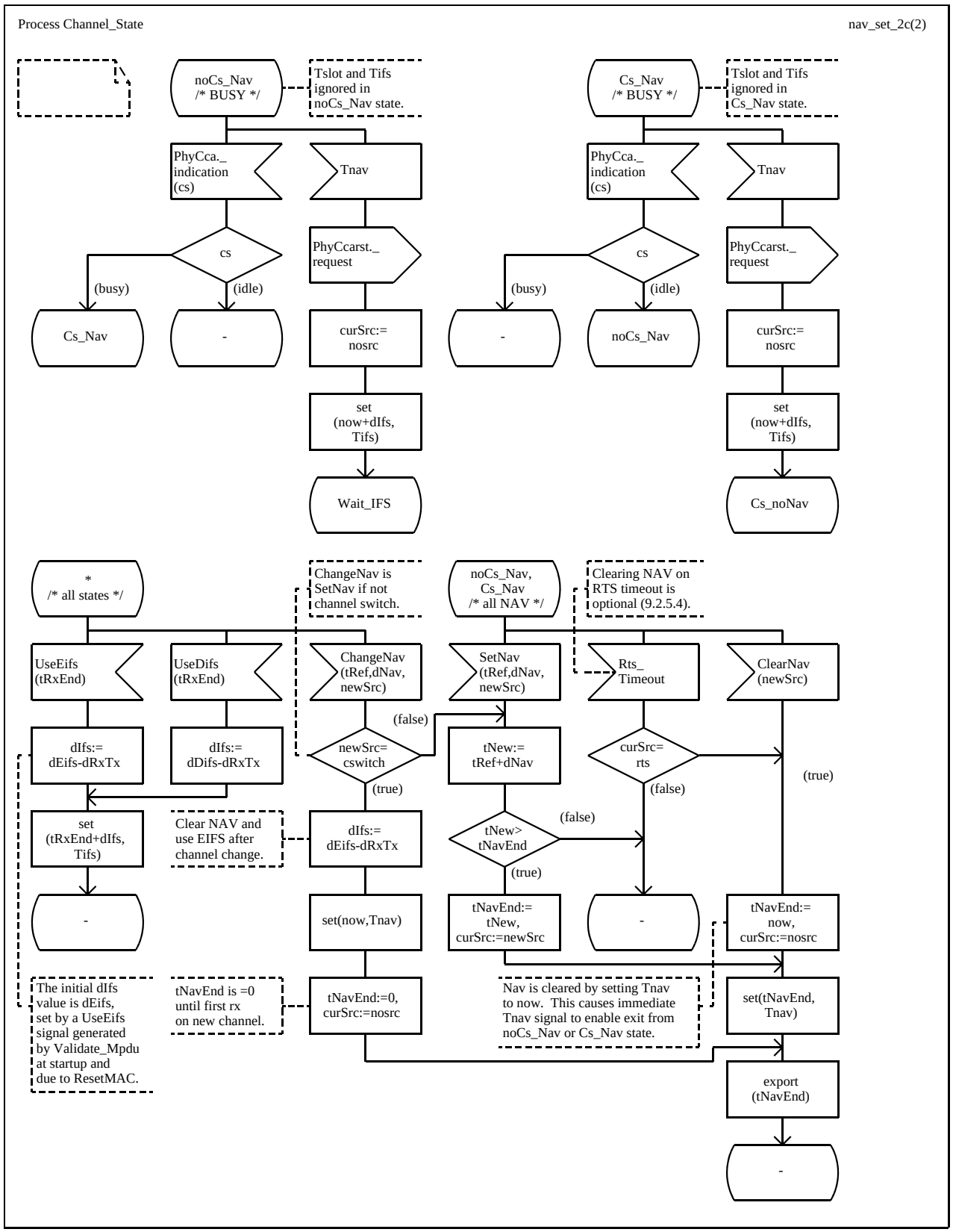



Copyright © 2007 IEEE. All rights reserved.

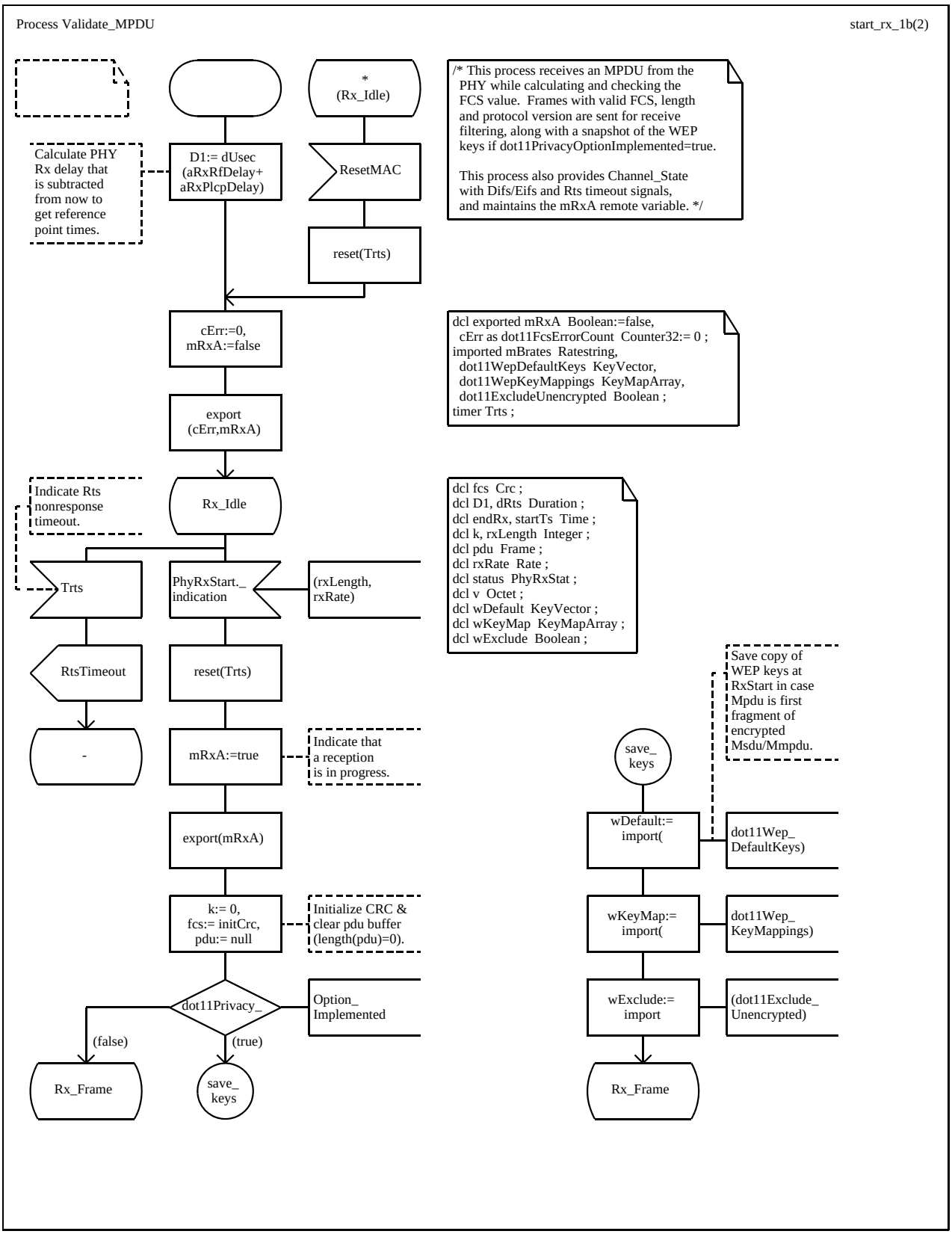




Copyright © 2007 IEEE. All rights reserved.

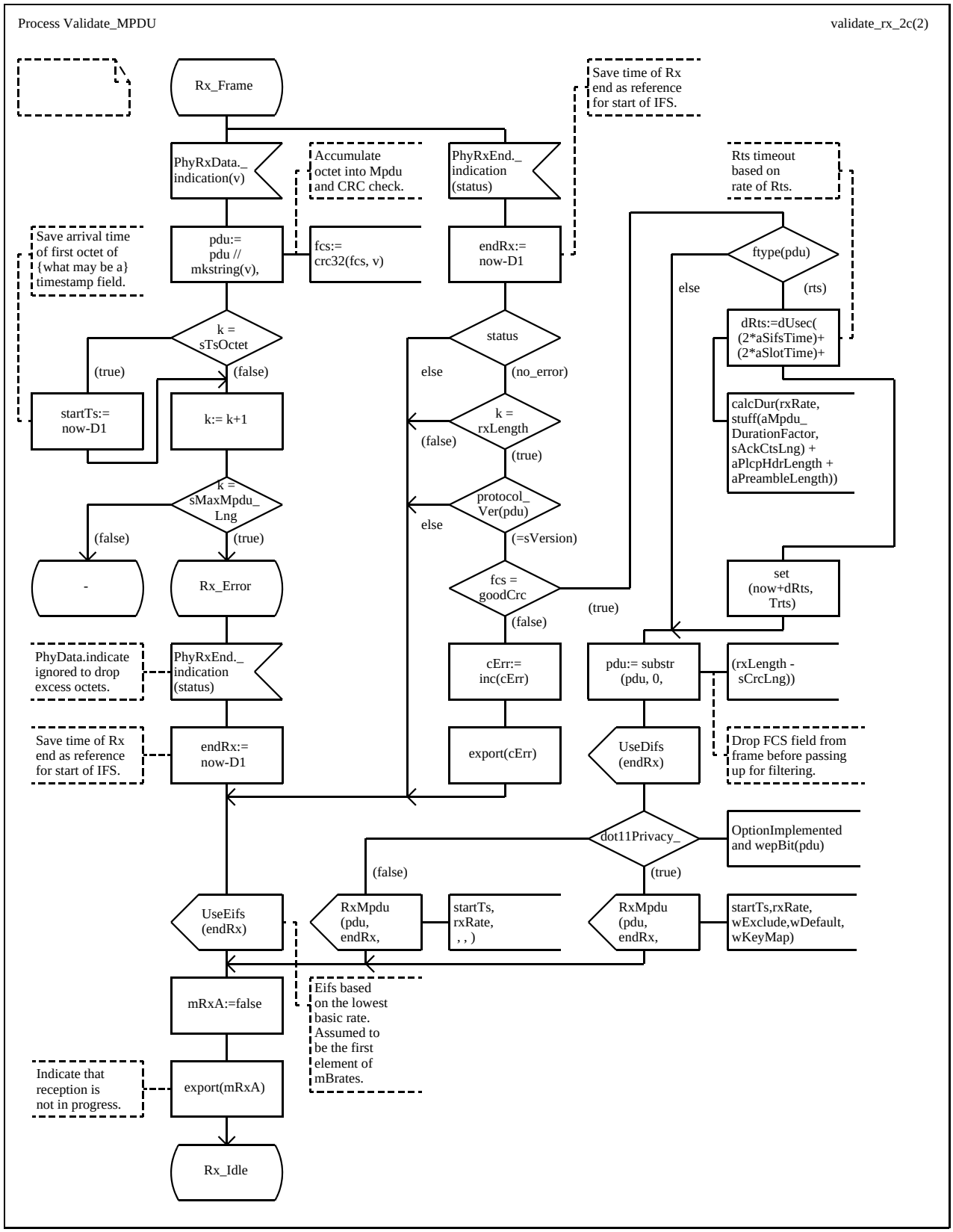



Copyright © 2007 IEEE. All rights reserved.

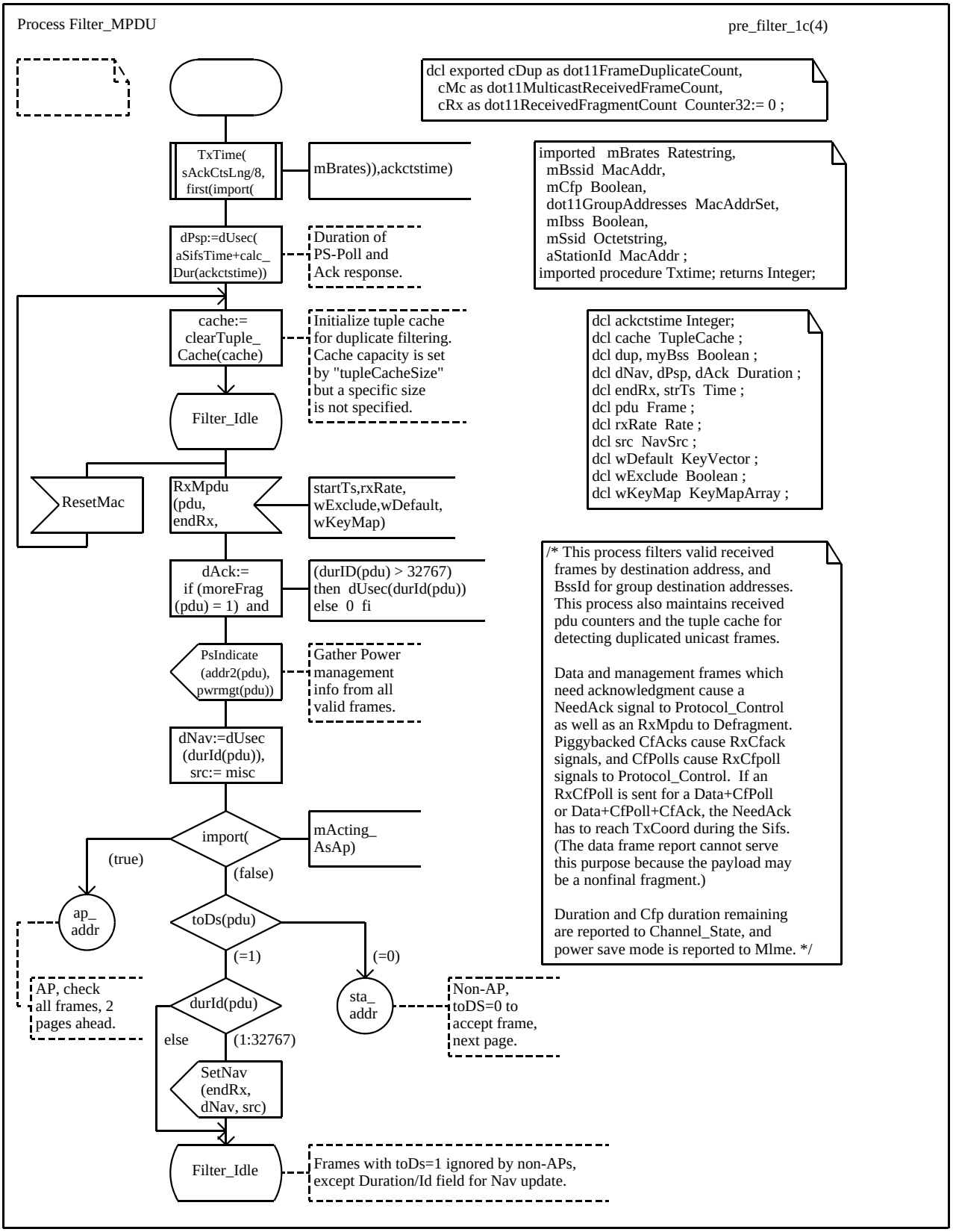




Copyright © 2007 IEEE. All rights reserved.

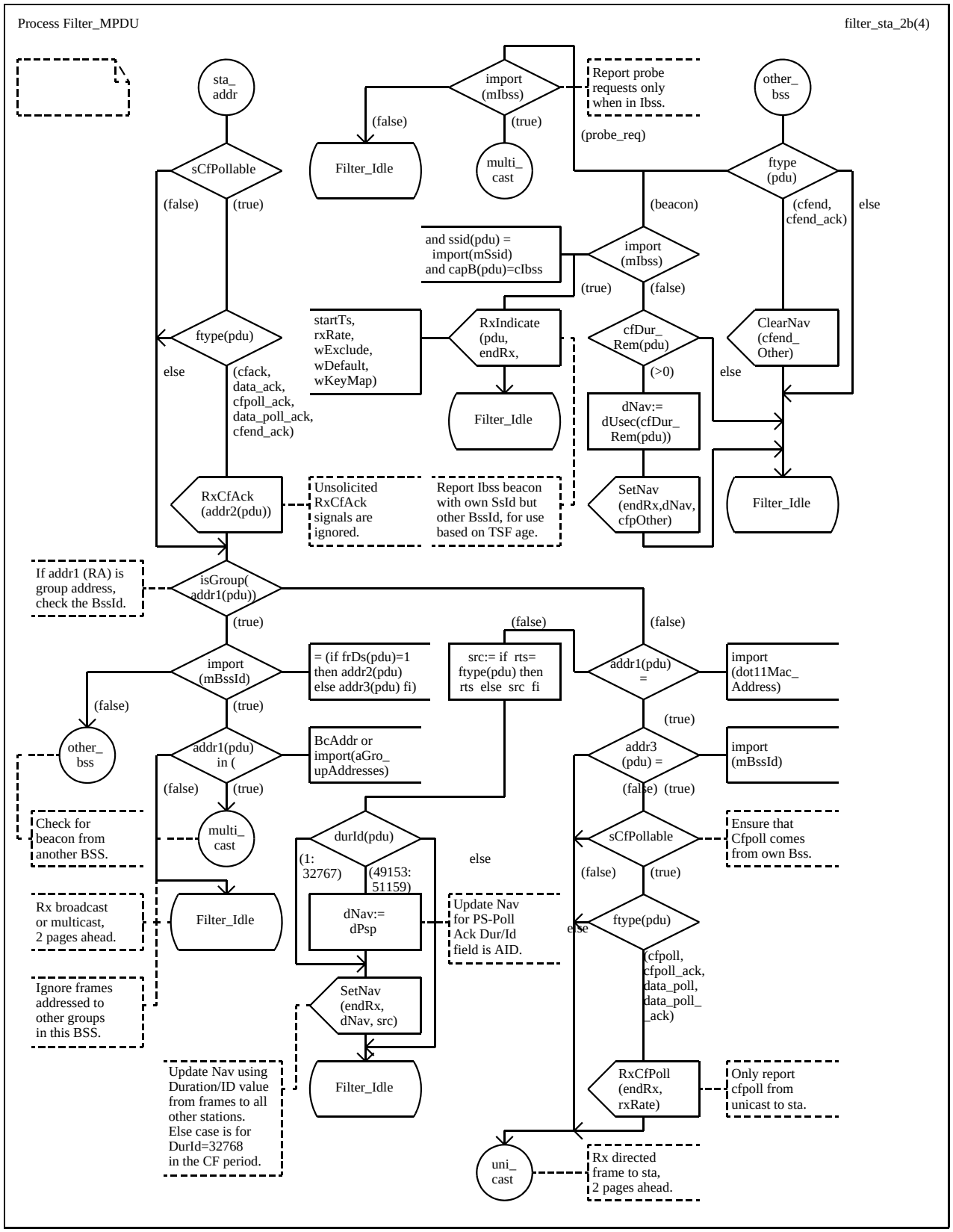



Copyright © 2007 IEEE. All rights reserved.

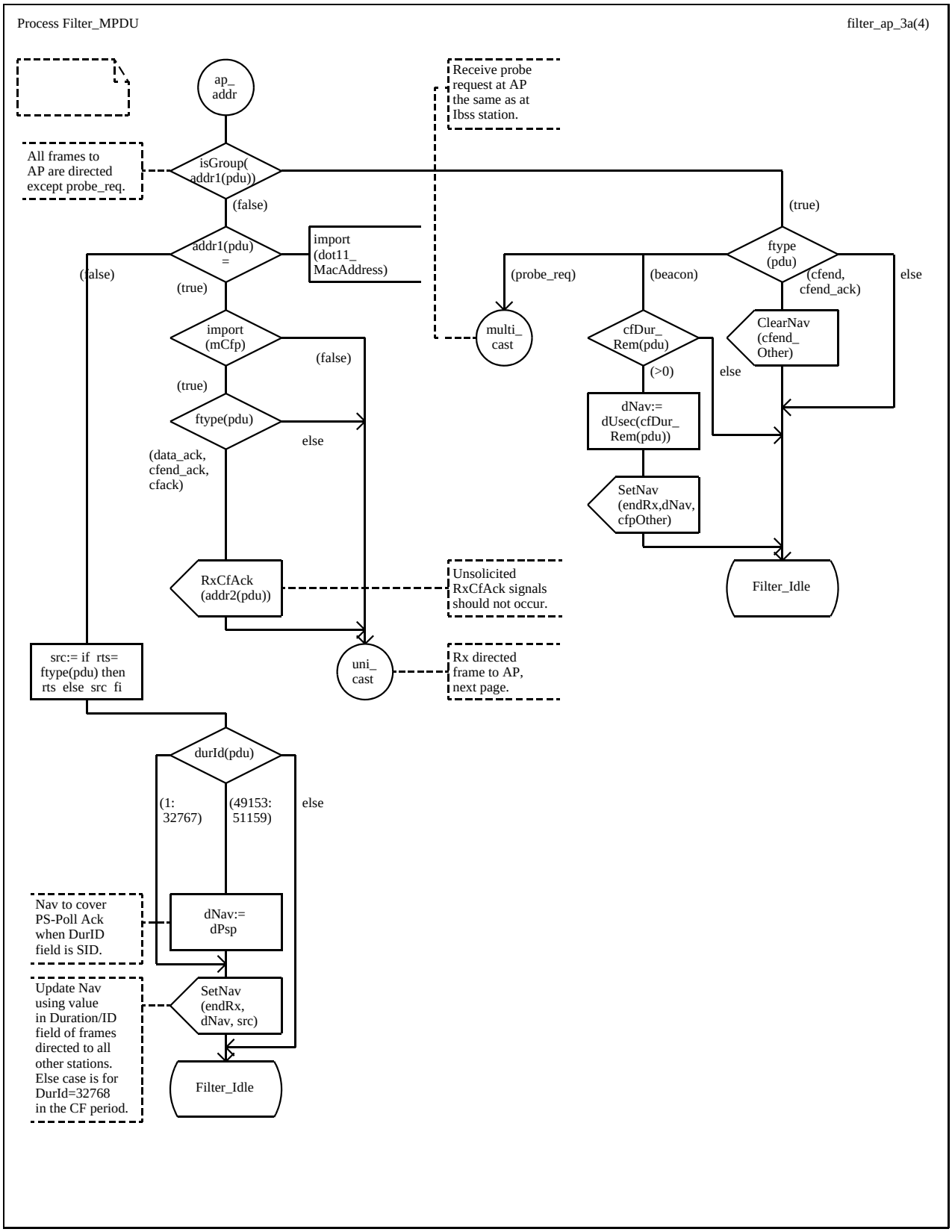




Copyright © 2007 IEEE. All rights reserved.

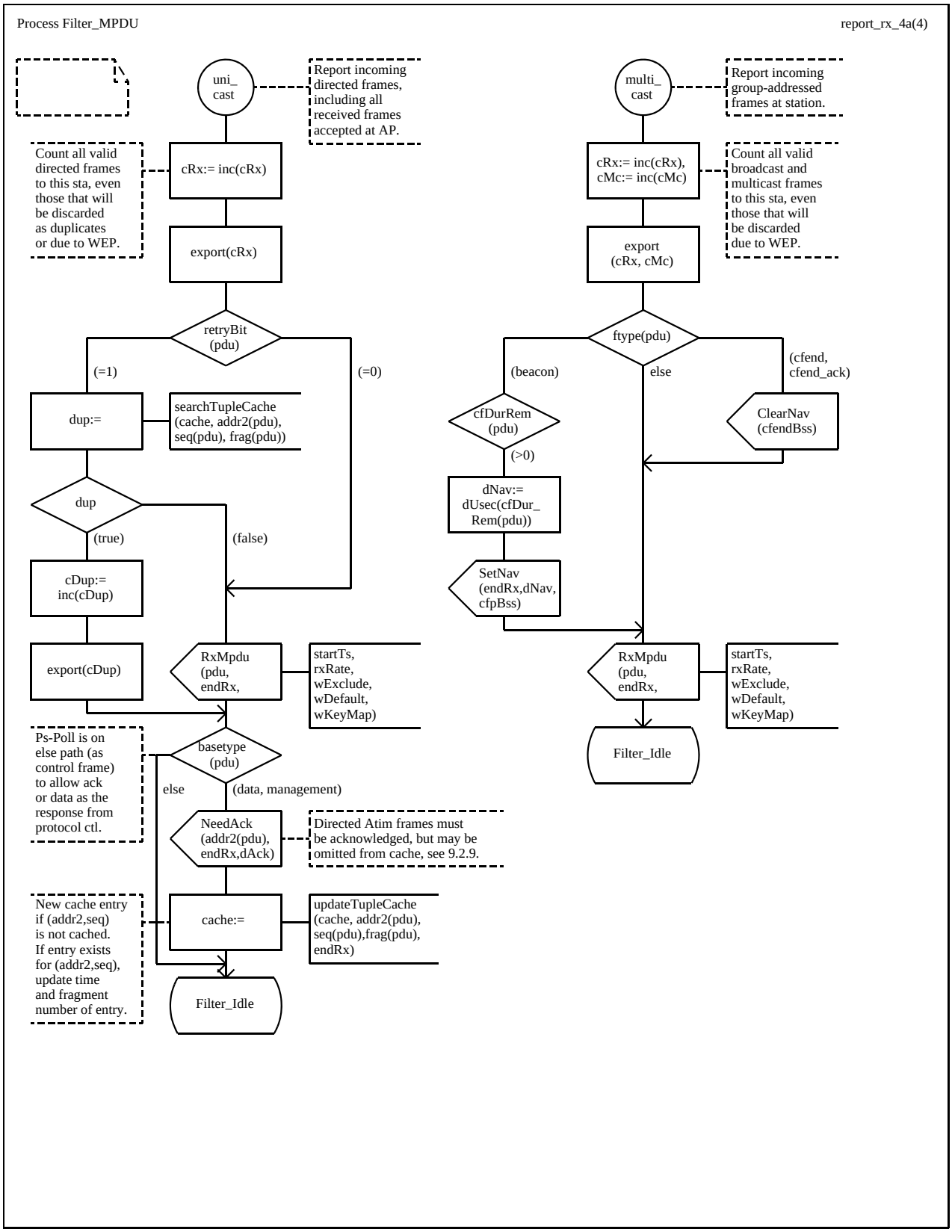



Copyright © 2007 IEEE. All rights reserved.

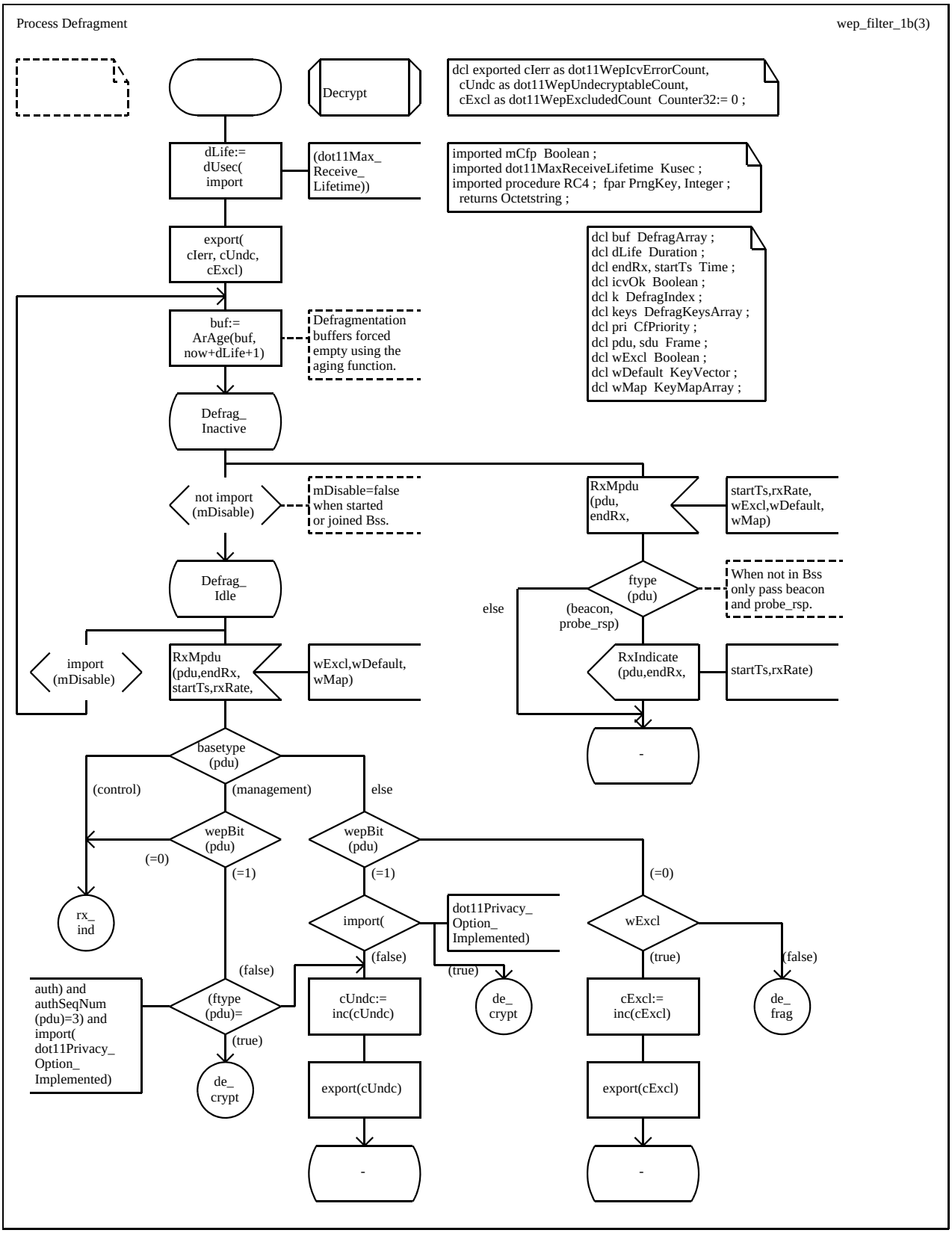




Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

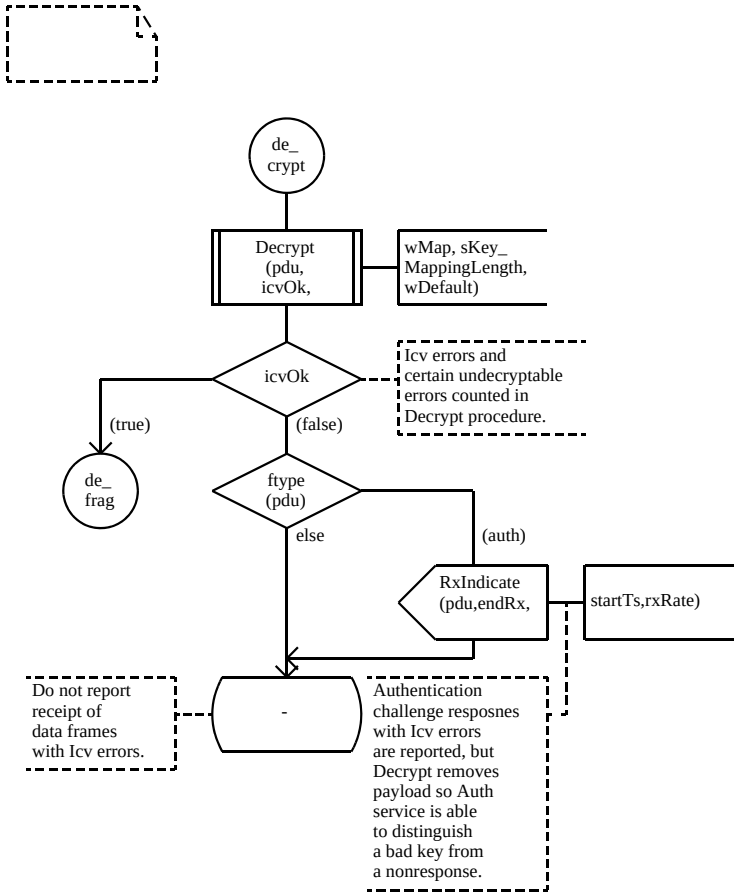




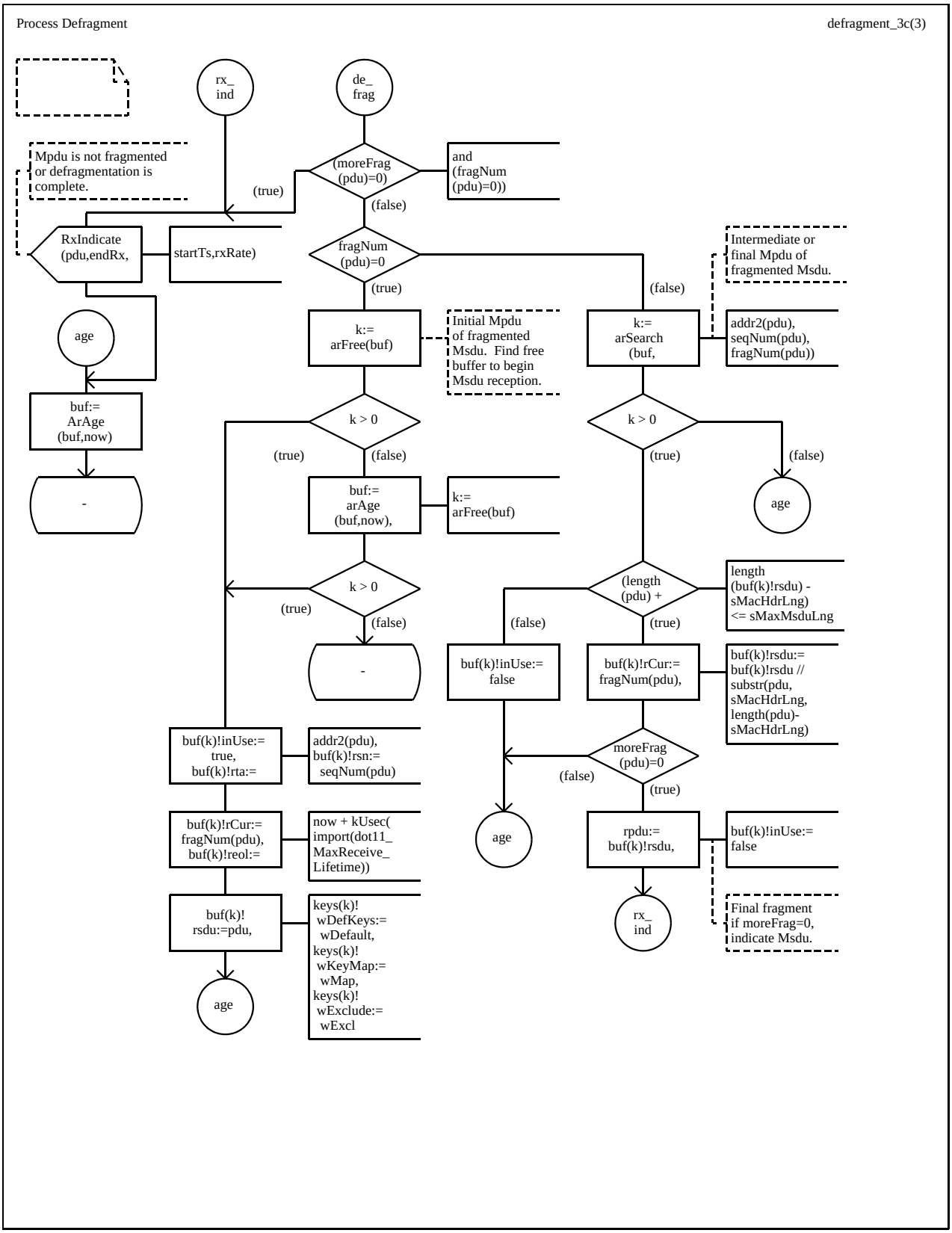
Copyright © 2007 IEEE. All rights reserved.

Process Defragment

wep_decrypt_2b(3)

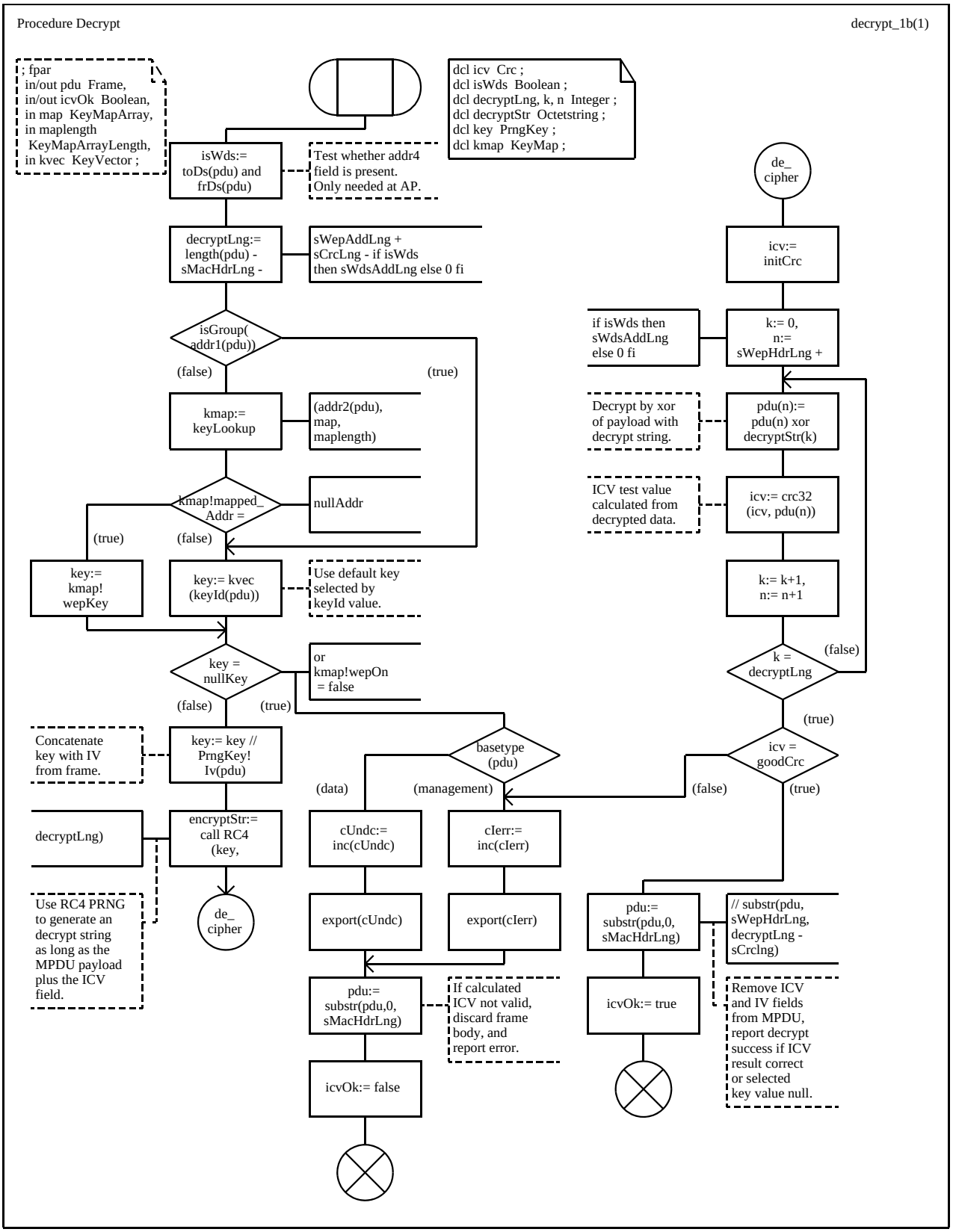



Copyright © 2007 IEEE. All rights reserved.





Copyright © 2007 IEEE. All rights reserved.




Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex D

(normative)

# ASN.1 encoding of the MAC and PHY MIB

```
-- ************************************************************************
-- * IEEE 802.11 MIB
-- ************************************************************************

IEEE802dot11-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        NOTIFICATION-TYPE,Integer32, Counter32,
        Unsigned32                      FROM SNMPv2-SMI

        DisplayString , MacAddress, RowStatus,
        TruthValue                      FROM SNMPv2-TC

        MODULE-COMPLIANCE, OBJECT-GROUP,
        NOTIFICATION-GROUP              FROM SNMPv2-CONF

        ifIndex                         FROM RFC1213-MIB;

-- ************************************************************************
-- *  Tree Definition
-- ************************************************************************

    member-body     OBJECT IDENTIFIER ::= { iso 2 }
    us              OBJECT IDENTIFIER ::= { member-body 840 }

-- ************************************************************************
-- *  MODULE IDENTITY
-- ************************************************************************

ieee802dot11 MODULE-IDENTITY
    LAST-UPDATED "0610200000Z"
    ORGANIZATION "IEEE 802.11"
    CONTACT-INFO
           "WG E-mail: stds-802-11@ieee.org

                Chair: Stuart J. Kerry
                Postal: NXP Semiconductors, Inc.
                        1109 McKay Drive
                        M/S 48A SJ
                        San Jose, CA 95130-1706  USA
                   Tel: +1 408 474 7356
                   Fax: +1 408 474 5343
                E-mail: stuart.kerry@nxp.com

                Editor: Terry L. Cole
                Postal: AMD, M/S PCS-1
                        5900 E. Ben White Blvd.
                        Austin, TX  78741 USA
                   Tel: +1 512 602 2454
                   Fax: +1 512 602 3712
```

Copyright © 2007 IEEE. All rights reserved.

```
                E-mail: terry.cole@amd.com"
    DESCRIPTION
        "The MIB module for IEEE 802.11 entities.
        iso(1).member-body(2).us(840).ieee802dot11(10036)"
    ::= { us 10036 }

-- ********************************************************************
-- *  Major sections
-- ********************************************************************

--  Station ManagemenT (SMT) Attributes
    --  DEFINED AS "The SMT object class provides the necessary support
    --  at the station to manage the processes in the station such that
    --  the station may work cooperatively as a part of an IEEE 802.11
    --  network."

    dot11smt OBJECT IDENTIFIER ::= { ieee802dot11 1 }

       --  dot11smt GROUPS
       --  dot11StationConfigTable                    ::= { dot11smt 1 }
       --  dot11AuthenticationAlgorithmsTable         ::= { dot11smt 2 }
       --  dot11WEPDefaultKeysTable                   ::= { dot11smt 3 }
       --  dot11WEPKeyMappingsTable                   ::= { dot11smt 4 }
       --  dot11PrivacyTable                          ::= { dot11smt 5 }
       --  dot11SMTnotification                       ::= { dot11smt 6 }
       --  dot11MultiDomainCapabilityTable            ::= { dot11smt 7 }
       --  dot11SpectrumManagementTable               ::= { dot11smt 8 }
       --  dot11RSNAConfigTable                       ::= { dot11smt 9 }
       --  dot11RSNAConfigPairwiseCiphersTable        ::= { dot11smt 10 }
       --  dot11RSNAConfigAuthenticationSuitesTable   ::= { dot11smt 11 }
       --  dot11RSNAStatsTable                        ::= { dot11smt 12 }
       --  dot11RegulatoryClassesTable                ::= { dot11smt 13 }

--  MAC Attributes
    --  DEFINED AS "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."

    dot11mac OBJECT IDENTIFIER ::= { ieee802dot11 2 }

      --  MAC GROUPS
      --  reference IEEE Std 802.1f-1993
       --  dot11OperationTable          ::= { dot11mac 1 }
       --  dot11CountersTable           ::= { dot11mac 2 }
       --  dot11GroupAddressesTable     ::= { dot11mac 3 }
       --  dot11EDCATable               ::= { dot11mac 4 }
       --  dot11QAPEDCATable            ::= { dot11mac 5 }
       --  dot11QosCountersTable        ::= { dot11mac 6 }

--  Resource Type ID
    dot11res      OBJECT IDENTIFIER ::= { ieee802dot11 3 }
    dot11resAttribute OBJECT IDENTIFIER ::= { dot11res 1 }

--  PHY Attributes
    --  DEFINED AS "The PHY object class provides the necessary support
    --  for required PHY operational information that may vary from PHY
    --  to PHY and from STA to STA to be communicated to upper layers."
```


Copyright © 2007 IEEE. All rights reserved.

```
    dot11phy OBJECT IDENTIFIER ::= { ieee802dot11 4 }

    --  PHY GROUPS
       --  dot11PhyOperationTable              ::= { dot11phy 1 }
       --  dot11PhyAntennaTable                ::= { dot11phy 2 }
       --  dot11PhyTxPowerTable                ::= { dot11phy 3 }
       --  dot11PhyFHSSTable                   ::= { dot11phy 4 }
       --  dot11PhyDSSSTable                   ::= { dot11phy 5 }
       --  dot11PhyIRTable                     ::= { dot11phy 6 }
       --  dot11RegDomainsSupportedTable       ::= { dot11phy 7 }
       --  dot11AntennasListTable              ::= { dot11phy 8 }
       --  dot11SupportedDataRatesTxTable      ::= { dot11phy 9 }
       --  dot11SupportedDataRatesRxTable      ::= { dot11phy 10 }
       --  dot11PhyOFDMTable                   ::= { dot11phy 11 }
       --  dot11PhyHRDSSSTable                 ::= { dot11phy 12 }
       --  dot11EHCCHoppingPatternTable        ::= { dot11phy 13 }
       --  dot11PhyERPTable                    ::= { dot11phy 14 }

-- ****************************************************************
-- *  Textual conventions from 802 definitions
-- ****************************************************************

    WEPKeytype ::= OCTET STRING (SIZE (5))

-- ****************************************************************
-- *  MIB attribute OBJECT-TYPE definitions follow
-- ****************************************************************

-- ****************************************************************
-- *  dotStationConfig  TABLE
-- ****************************************************************

dot11StationConfigTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11StationConfigEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Station Configuration attributes.  In tablular form to
            allow for multiple instances on an agent."
    ::= { dot11smt 1 }

dot11StationConfigEntry OBJECT-TYPE
        SYNTAX Dot11StationConfigEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11StationConfigTable.  It is
            possible for there to be multiple IEEE 802.11 interfaces
            on one agent, each with its unique MAC address. The
            relationship between an IEEE 802.11 interface and an
            interface in the context of the Internet-standard MIB is
            one-to-one.  As such, the value of an ifIndex object
            instance can be directly used to identify corresponding
            instances of the objects defined herein.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1035 of 1232

```
    ::= { dot11StationConfigTable 1 }

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 }

dot11StationID OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "The purpose of dot11StationID is to allow a manager to
            identify a station for its own purposes.  This attribute
            provides for that eventuality while keeping the true MAC
            address independent.  Its syntax is MAC address, and the
            default value is the station's assigned, unique
            MAC address."
    ::= { dot11StationConfigEntry 1 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11MediumOccupancyLimit OBJECT-TYPE
        SYNTAX INTEGER (0..1000)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall indicate the maximum amount of time,
            in TU, that a point coordinator (PC) may control the usage
            of the wireless medium (WM) without relinquishing control
            for long enough to allow at least one instance of DCF access
            to the medium.  The default value of this attribute shall
            be 100, and the maximum value shall be 1000."
    ::= { dot11StationConfigEntry 2 }

dot11CFPollable OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "When this attribute is TRUE, it shall indicate that
               the STA is able to respond to a CF-Poll with a data frame
               within a SIFS time. This attribute shall be FALSE if
               the STA is not able to respond to a CF-Poll with a data
               frame within a SIFS time."
    ::= { dot11StationConfigEntry 3 }

dot11CFPPeriod OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The attribute shall describe the number of DTIM intervals
            between the start of CFPs.  It is modified by
            MLME-START.request primitive."
    ::= { dot11StationConfigEntry 4 }

dot11CFPMaxDuration OBJECT-TYPE
        SYNTAX INTEGER (0..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
               "The attribute shall describe the maximum duration of
               the CFP in TU that may be generated by the PCF. It is
               modified by MLME-START.request primitive."
    ::= { dot11StationConfigEntry 5 }

dot11AuthenticationResponseTimeOut OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall specify the number of time units (TUs)
             that a responding STA should wait for the next frame in the
            authentication sequence."
    ::= { dot11StationConfigEntry 6 }

dot11PrivacyOptionImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1037 of 1232

```
        STATUS current
        DESCRIPTION
            "This attribute, when TRUE, shall indicate that the IEEE
            802.11 WEP option is implemented.  The default value of
            this attribute shall be FALSE."
    ::= { dot11StationConfigEntry 7 }

dot11PowerManagementMode OBJECT-TYPE
             SYNTAX INTEGER { active(1), powersave(2) }
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the power management
                mode of the STA. When set to active, it shall
                indicate that the station is not in power-save
                (PS) mode. When set to powersave, it shall indicate
                that the station is in power-save mode. The power
                management mode is transmitted in all frames
                according to the rules in  7.1.3.1.7."
       ::= { dot11StationConfigEntry 8 }

dot11DesiredSSID OBJECT-TYPE
             SYNTAX OCTET STRING (SIZE(0..32))
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute reflects the Service Set ID (SSID)
                used in the DesiredSSID parameter of the most recent
                MLME-SCAN.request.  This value may be modified
                by an external management entity and used by the
                local SME to make decisions about the Scanning
                process."
       ::= { dot11StationConfigEntry 9 }

dot11DesiredBSSType OBJECT-TYPE
             SYNTAX INTEGER { infrastructure(1), independent(2), any(3) }
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the type of BSS the
                station shall use when scanning for a BSS with
                which to synchronize. This value is used to filter
                Probe Response frames and Beacon frames. When set to
                infrastructure, the station shall only synchronize
                with a BSS whose Capability Information field has
                the ESS subfield set to 1. When set to independent,
                the station shall only synchronize with a BSS whose
                Capability Information field has the IBSS subfield
                set to 1. When set to any, the station may
                synchronize to either type of BSS."
       ::= { dot11StationConfigEntry 10 }

dot11OperationalRateSet OBJECT-TYPE
             SYNTAX OCTET STRING (SIZE(1..126))
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the set of data
                rates at which the station may transmit data.
```


Copyright © 2007 IEEE. All rights reserved.

```
                Each octet contains a value representing a rate.
                Each rate shall be within the range from 2 to 127,
                corresponding to data rates in increments of
                500 kbit/s from 1 Mb/s to 63.5 Mb/s, and shall
                be supported (as indicated in the supported rates
                table) for receiving data. This value is reported in
                transmitted Beacon, Probe Request, Probe Response,
                Association Request, Association Response,
                Reassociation Request, and Reassociation Response
                frames, and is used to determine whether a BSS
                with which the station desires to synchronize is
                suitable. It is also used when starting a BSS,
                as specified in  10.3."
       ::= { dot11StationConfigEntry 11 }

dot11BeaconPeriod OBJECT-TYPE
             SYNTAX INTEGER (1..65535)
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the number of TUs that
                a station shall use for scheduling Beacon
                transmissions. This value is transmitted in Beacon
                and Probe Response frames."
       ::= { dot11StationConfigEntry 12 }

dot11DTIMPeriod OBJECT-TYPE
             SYNTAX INTEGER(1..255)
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the number of beacon
                intervals that shall elapse between transmission of
                Beacon frames containing a TIM element whose DTIM
                Count field is 0. This value is transmitted in
                the DTIM Period field of Beacon frames."
       ::= { dot11StationConfigEntry 13 }

dot11AssociationResponseTimeOut OBJECT-TYPE
             SYNTAX Unsigned32 (1..4294967295)
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute shall specify the number of TU that a
                requesting STA should wait for a response to a
                transmitted association-request MMPDU."
       ::= { dot11StationConfigEntry 14 }

dot11DisassociateReason OBJECT-TYPE
             SYNTAX INTEGER(0..65535)
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the most recently
                transmitted Reason Code in a Disassociation
                frame.  If no Disassociation frame has been
                transmitted, the value of this attribute shall
                be 0."
             REFERENCE "IEEE Std 802.11-2007, 7.3.1.7"
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1039 of 1232

```
       ::= { dot11StationConfigEntry 15 }

dot11DisassociateStation OBJECT-TYPE
             SYNTAX MacAddress
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the MAC address from the
                Address 1 field of the most recently transmitted
                Disassociation frame.  If no Disassociation
                frame has been transmitted, the value of this
                attribute shall be 0."
       ::= { dot11StationConfigEntry 16 }

dot11DeauthenticateReason OBJECT-TYPE
             SYNTAX INTEGER(0..65535)
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the most recently
                transmitted Reason Code in a Deauthentication
                frame.  If no Deauthentication frame has been
                transmitted, the value of this attribute shall
                be 0."
             REFERENCE "IEEE Std 802.11-2007, 7.3.1.7"
       ::= { dot11StationConfigEntry 17 }

dot11DeauthenticateStation OBJECT-TYPE
             SYNTAX MacAddress
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the MAC address from the
                Address 1 field of the most recently transmitted
                Deauthentication frame.  If no Deauthentication
                frame has been transmitted, the value of this
                attribute shall be 0."
       ::= { dot11StationConfigEntry 18 }

dot11AuthenticateFailStatus OBJECT-TYPE
             SYNTAX INTEGER(0..65535)
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the most recently
                transmitted Status Code in a failed
                Authentication frame.  If no failed
                Authentication frame has been transmitted, the
                value of this attribute shall be 0."
       REFERENCE "IEEE Std 802.11-2007, 7.3.1.9"
       ::= { dot11StationConfigEntry 19 }

dot11AuthenticateFailStation OBJECT-TYPE
             SYNTAX MacAddress
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute holds the MAC address from the
                Address 1 field of the most recently transmitted
```


Copyright © 2007 IEEE. All rights reserved.

```
                failed Authentication frame.  If no failed
                Authentication frame has been transmitted, the
                value of this attribute shall be 0."
       ::= { dot11StationConfigEntry 20 }

dot11MultiDomainCapabilityImplemented OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute, when TRUE, indicates that the
                station implementation is capable of supporting
                multiple regulatory domains.  The capability is
                disabled, otherwise.  The default value of this
                attribute is FALSE."
       ::= { dot11StationConfigEntry 21 }

dot11MultiDomainCapabilityEnabled OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute, when TRUE, indicates that the
                capability of the station to operate in multiple
                regulatory domains is enabled.  The capability is
                disabled, otherwise.  The default value of this
                attribute is FALSE."
       ::= { dot11StationConfigEntry 22 }

dot11CountryString OBJECT-TYPE
             SYNTAX OCTET STRING (SIZE(3))
             MAX-ACCESS read-only
             STATUS current
             DESCRIPTION
                "This attribute identifies the country or non-country
                entity in which the station is operating. If it is a
                country, the first two octets of this
                string is the two character country code as described
                in document ISO/IEC 3166-1.  The third octet shall
                be one of the following:

                1. an ASCII space character, if the regulations under
                which the station is operating encompass all
                environments in the country,

                2. an ASCII 'O' character, if the regulations under
                which the station is operating are for an Outdoor
                environment only, or

                3. an ASCII 'I' character, if the regulations under
                which the station is operating are for an Indoor
                environment only.

                4. an ASCII 'X' character, if the station is operating
                under a non-country entity. The first two octets of the
                non-country entity shall be two ASCII 'XX' characters."
       ::= { dot11StationConfigEntry 23 }

dot11SpectrumManagementImplemented OBJECT-TYPE
```

Copyright © 2007 IEEE. All rights reserved.

```
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                 "This attribute, when TRUE, indicates that the station
                 implementation is capable of supporting spectrum management.
                 The capability is disabled otherwise. The default value of
                 this attribute is FALSE."
       ::= { dot11StationConfigEntry 24 }

dot11SpectrumManagementRequired OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                 "A STA will use the defined TPC and DFS procedures if and
                 only if this attribute is TRUE. The default value of this
                 attribute is FALSE."
       ::= { dot11StationConfigEntry 25 }

dot11RSNAOptionImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                 "This variable indicates whether the entity is RSNA-capable."
       ::= { dot11StationConfigEntry 26 }

dot11RSNAPreauthenticationImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-only
              STATUS current
              DESCRIPTION
                 "This variable indicates whether the entity supports RSNA
                 preauthentication. This cannot be TRUE unless
                 dot11RSNAOptionImplemented is TRUE."
       ::= { dot11StationConfigEntry 27 }

dot11RegulatoryClassesImplemented OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                 "This attribute, when TRUE, indicates that the station implemen-
                 tation is capable of supporting regulatory classes. The capabil-
                 ity is disabled otherwise. The default value of this attribute
                 is FALSE."
       ::= { dot11StationConfigEntry 28 }

dot11RegulatoryClassesRequired OBJECT-TYPE
              SYNTAX TruthValue
              MAX-ACCESS read-write
              STATUS current
              DESCRIPTION
                 "A STA will use the defined regulatory classes procedures if and
                 only if this attribute is TRUE. The default value of this
                 attribute is FALSE."
       ::= { dot11StationConfigEntry 29 }
```


Copyright © 2007 IEEE. All rights reserved.

```
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 STA in the BSS. 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 an AP. When TRUE,
                this attribute indicates that the AP 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
```

Copyright © 2007 IEEE. All rights reserved.

```
              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 STA 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 an AP. This attribute,
                 when TRUE, indicates that the AP implementation is capable
                 of generating and transmitting the BSS 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 an AP. This attribute,
                 when TRUE, indicates that the AP 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 an AP. This attribute,
                 when TRUE, indicates that the AP 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 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11AssociateinNQBSS OBJECT-TYPE
             SYNTAX TruthValue
             MAX-ACCESS read-write
             STATUS current
             DESCRIPTION
                "This attribute, when TRUE, indicates that the station may
                associate in a non-QoS BSS. When FALSE, this attribute
                indicates that the station shall not associate in a non-QoS
                BSS.
                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 AP, when TRUE, indicates
                that non-AP STAs may set up DLS and communicate with other
                non-AP STAs in the BSS. When FALSE, this attribute
                indicates that the stations shall not set up DLS nor
                communicate with other non-AP STAs in the BSS. 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 STAs, when TRUE,
                indicates that STA may set up DLS or accept DLS Requests,
                and communicate with other non-AP STAs in the BSS. When
                FALSE, this attribute indicates that the STA shall not set
                up DLS nor accept DLS, nor communicate with other non-AP
                STAs in the BSS. The default value of this attribute is
                TRUE."
       ::= { dot11StationConfigEntry 42}

-- ************************************************************************
-- *    End of dot11StationConfig  TABLE
-- ************************************************************************

-- ************************************************************************
-- *    dot11AuthenticationAlgorithms  TABLE
-- ************************************************************************

dot11AuthenticationAlgorithmsTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11AuthenticationAlgorithmsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "This (conceptual) table of attributes shall be a set of
            all the authentication algorithms supported by the
            stations.  The following are the default values and the
            associated algorithm:
                Value = 1: Open System
                Value = 2: Shared Key"
        REFERENCE "IEEE Std 802.11-2007, 7.3.1.1"
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1045 of 1232

```
    ::= { dot11smt 2 }

dot11AuthenticationAlgorithmsEntry OBJECT-TYPE
        SYNTAX Dot11AuthenticationAlgorithmsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the Authentication
            Algorithms Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11AuthenticationAlgorithmsIndex }
    ::= { dot11AuthenticationAlgorithmsTable  1 }

Dot11AuthenticationAlgorithmsEntry ::=
         SEQUENCE { dot11AuthenticationAlgorithmsIndex      Integer32,
                    dot11AuthenticationAlgorithm            INTEGER,
                    dot11AuthenticationAlgorithmsEnable     TruthValue }

dot11AuthenticationAlgorithmsIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The auxiliary variable used to identify instances
            of the columnar objects in the Authentication Algorithms Table."
    ::= { dot11AuthenticationAlgorithmsEntry 1 }

dot11AuthenticationAlgorithm OBJECT-TYPE
        SYNTAX INTEGER { openSystem(1), sharedKey(2) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This attribute shall be a set of all the authentication
            algorithms supported by the STAs. The following are the
            default values and the associated algorithm.
                      Value = 1: Open System
                      Value = 2: Shared Key"
    ::= { dot11AuthenticationAlgorithmsEntry 2 }

dot11AuthenticationAlgorithmsEnable  OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "This attribute, when TRUE at a station, shall enable the acceptance
       of the authentication algorithm described in the corresponding table
       entry in authentication frames received by the station that have odd
       authentication sequence numbers.  The default value of this attribute
       shall be true when the value of dot11AuthenticationAlgorithm is
       Open System and false otherwise."
    ::= { dot11AuthenticationAlgorithmsEntry 3 }

-- ************************************************************************
-- *    End of dot11AuthenticationAlgorithms  TABLE
-- ************************************************************************
```


Copyright © 2007 IEEE. All rights reserved.

```
-- ************************************************************************
-- *    dot11WEPDefaultKeys  TABLE
-- ************************************************************************

dot11WEPDefaultKeysTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11WEPDefaultKeysEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Conceptual table for WEP default keys.  This table shall
            contain the four WEP default secret key values
            corresponding to the four possible KeyID values.  The WEP
            default secret keys are logically WRITE-ONLY.  Attempts to
            read the entries in this table shall return unsuccessful
            status and values of null or zero.  The default value of
            each WEP default key shall be null."
        REFERENCE "IEEE Std 802.11-2007, 8.3.2"
    ::= { dot11smt 3 }

dot11WEPDefaultKeysEntry OBJECT-TYPE
        SYNTAX Dot11WEPDefaultKeysEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the WEP Default Keys Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
       INDEX { ifIndex,
               dot11WEPDefaultKeyIndex }
    ::= { dot11WEPDefaultKeysTable  1 }

Dot11WEPDefaultKeysEntry ::=
         SEQUENCE { dot11WEPDefaultKeyIndex     INTEGER,
                    dot11WEPDefaultKeyValue     WEPKeytype }

dot11WEPDefaultKeyIndex 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 WEP Default Keys Table.
            The value of this variable is equal to the WEPDefaultKeyID + 1"
    ::= { dot11WEPDefaultKeysEntry 1 }

dot11WEPDefaultKeyValue OBJECT-TYPE
        SYNTAX WEPKeytype
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "A WEP default secret key value."
    ::= { dot11WEPDefaultKeysEntry 2 }

-- ************************************************************************
-- *    End of dot11WEPDefaultKeys  TABLE
-- ************************************************************************
```

Copyright © 2007 IEEE. All rights reserved.

```
-- ************************************************************************
-- *    dot11WEPKeyMappings  TABLE
-- ************************************************************************

dot11WEPKeyMappingsTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11WEPKeyMappingsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Conceptual table for WEP Key Mappings.  The MIB supports
            the ability to share a separate WEP key for each RA/TA
            pair.  The Key Mappings Table contains zero or one entry
            for each MAC address and contains two fields for each
            entry: WEPOn and the corresponding WEP key.  The WEP key
            mappings are logically WRITE-ONLY.  Attempts to read the
            entries in this table shall return unsuccessful status and
            values of null or zero.  The default value for all WEPOn
            fields is FALSE."
        REFERENCE "IEEE Std 802.11-2007, 8.3.2"
    ::= { dot11smt 4 }

dot11WEPKeyMappingsEntry OBJECT-TYPE
        SYNTAX Dot11WEPKeyMappingsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the WEP Key Mappings Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11WEPKeyMappingIndex }
    ::= { dot11WEPKeyMappingsTable  1 }

Dot11WEPKeyMappingsEntry ::=
       SEQUENCE { dot11WEPKeyMappingIndex        Integer32,
                  dot11WEPKeyMappingAddress      MacAddress,
                  dot11WEPKeyMappingWEPOn        TruthValue,
                  dot11WEPKeyMappingValue        WEPKeytype,
                  dot11WEPKeyMappingStatus       RowStatus }

dot11WEPKeyMappingIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The auxiliary variable used to identify instances
            of the columnar objects in the WEP Key Mappings Table."
    ::= { dot11WEPKeyMappingsEntry 1 }

dot11WEPKeyMappingAddress OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "The MAC address of the STA for which the values from this
            key mapping entry are to be used."
```


Copyright © 2007 IEEE. All rights reserved.

```
    ::= { dot11WEPKeyMappingsEntry 2 }

dot11WEPKeyMappingWEPOn OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "Boolean as to whether WEP is to be used when communicating
            with the dot11WEPKeyMappingAddress STA."
    ::= { dot11WEPKeyMappingsEntry 3 }

dot11WEPKeyMappingValue OBJECT-TYPE
        SYNTAX WEPKeytype
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "A WEP secret key value."
    ::= { dot11WEPKeyMappingsEntry 4 }

dot11WEPKeyMappingStatus OBJECT-TYPE
        SYNTAX RowStatus
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "The status column used for creating, modifying, and
            deleting instances of the columnar objects in the WEP key
            mapping Table."
        DEFVAL { active }
    ::= { dot11WEPKeyMappingsEntry 5 }

-- ****************************************************************
-- *    End of dot11WEPKeyMappings  TABLE
-- ****************************************************************

-- ****************************************************************
-- *    dot11Privacy TABLE
-- ****************************************************************

dot11PrivacyTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PrivacyEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group containing attributes concerned with IEEE 802.11
            Privacy.  Created as a table to allow multiple
            instantiations on an agent."
    ::= { dot11smt 5 }

dot11PrivacyEntry OBJECT-TYPE
        SYNTAX Dot11PrivacyEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PrivacyTable Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1049 of 1232

```
    ::= { dot11PrivacyTable 1 }

Dot11PrivacyEntry ::=
        SEQUENCE { dot11PrivacyInvoked               TruthValue,
                   dot11WEPDefaultKeyID              INTEGER,
                   dot11WEPKeyMappingLength          Unsigned32,
                   dot11ExcludeUnencrypted           TruthValue,
                   dot11WEPICVErrorCount             Counter32,
                   dot11WEPExcludedCount             Counter32,
                   dot11RSNAEnabled                  TruthValue,
                   dot11RSNAPreauthenticationEnabled TruthValue }

dot11PrivacyInvoked OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "When this attribute is TRUE, it shall indicate that some level of security
             is invoked for transmitting data frames. For WEP-only clients, the
             security mechanism used is WEP.

       For RSNA-capable clients, an additional variable dot11RSNAEnabled indicates
             whether RSNA is enabled. If dot11RSNAEnabled is FALSE or the MIB
             variable does not exist, the security mechanism invoked is WEP; if
             dot11RSNAEnabled is TRUE, RSNA security mechanisms invoked are con-
             figured in the dot11RSNAConfigTable. The default value of this at-
             tribute shall be FALSE."
    ::= { dot11PrivacyEntry 1 }

dot11WEPDefaultKeyID  OBJECT-TYPE
        SYNTAX INTEGER (0..3)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This attribute shall indicate the use of the first,
            second, third, or fourth element of the WEPDefaultKeys
            array when set to values of zero, one, two, or three.  The
            default value of this attribute shall be 0."
        REFERENCE "IEEE Std 802.11-2007, 8.3.2"
    ::= { dot11PrivacyEntry 2 }

dot11WEPKeyMappingLength  OBJECT-TYPE
        SYNTAX Unsigned32 (10..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The maximum number of tuples that dot11WEPKeyMappings can hold."
        REFERENCE "IEEE Std 802.11-2007, 8.3.2"
    ::= { dot11PrivacyEntry 3 }

dot11ExcludeUnencrypted  OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "When this attribute is TRUE, the STA shall not indicate at the MAC service
             interface received MSDUs that have the Protected Frame subfield of
             the Frame Control field equal to zero. When this attribute is FALSE,
             the STA may accept MSDUs that have the Protected Frame subfield of
```


Copyright © 2007 IEEE. All rights reserved.

```
              the Frame Control field equal to zero. The default value of this at-
              tribute shall be FALSE."
    ::= { dot11PrivacyEntry 4 }

dot11WEPICVErrorCount  OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
       "This counter shall increment when a frame is received with the Protected
              Frame subfield of the Frame Control field set to one and the value
              of the ICV as received in the frame does not match the ICV value that
              is calculated for the contents of the received frame. ICV errors for
              TKIP are not counted in this variable but in
              dot11RSNAStatsTKIPICVErrors."
    ::= { dot11PrivacyEntry 5 }

dot11WEPExcludedCount  OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
       "This counter shall increment when a frame is received with the Protected
              Frame subfield of the Frame Control field set to zero and the value
              of dot11ExcludeUnencrypted causes that frame to be discarded."
    ::= { dot11PrivacyEntry 6 }

dot11RSNAEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "When this object is set to TRUE, this shall indicate that RSNA is
              enabled on this entity. The entity will advertise the RSN Informa-
              tion Element in its Beacon and Probe Response frames. Configuration
              variables for RSNA operation are found in the dot11RSNAConfigTable.
              This object requires that dot11PrivacyInvoked also be set to TRUE."
   ::= { dot11PrivacyEntry 7 }

dot11RSNAPreauthenticationEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "When this object is set to TRUE, this shall indicate that RSNA
              preauthentication is enabled on this entity.

              This object requires that dot11RSNAEnabled also be set to TRUE."
   ::= { dot11PrivacyEntry 8 }

-- ****************************************************************
-- *    End of dot11Privacy  TABLE
-- ****************************************************************

-- ****************************************************************
-- *    dot11SMTnotification Objects
-- ****************************************************************

dot11SMTnotification OBJECT IDENTIFIER ::= { dot11smt 6 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1051 of 1232

```
dot11Disassociate NOTIFICATION-TYPE
        OBJECTS { ifIndex, dot11DisassociateReason, dot11DisassociateStation }
        STATUS current
        DESCRIPTION
              "The disassociate notification shall be sent when the STA
              sends a Disassociation frame. The value of the notification
              shall include the MAC address of the MAC to which the Disassociation
              frame was sent and the reason for the disassociation.

              ifIndex - Each IEEE 802.11 interface is represented by an
              ifEntry.  Interface tables in this MIB module are indexed
              by ifIndex."
    ::= { dot11SMTnotification 0 1 }

dot11Deauthenticate NOTIFICATION-TYPE
       OBJECTS { ifIndex, dot11DeauthenticateReason,
              dot11DeauthenticateStation }
       STATUS current
       DESCRIPTION
              "The deauthenticate notification shall be sent when the STA
              sends a Deauthentication frame. The value of the notification
              shall include the MAC address of the MAC to which the
              Deauthentication frame was sent and the reason for the
              deauthentication.

              ifIndex - Each IEEE 802.11 interface is represented by an
              ifEntry.  Interface tables in this MIB module are indexed
              by ifIndex."
    ::= { dot11SMTnotification 0 2 }

dot11AuthenticateFail NOTIFICATION-TYPE
        OBJECTS { ifIndex, dot11AuthenticateFailStatus,
              dot11AuthenticateFailStation }
        STATUS current
        DESCRIPTION
              "The authenticate failure notification shall be sent when the STA
              sends an Authentication frame with a status code other than
              'successful'. The value of the notification
              shall include the MAC address of the MAC to which the Authentication
              frame was sent and the reason for the authentication failure.

              ifIndex - Each IEEE 802.11 interface is represented by an
              ifEntry.  Interface tables in this MIB module are indexed
              by ifIndex."
    ::= { dot11SMTnotification 0 3 }

-- ************************************************************************
-- *    End of dot11SMTnotification Objects
-- ************************************************************************

-- ************************************************************************
-- * dot11MultiDomainCapability TABLE
-- ************************************************************************

dot11MultiDomainCapabilityTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11MultiDomainCapabilityEntry
       MAX-ACCESS not-accessible
       STATUS current
```


Copyright © 2007 IEEE. All rights reserved.

```
       DESCRIPTION
              "This (conceptual) table of attributes for
              cross-domain mobility."
   ::= { dot11smt 7 }

dot11MultiDomainCapabilityEntry OBJECT-TYPE
       SYNTAX Dot11MultiDomainCapabilityEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry (conceptual row) in the Multiple Domain
              Capability Table.

              IfIndex - Each IEEE 802.11 interface is represented
              by an ifEntry. Interface tables in this MIB are
              indexed by ifIndex."
       INDEX { ifIndex,
               dot11MultiDomainCapabilityIndex }
   ::= { dot11MultiDomainCapabilityTable 1 }

Dot11MultiDomainCapabilityEntry ::=
      SEQUENCE { dot11MultiDomainCapabilityIndex  Integer32,
                 dot11FirstChannelNumber          Integer32,
                 dot11NumberofChannels            Integer32,
                 dot11MaximumTransmitPowerLevel   Integer32 }

dot11MultiDomainCapabilityIndex OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The auxiliary variable used to identify instances of
              the columnar objects in the Multi Domain Capability Table."
   ::= { dot11MultiDomainCapabilityEntry 1 }

dot11FirstChannelNumber OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the value of the lowest
              channel number in the subband for the associated domain
              country string. The default value of this attribute
              shall be zero."
   ::= { dot11MultiDomainCapabilityEntry 2 }

dot11NumberofChannels OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the value of the total
              number of channels allowed in the subband for the
              associated domain country string. The default value of
              this attribute shall be zero."
   ::= { dot11MultiDomainCapabilityEntry 3 }

dot11MaximumTransmitPowerLevel OBJECT-TYPE
       SYNTAX Integer32
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1053 of 1232

```
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the maximum transmit power,
              in dBm, allowed in the subband for the associated domain
              country string. The default value of this attribute shall
              be zero."
   ::= { dot11MultiDomainCapabilityEntry 4 }

-- ****************************************************************
-- * End of dot11MultiDomainCapability TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11SpectrumManagement TABLE
-- ****************************************************************

dot11SpectrumManagementTable OBJECT-TYPE
     SYNTAX SEQUENCE OF Dot11SpectrumManagementEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "(Conceptual) table of attributes for spectrum management"
  ::= {dot11smt 8}

dot11SpectrumManagementEntry OBJECT-TYPE
     SYNTAX Dot11SpectrumManagementEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "An entry (conceptual row) in the Spectrum Management Table.

          IfIndex - Each 802.11 interface is represented by an ifEntry.
          Interface tables in this MIB are indexed by ifIndex."
     INDEX {ifIndex, dot11SpectrumManagementIndex}
  ::= { dot11SpectrumManagementTable 1 }

Dot11SpectrumManagementEntry ::=
     SEQUENCE {
          dot11SpectrumManagementIndex        Integer32,
          dot11MitigationRequirement          Integer32,
          dot11ChannelSwitchTime              Integer32,
          dot11PowerCapabilityMax             Integer32,
          dot11PowerCapabilityMin             Integer32 }

dot11SpectrumManagementIndex OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          "The auxiliary variable used to identify instances of the columnar
          objects in the Spectrum Management Table."
  ::= { dot11SpectrumManagementEntry 1 }

dot11MitigationRequirement OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS read-write
     STATUS current
     DESCRIPTION
```


Copyright © 2007 IEEE. All rights reserved.

```
          "This attribute shall indicate the mitigation requirement in dB
          required. The default value of this attribute shall be 3 dB."
   ::= { dot11SpectrumManagementEntry 2 }

dot11ChannelSwitchTime OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS read-write
     STATUS current
     DESCRIPTION
          "This attribute shall indicate assumed channel switch time, measured
          in TUs. The unit of this attribute is TUs. The default value of this
          attribute shall be 2 TUs. The minimum value shall be 1 TU."
   ::= { dot11SpectrumManagementEntry 3 }

dot11PowerCapabilityMax OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "This attribute shall indicate the maximum transmit Power Capability
          of this station. The unit of this attribute is dBm. The default value
          of this attribute shall be 0 dBm."
   ::= { dot11SpectrumManagementEntry 4 }

dot11PowerCapabilityMin OBJECT-TYPE
     SYNTAX Integer32
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
          "This attribute shall indicate the minimum transmit Power Capability
          of this station. The unit of this attribute is dBm. The default value
          of this attribute shall be -100 dBm."
   ::= { dot11SpectrumManagementEntry 5 }

-- ****************************************************************
-- * End of dot11SpectrumManagement TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11RegulatoryClasses TABLE
-- ****************************************************************

dot11RegulatoryClassesTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11RegulatoryClassesEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
       "(Conceptual) table of attributes for regulatory classes"
   ::= {dot11smt 13}

dot11RegulatoryClassesEntry OBJECT-TYPE
       SYNTAX Dot11RegulatoryClassesEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
       "An entry (conceptual row) in the Regulatory Classes Table.

       IfIndex - Each 802.11 interface is represented by an ifEntry. Interface
                 tables in this MIB are indexed by ifIndex."
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1055 of 1232

```
       INDEX {ifIndex, dot11RegulatoryClassesIndex}
   ::= { dot11RegulatoryClassesTable 1 }

Dot11RegulatoryClassesEntry ::=
       SEQUENCE {
       dot11RegulatoryClassesIndex      Integer32,
       dot11RegulatoryClass             INTEGER,
       dot11CoverageClass               INTEGER }

dot11RegulatoryClassesIndex OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The auxiliary variable used to identify instances of the columnar
              objects in the Regulatory Classes Table."
   ::= { dot11RegulatoryClassesEntry 1 }

dot11RegulatoryClass OBJECT-TYPE
       SYNTAX INTEGER
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the regulatory class to be used. The
              default value of this attribute shall be zero."
   ::= { dot11RegulatoryClassesEntry 2 }

dot11CoverageClass OBJECT-TYPE
       SYNTAX INTEGER
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the coverage class to be used. The
              default value of this attribute shall be zero."
   ::= { dot11RegulatoryClassesEntry 3 }

-- ****************************************************************
-- * End of dot11RegulatoryClasses TABLE
-- ****************************************************************

-- ****************************************************************
-- * MAC Attribute Templates
-- ****************************************************************

-- ****************************************************************
-- * dot11Operation TABLE
-- ****************************************************************

dot11OperationTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11OperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group contains MAC attributes pertaining to the operation
            of the MAC.  This has been implemented as a table in order
            to allow for multiple instantiations on an agent."
    ::= { dot11mac 1 }

dot11OperationEntry OBJECT-TYPE
```


Copyright © 2007 IEEE. All rights reserved.

```
        SYNTAX Dot11OperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11OperationEntry Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11OperationTable 1 }

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,
                   dot11ChannelUtilizationBeaconInterval    INTEGER,
                   dot11ScheduleTimeout                     INTEGER,
                   dot11DLSResponseTimeout                  INTEGER,
                   dot11QAPMissingAckRetryLimit             INTEGER,
                   dot11EDCAAveragingPeriod                 INTEGER }

dot11MACAddress OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "Unique MAC Address assigned to the STA."
    ::= { dot11OperationEntry 1 }

dot11RTSThreshold OBJECT-TYPE
        SYNTAX INTEGER (0..3000)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall indicate the number of octets in an MPDU,
              below which an RTS/CTS handshake shall not be performed, except as
              RTS/CTS is used as a cross modulation protection mechanism as
              defined in 9.13. An RTS/CTS handshake shall be performed at the
              beginning of any frame exchange sequence where the MPDU is of type
              Data or Management, the MPDU has an individual address in the
              Address1 field, and the length of the MPDU is greater than this
              threshold. Setting this attribute to be larger than the maximum
              MSDU size shall have the effect of turning off the RTS/CTS
              handshake for frames of Data or Management type transmitted by
              this STA. Setting this attribute to zero shall have the effect
              of turning on the RTS/CTS handshake for all frames of Data or
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1057 of 1232

```
                Management type transmitted by this STA. The default value of
                this attribute shall be 3000."
    ::= { dot11OperationEntry 2 }

dot11ShortRetryLimit OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "This attribute shall indicate the maximum number of
                transmission attempts of a frame, the length of which is less
                than or equal to dot11RTSThreshold, that shall be made before a
                failure condition is indicated. The default value of this
                attribute shall be 7."
    ::= { dot11OperationEntry 3 }

dot11LongRetryLimit OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "This attribute shall indicate the maximum number of
                transmission attempts of a frame, the length of which is
                greater than dot11RTSThreshold, that shall be made before a
                failure condition is indicated. The default value of this
                attribute shall be 4."
    ::= { dot11OperationEntry 4 }

dot11FragmentationThreshold OBJECT-TYPE
        SYNTAX INTEGER (256..3000)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
              "This attribute shall specify the current maximum size, in
              octets, of the MPDU that may be delivered to the PHY. An MSDU
              shall be broken into fragments if its size exceeds the value
              of this attribute after adding MAC headers and trailers. An
              MSDU or MMPDU shall be fragmented when the resulting frame has
              an individual address in the Address1 field, and the length
              of the frame is larger than this threshold. The default value
              for this attribute shall be the lesser of 3000 or the
              aMPDUMaxLength of the attached PHY and shall never exceed the
              lesser of 3000 or the aMPDUMaxLength of the attached PHY.
              The value of this attribute shall never be less than 256."
    ::= { dot11OperationEntry 5 }

dot11MaxTransmitMSDULifetime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "The MaxTransmitMSDULifetime shall be the elapsed time in TU,
                after the initial transmission of an MSDU, after which further
                attempts to transmit the MSDU shall be terminated. The default
                value of this attribute shall be 512."
    ::= { dot11OperationEntry 6 }

dot11MaxReceiveLifetime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
```


Copyright © 2007 IEEE. All rights reserved.

```
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The MaxReceiveLifetime shall be the elapsed time in TU,
            after the initial reception of a fragmented MMPDU or MSDU,
            after which further attempts to reassemble the MMPDU or
            MSDU shall be terminated. The default value shall be
            512."
    ::= { dot11OperationEntry 7 }

dot11ManufacturerID OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The ManufacturerID shall include, at a minimum, the name
            of the manufacturer.  It may include additional
            information at the manufacturer's discretion.  The default
            value of this attribute shall be null."
    ::= { dot11OperationEntry 8 }

dot11ProductID OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The ProductID shall include, at a minimum, an identifier
            that is unique to the manufacturer.  It may include
            additional information at the manufacturer's discretion.
            The default value of this attribute shall be null."
    ::= { 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
```

Copyright © 2007 IEEE. All rights reserved.

```
              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
```


Copyright © 2007 IEEE. All rights reserved.

```
        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 AP 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 AP 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 }

-- ****************************************************************
-- *    End of dot11Operation TABLE
-- ****************************************************************

-- ****************************************************************
-- *    dot11Counters TABLE
-- ****************************************************************

dot11CountersTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11CountersEntry
        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 2 }

dot11CountersEntry OBJECT-TYPE
        SYNTAX Dot11CountersEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11CountersEntry Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11CountersTable 1 }
```

Copyright © 2007 IEEE. All rights reserved.

```
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 }

dot11TransmittedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall be incremented for an acknowledged MPDU
            with an individual address in the address 1 field or an MPDU
            with a multicast address in the address 1 field of type Data
            or Management."
    ::= { dot11CountersEntry 1 }

dot11MulticastTransmittedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment only when the multicast bit
            is set in the destination MAC address of a successfully
            transmitted MSDU.  When operating as a STA in an ESS, where
            these frames are directed to the AP, this implies having
            received an acknowledgment to all associated MPDUs."
    ::= { dot11CountersEntry 2 }

dot11FailedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "This counter shall increment when an MSDU is not transmitted
              successfully due to the number of transmit attempts exceeding
              either the  dot11ShortRetryLimit or dot11LongRetryLimit."
    ::= { dot11CountersEntry 3 }

dot11RetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
```


Copyright © 2007 IEEE. All rights reserved.

```
        DESCRIPTION
            "This counter shall increment when an MSDU is successfully
            transmitted after one or more retransmissions."
    ::= { dot11CountersEntry 4 }

dot11MultipleRetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment when an MSDU is successfully
            transmitted after more than one retransmission."
    ::= { dot11CountersEntry 5 }

dot11FrameDuplicateCount  OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment when a frame is received
            that the Sequence Control field indicates is a
            duplicate."
    ::= { dot11CountersEntry 6 }

dot11RTSSuccessCount 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."
    ::= {  dot11CountersEntry 7 }

dot11RTSFailureCount 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."
    ::= { dot11CountersEntry 8 }

dot11ACKFailureCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment when an ACK is not received
            when expected."
    ::= {  dot11CountersEntry 9 }

dot11ReceivedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall be incremented for each successfully
            received MPDU of type Data or Management."
    ::= { dot11CountersEntry 10 }
```

Copyright © 2007 IEEE. All rights reserved.

```
dot11MulticastReceivedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment when a MSDU is received
            with the multicast bit set in the destination
            MAC address."
    ::= { dot11CountersEntry 11 }

dot11FCSErrorCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "This counter shall increment when an FCS error is
            detected in a received MPDU."
    ::= { dot11CountersEntry 12 }

dot11TransmittedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "This counter shall increment for each successfully transmitted
             MSDU."
    ::= { dot11CountersEntry 13 }

dot11WEPUndecryptableCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "This counter shall increment when a frame is received with the
             Protected Frame subfield of the Frame Control field set to one and
             the WEPOn value for the key mapped to the transmitter's MAC address
             indicates that the frame should not have been encrypted or that frame
             is discarded due to the receiving STA not implementing the privacy
             option."
    ::= { 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 AP, shall increment when a
             station associates or reassociates. This counter shall decrement
             when a station disassociates."
```


Copyright © 2007 IEEE. All rights reserved.

```
    ::= { 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 }

-- ****************************************************************
-- * End of dot11Counters TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11GroupAddresses  TABLE
-- ****************************************************************

dot11GroupAddressesTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11GroupAddressesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "A conceptual table containing a set of MAC addresses
            identifying the multicast-group addresses for which this STA
            will receive frames.  The default value of this attribute
            shall be null."
    ::= { dot11mac 3 }

dot11GroupAddressesEntry OBJECT-TYPE
        SYNTAX Dot11GroupAddressesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the Group Addresses Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
```

Copyright © 2007 IEEE. All rights reserved.

```
        INDEX { ifIndex,
                dot11GroupAddressesIndex }
    ::= { dot11GroupAddressesTable  1 }

Dot11GroupAddressesEntry ::=
        SEQUENCE { dot11GroupAddressesIndex    Integer32,
                   dot11Address                MacAddress,
                   dot11GroupAddressesStatus   RowStatus }

dot11GroupAddressesIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The auxiliary variable used to identify instances
            of the columnar objects in the Group Addresses Table."
    ::= { dot11GroupAddressesEntry 1 }

dot11Address OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "MAC address identifying multicast-group addresses
            from which this STA will receive frames."
    ::= { dot11GroupAddressesEntry 2 }

dot11GroupAddressesStatus OBJECT-TYPE
        SYNTAX RowStatus
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "The status column used for creating, modifying, and
            deleting instances of the columnar objects in the Group
            Addresses Table."
        DEFVAL { active }
    ::= { dot11GroupAddressesEntry 3 }

-- ****************************************************************
-- *    End of dot11GroupAddresses TABLE
-- ****************************************************************

-- ****************************************************************
-- *    SMT EDCA Config TABLE
-- ****************************************************************

dot11EDCATable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11EDCAEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "Conceptual table for EDCA default parameter values at a non-AP
             STA. 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 802.11-2007, 9.1.3.1"
    ::= { dot11mac 4 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11EDCAEntry OBJECT-TYPE
        SYNTAX Dot11EDCAEntry
        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 }

Dot11EDCAEntry ::=
        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."
    ::= { dot11EDCAEntry 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 STA 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."
    ::= { dot11EDCAEntry 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 STA for a particular AC for
             generating a random number for the backoff. The value of this
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1067 of 1232

```
            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."
    ::= { dot11EDCAEntry 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 STA, 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."
    ::= { dot11EDCAEntry 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 STA. 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 Clause 17 and Clause 19 PHY and 6016
            microseconds for Clause 18 PHY, if dot11EDCATableIndex is 3,
            3) 1504 microseconds for Clause 17 and Clause 19 PHY and 3264
            microseconds for Clause 18 PHY, if dot11EDCATableIndex is 4."
    ::= { dot11EDCAEntry 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."
    ::= { dot11EDCAEntry 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."
    ::= { dot11EDCAEntry 7 }
```


Copyright © 2007 IEEE. All rights reserved.

```
-- ************************************************************************
-- *    End of SMT EDCA Config TABLE
-- ************************************************************************

-- ************************************************************************
-- *    SMT AP EDCA Config TABLE
-- ************************************************************************

dot11QAPEDCATable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11QAPEDCAEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "Conceptual table for EDCA default parameter values at the AP.
             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 802.11-2007, 9.9.1"
    ::= { dot11mac 5 }

dot11QAPEDCAEntry OBJECT-TYPE
        SYNTAX Dot11QAPEDCAEntry
        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 }

Dot11QAPEDCAEntry ::=
        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."
    ::= { dot11QAPEDCAEntry 1 }

dot11QAPEDCATableCWmin OBJECT-TYPE
        SYNTAX INTEGER (0..255)
```

Copyright © 2007 IEEE. All rights reserved.

```
        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 an AP 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."
    ::= { dot11QAPEDCAEntry 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 an AP 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."
    ::= { dot11QAPEDCAEntry 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 AP, 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."
    ::= { dot11QAPEDCAEntry 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 AP. 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 Clause 17 and Clause 19 PHY and 6016
             microseconds for Clause 18 PHY, if dot11QAPEDCATableIndex is 3,
             3) 1504 microseconds for Clause 17 and Clause 19 PHY and 3264
```


Copyright © 2007 IEEE. All rights reserved.

```
              microseconds for Clause 18 PHY, if dot11QAPEDCATableIndex is 4."
    ::= { dot11QAPEDCAEntry 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 AP before
             it is discarded. The default value for this parameter shall be
             500."
    ::= { dot11QAPEDCAEntry 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."
    ::= { dot11QAPEDCAEntry 7 }

-- ************************************************************************
-- *    End of SMT AP 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,
```

Copyright © 2007 IEEE. All rights reserved.

```
                    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
```


Copyright © 2007 IEEE. All rights reserved.

```
        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 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1073 of 1232

```
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
-- ************************************************************************

-- ************************************************************************
-- *    Resource Type Attribute Templates
-- ************************************************************************

dot11ResourceTypeIDName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Contains the name of the Resource Type ID managed object.
            The attribute is read-only and always contains the value
            RTID.  This attribute value shall not be used as a naming
            attribute for any other managed object class."
        REFERENCE "IEEE Std 802.1F-1993,  A.7"
        DEFVAL { "RTID" }
    ::= { dot11resAttribute 1 }
```


Copyright © 2007 IEEE. All rights reserved.

```
-- ************************************************************************
-- *    dot11ResourceInfo  TABLE
-- ************************************************************************

dot11ResourceInfoTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11ResourceInfoEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Provides a means of indicating, in data readable from a
            managed object, information that identifies the source of
            the implementation."
        REFERENCE "IEEE Std 802.1F-1993,  A.7"
    ::= { dot11resAttribute 2 }

dot11ResourceInfoEntry OBJECT-TYPE
        SYNTAX Dot11ResourceInfoEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11ResourceInfo Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11ResourceInfoTable 1 }

Dot11ResourceInfoEntry ::=
        SEQUENCE { dot11manufacturerOUI             OCTET STRING,
                   dot11manufacturerName            DisplayString,
                   dot11manufacturerProductName     DisplayString,
                   dot11manufacturerProductVersion  DisplayString }

dot11manufacturerOUI OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(3))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Takes the value of an organizationally unique identifier."
    ::= { dot11ResourceInfoEntry 1 }

dot11manufacturerName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "A printable string used to identify the manufacturer of the
            resource.  Maximum string length is 128 octets."
    ::= { dot11ResourceInfoEntry 2 }

dot11manufacturerProductName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "A printable string used to identify the manufacturer's product
            name of the resource.  Maximum string length is 128 octets."
    ::= { dot11ResourceInfoEntry 3 }
```

Copyright © 2007 IEEE. All rights reserved.

```
dot11manufacturerProductVersion OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "Printable string used to identify the manufacturer's product
            version of the resource.  Maximum string length is 128 octets."
    ::= { dot11ResourceInfoEntry 4 }

-- ****************************************************************
-- * End of dot11ResourceInfo  TABLE
-- ****************************************************************

-- ****************************************************************
-- * PHY Attribute Templates
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyOperation  TABLE
-- ****************************************************************

dot11PhyOperationTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyOperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
               "PHY level attributes concerned with
               operation.  Implemented as a table indexed on
               ifIndex to allow for multiple instantiations on an
               Agent."
    ::= { dot11phy 1 }

dot11PhyOperationEntry OBJECT-TYPE
        SYNTAX Dot11PhyOperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyOperation Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyOperationTable 1 }

Dot11PhyOperationEntry ::=
        SEQUENCE { dot11PHYType            INTEGER,
                   dot11CurrentRegDomain   Integer32,
                   dot11TempType           INTEGER }

dot11PHYType OBJECT-TYPE
        SYNTAX INTEGER { fhss(1), dsss(2), irbaseband(3), ofdm(4),
                         hrdsss(5), erp(6)}
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "This is an 8-bit integer value that identifies the PHY type
               supported by the attached PLCP and PMD. Currently defined
```


Copyright © 2007 IEEE. All rights reserved.

```
                values and their corresponding PHY types are:

                FHSS 2.4 GHz = 01, DSSS 2.4 GHz = 02, IR Baseband = 03,
                OFDM 5GHz = 04, HRDSSS = 05, ERP = 06"
    ::= { dot11PhyOperationEntry 1 }

dot11CurrentRegDomain OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current regulatory domain this instance of the PMD is
            supporting.  This object corresponds to one of the
            RegDomains listed in dot11RegDomainsSupported."
    ::= { dot11PhyOperationEntry 2 }

dot11TempType OBJECT-TYPE
        SYNTAX INTEGER { tempType1(1), tempType2(2) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
               "There are different operating temperature requirements
               dependent on the anticipated environmental conditions. This
               attribute describes the current PHY's operating temperature
               range capability. Currently defined values and their
               corresponding temperature ranges are:

               Type 1 = X'01'-Commercial range of 0 to 40 degrees C,

               Type 2 = X'02'-Industrial range of -30 to 70 degrees C."
    ::= { dot11PhyOperationEntry 3 }

-- ****************************************************************
-- * End of dot11PhyOperation  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyAntenna  TABLE
-- ****************************************************************

dot11PhyAntennaTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyAntennaEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group of attributes for PhyAntenna.  Implemented as a
            table indexed on ifIndex to allow for multiple instances on
            an agent."
    ::= { dot11phy 2}

dot11PhyAntennaEntry OBJECT-TYPE
        SYNTAX Dot11PhyAntennaEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyAntenna Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1077 of 1232

```
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyAntennaTable 1 }

Dot11PhyAntennaEntry ::=
        SEQUENCE { dot11CurrentTxAntenna  Integer32,
                   dot11DiversitySupport  INTEGER,
                   dot11CurrentRxAntenna  Integer32 }

dot11CurrentTxAntenna OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The current antenna being used to transmit.  This value
       is one of the values appearing in dot11SupportedTxAntenna. This
       may be used by a management agent to control which antenna is
       used for transmission. "
    ::= { dot11PhyAntennaEntry 1 }

dot11DiversitySupport OBJECT-TYPE
        SYNTAX INTEGER { fixedlist(1), notsupported(2), dynamic(3) }
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
       "This implementation's support for diversity, encoded as:

              X'01'-diversity is available and is performed over the fixed
                  list of antennas defined in dot11DiversitySelectionRx.

              X'02'-diversity is not supported.

              X'03'-diversity is supported and control of diversity is also
                  available, in which case the attribute
                  dot11DiversitySelectionRx can be dynamically modified by
                  the LME."
    ::= { dot11PhyAntennaEntry 2 }

dot11CurrentRxAntenna OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The current antenna being used to receive, if the dot11
       DiversitySupport indicates that diversity is not supported.
       The selected antenna shall be one of the antennae marked
       for receive in the dot11AntennasListTable."
    ::= { dot11PhyAntennaEntry 3 }

-- ****************************************************************
-- * End of dot11PhyAntenna  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyTxPower  TABLE
-- ****************************************************************

dot11PhyTxPowerTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyTxPowerEntry
```


Copyright © 2007 IEEE. All rights reserved.

```
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group of attributes for dot11PhyTxPowerTable.  Implemented
            as a table indexed on STA ID to allow for multiple
            instances on an Agent."
    ::= { dot11phy 3}

dot11PhyTxPowerEntry OBJECT-TYPE
        SYNTAX Dot11PhyTxPowerEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyTxPower Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyTxPowerTable 1 }

Dot11PhyTxPowerEntry ::=
        SEQUENCE { dot11NumberSupportedPowerLevels  INTEGER,
                   dot11TxPowerLevel1               INTEGER,
                   dot11TxPowerLevel2               INTEGER,
                   dot11TxPowerLevel3               INTEGER,
                   dot11TxPowerLevel4               INTEGER,
                   dot11TxPowerLevel5               INTEGER,
                   dot11TxPowerLevel6               INTEGER,
                   dot11TxPowerLevel7               INTEGER,
                   dot11TxPowerLevel8               INTEGER,
                   dot11CurrentTxPowerLevel         INTEGER }

dot11NumberSupportedPowerLevels OBJECT-TYPE
        SYNTAX INTEGER (1..8)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The number of power levels supported by the PMD.
            This attribute can have a value of 1 to 8."
    ::= { dot11PhyTxPowerEntry 1 }

dot11TxPowerLevel1 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL1 in mW.
            This is also the default power level."
    ::= { dot11PhyTxPowerEntry 2 }

dot11TxPowerLevel2 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL2 in mW."
    ::= { dot11PhyTxPowerEntry 3 }
```

Copyright © 2007 IEEE. All rights reserved.

```
dot11TxPowerLevel3 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL3 in mW."
    ::= { dot11PhyTxPowerEntry 4 }

dot11TxPowerLevel4 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL4 in mW."
    ::= { dot11PhyTxPowerEntry 5 }

dot11TxPowerLevel5 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL5 in mW."
    ::= { dot11PhyTxPowerEntry 6 }

dot11TxPowerLevel6 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL6 in mW."
    ::= { dot11PhyTxPowerEntry 7 }

dot11TxPowerLevel7 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL7 in mW."
    ::= { dot11PhyTxPowerEntry 8 }

dot11TxPowerLevel8 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The transmit output power for LEVEL8 in mW."
    ::= { dot11PhyTxPowerEntry 9 }

dot11CurrentTxPowerLevel OBJECT-TYPE
        SYNTAX INTEGER (1..8)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The TxPowerLevel N currently being used to transmit data.
            Some PHYs also use this value to determine the receiver
            sensitivity requirements for CCA."
    ::= { dot11PhyTxPowerEntry 10 }

-- ****************************************************************
```


Copyright © 2007 IEEE. All rights reserved.

```
-- * End of dot11PhyTxPower  TABLE
-- ************************************************************************

-- ************************************************************************
-- * dot11PhyFHSS  TABLE
-- ************************************************************************

dot11PhyFHSSTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyFHSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group of attributes for dot11PhyFHSSTable.  Implemented as a
            table indexed on STA ID to allow for multiple instances on
            an Agent."
    ::= { dot11phy 4 }

dot11PhyFHSSEntry OBJECT-TYPE
        SYNTAX Dot11PhyFHSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyFHSS Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyFHSSTable 1 }

Dot11PhyFHSSEntry ::=
        SEQUENCE { dot11HopTime                          INTEGER,
                   dot11CurrentChannelNumber             INTEGER,
                   dot11MaxDwellTime                     INTEGER,
                   dot11CurrentDwellTime                 INTEGER,
                   dot11CurrentSet                       INTEGER,
                   dot11CurrentPattern                   INTEGER,
                   dot11CurrentIndex                     INTEGER,
                   dot11EHCCPrimeRadix                   Integer32,
                   dot11EHCCNumberofChannelsFamilyIndex  Integer32,
                   dot11EHCCCapabilityImplemented        TruthValue,
                   dot11EHCCCapabilityEnabled            TruthValue,
                   dot11HopAlgorithmAdopted              INTEGER,
                   dot11RandomTableFlag                  TruthValue,
                   dot11NumberofHoppingSets              Integer32,
                   dot11HopModulus                       Integer32,
                   dot11HopOffset                        Integer32 }

dot11HopTime OBJECT-TYPE
        SYNTAX INTEGER (224)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The time in microseconds for the PMD to change from
            channel 2 to channel 80."
    ::= { dot11PhyFHSSEntry 1 }

dot11CurrentChannelNumber OBJECT-TYPE
        SYNTAX INTEGER (0..200)
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1081 of 1232

```
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The current channel number of the frequency output by the RF
       synthesizer."
    ::= { dot11PhyFHSSEntry 2 }

dot11MaxDwellTime OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The maximum time in TU that the transmitter
            is permitted to operate on a single channel."
    ::= { dot11PhyFHSSEntry 3 }

dot11CurrentDwellTime OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current time in TU that the transmitter shall operate
            on a single channel, as set by the MAC.  Default is 19 TU."
    ::= { dot11PhyFHSSEntry 4 }

dot11CurrentSet OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The current set of patterns the PLME
       is using to determine the hopping sequence. "
    ::= { dot11PhyFHSSEntry 5 }

dot11CurrentPattern OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current pattern the PLME is
            using to determine the hop sequence."
    ::= { dot11PhyFHSSEntry 6 }

dot11CurrentIndex OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current index value the PLME is using to determine
            the CurrentChannelNumber."
    ::= { dot11PhyFHSSEntry 7 }

dot11EHCCPrimeRadix OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute indicates the value to be
              used as the prime radix (N) in the HCC and
```


Copyright © 2007 IEEE. All rights reserved.

```
              EHCC algorithms."
    ::= { dot11PhyFHSSEntry 8 }

dot11EHCCNumberofChannelsFamilyIndex OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute indicates the value to be
              used as the maximum for the family index (a)
              in the HCC and EHCC algorithms. The value of
              this field shall not be less than the prime
              radix minus 3 (N - 3). The valid range of
              allowed values is (N - 1), (N - 2), and (N - 3)."
    ::= { dot11PhyFHSSEntry 9 }

dot11EHCCCapabilityImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, indicates that the
              station implementation is capable of generating
              the HCC or EHCC algorithms for determining Hopping
              patterns.  The capability is disabled, otherwise.
              The default value of this attribute is FALSE."
    ::= { dot11PhyFHSSEntry 10 }

dot11EHCCCapabilityEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, indicates that the
              capability of the station to operate using the HCC
              or EHCC algorithms for determining Hopping Patterns
              is enabled.  The capability is disabled, otherwise.
              The default value of this attribute is FALSE."
    ::= { dot11PhyFHSSEntry 11 }

dot11HopAlgorithmAdopted OBJECT-TYPE
       SYNTAX INTEGER { crnt(1), hopindex(2), hcc(3) }
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute, indicates which of the algorithms
              will be used to generate the Hopping Patterns.
              Valid values are:

              1 - hopping patterns as defined in Clause 14
              2 - hop index method (with or without table)
              3 - HCC/EHCC method"
    ::= { dot11PhyFHSSEntry 12 }

dot11RandomTableFlag OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1083 of 1232

```
              "This attribute, indicates that a Random Table is
              present when the value is TRUE. When the value is
              FALSE it indicates that a Random Table is not
              present and that the hop index method is to be
              used to determine the hopping sequence. The default
              value of this attribute is TRUE."
    ::= { dot11PhyFHSSEntry 13 }

dot11NumberofHoppingSets OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The Number of Sets field indicates the total
              number of sets within the hopping patterns."
    ::= { dot11PhyFHSSEntry 14 }

dot11HopModulus OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The number of allowed channels for the hopping
              set. This is defined by the governing regulatory
              agency for the country code of the country
              in which this device is operating."
    ::= { dot11PhyFHSSEntry 15 }

dot11HopOffset OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The next position in the hopping set."
    ::= { dot11PhyFHSSEntry 16 }

-- ****************************************************************
-- * End of dot11PhyFHSS  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyDSSSEntry  TABLE
-- ****************************************************************

dot11PhyDSSSTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyDSSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Entry of attributes for dot11PhyDSSSEntry.  Implemented as a
            table indexed on ifIndex to allow for multiple instances on
            an Agent."
    ::= { dot11phy 5 }

dot11PhyDSSSEntry OBJECT-TYPE
        SYNTAX Dot11PhyDSSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
```


Copyright © 2007 IEEE. All rights reserved.

```
            "An entry in the dot11PhyDSSSEntry Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyDSSSTable 1 }

Dot11PhyDSSSEntry ::=
        SEQUENCE { dot11CurrentChannel    INTEGER,
                   dot11CCAModeSupported  INTEGER,
                   dot11CurrentCCAMode    INTEGER,
                   dot11EDThreshold       Integer32 }

dot11CurrentChannel OBJECT-TYPE
        SYNTAX INTEGER (1..14)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current operating frequency channel of the DSSS
            PHY. Valid channel numbers are as defined in 15.4.6.2"
    ::= { dot11PhyDSSSEntry 1 }

dot11CCAModeSupported OBJECT-TYPE
        SYNTAX INTEGER (1..7)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
       "dot11CCAModeSupported is a bit-significant value,
       representing all of the CCA modes supported by the PHY.
       Valid values are:

          energy detect only (ED_ONLY) = 01,
          carrier sense only (CS_ONLY) = 02,
          carrier sense and energy detect (ED_and_CS)= 04

       or the logical sum of any of these values.  This
       attribute shall not be used to indicate the CCA modes
       supported by a higher rate extension PHY.  Rather, the
       dot11HRCCAModeSupported attribute shall be used to
       indicate the CCA modes of the higher rate extension PHY."
    ::= { dot11PhyDSSSEntry 2 }

dot11CurrentCCAMode OBJECT-TYPE
        SYNTAX INTEGER { edonly(1), csonly(2), edandcs(4), cswithtimer(8),
                  hrcsanded(16) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current CCA method in operation.   Valid values are:
                energy detect only (edonly) = 01,
                carrier sense only (csonly) = 02,
                carrier sense and energy detect (edandcs)= 04
                carrier sense with timer (cswithtimer)= 08
                high rate carrier sense and energy detect (hrcsanded)=16."
    ::= { dot11PhyDSSSEntry 3 }

dot11EDThreshold OBJECT-TYPE
        SYNTAX Integer32
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1085 of 1232

```
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The current Energy Detect Threshold being used by the DSSS PHY."
    ::= { dot11PhyDSSSEntry 4 }

-- ****************************************************************
-- * End of dot11PhyDSSSEntry  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyIR  TABLE
-- ****************************************************************

dot11PhyIRTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyIREntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group of attributes for dot11PhyIRTable.  Implemented as a
            table indexed on ifIndex to allow for multiple instances on
            an Agent."
    ::= { dot11phy 6 }

dot11PhyIREntry OBJECT-TYPE
        SYNTAX Dot11PhyIREntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyIR Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyIRTable 1 }

Dot11PhyIREntry ::=
        SEQUENCE { dot11CCAWatchdogTimerMax       Integer32,
                   dot11CCAWatchdogCountMax       Integer32,
                   dot11CCAWatchdogTimerMin       Integer32,
                   dot11CCAWatchdogCountMin       Integer32 }

dot11CCAWatchdogTimerMax OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This parameter, together with CCAWatchdogCountMax,
            determines when energy detected in the channel can be
            ignored."
    ::= { dot11PhyIREntry 1 }

dot11CCAWatchdogCountMax OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "This parameter, together with CCAWatchdogTimerMax,
```


Copyright © 2007 IEEE. All rights reserved.

```
            determines when energy detected in the channel can be
            ignored."
    ::= { dot11PhyIREntry 2 }

dot11CCAWatchdogTimerMin OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The minimum value to which CCAWatchdogTimerMax can be
            set."
    ::= { dot11PhyIREntry 3 }

dot11CCAWatchdogCountMin OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "The minimum value to which CCAWatchdogCount can be set."
    ::= { dot11PhyIREntry 4 }

-- ****************************************************************
-- * End of dot11PhyIR  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11RegDomainsSupported  TABLE
-- ****************************************************************

dot11RegDomainsSupportedTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11RegDomainsSupportedEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "There are different operational requirements dependent on
            the regulatory domain.  This attribute list describes the
            regulatory domains the PLCP and PMD support in this
            implementation.  Currently defined values and their
            corresponding Regulatory Domains are:

            FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of
            Europe) = X'30', Spain = X'31', France = X'32',
            Japan = X'40', China = X'50', Other = X'00' "
    ::= { dot11phy 7}

dot11RegDomainsSupportedEntry OBJECT-TYPE
        SYNTAX Dot11RegDomainsSupportedEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11RegDomainsSupportedTable.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11RegDomainsSupportedIndex }
    ::= { dot11RegDomainsSupportedTable 1 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1087 of 1232

```
Dot11RegDomainsSupportedEntry ::=
        SEQUENCE { dot11RegDomainsSupportedIndex    Integer32,
                   dot11RegDomainsSupportedValue    INTEGER }

dot11RegDomainsSupportedIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The auxiliary variable used to identify instances
            of the columnar objects in the RegDomainsSupport Table."
    ::= { dot11RegDomainsSupportedEntry 1 }

dot11RegDomainsSupportedValue OBJECT-TYPE
        SYNTAX INTEGER { fcc(16), doc(32), etsi(48), spain (49), france(50),
                         japan (64), other (0)}
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "There are different operational requirements dependent on
            the regulatory domain.  This attribute list describes the
            regulatory domains the PLCP and PMD support in this
            implementation.  Currently defined values and their
            corresponding Regulatory Domains are:

            FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of
            Europe) = X'30', Spain = X'31', France = X'32',
            Japan = X'40', China = X'50' "
    ::= { dot11RegDomainsSupportedEntry 2 }

-- ****************************************************************
-- * End of dot11RegDomainsSupported  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11AntennasList  TABLE
-- ****************************************************************

dot11AntennasListTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11AntennasListEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
             "This table represents the list of antennae.  An antenna can be
             marked to be capable of transmitting, receiving, and/or for
             participation in receive diversity.  Each entry in this table
             represents a single antenna with its properties.  The maximum
             number of antennae that can be contained in this table is 255."
    ::= { dot11phy 8 }

dot11AntennasListEntry OBJECT-TYPE
        SYNTAX Dot11AntennasListEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
           "An entry in the dot11AntennasListTable, representing the properties
            of a single antenna.

            ifIndex - Each IEEE 802.11 interface is represented by an
```


Copyright © 2007 IEEE. All rights reserved.

```
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11AntennaListIndex }
    ::= { dot11AntennasListTable 1 }

Dot11AntennasListEntry ::=
        SEQUENCE { dot11AntennaListIndex     Integer32,
                   dot11SupportedTxAntenna   TruthValue,
                   dot11SupportedRxAntenna   TruthValue,
                   dot11DiversitySelectionRx TruthValue }

dot11AntennaListIndex OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
           "The unique index of an antenna which is used to identify the columnar
            objects in the dot11AntennasList Table."
    ::= { dot11AntennasListEntry 1 }

dot11SupportedTxAntenna OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "When TRUE, this object indicates that the antenna represented by
            dot11AntennaIndex can be used as a transmit antenna."
    ::= { dot11AntennasListEntry 2 }

dot11SupportedRxAntenna OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
           "When TRUE, this object indicates that the antenna represented by the
            dot11AntennaIndex can be used as a receive antenna."
    ::= { dot11AntennasListEntry 3 }

dot11DiversitySelectionRx OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "When TRUE, this object indicates that the antenna represented by
            dot11AntennaIndex can be used for receive diversity.  This object
            shall be TRUE only if the antenna can be used as a receive antenna,
            as indicated by dot11SupportedRxAntenna."
    ::= { dot11AntennasListEntry 4 }

-- ****************************************************************
-- * End of dot11AntennasList  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11SupportedDataRatesTx  TABLE
-- ****************************************************************

dot11SupportedDataRatesTxTable OBJECT-TYPE
```

Copyright © 2007 IEEE. All rights reserved.

```
        SYNTAX SEQUENCE OF Dot11SupportedDataRatesTxEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The Transmit bit rates supported by the PLCP and PMD,
            represented by a count from X'02-X'7f, corresponding to data
            rates in increments of 500kbit/s from 1 Mb/s to 63.5 Mb/s subject
            to limitations of each individual PHY."
    ::= { dot11phy 9 }

dot11SupportedDataRatesTxEntry OBJECT-TYPE
        SYNTAX Dot11SupportedDataRatesTxEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the dot11SupportedDataRatesTx
            Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11SupportedDataRatesTxIndex }
    ::= { dot11SupportedDataRatesTxTable  1 }

Dot11SupportedDataRatesTxEntry ::=
        SEQUENCE { dot11SupportedDataRatesTxIndex  Integer32,
                   dot11SupportedDataRatesTxValue  Integer32 }

dot11SupportedDataRatesTxIndex OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Index object that identifies which data rate to access.
            Range is 1..255."
    ::= { dot11SupportedDataRatesTxEntry 1 }

dot11SupportedDataRatesTxValue OBJECT-TYPE
        SYNTAX Integer32 (2..127)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The Transmit bit rates supported by the PLCP and PMD,
            represented by a count from X'02-X'7f, corresponding to data
            rates in increments of 500kbit/s from 1 Mb/s to 63.5 Mb/s subject
            to limitations of each individual PHY."
    ::= { dot11SupportedDataRatesTxEntry 2 }

-- ************************************************************************
-- * End of dot11SupportedDataRatesTx  TABLE
-- ************************************************************************

-- ************************************************************************
-- * dot11SupportedDataRatesRx  TABLE
-- ************************************************************************

dot11SupportedDataRatesRxTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11SupportedDataRatesRxEntry
```


Copyright © 2007 IEEE. All rights reserved.

```
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "The receive bit rates supported by the PLCP and PMD,
            represented by a count from X'002-X'7f, corresponding to data
            rates in increments of 500kbit/s from 1 Mb/s to 63.5 Mb/s."
    ::= { dot11phy 10 }

dot11SupportedDataRatesRxEntry OBJECT-TYPE
        SYNTAX Dot11SupportedDataRatesRxEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An Entry (conceptual row) in the dot11SupportedDataRatesRx Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry.  Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex,
                dot11SupportedDataRatesRxIndex }
    ::= { dot11SupportedDataRatesRxTable  1 }

Dot11SupportedDataRatesRxEntry ::=
        SEQUENCE { dot11SupportedDataRatesRxIndex  Integer32,
                   dot11SupportedDataRatesRxValue  Integer32 }

dot11SupportedDataRatesRxIndex OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Index object that identifies which data rate to access.
            Range is 1..255."
    ::= { dot11SupportedDataRatesRxEntry 1 }

dot11SupportedDataRatesRxValue OBJECT-TYPE
        SYNTAX Integer32 (2..127)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "The receive bit rates supported by the PLCP and PMD,
            represented by a count from X'02-X'7f, corresponding to data
            rates in increments of 500kbit/s from 1 Mb/s to 63.5 Mb/s."
    ::= { dot11SupportedDataRatesRxEntry 2 }

-- ****************************************************************
-- * End of dot11SupportedDataRatesRx  TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyOFDM TABLE
-- ****************************************************************

dot11PhyOFDMTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyOFDMEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "Group of attributes for dot11PhyOFDMTable. Implemented as a
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1091 of 1232

```
            table indexed on ifindex to allow for multiple instances on
            an Agent."
    ::= { dot11phy 11 }

dot11PhyOFDMEntry OBJECT-TYPE
        SYNTAX Dot11PhyOFDMEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "An entry in the dot11PhyOFDM Table.

            ifIndex - Each IEEE 802.11 interface is represented by an
            ifEntry. Interface tables in this MIB module are indexed
            by ifIndex."
        INDEX { ifIndex }
    ::= { dot11PhyOFDMTable 1 }

Dot11PhyOFDMEntry ::=
        SEQUENCE { dot11CurrentFrequency                    INTEGER,
                   dot11TIThreshold                         Integer32,
                   dot11FrequencyBandsSupported             INTEGER,
                   dot11ChannelStartingFactor               Integer32,
                   dot11FiveMHzOperationImplemented         TruthValue,
                   dot11TenMHzOperationImplemented          TruthValue,
                   dot11TwentyMHzOperationImplemented       TruthValue,
                   dot11PhyOFDMChannelWidth                 INTEGER }

dot11CurrentFrequency OBJECT-TYPE
        SYNTAX INTEGER (0..200)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
       "The number of the current operating frequency channel of the OFDM
       PHY."
    ::= { dot11PhyOFDMEntry 1 }

dot11TIThreshold OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "The Threshold being used to detect a busy medium (frequency).
            CCA shall report a busy medium upon detecting the RSSI above
          this threshold."
    ::= { dot11PhyOFDMEntry 2 }

dot11FrequencyBandsSupported OBJECT-TYPE
        SYNTAX INTEGER (1..127)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
             "The capability of the OFDM PHY implementation to operate in the
             4.9 GHz and 5 GHz bands. Coded as an integer value with bit 0 LSB as
             follows:
                bit 0 .. capable of operating in the 5.15-5.25 GHz band
                bit 1 .. capable of operating in the 5.25-5.35 GHz band
                bit 2 .. capable of operating in the 5.725-5.825 GHz band
                bit 3 .. capable of operating in the 5.47-5.725 GHz band
                bit 4 .. capable of operating in the lower Japanese (5.15-
```


Copyright © 2007 IEEE. All rights reserved.

```
                    5.25 GHz) band
                 bit 5 .. capable of operating in the 5.03-5.091 GHz band
                 bit 6 .. capable of operating in the 4.94-4.99 GHz band
              For example, for an implementation capable of operating in the
              5.15-5.35 GHz bands this attribute would take the value 3."
::= { dot11PhyOFDMEntry 3 }

dot11ChannelStartingFactor OBJECT-TYPE
       SYNTAX Integer32 (8000..10000)
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The base factor from which channel center frequencies are calculat-
              ed. This number is multiplied by 500 kHz to form the base frequency
              to be added to the channel number x 5 MHz. The default value of this
              attribute shall be 10 000."
    ::= { dot11PhyOFDMEntry 4 }

dot11FiveMHzOperationImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the 5 MHz
              Operation is implemented. The default value of this attribute is
              FALSE."
    ::= { dot11PhyOFDMEntry 5 }

dot11TenMHzOperationImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the 10 MHz
              Operation is implemented. The default value of this attribute is
              FALSE."
    ::= { dot11PhyOFDMEntry 6 }

dot11TwentyMHzOperationImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the 20 MHz
              Operation is implemented. The default value of this attribute is
              TRUE."
    ::= { dot11PhyOFDMEntry 7 }

dot11PhyOFDMChannelWidth OBJECT-TYPE
       SYNTAX INTEGER { width5MHz(1), width10MHz(2), width20MHz(3)}
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This is an 8-bit integer value that identifies the OFDM
              PHY channel width. Currently defined values and their corresponding
              Channel widths are:
                 5MHz = 01, 10MHz = 02, 20MHz = 03"
    ::= { dot11PhyOFDMEntry 8 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1093 of 1232

```
-- ************************************************************************
-- * End of dot11PhyOFDM TABLE
-- ************************************************************************

-- ************************************************************************
-- * dot11PhyHRDSSSEntry TABLE
-- ************************************************************************

dot11PhyHRDSSSTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11PhyHRDSSSEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Entry of attributes for dot11PhyHRDSSSEntry.
              Implemented as a table indexed on ifIndex to allow for
              multiple instances on an Agent."
    ::= { dot11phy 12 }

dot11PhyHRDSSSEntry OBJECT-TYPE
       SYNTAX Dot11PhyHRDSSSEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry in the dot11PhyHRDSSSEntry Table.

              ifIndex - Each IEEE 802.11 interface is represented by an
              ifEntry. Interface tables in this MIB module are indexed
              by ifIndex."
       INDEX { ifIndex }
    ::= { dot11PhyHRDSSSTable 1 }

Dot11PhyHRDSSSEntry ::=
        SEQUENCE { dot11ShortPreambleOptionImplemented      TruthValue,
                   dot11PBCCOptionImplemented               TruthValue,
                   dot11ChannelAgilityPresent               TruthValue,
                   dot11ChannelAgilityEnabled               TruthValue,
                   dot11HRCCAModeSupported                  INTEGER }

dot11ShortPreambleOptionImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the
              short preamble option as defined in 18.2.2.2
              is implemented.  The default value of this attribute
              shall be FALSE."
    ::= { dot11PhyHRDSSSEntry 1 }

dot11PBCCOptionImplemented OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the PBCC
              modulation option as defined in  18.4.6.6 is
              implemented.  The default value of this attribute shall
              be FALSE."
    ::= { dot11PhyHRDSSSEntry 2 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11ChannelAgilityPresent OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute indicates that the PHY is capable of
              channel agility."
    ::= { dot11PhyHRDSSSEntry 3 }

dot11ChannelAgilityEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This attribute indicates that the PHY channel agility
              functionality is enabled."
    ::= { dot11PhyHRDSSSEntry 4 }

dot11HRCCAModeSupported OBJECT-TYPE
       SYNTAX INTEGER (1..31)
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "dot11HRCCAModeSupported is a bit-significant value,
              representing all of the CCA modes supported by the PHY.
              Valid values are:
                  energy detect only (ED_ONLY) = 01,
                  carrier sense only (CS_ONLY) = 02,
                  carrier sense and energy detect (ED_and_CS)= 04,
                  carrier sense with timer (CS_and_Timer)= 08,
                  high rate carrier sense and energy detect
              (HRCS_and_ED)= 16
              or the logical sum of any of these values.  In
              the high rate extension PHY, this attribute shall
              be used in preference to the dot11CCAModeSupported
              attribute."
    ::= { dot11PhyHRDSSSEntry 5 }

-- ****************************************************************
-- * End of dot11PhyHRDSSSEntry TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11HoppingPattern TABLE
-- ****************************************************************

dot11HoppingPatternTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11HoppingPatternEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The (conceptual) table of attributes necessary for
              a frequency hopping implementation to be able to
              create the hopping sequences necessary to operate
              in the subband for the associated domain country string."
       ::= { dot11phy 13 }

dot11HoppingPatternEntry OBJECT-TYPE
```

Copyright © 2007 IEEE. All rights reserved.

```
       SYNTAX Dot11HoppingPatternEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry (conceptual row) in the Hopping Pattern Table
              that indicates the random hopping sequence to be followed.

              IfIndex - Each IEEE 802.11 interface is represented
              by an ifEntry. Interface tables in this MIB are indexed
              by ifIndex."
       INDEX { ifIndex,
                dot11HoppingPatternIndex }
    ::= { dot11HoppingPatternTable 1 }

Dot11HoppingPatternEntry ::=
            SEQUENCE { dot11HoppingPatternIndex      Integer32,
                       dot11RandomTableFieldNumber   Integer32 }

dot11HoppingPatternIndex OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The auxiliary variable used to identify instances of
              the columnar objects in the Hopping Pattern Table."
    ::= { dot11HoppingPatternEntry 1 }

dot11RandomTableFieldNumber OBJECT-TYPE
       SYNTAX Integer32
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute shall indicate the value of the
              starting channel number in the hopping sequence of
              the subband for the associated domain country string.
              The default value of this attribute shall be zero."
    ::= { dot11HoppingPatternEntry 2 }

-- ****************************************************************
-- * End of dot11HoppingPattern TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11PhyERP TABLE
-- ****************************************************************

dot11PhyERPTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11PhyERPEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Entry of attributes for dot11PhyERPEntry. Implemented as a
              table indexed on ifIndex to allow for multiple instances on an
              Agent."
    ::= { dot11phy 14 }

dot11PhyERPEntry OBJECT-TYPE
       SYNTAX Dot11PhyERPEntry
       MAX-ACCESS not-accessible
```


Copyright © 2007 IEEE. All rights reserved.

```
        STATUS current
        DESCRIPTION
                "An entry in the dot11PhyERPEntry Table.
                ifIndex - Each 802.11 interface is represented by an ifEntry.
                Interface tables in this MIB module are indexed by ifIndex."
        INDEX {ifIndex}
    ::= { dot11PhyERPTable 1 }

Dot11PhyERPEntry ::= SEQUENCE {
        dot11ERPPBCCOptionImplemented              TruthValue,
        dot11ERPBCCOptionEnabled                   TruthValue,
        dot11DSSSOFDMOptionImplemented             TruthValue,
        dot11DSSSOFDMOptionEnabled                 TruthValue,
        dot11ShortSlotTimeOptionImplemented        TruthValue,
        dot11ShortSlotTimeOptionEnabled            TruthValue }

dot11ERPPBCCOptionImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This attribute, when TRUE, shall indicate that the ERP-PBCC
                modulation option as defined in 19.6 is implemented. The default
                value of this attribute is FALSE."
    ::= { dot11PhyERPEntry 1 }

dot11ERPBCCOptionEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "This attribute, when TRUE, shall indicate that the
                ERP-PBCC option as defined in 19.6 is enabled. The default value
                of this attribute is FALSE."
    ::= { dot11PhyERPEntry 2 }

dot11DSSSOFDMOptionImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This attribute, when TRUE, shall indicate that the DSSS-OFDM
                option as defined in 19.7 is implemented. The default value of
                this attribute is FALSE."
    ::= { dot11PhyERPEntry 3 }

dot11DSSSOFDMOptionEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "This attribute, when TRUE, shall indicate that the DSSS-OFDM
                option as defined in 19.7 is enabled. The default value of this
                attribute is FALSE."
    ::= { dot11PhyERPEntry 4 }

dot11ShortSlotTimeOptionImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1097 of 1232

```
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the Short Slot
              Time option as defined in 7.3.1.4 is implemented. The default
              value of this attribute is FALSE."
    ::= { dot11PhyERPEntry 5}

dot11ShortSlotTimeOptionEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This attribute, when TRUE, shall indicate that the Short Slot
              Time option as defined in 7.3.1.4 is enabled. The default value
              of this attribute is FALSE."
    ::= { dot11PhyERPEntry 6 }

-- ****************************************************************
-- * End of dot11PhyERP TABLE
-- ****************************************************************

-- ****************************************************************
-- * Conformance Information
-- ****************************************************************

dot11Conformance    OBJECT IDENTIFIER ::= { ieee802dot11 5 }
dot11Groups         OBJECT IDENTIFIER ::= { dot11Conformance 1 }
dot11Compliances    OBJECT IDENTIFIER ::= { dot11Conformance 2 }

-- ****************************************************************
-- * Compliance Statements
-- ****************************************************************

dot11Compliance MODULE-COMPLIANCE
        STATUS  current
        DESCRIPTION
            "The compliance statement for SNMPv2 entities
            that implement the IEEE 802.11 MIB."
       MODULE  -- this module
       MANDATORY-GROUPS {
       dot11SMTbase6,
       dot11MACbase2, dot11CountersGroup2,
       dot11SmtAuthenticationAlgorithms,
       dot11ResourceTypeID, dot11PhyOperationComplianceGroup }

GROUP dot11PhyDSSSComplianceGroup
        DESCRIPTION
              "Implementation of this group is required when object
              dot11PHYType has the value of dsss.  This group is
              mutually exclusive with the groups dot11PhyIRComplianceGroup,
              dot11PhyFHSSComplianceGroup2, dot11PhyOFDMComplianceGroup2
              and dot11PhyHRDSSSComplianceGroup."

    GROUP dot11PhyIRComplianceGroup
        DESCRIPTION
              "Implementation of this group is required when object
              dot11PHYType has the value of irbaseband.  This group is
              mutually exclusive with the groups dot11PhyDSSSComplianceGroup,
              dot11PhyFHSSComplianceGroup2, dot11PhyOFDMComplianceGroup2
```


Copyright © 2007 IEEE. All rights reserved.

```
            and dot11PhyHRDSSSComplianceGroup."

    GROUP dot11PhyFHSSComplianceGroup2
        DESCRIPTION
            "Implementation of this group is required when object
            dot11PHYType has the value of fhss.  This group is
            mutually exclusive with the groups dot11PhyDSSSComplianceGroup,
            dot11PhyIRComplianceGroup, dot11PhyOFDMComplianceGroup2
            and dot11PhyHRDSSSComplianceGroup."

    GROUP dot11PhyOFDMComplianceGroup2
        DESCRIPTION
            "Implementation of this group is required when object
            dot11PHYType has the value of ofdm. This group is
            mutually exclusive with the groups dot11PhyDSSSComplianceGroup,
            dot11PhyIRComplianceGroup, dot11PhyFHSSComplianceGroup2
            and dot11PhyHRDSSSComplianceGroup."

    GROUP dot11PhyHRDSSSComplianceGroup
       DESCRIPTION
            "Implementation of this group is required when object
            dot11PHYType has the value of hrdsss. This group is
            mutually exclusive with the groups
            dot11PhyDSSSComplianceGroup, dot11PhyIRComplianceGroup,
            dot11PhyFHSSComplianceGroup2 and dot11PhyOFDMComplianceGroup2."

    GROUP dot11PhyERPComplianceGroup
       DESCRIPTION
            "Implementation of this group is required when object
            dot11PHYType has the value of ERP. This group is mutually
            exclusive with the groups dot11PhyIRComplianceGroup and
            dot11PhyFHSSComplianceGroup2."

    -- OPTIONAL-GROUPS { dot11SMTprivacy, dot11MACStatistics,
    --    dot11PhyAntennaComplianceGroup, dot11PhyTxPowerComplianceGroup,
    --    dot11PhyRegDomainsSupportGroup,
    --    dot11PhyAntennasListGroup, dot11PhyRateGroup,
    --    dot11MultiDomainCapabilityGroup,
    --    dot11PhyFHSSComplianceGroup2, dot11RSNAadditions,
    --    dot11RegulatoryClassesGroup, dot11Qosadditions }
    --

    ::= { dot11Compliances 1 }

-- ****************************************************************
-- *   Groups - units of conformance
-- ****************************************************************

dot11SMTbase OBJECT-GROUP
       OBJECTS { dot11StationID, dot11MediumOccupancyLimit,
                 dot11CFPollable,
                 dot11CFPPeriod,
                 dot11CFPMaxDuration,
                 dot11AuthenticationResponseTimeOut,
                 dot11PrivacyOptionImplemented,
                 dot11PowerManagementMode,
                 dot11DesiredSSID, dot11DesiredBSSType,
                 dot11OperationalRateSet,
                 dot11BeaconPeriod, dot11DTIMPeriod,
```

Copyright © 2007 IEEE. All rights reserved.

```
                dot11AssociationResponseTimeOut }
        STATUS deprecated
        DESCRIPTION
             "The SMT 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 1 }

dot11SMTprivacy OBJECT-GROUP
       OBJECTS { dot11PrivacyInvoked,
                 dot11WEPKeyMappingLength, dot11ExcludeUnencrypted,
                 dot11WEPICVErrorCount , dot11WEPExcludedCount ,
                 dot11WEPDefaultKeyID,
                 dot11WEPDefaultKeyValue,
                 dot11WEPKeyMappingWEPOn,
                 dot11WEPKeyMappingValue , dot11WEPKeyMappingAddress,
                 dot11WEPKeyMappingStatus }
        STATUS current
        DESCRIPTION
            "The SMTPrivacy package is a set of attributes that shall be
            present if WEP is implemented in the STA."
    ::= { dot11Groups 2 }

dot11MACbase OBJECT-GROUP
       OBJECTS { dot11MACAddress, dot11Address,
                 dot11GroupAddressesStatus,
                 dot11RTSThreshold, dot11ShortRetryLimit,
                 dot11LongRetryLimit, dot11FragmentationThreshold,
                 dot11MaxTransmitMSDULifetime,
                 dot11MaxReceiveLifetime, dot11ManufacturerID,
                 dot11ProductID }
        STATUS 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 }

dot11MACStatistics OBJECT-GROUP
       OBJECTS { dot11RetryCount, dot11MultipleRetryCount,
                 dot11RTSSuccessCount, dot11RTSFailureCount,
                 dot11ACKFailureCount, dot11FrameDuplicateCount }
        STATUS current
        DESCRIPTION
            "The MACStatistics package provides extended statistical
            information on the operation of the MAC.  This
            package is completely optional."
    ::= { dot11Groups 4 }

dot11ResourceTypeID OBJECT-GROUP
       OBJECTS { dot11ResourceTypeIDName, dot11manufacturerOUI,
                 dot11manufacturerName, dot11manufacturerProductName,
                 dot11manufacturerProductVersion }
        STATUS current
        DESCRIPTION
             "Attributes used to identify a STA, its manufacturer,
             and various product names and versions."
    ::= { dot11Groups 5 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11SmtAuthenticationAlgorithms OBJECT-GROUP
       OBJECTS { dot11AuthenticationAlgorithm,
                 dot11AuthenticationAlgorithmsEnable }
        STATUS current
        DESCRIPTION
             "Authentication Algorithm Table."
    ::= { dot11Groups 6 }

dot11PhyOperationComplianceGroup OBJECT-GROUP
       OBJECTS { dot11PHYType, dot11CurrentRegDomain, dot11TempType }
        STATUS current
        DESCRIPTION
            "PHY layer operations attributes."
    ::= { dot11Groups 7 }

dot11PhyAntennaComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CurrentTxAntenna, dot11DiversitySupport,
       dot11CurrentRxAntenna }
        STATUS current
        DESCRIPTION
            "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 8 }

dot11PhyTxPowerComplianceGroup OBJECT-GROUP
       OBJECTS { dot11NumberSupportedPowerLevels, dot11TxPowerLevel1,
                 dot11TxPowerLevel2, dot11TxPowerLevel3, dot11TxPowerLevel4,
                 dot11TxPowerLevel5, dot11TxPowerLevel6, dot11TxPowerLevel7,
                 dot11TxPowerLevel8, dot11CurrentTxPowerLevel }
        STATUS current
        DESCRIPTION
            "Attributes for Control and Management of transmit power."
    ::= { dot11Groups 9 }

dot11PhyFHSSComplianceGroup OBJECT-GROUP
       OBJECTS { dot11HopTime, dot11CurrentChannelNumber,
                 dot11MaxDwellTime, dot11CurrentDwellTime, dot11CurrentSet,
                 dot11CurrentPattern, dot11CurrentIndex }
        STATUS deprecated
        DESCRIPTION
            "Attributes that configure the Frequency Hopping for IEEE
            802.11."
    ::= { dot11Groups 10 }

dot11PhyDSSSComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CurrentChannel, dot11CCAModeSupported,
                 dot11CurrentCCAMode, dot11EDThreshold }
        STATUS current
        DESCRIPTION
            "Attributes that configure the DSSS for IEEE 802.11."
    ::= { dot11Groups 11 }

dot11PhyIRComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CCAWatchdogTimerMax, dot11CCAWatchdogCountMax,
                 dot11CCAWatchdogTimerMin, dot11CCAWatchdogCountMin }
        STATUS current
        DESCRIPTION
            "Attributes that configure the baseband IR for IEEE 802.11."
    ::= { dot11Groups 12 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1101 of 1232

```
dot11PhyRegDomainsSupportGroup OBJECT-GROUP
        OBJECTS { dot11RegDomainsSupportedTable }
        STATUS current
        DESCRIPTION
            "Attributes that specify the supported Regulation Domains."
    ::= { dot11Groups 13 }

dot11PhyAntennasListGroup OBJECT-GROUP
       OBJECTS { dot11SupportedTxAntenna,
                 dot11SupportedRxAntenna, dot11DiversitySelectionRx }
        STATUS current
        DESCRIPTION
            "Attributes that specify the supported Regulation Domains."
    ::= { dot11Groups 14 }

dot11PhyRateGroup OBJECT-GROUP
       OBJECTS { dot11SupportedDataRatesTxValue,
                 dot11SupportedDataRatesRxValue }
        STATUS current
        DESCRIPTION
            "Attributes for Data Rates for IEEE 802.11."
    ::= { dot11Groups 15 }

dot11CountersGroup OBJECT-GROUP
       OBJECTS { dot11TransmittedFragmentCount,
                 dot11MulticastTransmittedFrameCount,
                 dot11FailedCount, dot11ReceivedFragmentCount,
                 dot11MulticastReceivedFrameCount,
                 dot11FCSErrorCount,
                 dot11WEPUndecryptableCount,
                 dot11TransmittedFrameCount }
        STATUS deprecated
        DESCRIPTION
            "Attributes from the dot11CountersGroup that are not described
            in the dot11MACStatistics group.  These objects are
            mandatory."
    ::= { dot11Groups 16 }

dot11NotificationGroup NOTIFICATION-GROUP
       NOTIFICATIONS { dot11Disassociate,
                 dot11Deauthenticate,
                 dot11AuthenticateFail }
       STATUS current
       DESCRIPTION
              "IEEE 802.11 notifications"
   ::= { dot11Groups 17 }

dot11SMTbase2 OBJECT-GROUP
       OBJECTS { dot11MediumOccupancyLimit,
                 dot11CFPollable,
                 dot11CFPPeriod,
                 dot11CFPMaxDuration,
                 dot11AuthenticationResponseTimeOut,
                 dot11PrivacyOptionImplemented,
                 dot11PowerManagementMode,
                 dot11DesiredSSID, dot11DesiredBSSType,
                 dot11OperationalRateSet,
                 dot11BeaconPeriod, dot11DTIMPeriod,
```


Copyright © 2007 IEEE. All rights reserved.

```
                dot11AssociationResponseTimeOut,
                dot11DisassociateReason,
                dot11DisassociateStation,
                dot11DeauthenticateReason,
                dot11DeauthenticateStation,
                dot11AuthenticateFailStatus,
                dot11AuthenticateFailStation }
        STATUS deprecated
        DESCRIPTION
             "The SMTbase2 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 18 }

dot11PhyOFDMComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CurrentFrequency,
                 dot11TIThreshold,
                 dot11FrequencyBandsSupported,
                 dot11ChannelStartingFactor }
        STATUS deprecated
        DESCRIPTION
            "Attributes that configure the OFDM for IEEE 802.11."
    ::= { dot11Groups 19 }

dot11SMTbase3 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 }
        STATUS deprecated
        DESCRIPTION
             "The SMTbase3 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, when the STA
             is capable of multi-domain operation. This object group should be
             implemented when the multi-domain capability option is implemented."
   ::= { dot11Groups 20 }

dot11MultiDomainCapabilityGroup OBJECT-GROUP
       OBJECTS { dot11FirstChannelNumber,
                 dot11NumberofChannels,
                 dot11MaximumTransmitPowerLevel }
       STATUS current
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1103 of 1232

```
       DESCRIPTION
              "The dot11MultiDomainCapabilityGroup object class provides
              the objects necessary to manage the channels usable by a STA,
              when the multi-domain capability option is implemented."
   ::= { dot11Groups 21 }

dot11PhyFHSSComplianceGroup2 OBJECT-GROUP
       OBJECTS { dot11HopTime, dot11CurrentChannelNumber,
                 dot11MaxDwellTime, dot11CurrentDwellTime, dot11CurrentSet,
                 dot11CurrentPattern, dot11CurrentIndex, dot11EHCCPrimeRadix,
                 dot11EHCCNumberofChannelsFamilyIndex,
                 dot11EHCCCapabilityImplemented, dot11EHCCCapabilityEnabled,
                 dot11HopAlgorithmAdopted, dot11RandomTableFlag,
                 dot11NumberofHoppingSets, dot11HopModulus,
                 dot11HopOffset, dot11RandomTableFieldNumber }
        STATUS current
        DESCRIPTION
            "Attributes that configure the Frequency Hopping for IEEE
            802.11 when multi-domain capability option is implemented."
    ::= { dot11Groups 22 }

dot11PhyHRDSSSComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CurrentChannel, dot11CCAModeSupported,
                 dot11CurrentCCAMode, dot11EDThreshold,
                 dot11ShortPreambleOptionImplemented,
                 dot11PBCCOptionImplemented, dot11ChannelAgilityPresent,
                 dot11ChannelAgilityEnabled, dot11HRCCAModeSupported }
        STATUS current
        DESCRIPTION
              "Attributes that configure the HRDSSS for IEEE 802.11."
    ::= { dot11Groups 23 }

dot11PhyERPComplianceGroup OBJECT-GROUP
       OBJECTS { dot11CurrentChannel,
                 dot11ShortPreambleOptionImplemented,
                 dot11ChannelAgilityPresent,
                 dot11ChannelAgilityEnabled,
                 dot11DSSSOFDMOptionImplemented,
                 dot11DSSSOFDMOptionEnabled,
                 dot11PBCCOptionImplemented,
                 dot11ERPPBCCOptionImplemented,
                 dot11ShortSlotTimeOptionImplemented,
                 dot11ShortSlotTimeOptionEnabled }
        STATUS current
        DESCRIPTION
              "Attributes that configure the ERP."
    ::= { dot11Groups 24 }

dot11RSNAadditions OBJECT-GROUP
       OBJECTS { dot11RSNAEnabled,
              dot11RSNAPreauthenticationEnabled }
       STATUS current
       DESCRIPTION
              "This object class provides the objects from the IEEE 802.11 MIB
              required to manage RSNA functionality. Note that additional objects
              for managing this functionality are located in the IEEE 802.11 RSN
              MIB."
   ::= { dot11Groups 25 }
```


Copyright © 2007 IEEE. All rights reserved.

```
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 deprecated
       DESCRIPTION
              "The SMTbase4 object class provides the necessary support at the
              IEEE 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 }

dot11RegulatoryClassesGroup OBJECT-GROUP
       OBJECTS { dot11RegulatoryClass,
              dot11CoverageClass }
       STATUS current
       DESCRIPTION
              "Attributes that configure the OFDM for IEEE 802.11 in many regula-
              tory domains."
   ::= { dot11Groups 29 }

dot11SMTbase5 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,
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1105 of 1232

```
              dot11SpectrumManagementImplemented,
              dot11SpectrumManagementRequired,
              dot11RSNAOptionImplemented,
              dot11RegulatoryClassesImplemented,
              dot11RegulatoryClassesRequired }
       STATUS current
       DESCRIPTION
              "The SMTbase5 object class provides the necessary support at the STA
              to manage the processes in the STA so that the STA may work cooper-
              atively as a part of an IEEE 802.11 network, when the STA is capable
              of multidomain operation. This object group should be implemented
              when the multidomain capability option is implemented."
   ::= { dot11Groups 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,
              dot11QosCountersTable }
       STATUS current
       DESCRIPTION
```


Copyright © 2007 IEEE. All rights reserved.

```
          "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 }

dot11PhyOFDMComplianceGroup2 OBJECT-GROUP
       OBJECTS { dot11CurrentFrequency,
                 dot11TIThreshold,
                 dot11FrequencyBandsSupported,
                 dot11ChannelStartingFactor,
                 dot11FiveMHzOperationImplemented,
                 dot11TenMHzOperationImplemented,
                 dot11TwentyMHzOperationImplemented,
                 dot11PhyOFDMChannelWidth }
        STATUS current
        DESCRIPTION
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1107 of 1232

```
            "Attributes that configure the OFDM for IEEE 802.11."
    ::= { dot11Groups 35}

-- ************************************************************************
-- * dot11RSNAConfig TABLE (RSNA and TSN)
-- ************************************************************************

dot11RSNAConfigTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11RSNAConfigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The table containing RSNA configuration objects."
   ::= { dot11smt 9 }

dot11RSNAConfigEntry OBJECT-TYPE
       SYNTAX Dot11RSNAConfigEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry in the dot11RSNAConfigTable."
       INDEX { ifIndex }
   ::= { dot11RSNAConfigTable 1 }

Dot11RSNAConfigEntry ::=
       SEQUENCE {
              dot11RSNAConfigVersion                          Integer32,
              dot11RSNAConfigPairwiseKeysSupported            Unsigned32,
              dot11RSNAConfigGroupCipher                      OCTET STRING,
              dot11RSNAConfigGroupRekeyMethod                 INTEGER,
              dot11RSNAConfigGroupRekeyTime                   Unsigned32,
              dot11RSNAConfigGroupRekeyPackets                Unsigned32,
              dot11RSNAConfigGroupRekeyStrict                 TruthValue,
              dot11RSNAConfigPSKValue                         OCTET STRING,
              dot11RSNAConfigPSKPassPhrase                    DisplayString,
              dot11RSNAConfigGroupUpdateCount                 Unsigned32,
              dot11RSNAConfigPairwiseUpdateCount              Unsigned32,
              dot11RSNAConfigGroupCipherSize                  Unsigned32,
              dot11RSNAConfigPMKLifetime                      Unsigned32,
              dot11RSNAConfigPMKReauthThreshold               Unsigned32,
              dot11RSNAConfigNumberOfPTKSAReplayCounters      INTEGER,
              dot11RSNAConfigSATimeout                        Unsigned32,
              dot11RSNAAuthenticationSuiteSelected            OCTET STRING,
              dot11RSNAPairwiseCipherSelected                 OCTET STRING,
              dot11RSNAGroupCipherSelected                    OCTET STRING,
              dot11RSNAPMKIDUsed                              OCTET STRING,
              dot11RSNAAuthenticationSuiteRequested           OCTET STRING,
              dot11RSNAPairwiseCipherRequested                OCTET STRING,
              dot11RSNAGroupCipherRequested                   OCTET STRING,
              dot11RSNATKIPCounterMeasuresInvoked             Unsigned32,
              dot11RSNA4WayHandshakeFailures                  Unsigned32,
              dot11RSNAConfigNumberOfGTKSAReplayCounters      INTEGER,
              dot11RSNAConfigSTKKeysSupported                 Unsigned32,
              dot11RSNAConfigSTKCipher                        OCTET STRING,
              dot11RSNAConfigSTKRekeyTime                     Unsigned32,
              dot11RSNAConfigSMKUpdateCount                   Unsigned32,
              dot11RSNAConfigSTKCipherSize                    Unsigned32,
              dot11RSNAConfigSMKLifetime                      Unsigned32,
              dot11RSNAConfigSMKReauthThreshold               Unsigned32,
```


Copyright © 2007 IEEE. All rights reserved.

```
            dot11RSNAConfigNumberOfSTKSAReplayCounters     INTEGER,
            dot11RSNAPairwiseSTKSelected                   OCTET STRING,
            dot11RSNASMKHandshakeFailures                  Unsigned32 }

-- dot11RSNAConfigEntry 1 has been deprecated.

dot11RSNAConfigVersion OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The highest RSNA version this entity supports. See 7.3.2.9."
   ::= { dot11RSNAConfigEntry 2 }

dot11RSNAConfigPairwiseKeysSupported OBJECT-TYPE
      SYNTAX Unsigned32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "This object indicates how many pairwise keys the entity supports
            for RSNA."
   ::= { dot11RSNAConfigEntry 3 }

dot11RSNAConfigGroupCipher OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(4))
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This object indicates the group cipher suite selector the entity
            must use. The group cipher suite in the RSN Information Element
            shall take its value from this variable. It consists of an OUI (the
            first 3 octets) and a cipher suite identifier (the last octet)."
   ::= { dot11RSNAConfigEntry 4 }

dot11RSNAConfigGroupRekeyMethod OBJECT-TYPE
      SYNTAX INTEGER { disabled(1), timeBased(2), packetBased(3), timepacket-
            Based(4) }
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This object selects a mechanism for rekeying the RSNA GTK. The
            default is time-based, once per day. Rekeying the GTK is only
            applicable to an entity acting in the Authenticator role (an AP in
            an ESS)."
      DEFVAL { timeBased }
   ::= { dot11RSNAConfigEntry 5 }

dot11RSNAConfigGroupRekeyTime OBJECT-TYPE
      SYNTAX Unsigned32 (1..4294967295)
      UNITS "seconds"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "The time in seconds after which the RSNA GTK shall be refreshed.
            The timer shall start at the moment the GTK was set using the MLME-
            SETKEYS.request primitive."
      DEFVAL { 86400 } -- once per day
   ::= { dot11RSNAConfigEntry 6 }
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1109 of 1232

```
dot11RSNAConfigGroupRekeyPackets OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       UNITS "1000 packets"
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "A packet count (in 1000s of packets) after which the RSNA GTK
              shall be refreshed. The packet counter shall start at the moment
              the GTK was set using the MLME-SETKEYS.request primitive and it
              shall count all packets encrypted using the current GTK."
   ::= { dot11RSNAConfigEntry 7 }

dot11RSNAConfigGroupRekeyStrict OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This object signals that the GTK shall be refreshed whenever a STA
              leaves the BSS that possesses the GTK."
   ::= { dot11RSNAConfigEntry 8 }

dot11RSNAConfigPSKValue OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(32))
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The PSK for when RSNA in PSK mode is the selected AKM suite. In
              that case, the PMK will obtain its value from this object.

              This object is logically write-only. Reading this variable shall
              return unsuccessful status or null or zero."
   ::= { dot11RSNAConfigEntry 9 }

dot11RSNAConfigPSKPassPhrase OBJECT-TYPE
       SYNTAX DisplayString
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The PSK, for when RSNA in PSK mode is the selected AKM suite, is
              configured by dot11RSNAConfigPSKValue.

              An alternative manner of setting the PSK uses the password-to-key
              algorithm defined in H.4. This variable provides a means to enter a
              pass-phrase. When this object is written, the RSNA entity shall use
              the password-to-key algorithm specified in H.4 to derive a pre-
              shared and populate dot11RSNAConfigPSKValue with this key.
              This object is logically write-only. Reading this variable shall
              return unsuccessful status or null or zero."
   ::= { dot11RSNAConfigEntry 10 }

-- dot11RSNAConfigEntry 11 and dot11RSNAConfigEntry 12 have been
-- deprecated.

dot11RSNAConfigGroupUpdateCount OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
```


Copyright © 2007 IEEE. All rights reserved.

```
              "The number of times Message 1 in the RSNA Group Key Handshake will
              be retried per GTK Handshake attempt."
       DEFVAL { 3 } --
   ::= { dot11RSNAConfigEntry 13 }

dot11RSNAConfigPairwiseUpdateCount OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The number of times Message 1 and Message 3 in the RSNA 4-Way Hand-
              shake will be retried per 4-Way Handshake attempt."
       DEFVAL { 3 } --
   ::= { dot11RSNAConfigEntry 14 }

dot11RSNAConfigGroupCipherSize OBJECT-TYPE
       SYNTAX Unsigned32 (0..4294967295)
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This object indicates the length in bits of the group cipher key."
   ::= { dot11RSNAConfigEntry 15 }

dot11RSNAConfigPMKLifetime OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       UNITS "seconds"
       MAX-ACCESSread-write
       STATUS current
       DESCRIPTION
              "The maximum lifetime of a PMK in the PMK cache."
       DEFVAL { 43200 } --
   ::= { dot11RSNAConfigEntry 16 }

dot11RSNAConfigPMKReauthThreshold OBJECT-TYPE
       SYNTAX Unsigned32 (1..100)
       UNITS "percentage"
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The percentage of the PMK lifetime that should expire before an
              IEEE 802.1X reauthentication occurs."
       DEFVAL { 70 } --
   ::= { dot11RSNAConfigEntry 17 }

dot11RSNAConfigNumberOfPTKSAReplayCounters OBJECT-TYPE
       SYNTAX INTEGER
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Specifies the number of PTKSA replay counters per association:
                     0 –> 1 replay counter,
                     1 –> 2 replay counters,
                     2 –> 4 replay counters,
                     3 –> 16 replay counters"
   ::= { dot11RSNAConfigEntry 18 }

dot11RSNAConfigSATimeout OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       UNITS "seconds"
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1111 of 1232

```
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The maximum time a security association shall take to set up."
       DEFVAL { 60 } --
   ::= { dot11RSNAConfigEntry 19 }

dot11RSNAAuthenticationSuiteSelected OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last AKM suite negotiated."
   ::= { dot11RSNAConfigEntry 20 }

dot11RSNAPairwiseCipherSelected OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last pairwise cipher negotiated."
   ::= { dot11RSNAConfigEntry 21 }

dot11RSNAGroupCipherSelected OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last group cipher negotiated."
   ::= { dot11RSNAConfigEntry 22 }

dot11RSNAPMKIDUsed OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(16))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last PMKID used in the last 4-Way Handshake."
   ::= { dot11RSNAConfigEntry 23 }

dot11RSNAAuthenticationSuiteRequested OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last AKM suite requested."
   ::= { dot11RSNAConfigEntry 24 }

dot11RSNAPairwiseCipherRequested OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last pairwise cipher requested."
   ::= { dot11RSNAConfigEntry 25 }

dot11RSNAGroupCipherRequested OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
```


Copyright © 2007 IEEE. All rights reserved.

```
        DESCRIPTION
                "The selector of the last group cipher requested."
    ::= { dot11RSNAConfigEntry 26 }

dot11RSNATKIPCounterMeasuresInvoked OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "Counts the number of times that a TKIP MIC failure occurred two
                times within 60 s and TKIP countermeasures were invoked. This
                attribute counts both local and remote MIC failure events reported
                to this STA. It increments every time TKIP countermeasures are
                invoked"
    ::= { dot11RSNAConfigEntry 27 }

dot11RSNA4WayHandshakeFailures OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "Counts the number of 4-Way Handshake failures."
    ::= { dot11RSNAConfigEntry 28 }

dot11RSNAConfigNumberOfGTKSAReplayCounters OBJECT-TYPE
        SYNTAX INTEGER
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "Specifies the number of GTKSA replay counters per association:
                        0 –> 1 replay counter,
                        1 –> 2 replay counters,
                        2 –> 4 replay counters,
                        3 –> 16 replay counters"
    ::= { dot11RSNAConfigEntry 29 }

dot11RSNAConfigSTKKeysSupported OBJECT-TYPE
        SYNTAX Unsigned32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This object indicates how many STK keys the entity supports for
                RSNA."
    ::= { dot11RSNAConfigEntry 30 }

dot11RSNAConfigSTKCipher OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This object specifies the ciphersuite used by the STK for a DLS
                link."
    ::= { dot11RSNAConfigEntry 31}


dot11RSNAConfigSTKRekeyTime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        UNITS "seconds"
        MAX-ACCESS read-write
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1113 of 1232

```
        STATUS current
        DESCRIPTION
                "The time in seconds after which an RSNA STK shall be refreshed.
                The timer shall start at the moment the STK was set using the MLM-
                ESETKEYS. request primitive."
        DEFVAL { 86400 } -- once per day
    ::= { dot11RSNAConfigEntry 32 }

dot11RSNAConfigSMKUpdateCount OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "The number of times Message 1 in the RSNA SMK Handshake will be
                retried per SMK Handshake attempt."
        DEFVAL { 3 }
    ::= { dot11RSNAConfigEntry 33 }

dot11RSNAConfigSTKCipherSize OBJECT-TYPE
        SYNTAX Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This object indicates the length in bits of the STK cipher key."
    ::= { dot11RSNAConfigEntry 34 }

dot11RSNAConfigSMKLifetime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        UNITS "seconds"
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "The maximum lifetime of an SMK in the SMK cache."
        DEFVAL { 43200 }
    ::= { dot11RSNAConfigEntry 35 }

dot11RSNAConfigSMKReauthThreshold OBJECT-TYPE
        SYNTAX Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "This attribute indicates the number of seconds for which an SMK
                authentication is valid. A new SMK authentication must be completed
                successfully before the number of seconds indicated by this
                attribute elapse, from the prior authentication, before the STAs
                become unauthenticated."
    ::= { dot11RSNAConfigEntry 36 }

dot11RSNAConfigNumberOfSTKSAReplayCounters OBJECT-TYPE
        SYNTAX INTEGER
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "Specifies the number of STKSA replay counters per association:
                0 -> 1 replay counter,
                1 -> 2 replay counters,
                2 -> 4 replay counters,
                3 -> 16 replay counters"
    ::= { dot11RSNAConfigEntry 37 }
```


Copyright © 2007 IEEE. All rights reserved.

```
dot11RSNAPairwiseSTKSelected OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of the last STK cipher negotiated."
   ::= { dot11RSNAConfigEntry 38 }

dot11RSNASMKHandshakeFailures OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "Counts the number of SMK Handshake failures."
   ::= { dot11RSNAConfigEntry 39 }

-- ****************************************************************
-- * End of dot11RSNAConfig TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11RSNAConfigPairwiseCiphers TABLE
-- ****************************************************************

dot11RSNAConfigPairwiseCiphersTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11RSNAConfigPairwiseCiphersEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table lists the pairwise ciphers supported by this entity. It
              allows enabling and disabling of each pairwise cipher by network
              management. The pairwise cipher suite list in the RSN Information
              Element is formed using the information in this table."
   ::= { dot11smt 10 }

dot11RSNAConfigPairwiseCiphersEntry OBJECT-TYPE
       SYNTAX Dot11RSNAConfigPairwiseCiphersEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The table entry, indexed by the interface index (or all inter-
              faces) and the pairwise cipher."
       INDEX { dot11RSNAConfigIndex, dot11RSNAConfigPairwiseCipherIndex }
   ::= { dot11RSNAConfigPairwiseCiphersTable 1 }

Dot11RSNAConfigPairwiseCiphersEntry ::=
       SEQUENCE {
              dot11RSNAConfigPairwiseCipherIndex      Unsigned32,
              dot11RSNAConfigPairwiseCipher           OCTET STRING,
              dot11RSNAConfigPairwiseCipherEnabled    TruthValue,
              dot11RSNAConfigPairwiseCipherSize       Unsigned32  }

dot11RSNAConfigPairwiseCipherIndex OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The auxiliary index into the dot11RSNAConfigPairwiseCiphersTable."
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1115 of 1232

```
   ::= { dot11RSNAConfigPairwiseCiphersEntry 1 }

dot11RSNAConfigPairwiseCipher OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of a supported pairwise cipher. It consists of an OUI
              (the first 3 octets) and a cipher suite identifier (the last
              octet)."
   ::= { dot11RSNAConfigPairwiseCiphersEntry 2 }

dot11RSNAConfigPairwiseCipherEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This object enables or disables the pairwise cipher."
   ::= { dot11RSNAConfigPairwiseCiphersEntry 3 }

dot11RSNAConfigPairwiseCipherSize OBJECT-TYPE
       SYNTAX Unsigned32 (0..4294967295)
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "This object indicates the length in bits of the pairwise cipher
              key. This should be 256 for TKIP and 128 for CCMP."
   ::= { dot11RSNAConfigPairwiseCiphersEntry 4 }

-- ****************************************************************
-- * End of dot11RSNAConfigPairwiseCiphers TABLE
-- ****************************************************************

-- ****************************************************************
-- * dot11RSNAConfigAuthenticationSuites TABLE
-- ****************************************************************

dot11RSNAConfigAuthenticationSuitesTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11RSNAConfigAuthenticationSuitesEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table lists the AKM suites supported by this entity. Each AKM
              suite can be individually enabled and disabled. The AKM suite list
              in the RSN information element is formed using the information in
              this table."
   ::= { dot11smt 11 }

dot11RSNAConfigAuthenticationSuitesEntry OBJECT-TYPE
       SYNTAX Dot11RSNAConfigAuthenticationSuitesEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry (row) in the dot11RSNAConfigAuthenticationSuitesTable."
       INDEX { dot11RSNAConfigAuthenticationSuiteIndex }
   ::= { dot11RSNAConfigAuthenticationSuitesTable 1 }

Dot11RSNAConfigAuthenticationSuitesEntry ::=
       SEQUENCE {
```


Copyright © 2007 IEEE. All rights reserved.

```
              dot11RSNAConfigAuthenticationSuiteIndex       Unsigned32,
              dot11RSNAConfigAuthenticationSuite            OCTET STRING,
              dot11RSNAConfigAuthenticationSuiteEnabled     TruthValue }

dot11RSNAConfigAuthenticationSuiteIndex OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The auxiliary variable used as an index into the
              dot11RSNAConfigAuthenticationSuitesTable."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 1 }

dot11RSNAConfigAuthenticationSuite OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The selector of an AKM suite. It consists of an OUI (the first 3
              octets) and a cipher suite identifier (the last octet)."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 2 }

dot11RSNAConfigAuthenticationSuiteEnabled OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "This variable indicates whether the corresponding AKM suite is
              enabled/disabled."
   ::= { dot11RSNAConfigAuthenticationSuitesEntry 3 }

-- ************************************************************************
-- * End of dot11RSNAConfigAuthenticationSuites TABLE
-- ************************************************************************

-- ************************************************************************
-- * dot11RSNAStats TABLE
-- ************************************************************************

dot11RSNAStatsTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Dot11RSNAStatsEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table maintains per-STA statistics in an RSN. The entry with
              dot11RSNAStatsSTAAddress set to FF-FF-FF-FF-FF-FF shall contain
              statistics for broadcast/multicast traffic."
   ::= { dot11smt 12 }

dot11RSNAStatsEntry OBJECT-TYPE
       SYNTAX Dot11RSNAStatsEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An entry in the dot11RSNAStatsTable."
       INDEX { dot11RSNAConfigIndex, dot11RSNAStatsIndex }
   ::= { dot11RSNAStatsTable 1 }

Dot11RSNAStatsEntry ::=
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1117 of 1232

```
       SEQUENCE {
              dot11RSNAStatsIndex                    Unsigned32,
              dot11RSNAStatsSTAAddress               MacAddress,
              dot11RSNAStatsVersion                  Unsigned32,
              dot11RSNAStatsSelectedPairwiseCipher   OCTET STRING,
              dot11RSNAStatsTKIPICVErrors            Counter32,
              dot11RSNAStatsTKIPLocalMICFailures     Counter32,
              dot11RSNAStatsTKIPRemoteMICFailures    Counter32,
              dot11RSNAStatsCCMPReplays              Counter32,
              dot11RSNAStatsCCMPDecryptErrors        Counter32,
              dot11RSNAStatsTKIPReplays              Counter32 }

dot11RSNAStatsIndex OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An auxiliary index into the dot11RSNAStatsTable."
   ::= { dot11RSNAStatsEntry 1 }

dot11RSNAStatsSTAAddress OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The MAC address of the STA to which the statistics in this
              conceptual row belong."
   ::= { dot11RSNAStatsEntry 2 }

dot11RSNAStatsVersion OBJECT-TYPE
       SYNTAX Unsigned32 (1..4294967295)
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The RSNA version with which the STA associated."
   ::= { dot11RSNAStatsEntry 3 }

dot11RSNAStatsSelectedPairwiseCipher OBJECT-TYPE
       SYNTAX OCTET STRING (SIZE(4))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The pairwise cipher suite Selector (as defined in 7.3.29.1) used
              during association, in transmission order."
   ::= { dot11RSNAStatsEntry 4 }

dot11RSNAStatsTKIPICVErrors OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Counts the number of TKIP ICV errors encountered when decrypting
              packets for the STA."
   ::= { dot11RSNAStatsEntry 5 }

dot11RSNAStatsTKIPLocalMICFailures OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
```


Copyright © 2007 IEEE. All rights reserved.

```
       DESCRIPTION
              "Counts the number of MIC failures encountered when checking the
              integrity of packets received from the STA at this entity."
   ::= { dot11RSNAStatsEntry 6 }

dot11RSNAStatsTKIPRemoteMICFailures OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Counts the number of MIC failures encountered by the STA identi-
              fied by dot11StatsSTAAddress and reported back to this entity."
   ::= { dot11RSNAStatsEntry 7 }

dot11RSNAStatsCCMPReplays OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The number of received CCMP MPDUs discarded by the replay
              mechanism."
   ::= { dot11RSNAStatsEntry 8 }

dot11RSNAStatsCCMPDecryptErrors OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The number of received MPDUs discarded by the CCMP decryption
              algorithm."
   ::= { dot11RSNAStatsEntry 9 }

dot11RSNAStatsTKIPReplays OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Counts the number of TKIP replay errors detected."
   ::= { dot11RSNAStatsEntry 10 }

-- ************************************************************************
-- * End of dot11RSNAStats TABLE
-- ************************************************************************

-- ************************************************************************
-- * Conformance information - RSN
-- ************************************************************************


-- ************************************************************************
-- * Compliance Statements - RSN
-- ************************************************************************
dot11RSNCompliance MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
              "The compliance statement for SNMPv2 entities that implement the
              IEEE 802.11 RSN MIB."
       MODULE -- this module
       MANDATORY-GROUPS {
```

Copyright © 2007 IEEE. All rights reserved.

```
              dot11RSNBase }

-- OPTIONAL-GROUPS {dot11RSNPMKcachingGroup }
::= { dot11Compliances 2 }

-- ************************************************************************
-- * Groups - units of conformance - RSN
-- ************************************************************************

dot11RSNBase OBJECT-GROUP
       OBJECTS {
              dot11RSNAConfigVersion,
              dot11RSNAConfigPairwiseKeysSupported,
              dot11RSNAConfigGroupCipher,
              dot11RSNAConfigGroupRekeyMethod,
              dot11RSNAConfigGroupRekeyTime,
              dot11RSNAConfigGroupRekeyPackets,
              dot11RSNAConfigGroupRekeyStrict,
              dot11RSNAConfigPSKValue,
              dot11RSNAConfigPSKPassPhrase,
              dot11RSNAConfigGroupUpdateCount,
              dot11RSNAConfigPairwiseUpdateCount,
              dot11RSNAConfigGroupCipherSize,
              dot11RSNAConfigPairwiseCipher,
              dot11RSNAConfigPairwiseCipherEnabled,
              dot11RSNAConfigPairwiseCipherSize,
              dot11RSNAConfigAuthenticationSuite,
              dot11RSNAConfigAuthenticationSuiteEnabled,
              dot11RSNAConfigNumberOfPTKSAReplayCounters,
              dot11RSNAConfigSATimeout,
              dot11RSNAConfigNumberOfGTKSAReplayCounters,
              dot11RSNAAuthenticationSuiteSelected,
              dot11RSNAPairwiseCipherSelected,
              dot11RSNAGroupCipherSelected,
              dot11RSNAPMKIDUsed,
              dot11RSNAAuthenticationSuiteRequested,
              dot11RSNAPairwiseCipherRequested,
              dot11RSNAGroupCipherRequested,
              dot11RSNAStatsSTAAddress,
              dot11RSNAStatsVersion,
              dot11RSNAStatsSelectedPairwiseCipher,
              dot11RSNAStatsTKIPICVErrors,
              dot11RSNAStatsTKIPLocalMICFailures,
              dot11RSNAStatsTKIPRemoteMICFailures,
              dot11RSNAStatsTKIPCounterMeasuresInvoked,
              dot11RSNAStatsCCMPReplays,
              dot11RSNAStatsCCMPDecryptErrors,
              dot11RSNAStatsTKIPReplays,
              dot11RSNAStats4WayHandshakeFailures,
              dot11RSNAConfigSTKKeysSupported,
              dot11RSNAConfigSTKCipher,
              dot11RSNAConfigSTKRekeyTime,
              dot11RSNAConfigSTKUpdateCount,
              dot11RSNAConfigSTKCipherSize,
              dot11RSNAConfigNumberOfSTKSAReplayCounters,
              dot11RSNAPairwiseSTKSelected,
              dot11RSNASMKHandshakeFailures }
       STATUS current
       DESCRIPTION
```


Copyright © 2007 IEEE. All rights reserved.

```
              "The dot11RSNBase object class provides the necessary support for
              managing RSNA functionality in the STA."
   ::= { dot11Groups 27 }

dot11RSNPMKcachingGroup OBJECT-GROUP
       OBJECTS {
              dot11RSNAConfigPMKLifetime,
              dot11RSNAConfigPMKReauthThreshold
              }
       STATUS current
       DESCRIPTION
              "The dot11RSNPMKcachingGroup object class provides the necessary
              support for managing PMK caching functionality in the STA"
   ::= { dot11Groups 28 }

dot11RSNSMKcachingGroup OBJECT-GROUP
       OBJECTS {
              dot11RSNAConfigSMKLifetime,
              dot11RSNAConfigSMKReauthThreshold
              }
       STATUS current
       DESCRIPTION
              "The dot11RSNSMKcachingGroup object class provides the necessary
              support for managing SMK caching functionality in the STA."
   ::= { dot11Groups 29 }

-- ************************************************************************
-- *   End of 802.11 MIB
-- ************************************************************************

END
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1121 of 1232

Copyright © 2007 IEEE. All rights reserved.

# Annex E

(informative)

## (Reserved for future use)

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1123 of 1232

Copyright © 2007 IEEE. All rights reserved.

# Annex F

(informative)

# High Rate PHY/FH interoperability

The Channel Agility option described in 18.4.6.7 provides for IEEE 802.11 FH PHY interoperability with the High Rate PHY. The FH patterns, as defined within this annex, enable synchronization with an FH-PHY-compliant BSS in North America and most of Europe. In addition, CCA requirements on a High Rate STA using this mode provide for CCA detection of 1 MHz wide FH signals within the wideband DS channel selected. FH PHY STAs operating in mixed mode FH/DS environments are advised to use similar cross PHY CCA mechanisms. The FH (Channel Agility) and cross CCA mechanisms provide the basic mechanisms to enable coexistence and interoperability.

The MAC elements include both DS and FH elements in Beacon frames and Probe Response frames when the Channel Agility option is turned on. Added capability fields indicate the ability to support the Channel Agility option and to indicate whether the option is turned on. These fields allow synchronization to the hopping sequence and timing, identification of what modes are being used within a BSS when joining on either High Rate or FHSS sides, and rejection of an association request in some cases.

Interoperability within an infrastructure BSS can be achieved, as an example, using a virtual dual AP. A virtual dual AP is defined, for purposes of discussion, as two logically separate APs that exist within a single physical AP with a single radio (one transmit and one receive path). Both FHSS and High Rate logical APs send out their own Beacon frames, DTIMs, and other nondirected packets. The two sides interact in the sharing of the medium and the AP's processor and radio. Addressing and association issues may be handled in one of several ways and are left as an implementation choice.

Minimal interoperability with a nonhopping High Rate or legacy DSSS is provided by the use of a channel at least 1/7 or more of the time. While throughput would be significantly reduced by having a channel only 1/7 of the time, connection and minimal throughput can be provided.

## F.1 Additional CCA recommendations

When the FH option is utilized, the HR/DSSS PHY should provide the CCA capability to detect 1 MHz wide FH PHY signals operating within the wideband DS channel at levels 10 dB higher than that specified in 18.4.8.4 for wideband HR/DSSS signals. This is in addition to the primary CCA requirements in 18.4.8.4. A timeout mechanism to avoid excessive deferral to constant CW or other non-IEEE-802.11 type signals is allowed.

FH PHY STAs operating in mixed environments should provide similar CCA mechanisms to detect wideband DSSS signals at levels specified in 18.4.8.4, but measured within a 1 MHz bandwidth. Signal levels measured in a full DSSS channel will be generally 10 dB or higher.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1125 of 1232

Copyright © 2007 IEEE. All rights reserved.

# Annex G

(informative)

# An example of encoding a frame for OFDM PHY

## G.1 Introduction

The purpose of this annex is to show an example of encoding a frame for the OFDM PHY, as described in Clause 17. This example covers all the encoding details defined by this standard.

The encoding illustration goes through the following stages:

a) Generating the short training sequence section of the preamble;
b) Generating the long preamble sequence section of the preamble;
c) Generating the SIGNAL field bits;
d) Coding and interleaving the SIGNAL field bits;
e) Mapping the SIGNAL field into frequency domain;
f) Pilot insertion;
g) Transforming into time domain;
h) Delineating the data octet stream into a bit stream;
i) Prepending the SERVICE field and adding the pad bits, thus forming the DATA;
j) Scrambling and zeroing the tail bits;
k) Encoding the DATA with a convolutional encoder and puncturing;
l) Mapping into complex 16-QAM symbols;
m) Pilot insertion;
n) Transforming from frequency to time and adding a circular prefix;
o) Concatenating the OFDM symbols into a single, time-domain signal.

In the description of time domain waveforms, a complex baseband signal at 20 Msample/s shall be used.

This example uses the 36 Mb/s data rate and a message of 100 octets. These parameters are chosen in order to illustrate as many nontrivial aspects of the processing as possible.

a) Use of several bits per symbol (4 in this case);
b) Puncturing of the convolutional code;
c) Interleaving, which uses the LSB–MSB swapping stage;
d) Scrambling of the pilot subcarriers.

## G.2 The message

The message being encoded consists of the first 72 characters of the well-known “Ode to Joy” by F. Schiller:

Joy, bright spark of divinity,
Daughter of Elysium,
Fire-insired we tread
Thy sanctuary.

Copyright © 2007 IEEE. All rights reserved.

Thy magic power re-unites
All that custom has divided,
All men become brothers
Under the sway of thy gentle wings…

The message is converted to ASCII; then it is prepended with an appropriate MAC header and a CRC32 is added. The resulting 100 octets PSDU is shown in Table G.1.

**Table G.1—The message**

| ## | Val | Val | Val | Val | Val |
|---|---|---|---|---|---|
| 1…5 | 04 | 02 | 00 | 2e | 00 |
| 6…10 | 60 | 08 | cd | 37 | a6 |
| 11…15 | 00 | 20 | d6 | 01 | 3c |
| 16…20 | f1 | 00 | 60 | 08 | ad |
| 21…25 | 3b | af | 00 | 00 | 4a |
| 26…30 | 6f | 79 | 2c | 20 | 62 |
| 31…35 | 72 | 69 | 67 | 68 | 74 |
| 36…45 | 20 | 73 | 70 | 61 | 72 |
| 41…45 | 6b | 20 | 6f | 66 | 20 |
| 46…50 | 64 | 69 | 76 | 69 | 6e |
| 51…55 | 69 | 74 | 79 | 2c | 0a |
| 56…60 | 44 | 61 | 75 | 67 | 68 |
| 61…55 | 74 | 65 | 72 | 20 | 6f |
| 66…60 | 66 | 20 | 45 | 6c | 79 |
| 71…55 | 73 | 69 | 75 | 6d | 2c |
| 76…60 | 0a | 46 | 69 | 72 | 65 |
| 81…55 | 2d | 69 | 6e | 73 | 69 |
| 86…60 | 72 | 65 | 64 | 20 | 77 |
| 91…55 | 65 | 20 | 74 | 72 | 65 |
| 96…100 | 61 | da | 57 | 99 | ed |

## G.3 Generation of the preamble

### G.3.1 Generation of the short sequences

The short sequences section of the preamble is described by its frequency domain representation, given in Table G.2.


Copyright © 2007 IEEE. All rights reserved.

**Table G.2—Frequency domain representation of the short sequences**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| –32 | 0.0 | 0.0 | –16 | 1.472 | 1.472 | 0 | 0.0 | 0.0 | 16 | 1.472 | 1.472 |
| –31 | 0.0 | 0.0 | –15 | 0.0 | 0.0 | 1 | 0.0 | 0.0 | 17 | 0.0 | 0.0 |
| –30 | 0.0 | 0.0 | –14 | 0.0 | 0.0 | 2 | 0.0 | 0.0 | 18 | 0.0 | 0.0 |
| –29 | 0.0 | 0.0 | –13 | 0.0 | 0.0 | 3 | 0.0 | 0.0 | 19 | 0.0 | 0.0 |
| –28 | 0.0 | 0.0 | –12 | –1.472 | –1.472 | 4 | –1.472 | –1.472 | 20 | 1.472 | 1.472 |
| –27 | 0.0 | 0.0 | –11 | 0.0 | 0.0 | 5 | 0.0 | 0.0 | 21 | 0.0 | 0.0 |
| –26 | 0.0 | 0.0 | –10 | 0.0 | 0.0 | 6 | 0.0 | 0.0 | 22 | 0.0 | 0.0 |
| –25 | 0.0 | 0.0 | –9 | 0.0 | 0.0 | 7 | 0.0 | 0.0 | 23 | 0.0 | 0.0 |
| –24 | 1.472 | 1.472 | –8 | –1.472 | –1.472 | 8 | –1.472 | –1.472 | 24 | 1.472 | 1.472 |
| –23 | 0.0 | 0.0 | –7 | 0.0 | 0.0 | 9 | 0.0 | 0.0 | 25 | 0.0 | 0.0 |
| –22 | 0.0 | 0.0 | –6 | 0.0 | 0.0 | 10 | 0.0 | 0.0 | 26 | 0.0 | 0.0 |
| –21 | 0.0 | 0.0 | –5 | 0.0 | 0.0 | 11 | 0.0 | 0.0 | 27 | 0.0 | 0.0 |
| –20 | –1.472 | –1.472 | –4 | 1.472 | 1.472 | 12 | 1.472 | 1.472 | 28 | 0.0 | 0.0 |
| –19 | 0.0 | 0.0 | –3 | 0.0 | 0.0 | 13 | 0.0 | 0.0 | 29 | 0.0 | 0.0 |
| –18 | 0.0 | 0.0 | –2 | 0.0 | 0.0 | 14 | 0.0 | 0.0 | 30 | 0.0 | 0.0 |
| –17 | 0.0 | 0.0 | –1 | 0.0 | 0.0 | 15 | 0.0 | 0.0 | 31 | 0.0 | 0.0 |

One period of the IFFT on the contents of Table G.2 is given in Table G.3.

**Table G.3—One period of IFFT of the short sequences**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.046 | 0.046 | 1 | –0.132 | 0.002 | 2 | –0.013 | –0.079 | 3 | 0.143 | –0.013 |
| 4 | 0.092 | 0.000 | 5 | 0.143 | –0.013 | 6 | –0.013 | –0.079 | 7 | –0.132 | 0.002 |
| 8 | 0.046 | 0.046 | 9 | 0.002 | –0.132 | 10 | –0.079 | –0.013 | 11 | –0.013 | 0.143 |
| 12 | 0.000 | 0.092 | 13 | –0.013 | 0.143 | 14 | –0.079 | –0.013 | 15 | 0.002 | –0.132 |
| 16 | 0.046 | 0.046 | 17 | –0.132 | 0.002 | 18 | –0.013 | –0.079 | 19 | 0.143 | –0.013 |
| 20 | 0.092 | 0.000 | 21 | 0.143 | –0.013 | 22 | –0.013 | –0.079 | 23 | –0.132 | 0.002 |
| 24 | 0.046 | 0.046 | 25 | 0.002 | –0.132 | 26 | –0.079 | –0.013 | 27 | –0.013 | 0.143 |
| 28 | 0.000 | 0.092 | 29 | –0.013 | 0.143 | 30 | –0.079 | –0.013 | 31 | 0.002 | –0.132 |
| 32 | 0.046 | 0.046 | 33 | –0.132 | 0.002 | 34 | –0.013 | –0.079 | 35 | 0.143 | –0.013 |
| 36 | 0.092 | 0.000 | 37 | 0.143 | –0.013 | 38 | –0.013 | –0.079 | 39 | –0.132 | 0.002 |
| 40 | 0.046 | 0.046 | 41 | 0.002 | –0.132 | 42 | –0.079 | –0.013 | 43 | –0.013 | 0.143 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.3—One period of IFFT of the short sequences *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 44 | 0.000 | 0.092 | 45 | –0.013 | 0.143 | 46 | –0.079 | –0.013 | 47 | 0.002 | –0.132 |
| 48 | 0.046 | 0.046 | 49 | –0.132 | 0.002 | 50 | –0.013 | –0.079 | 51 | 0.143 | –0.013 |
| 52 | 0.092 | 0.000 | 53 | 0.143 | –0.013 | 54 | –0.013 | –0.079 | 55 | –0.132 | 0.002 |
| 56 | 0.046 | 0.046 | 57 | 0.002 | –0.132 | 58 | –0.079 | –0.013 | 59 | –0.013 | 0.143 |
| 60 | 0.000 | 0.092 | 61 | –0.013 | 0.143 | 62 | –0.079 | –0.013 | 63 | 0.002 | –0.132 |

The single period of the short training sequence is extended periodically for 161 samples (about 8 μs), and then multiplied by the window function:

$$W(k) = \begin{bmatrix} 0.5 & k = 0 \\ 1 & 1 \le k \le 159 \\ 0.5 & k = 160 \end{bmatrix}$$

The last sample serves as an overlap with the following OFDM symbol. The 161 samples vector is shown in Table G.4. The time-windowing feature illustrated here is not part of the normative specifications.

**Table G.4—Time domain representation of the short sequence**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.023 | 0.023 | 1 | –0.132 | 0.002 | 2 | –0.013 | –0.079 | 3 | 0.143 | –0.013 |
| 4 | 0.092 | 0.000 | 5 | 0.143 | –0.013 | 6 | –0.013 | –0.079 | 7 | –0.132 | 0.002 |
| 8 | 0.046 | 0.046 | 9 | 0.002 | –0.132 | 10 | –0.079 | –0.013 | 11 | –0.013 | 0.143 |
| 12 | 0.000 | 0.092 | 13 | –0.013 | 0.143 | 14 | –0.079 | –0.013 | 15 | 0.002 | –0.132 |
| 16 | 0.046 | 0.046 | 17 | –0.132 | 0.002 | 18 | –0.013 | –0.079 | 19 | 0.143 | –0.013 |
| 20 | 0.092 | 0.000 | 21 | 0.143 | –0.013 | 22 | –0.013 | –0.079 | 23 | –0.132 | 0.002 |
| 24 | 0.046 | 0.046 | 25 | 0.002 | –0.132 | 26 | –0.079 | –0.013 | 27 | –0.013 | 0.143 |
| 28 | 0.000 | 0.092 | 29 | –0.013 | 0.143 | 30 | –0.079 | –0.013 | 31 | 0.002 | –0.132 |
| 32 | 0.046 | 0.046 | 33 | –0.132 | 0.002 | 34 | –0.013 | –0.079 | 35 | 0.143 | –0.013 |
| 36 | 0.092 | 0.000 | 37 | 0.143 | –0.013 | 38 | –0.013 | –0.079 | 39 | –0.132 | 0.002 |
| 40 | 0.046 | 0.046 | 41 | 0.002 | –0.132 | 42 | –0.079 | –0.013 | 43 | –0.013 | 0.143 |
| 44 | 0.000 | 0.092 | 45 | –0.013 | 0.143 | 46 | –0.079 | –0.013 | 47 | 0.002 | –0.132 |
| 48 | 0.046 | 0.046 | 49 | –0.132 | 0.002 | 50 | –0.013 | –0.079 | 51 | 0.143 | –0.013 |
| 52 | 0.092 | 0.000 | 53 | 0.143 | –0.013 | 54 | –0.013 | –0.079 | 55 | –0.132 | 0.002 |
| 56 | 0.046 | 0.046 | 57 | 0.002 | –0.132 | 58 | –0.079 | –0.013 | 59 | –0.013 | 0.143 |
| 60 | 0.000 | 0.092 | 61 | –0.013 | 0.143 | 62 | –0.079 | –0.013 | 63 | 0.002 | –0.132 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.4—Time domain representation of the short sequence *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 64 | 0.046 | 0.046 | 65 | –0.132 | 0.002 | 66 | –0.013 | –0.079 | 67 | 0.143 | –0.013 |
| 68 | 0.092 | 0.000 | 69 | 0.143 | –0.013 | 70 | –0.013 | –0.079 | 71 | –0.132 | 0.002 |
| 72 | 0.046 | 0.046 | 73 | 0.002 | –0.132 | 74 | –0.079 | –0.013 | 75 | –0.013 | 0.143 |
| 76 | 0.000 | 0.092 | 77 | –0.013 | 0.143 | 78 | –0.079 | –0.013 | 79 | 0.002 | –0.132 |
| 80 | 0.046 | 0.046 | 81 | –0.132 | 0.002 | 82 | –0.013 | –0.079 | 83 | 0.143 | –0.013 |
| 84 | 0.092 | 0.000 | 85 | 0.143 | –0.013 | 86 | –0.013 | –0.079 | 87 | –0.132 | 0.002 |
| 88 | 0.046 | 0.046 | 89 | 0.002 | –0.132 | 90 | –0.079 | –0.013 | 91 | –0.013 | 0.143 |
| 92 | 0.000 | 0.092 | 93 | –0.013 | 0.143 | 94 | –0.079 | –0.013 | 95 | 0.002 | –0.132 |
| 96 | 0.046 | 0.046 | 97 | –0.132 | 0.002 | 98 | –0.013 | –0.079 | 99 | 0.143 | –0.013 |
| 100 | 0.092 | 0.000 | 101 | 0.143 | –0.013 | 102 | –0.013 | –0.079 | 103 | –0.132 | 0.002 |
| 104 | 0.046 | 0.046 | 105 | 0.002 | –0.132 | 106 | –0.079 | –0.013 | 107 | –0.013 | 0.143 |
| 108 | 0.000 | 0.092 | 109 | –0.013 | 0.143 | 110 | –0.079 | –0.013 | 111 | 0.002 | –0.132 |
| 112 | 0.046 | 0.046 | 113 | –0.132 | 0.002 | 114 | –0.013 | –0.079 | 115 | 0.143 | –0.013 |
| 116 | 0.092 | 0.000 | 117 | 0.143 | –0.013 | 118 | –0.013 | –0.079 | 119 | –0.132 | 0.002 |
| 120 | 0.046 | 0.046 | 121 | 0.002 | –0.132 | 122 | –0.079 | –0.013 | 123 | –0.013 | 0.143 |
| 124 | 0.000 | 0.092 | 125 | –0.013 | 0.143 | 126 | –0.079 | –0.013 | 127 | 0.002 | –0.132 |
| 128 | 0.046 | 0.046 | 129 | –0.132 | 0.002 | 130 | –0.013 | –0.079 | 131 | 0.143 | –0.013 |
| 132 | 0.092 | 0.000 | 133 | 0.143 | –0.013 | 134 | –0.013 | –0.079 | 135 | –0.132 | 0.002 |
| 136 | 0.046 | 0.046 | 137 | 0.002 | –0.132 | 138 | –0.079 | –0.013 | 139 | –0.013 | 0.143 |
| 140 | 0.000 | 0.092 | 141 | –0.013 | 0.143 | 142 | –0.079 | –0.013 | 143 | 0.002 | –0.132 |
| 144 | 0.046 | 0.046 | 145 | –0.132 | 0.002 | 146 | –0.013 | –0.079 | 147 | 0.143 | –0.013 |
| 148 | 0.092 | 0.000 | 149 | 0.143 | –0.013 | 150 | –0.013 | –0.079 | 151 | –0.132 | 0.002 |
| 152 | 0.046 | 0.046 | 153 | 0.002 | –0.132 | 154 | –0.079 | –0.013 | 155 | –0.013 | 0.143 |
| 156 | 0.000 | 0.092 | 157 | –0.013 | 0.143 | 158 | –0.079 | –0.013 | 159 | 0.002 | –0.132 |
| 160 | 0.023 | 0.023 | | | | | | | | | |

### G.3.2 Generation of the long sequences

The frequency domain representation of the long training sequence part of the preamble is given in Table G.5.

Copyright © 2007 IEEE. All rights reserved.

**Table G.5—Frequency domain representation of the long sequences**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| –32 | 0.000 | 0.000 | -16 | 1.000 | 0.000 | 0 | 0.000 | 0.000 | 16 | 1.000 | 0.000 |
| –31 | 0.000 | 0.000 | -15 | 1.000 | 0.000 | 1 | 1.000 | 0.000 | 17 | –1.000 | 0.000 |
| –30 | 0.000 | 0.000 | -14 | 1.000 | 0.000 | 2 | –1.000 | 0.000 | 18 | –1.000 | 0.000 |
| –29 | 0.000 | 0.000 | -13 | 1.000 | 0.000 | 3 | –1.000 | 0.000 | 19 | 1.000 | 0.000 |
| –28 | 0.000 | 0.000 | -12 | 1.000 | 0.000 | 4 | 1.000 | 0.000 | 20 | –1.000 | 0.000 |
| –27 | 0.000 | 0.000 | -11 | –1.000 | 0.000 | 5 | 1.000 | 0.000 | 21 | 1.000 | 0.000 |
| –26 | 1.000 | 0.000 | -10 | –1.000 | 0.000 | 6 | –1.000 | 0.000 | 22 | –1.000 | 0.000 |
| –25 | 1.000 | 0.000 | -9 | 1.000 | 0.000 | 7 | 1.000 | 0.000 | 23 | 1.000 | 0.000 |
| –24 | –1.000 | 0.000 | -8 | 1.000 | 0.000 | 8 | –1.000 | 0.000 | 24 | 1.000 | 0.000 |
| –23 | –1.000 | 0.000 | -7 | –1.000 | 0.000 | 9 | 1.000 | 0.000 | 25 | 1.000 | 0.000 |
| –22 | 1.000 | 0.000 | -6 | 1.000 | 0.000 | 10 | –1.000 | 0.000 | 26 | 1.000 | 0.000 |
| –21 | 1.000 | 0.000 | -5 | –1.000 | 0.000 | 11 | –1.000 | 0.000 | 27 | 0.000 | 0.000 |
| –20 | –1.000 | 0.000 | -4 | 1.000 | 0.000 | 12 | –1.000 | 0.000 | 28 | 0.000 | 0.000 |
| –19 | 1.000 | 0.000 | -3 | 1.000 | 0.000 | 13 | –1.000 | 0.000 | 29 | 0.000 | 0.000 |
| –18 | –1.000 | 0.000 | -2 | 1.000 | 0.000 | 14 | –1.000 | 0.000 | 30 | 0.000 | 0.000 |
| –17 | 1.000 | 0.000 | -1 | 1.000 | 0.000 | 15 | 1.000 | 0.000 | 31 | 0.000 | 0.000 |

The time domain representation is derived by performing IFFT on the contents of Table G.5, cyclically extending the result to get the cyclic prefix, and then multiplying with the window function given in G.3.1. The resulting 161 points vector is shown in Table G.6. The samples are appended to the short sequence section by overlapping and adding element 160 of Table G.4 to element 0 of Table G.6.

**Table G.6—Time domain representation of the long sequence**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | –0.078 | 0.000 | 1 | 0.012 | –0.098 | 2 | 0.092 | –0.106 | 3 | –0.092 | –0.115 |
| 4 | –0.003 | –0.054 | 5 | 0.075 | 0.074 | 6 | –0.127 | 0.021 | 7 | –0.122 | 0.017 |
| 8 | –0.035 | 0.151 | 9 | –0.056 | 0.022 | 10 | –0.060 | –0.081 | 11 | 0.070 | –0.014 |
| 12 | 0.082 | –0.092 | 13 | –0.131 | –0.065 | 14 | –0.057 | –0.039 | 15 | 0.037 | –0.098 |
| 16 | 0.062 | 0.062 | 17 | 0.119 | 0.004 | 18 | –0.022 | –0.161 | 19 | 0.059 | 0.015 |
| 20 | 0.024 | 0.059 | 21 | –0.137 | 0.047 | 22 | 0.001 | 0.115 | 23 | 0.053 | –0.004 |
| 24 | 0.098 | 0.026 | 25 | –0.038 | 0.106 | 26 | –0.115 | 0.055 | 27 | 0.060 | 0.088 |
| 28 | 0.021 | –0.028 | 29 | 0.097 | –0.083 | 30 | 0.040 | 0.111 | 31 | –0.005 | 0.120 |
| 32 | 0.156 | 0.000 | 33 | –0.005 | –0.120 | 34 | 0.040 | –0.111 | 35 | 0.097 | 0.083 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.6—Time domain representation of the long sequence *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 0.021 | 0.028 | 37 | 0.060 | –0.088 | 38 | –0.115 | –0.055 | 39 | –0.038 | –0.106 |
| 40 | 0.098 | –0.026 | 41 | 0.053 | 0.004 | 42 | 0.001 | –0.115 | 43 | –0.137 | –0.047 |
| 44 | 0.024 | –0.059 | 45 | 0.059 | –0.015 | 46 | –0.022 | 0.161 | 47 | 0.119 | –0.004 |
| 48 | 0.062 | –0.062 | 49 | 0.037 | 0.098 | 50 | –0.057 | 0.039 | 51 | –0.131 | 0.065 |
| 52 | 0.082 | 0.092 | 53 | 0.070 | 0.014 | 54 | –0.060 | 0.081 | 55 | –0.056 | –0.022 |
| 56 | –0.035 | –0.151 | 57 | –0.122 | –0.017 | 58 | –0.127 | –0.021 | 59 | 0.075 | –0.074 |
| 60 | –0.003 | 0.054 | 61 | –0.092 | 0.115 | 62 | 0.092 | 0.106 | 63 | 0.012 | 0.098 |
| 64 | –0.156 | 0.000 | 65 | 0.012 | –0.098 | 66 | 0.092 | –0.106 | 67 | –0.092 | –0.115 |
| 68 | –0.003 | –0.054 | 69 | 0.075 | 0.074 | 70 | –0.127 | 0.021 | 71 | –0.122 | 0.017 |
| 72 | –0.035 | 0.151 | 73 | –0.056 | 0.022 | 74 | –0.060 | –0.081 | 75 | 0.070 | –0.014 |
| 76 | 0.082 | –0.092 | 77 | –0.131 | –0.065 | 78 | –0.057 | –0.039 | 79 | 0.037 | –0.098 |
| 80 | 0.062 | 0.062 | 81 | 0.119 | 0.004 | 82 | –0.022 | –0.161 | 83 | 0.059 | 0.015 |
| 84 | 0.024 | 0.059 | 85 | –0.137 | 0.047 | 86 | 0.001 | 0.115 | 87 | 0.053 | –0.004 |
| 88 | 0.098 | 0.026 | 89 | –0.038 | 0.106 | 90 | –0.115 | 0.055 | 91 | 0.060 | 0.088 |
| 92 | 0.021 | –0.028 | 93 | 0.097 | –0.083 | 94 | 0.040 | 0.111 | 95 | –0.005 | 0.120 |
| 96 | 0.156 | 0.000 | 97 | –0.005 | –0.120 | 98 | 0.040 | –0.111 | 99 | 0.097 | 0.083 |
| 100 | 0.021 | 0.028 | 101 | 0.060 | –0.088 | 102 | –0.115 | –0.055 | 103 | –0.038 | –0.106 |
| 104 | 0.098 | –0.026 | 105 | 0.053 | 0.004 | 106 | 0.001 | –0.115 | 107 | –0.137 | –0.047 |
| 108 | 0.024 | –0.059 | 109 | 0.059 | –0.015 | 110 | –0.022 | 0.161 | 111 | 0.119 | –0.004 |
| 112 | 0.062 | –0.062 | 113 | 0.037 | 0.098 | 114 | –0.057 | 0.039 | 115 | –0.131 | 0.065 |
| 116 | 0.082 | 0.092 | 117 | 0.070 | 0.014 | 118 | –0.060 | 0.081 | 119 | –0.056 | –0.022 |
| 120 | –0.035 | –0.151 | 121 | –0.122 | –0.017 | 122 | –0.127 | –0.021 | 123 | 0.075 | –0.074 |
| 124 | –0.003 | 0.054 | 125 | –0.092 | 0.115 | 126 | 0.092 | 0.106 | 127 | 0.012 | 0.098 |
| 128 | –0.156 | 0.000 | 129 | 0.012 | –0.098 | 130 | 0.092 | –0.106 | 131 | –0.092 | –0.115 |
| 132 | –0.003 | –0.054 | 133 | 0.075 | 0.074 | 134 | –0.127 | 0.021 | 135 | –0.122 | 0.017 |
| 136 | –0.035 | 0.151 | 137 | –0.056 | 0.022 | 138 | –0.060 | –0.081 | 139 | 0.070 | –0.014 |
| 140 | 0.082 | –0.092 | 141 | –0.131 | –0.065 | 142 | –0.057 | –0.039 | 143 | 0.037 | –0.098 |
| 144 | 0.062 | 0.062 | 145 | 0.119 | 0.004 | 146 | –0.022 | –0.161 | 147 | 0.059 | 0.015 |
| 148 | 0.024 | 0.059 | 149 | –0.137 | 0.047 | 150 | 0.001 | 0.115 | 151 | 0.053 | –0.004 |
| 152 | 0.098 | 0.026 | 153 | –0.038 | 0.106 | 154 | –0.115 | 0.055 | 155 | 0.060 | 0.088 |
| 156 | 0.021 | –0.028 | 157 | 0.097 | –0.083 | 158 | 0.040 | 0.111 | 159 | –0.005 | 0.120 |
| 160 | 0.078 | 0 | | | | | | | | | |

Copyright © 2007 IEEE. All rights reserved.

## G.4 Generation of the SIGNAL field

### G.4.1 SIGNAL field bit assignment

The SIGNAL field bit assignment follows 17.3.4 and Figure 17-5. The transmitted bits are shown in Table G.7, where bit 0 is transmitted first.

**Table G.7—Bit assignment for SIGNAL field**

| ## | Bit | Meaning | | ## | Bit | Meaning |
|---|---|---|---|---|---|---|
| 0 | 1 | RATE: R1 | | 12 | 0 | — |
| 1 | 0 | RATE: R2 | | 13 | 0 | — |
| 2 | 1 | RATE: R3 | | 14 | 0 | — |
| 3 | 1 | RATE: R4 | | 15 | 0 | — |
| 4 | 0 | Reserved | | 16 | 0 | LENGTH (MSB) |
| 5 | 0 | LENGTH (LSB) | | 17 | 0 | Parity |
| 6 | 0 | — | | 18 | 0 | SIGNAL TAIL |
| 7 | 1 | — | | 19 | 0 | SIGNAL TAIL |
| 8 | 0 | — | | 20 | 0 | SIGNAL TAIL |
| 9 | 0 | — | | 21 | 0 | SIGNAL TAIL |
| 10 | 1 | — | | 22 | 0 | SIGNAL TAIL |
| 11 | 1 | — | | 23 | 0 | SIGNAL TAIL |

### G.4.2 Coding the SIGNAL field bits

The bits are encoded by the rate 1/2 convolutional encoder to yield the 48 bits given in Table G.8.

**Table G.8—SIGNAL field bits after encoding**

| ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | | 8 | 1 | | 16 | 0 | | 24 | 0 | | 32 | 0 | | 40 | 0 |
| 1 | 1 | | 9 | 0 | | 17 | 0 | | 25 | 0 | | 33 | 1 | | 41 | 0 |
| 2 | 0 | | 10 | 1 | | 18 | 0 | | 26 | 1 | | 34 | 1 | | 42 | 0 |
| 3 | 1 | | 11 | 0 | | 19 | 0 | | 27 | 1 | | 35 | 1 | | 43 | 0 |
| 4 | 0 | | 12 | 0 | | 20 | 0 | | 28 | 1 | | 36 | 0 | | 44 | 0 |
| 5 | 0 | | 13 | 0 | | 21 | 0 | | 29 | 1 | | 37 | 0 | | 45 | 0 |
| 6 | 0 | | 14 | 0 | | 22 | 1 | | 30 | 1 | | 38 | 0 | | 46 | 0 |
| 7 | 1 | | 15 | 1 | | 23 | 0 | | 31 | 0 | | 39 | 0 | | 47 | 0 |


Copyright © 2007 IEEE. All rights reserved.

### G.4.3 Interleaving the SIGNAL field bits

The encoded bits are interleaved according to the interleaver of 17.3.5.6. A detailed breakdown of the interleaving operation is described in G.7. The interleaved SIGNAL field bits are shown in Table G.9.

**Table G.9—SIGNAL field bits after interleaving**

| ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | | 8 | 1 | | 16 | 0 | | 24 | 1 | | 32 | 0 | | 40 | 1 |
| 1 | 0 | | 9 | 1 | | 17 | 0 | | 25 | 0 | | 33 | 0 | | 41 | 0 |
| 2 | 0 | | 10 | 0 | | 18 | 0 | | 26 | 0 | | 34 | 1 | | 42 | 0 |
| 3 | 1 | | 11 | 1 | | 19 | 1 | | 27 | 0 | | 35 | 0 | | 43 | 1 |
| 4 | 0 | | 12 | 0 | | 20 | 0 | | 28 | 0 | | 36 | 0 | | 44 | 0 |
| 5 | 1 | | 13 | 0 | | 21 | 1 | | 29 | 0 | | 37 | 1 | | 45 | 1 |
| 6 | 0 | | 14 | 0 | | 22 | 0 | | 30 | 1 | | 38 | 0 | | 46 | 0 |
| 7 | 0 | | 15 | 0 | | 23 | 0 | | 31 | 1 | | 39 | 0 | | 47 | 0 |

### G.4.4 SIGNAL field frequency domain

The encoded and interleaved bits are BPSK modulated to yield the frequency domain representation given in Table G.10. Locations –21, –7, 7, and 21 are skipped and will be used for pilot insertion.

**Table G.10—Frequency domain representation of SIGNAL field**

| ## | Re | Im | | ## | Re | Im | | ## | Re | Im | | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| –32 | 0.000 | 0.000 | | –16 | 1.000 | 0.000 | | 0 | 0.000 | 0.000 | | 16 | –1.000 | 0.000 |
| –31 | 0.000 | 0.000 | | –15 | –1.000 | 0.000 | | 1 | 1.000 | 0.000 | | 17 | –1.000 | 0.000 |
| –30 | 0.000 | 0.000 | | –14 | 1.000 | 0.000 | | 2 | –1.000 | 0.000 | | 18 | 1.000 | 0.000 |
| –29 | 0.000 | 0.000 | | –13 | –1.000 | 0.000 | | 3 | –1.000 | 0.000 | | 19 | –1.000 | 0.000 |
| –28 | 0.000 | 0.000 | | –12 | –1.000 | 0.000 | | 4 | –1.000 | 0.000 | | 20 | –1.000 | 0.000 |
| –27 | 0.000 | 0.000 | | –11 | –1.000 | 0.000 | | 5 | –1.000 | 0.000 | | 21 | X | X |
| –26 | 1.000 | 0.000 | | –10 | –1.000 | 0.000 | | 6 | –1.000 | 0.000 | | 22 | 1.000 | 0.000 |
| –25 | –1.000 | 0.000 | | –9 | –1.000 | 0.000 | | 7 | X | X | | 23 | –1.000 | 0.000 |
| –24 | –1.000 | 0.000 | | –8 | –1.000 | 0.000 | | 8 | 1.000 | 0.000 | | 24 | 1.000 | 0.000 |
| –23 | 1.000 | 0.000 | | –7 | X | X | | 9 | 1.000 | 0.000 | | 25 | –1.000 | 0.000 |
| –22 | –1.000 | 0.000 | | –6 | –1.000 | 0.000 | | 10 | –1.000 | 0.000 | | 26 | –1.000 | 0.000 |
| –21 | X | X | | –5 | 1.000 | 0.000 | | 11 | –1.000 | 0.000 | | 27 | 0.000 | 0.000 |
| –20 | 1.000 | 0.000 | | –4 | –1.000 | 0.000 | | 12 | 1.000 | 0.000 | | 28 | 0.000 | 0.000 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.10—Frequency domain representation of SIGNAL field *(continued)***

| ## | Re | Im | | ## | Re | Im | | ## | Re | Im | | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| –19 | –1.000 | 0.000 | | –3 | 1.000 | 0.000 | | 13 | –1.000 | 0.000 | | 29 | 0.000 | 0.000 |
| –18 | –1.000 | 0.000 | | –2 | –1.000 | 0.000 | | 14 | –1.000 | 0.000 | | 30 | 0.000 | 0.000 |
| –17 | 1.000 | 0.000 | | –1 | –1.000 | 0.000 | | 15 | 1.000 | 0.000 | | 31 | 0.000 | 0.000 |

Four pilot subcarriers are added by taking the values {1.0,1.0,1.0,-1.0}, multiplying them by the first element of sequence $p_{0\ldots126}$, given in Equation (17-22) (in 17.3.5.9), and inserting them into location {–21, –7,7,21}, respectively. The resulting frequency domain values are given in Table G.11.

**Table G.11—Frequency domain representation of SIGNAL field with pilots inserted**

| ## | Re | Im | | ## | Re | Im | | ## | Re | Im | | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| –32 | 0.000 | 0.000 | | –16 | 1.000 | 0.000 | | 0 | 0.000 | 0.000 | | 16 | –1.000 | 0.000 |
| –31 | 0.000 | 0.000 | | –15 | –1.000 | 0.000 | | 1 | 1.000 | 0.000 | | 17 | –1.000 | 0.000 |
| –30 | 0.000 | 0.000 | | –14 | 1.000 | 0.000 | | 2 | –1.000 | 0.000 | | 18 | 1.000 | 0.000 |
| –29 | 0.000 | 0.000 | | –13 | –1.000 | 0.000 | | 3 | –1.000 | 0.000 | | 19 | –1.000 | 0.000 |
| –28 | 0.000 | 0.000 | | –12 | –1.000 | 0.000 | | 4 | –1.000 | 0.000 | | 20 | –1.000 | 0.000 |
| –27 | 0.000 | 0.000 | | –11 | –1.000 | 0.000 | | 5 | –1.000 | 0.000 | | 21 | –1.000 | 0.000 |
| –26 | 1.000 | 0.000 | | –10 | –1.000 | 0.000 | | 6 | –1.000 | 0.000 | | 22 | 1.000 | 0.000 |
| –25 | –1.000 | 0.000 | | –9 | –1.000 | 0.000 | | 7 | 1.000 | 0.000 | | 23 | –1.000 | 0.000 |
| –24 | –1.000 | 0.000 | | –8 | –1.000 | 0.000 | | 8 | 1.000 | 0.000 | | 24 | 1.000 | 0.000 |
| –23 | 1.000 | 0.000 | | –7 | 1.000 | 0.000 | | 9 | 1.000 | 0.000 | | 25 | –1.000 | 0.000 |
| –22 | –1.000 | 0.000 | | –6 | –1.000 | 0.000 | | 10 | –1.000 | 0.000 | | 26 | –1.000 | 0.000 |
| –21 | 1.000 | 0.000 | | –5 | 1.000 | 0.000 | | 11 | –1.000 | 0.000 | | 27 | 0.000 | 0.000 |
| –20 | 1.000 | 0.000 | | –4 | –1.000 | 0.000 | | 12 | 1.000 | 0.000 | | 28 | 0.000 | 0.000 |
| –19 | –1.000 | 0.000 | | –3 | 1.000 | 0.000 | | 13 | –1.000 | 0.000 | | 29 | 0.000 | 0.000 |
| –18 | –1.000 | 0.000 | | –2 | –1.000 | 0.000 | | 14 | –1.000 | 0.000 | | 30 | 0.000 | 0.000 |
| –17 | 1.000 | 0.000 | | –1 | –1.000 | 0.000 | | 15 | 1.000 | 0.000 | | 31 | 0.000 | 0.000 |

### G.4.5 SIGNAL field time domain

The time domain representation is derived by performing IFFT on the contents of Table G.11, extending cyclically, and multiplying by the window function


Copyright © 2007 IEEE. All rights reserved.

$$W(k) = \begin{bmatrix} 0.5 & k = 0 \\ 1 & 1 \le k \le 79 \\ 0.5 & k = 80 \end{bmatrix}$$

The resulting 81 samples vector is shown in Table G.12. Note that the time-windowing feature illustrated here is not a part of the normative specifications.

**Table G.12—Time domain representation of SIGNAL field**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.031 | 0.000 | 1 | 0.033 | –0.044 | 2 | –0.002 | –0.038 | 3 | –0.081 | 0.084 |
| 4 | 0.007 | –0.100 | 5 | –0.001 | –0.113 | 6 | –0.021 | –0.005 | 7 | 0.136 | –0.105 |
| 8 | 0.098 | –0.044 | 9 | 0.011 | –0.002 | 10 | –0.033 | 0.044 | 11 | –0.060 | 0.124 |
| 12 | 0.010 | 0.097 | 13 | 0.000 | –0.008 | 14 | 0.018 | –0.083 | 15 | –0.069 | 0.027 |
| 16 | –0.219 | 0.000 | 17 | –0.069 | –0.027 | 18 | 0.018 | 0.083 | 19 | 0.000 | 0.008 |
| 20 | 0.010 | –0.097 | 21 | –0.060 | –0.124 | 22 | –0.033 | –0.044 | 23 | 0.011 | 0.002 |
| 24 | 0.098 | 0.044 | 25 | 0.136 | 0.105 | 26 | –0.021 | 0.005 | 27 | –0.001 | 0.113 |
| 28 | 0.007 | 0.100 | 29 | –0.081 | –0.084 | 30 | –0.002 | 0.038 | 31 | 0.033 | 0.044 |
| 32 | 0.062 | 0.000 | 33 | 0.057 | 0.052 | 34 | 0.016 | 0.174 | 35 | 0.035 | 0.116 |
| 36 | –0.051 | –0.202 | 37 | 0.011 | 0.036 | 38 | 0.089 | 0.209 | 39 | –0.049 | –0.008 |
| 40 | –0.035 | 0.044 | 41 | 0.017 | –0.059 | 42 | 0.053 | –0.017 | 43 | 0.099 | 0.100 |
| 44 | 0.034 | –0.148 | 45 | –0.003 | –0.094 | 46 | –0.120 | 0.042 | 47 | –0.136 | –0.070 |
| 48 | –0.031 | 0.000 | 49 | –0.136 | 0.070 | 50 | –0.120 | –0.042 | 51 | –0.003 | 0.094 |
| 52 | 0.034 | 0.148 | 53 | 0.099 | –0.100 | 54 | 0.053 | 0.017 | 55 | 0.017 | 0.059 |
| 56 | –0.035 | –0.044 | 57 | –0.049 | 0.008 | 58 | 0.089 | –0.209 | 59 | 0.011 | –0.036 |
| 60 | –0.051 | 0.202 | 61 | 0.035 | –0.116 | 62 | 0.016 | –0.174 | 63 | 0.057 | –0.052 |
| 64 | 0.062 | 0.000 | 65 | 0.033 | –0.044 | 66 | –0.002 | –0.038 | 67 | –0.081 | 0.084 |
| 68 | 0.007 | –0.100 | 69 | –0.001 | –0.113 | 70 | –0.021 | –0.005 | 71 | 0.136 | –0.105 |
| 72 | 0.098 | –0.044 | 73 | 0.011 | –0.002 | 74 | –0.033 | 0.044 | 75 | –0.060 | 0.124 |
| 76 | 0.010 | 0.097 | 77 | 0.000 | –0.008 | 78 | 0.018 | –0.083 | 79 | –0.069 | 0.027 |
| 80 | –0.109 | 0.000 | | | | | | | | | |

The SIGNAL field samples are appended with one sample overlap to the preamble, given in Table G.6.

Copyright © 2007 IEEE. All rights reserved.

## G.5 Generating the DATA bits

### G.5.1 Delineating, SERVICE field prepending, and zero padding

The transmitted message shown in Table G.1 contains 100 octets or, equivalently, 800 bits. The bits are prepended by the 16 SERVICE field bits and are appended by 6 tail bits. The resulting 822 bits are appended by zero bits to yield an integer number of OFDM symbols. For the 36 Mb/s mode, there are 144 data bits per OFDM symbol; the overall number of bits is ceiling (822/144) × 144 = 864. Hence, 864 – 822 = 42 zero bits are appended.

The data bits are shown in Table G.13 and Table G.14. For clarity, only the first and last 144 bits are shown.

**Table G.13—First 144 DATA bits**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 0 | 48 | 0 | 72 | 1 | 96 | 0 | 120 | 1 |
| 1 | 0 | 25 | 1 | 49 | 0 | 73 | 0 | 97 | 0 | 121 | 0 |
| 2 | 0 | 26 | 0 | 50 | 0 | 74 | 1 | 98 | 0 | 122 | 0 |
| 3 | 0 | 27 | 0 | 51 | 0 | 75 | 1 | 99 | 0 | 123 | 0 |
| 4 | 0 | 28 | 0 | 52 | 0 | 76 | 0 | 100 | 0 | 124 | 0 |
| 5 | 0 | 29 | 0 | 53 | 0 | 77 | 0 | 101 | 0 | 125 | 0 |
| 6 | 0 | 30 | 0 | 54 | 0 | 78 | 1 | 102 | 0 | 126 | 0 |
| 7 | 0 | 31 | 0 | 55 | 0 | 79 | 1 | 103 | 0 | 127 | 0 |
| 8 | 0 | 32 | 0 | 56 | 0 | 80 | 1 | 104 | 0 | 128 | 0 |
| 9 | 0 | 33 | 0 | 57 | 0 | 81 | 1 | 105 | 0 | 129 | 0 |
| 10 | 0 | 34 | 0 | 58 | 0 | 82 | 1 | 106 | 0 | 130 | 1 |
| 11 | 0 | 35 | 0 | 59 | 0 | 83 | 0 | 107 | 0 | 131 | 1 |
| 12 | 0 | 36 | 0 | 60 | 0 | 84 | 1 | 108 | 0 | 132 | 1 |
| 13 | 0 | 37 | 0 | 61 | 1 | 85 | 1 | 109 | 1 | 133 | 1 |
| 14 | 0 | 38 | 0 | 62 | 1 | 86 | 0 | 110 | 0 | 134 | 0 |
| 15 | 0 | 39 | 0 | 63 | 0 | 87 | 0 | 111 | 0 | 135 | 0 |
| 16 | 0 | 40 | 0 | 64 | 0 | 88 | 0 | 112 | 0 | 136 | 1 |
| 17 | 0 | 41 | 1 | 65 | 0 | 89 | 1 | 113 | 1 | 137 | 0 |
| 18 | 1 | 42 | 1 | 66 | 0 | 90 | 1 | 114 | 1 | 138 | 0 |
| 19 | 0 | 43 | 1 | 67 | 1 | 91 | 0 | 115 | 0 | 139 | 0 |
| 20 | 0 | 44 | 0 | 68 | 0 | 92 | 0 | 116 | 1 | 140 | 1 |
| 21 | 0 | 45 | 1 | 69 | 0 | 93 | 1 | 117 | 0 | 141 | 1 |
| 22 | 0 | 46 | 0 | 70 | 0 | 94 | 0 | 118 | 1 | 142 | 1 |
| 23 | 0 | 47 | 0 | 71 | 0 | 95 | 1 | 119 | 1 | 143 | 1 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.14—Last 144 DATA bits**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 720 | 0 | 744 | 0 | 768 | 1 | 792 | 1 | 816 | 0 | 840 | 0 |
| 721 | 0 | 745 | 0 | 769 | 0 | 793 | 1 | 817 | 0 | 841 | 0 |
| 722 | 0 | 746 | 0 | 770 | 1 | 794 | 1 | 818 | 0 | 842 | 0 |
| 723 | 0 | 747 | 0 | 771 | 0 | 795 | 0 | 819 | 0 | 843 | 0 |
| 724 | 0 | 748 | 0 | 772 | 0 | 796 | 1 | 820 | 0 | 844 | 0 |
| 725 | 1 | 749 | 1 | 773 | 1 | 797 | 0 | 821 | 0 | 845 | 0 |
| 726 | 0 | 750 | 0 | 774 | 1 | 798 | 1 | 822 | 0 | 846 | 0 |
| 727 | 0 | 751 | 0 | 775 | 0 | 799 | 0 | 823 | 0 | 847 | 0 |
| 728 | 1 | 752 | 0 | 776 | 1 | 800 | 1 | 824 | 0 | 848 | 0 |
| 729 | 1 | 753 | 0 | 777 | 0 | 801 | 0 | 825 | 0 | 849 | 0 |
| 730 | 1 | 754 | 1 | 778 | 0 | 802 | 0 | 826 | 0 | 850 | 0 |
| 731 | 0 | 755 | 0 | 779 | 0 | 803 | 1 | 827 | 0 | 851 | 0 |
| 732 | 1 | 756 | 1 | 780 | 0 | 804 | 1 | 828 | 0 | 852 | 0 |
| 733 | 1 | 757 | 1 | 781 | 1 | 805 | 0 | 829 | 0 | 853 | 0 |
| 734 | 1 | 758 | 1 | 782 | 1 | 806 | 0 | 830 | 0 | 854 | 0 |
| 735 | 0 | 759 | 0 | 783 | 0 | 807 | 1 | 831 | 0 | 855 | 0 |
| 736 | 1 | 760 | 0 | 784 | 0 | 808 | 1 | 832 | 0 | 856 | 0 |
| 737 | 0 | 761 | 1 | 785 | 1 | 809 | 0 | 833 | 0 | 857 | 0 |
| 738 | 1 | 762 | 0 | 786 | 0 | 810 | 1 | 834 | 0 | 858 | 0 |
| 739 | 0 | 763 | 0 | 787 | 1 | 811 | 1 | 835 | 0 | 859 | 0 |
| 740 | 0 | 764 | 1 | 788 | 1 | 812 | 0 | 836 | 0 | 860 | 0 |
| 741 | 1 | 765 | 1 | 789 | 0 | 813 | 1 | 837 | 0 | 861 | 0 |
| 742 | 1 | 766 | 1 | 790 | 1 | 814 | 1 | 838 | 0 | 862 | 0 |
| 743 | 0 | 767 | 0 | 791 | 1 | 815 | 1 | 839 | 0 | 863 | 0 |

### G.5.2 Scrambling

The 864 bits are scrambled by the scrambler of Figure 17-7 (in 17.3.5.4). The initial state of the scrambler is the state 1011101. The generated scrambling sequence is given in Table G.15.

After scrambling, the 6 bits in location 816 (i.e., the 817$^{th}$ bit) to 821 (the 822$^{nd}$ bit) are zeroed. The first and last 144 scrambled bits are shown in Table G.16 and Table G.17, respectively.

Copyright © 2007 IEEE. All rights reserved.

**Table G.15—Scrambling sequence for seed 1011101**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 16 | 1 | 32 | 0 | 48 | 1 | 64 | 0 | 80 | 0 | 96 | 0 | 112 | 1 |
| 1 | 1 | 17 | 0 | 33 | 1 | 49 | 1 | 65 | 1 | 81 | 0 | 97 | 0 | 113 | 0 |
| 2 | 1 | 18 | 1 | 34 | 1 | 50 | 1 | 66 | 1 | 82 | 1 | 98 | 1 | 114 | 0 |
| 3 | 0 | 19 | 0 | 35 | 0 | 51 | 1 | 67 | 1 | 83 | 1 | 99 | 0 | 115 | 1 |
| 4 | 1 | 20 | 1 | 36 | 1 | 52 | 0 | 68 | 0 | 84 | 1 | 100 | 0 | 116 | 1 |
| 5 | 1 | 21 | 0 | 37 | 0 | 53 | 1 | 69 | 0 | 85 | 0 | 101 | 1 | 117 | 0 |
| 6 | 0 | 22 | 0 | 38 | 0 | 54 | 0 | 70 | 0 | 86 | 1 | 102 | 0 | 118 | 0 |
| 7 | 0 | 23 | 1 | 39 | 0 | 55 | 0 | 71 | 1 | 87 | 1 | 103 | 0 | 119 | 0 |
| 8 | 0 | 24 | 1 | 40 | 0 | 56 | 1 | 72 | 1 | 88 | 1 | 104 | 0 | 120 | 1 |
| 9 | 0 | 25 | 1 | 41 | 1 | 57 | 0 | 73 | 1 | 89 | 1 | 105 | 0 | 121 | 0 |
| 10 | 0 | 26 | 0 | 42 | 0 | 58 | 1 | 74 | 1 | 90 | 0 | 106 | 0 | 122 | 1 |
| 11 | 1 | 27 | 0 | 43 | 1 | 59 | 0 | 75 | 1 | 91 | 0 | 107 | 0 | 123 | 1 |
| 12 | 1 | 28 | 1 | 44 | 0 | 60 | 0 | 76 | 1 | 92 | 1 | 108 | 1 | 124 | 1 |
| 13 | 0 | 29 | 1 | 45 | 1 | 61 | 0 | 77 | 1 | 93 | 0 | 109 | 0 | 125 | 0 |
| 14 | 0 | 30 | 1 | 46 | 0 | 62 | 1 | 78 | 0 | 94 | 1 | 110 | 0 | 126 | 1 |
| 15 | 1 | 31 | 1 | 47 | 1 | 63 | 1 | 79 | 0 | 95 | 1 | 111 | 0 | | |

**Table G.16—First 144 bits after scrambling**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 1 | 48 | 1 | 72 | 0 | 96 | 0 | 120 | 0 |
| 1 | 1 | 25 | 0 | 49 | 1 | 73 | 1 | 97 | 0 | 121 | 0 |
| 2 | 1 | 26 | 0 | 50 | 1 | 74 | 0 | 98 | 1 | 122 | 1 |
| 3 | 0 | 27 | 0 | 51 | 1 | 75 | 0 | 99 | 0 | 123 | 1 |
| 4 | 1 | 28 | 1 | 52 | 0 | 76 | 1 | 100 | 0 | 124 | 1 |
| 5 | 1 | 29 | 1 | 53 | 1 | 77 | 1 | 101 | 1 | 125 | 0 |
| 6 | 0 | 30 | 1 | 54 | 0 | 78 | 1 | 102 | 0 | 126 | 1 |
| 7 | 0 | 31 | 1 | 55 | 0 | 79 | 1 | 103 | 0 | 127 | 0 |
| 8 | 0 | 32 | 0 | 56 | 1 | 80 | 1 | 104 | 0 | 128 | 1 |
| 9 | 0 | 33 | 1 | 57 | 0 | 81 | 1 | 105 | 0 | 129 | 1 |
| 10 | 0 | 34 | 1 | 58 | 1 | 82 | 0 | 106 | 0 | 130 | 1 |
| 11 | 1 | 35 | 0 | 59 | 0 | 83 | 1 | 107 | 0 | 131 | 0 |
| 12 | 1 | 36 | 1 | 60 | 0 | 84 | 0 | 108 | 1 | 132 | 0 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.16—First 144 bits after scrambling *(continued)***

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 0 | 37 | 0 | 61 | 1 | 85 | 1 | 109 | 1 | 133 | 1 |
| 14 | 0 | 38 | 0 | 62 | 0 | 86 | 1 | 110 | 0 | 134 | 0 |
| 15 | 1 | 39 | 0 | 63 | 1 | 87 | 1 | 111 | 0 | 135 | 0 |
| 16 | 1 | 40 | 0 | 64 | 0 | 88 | 1 | 112 | 1 | 136 | 1 |
| 17 | 0 | 41 | 0 | 65 | 1 | 89 | 0 | 113 | 1 | 137 | 0 |
| 18 | 0 | 42 | 1 | 66 | 1 | 90 | 1 | 114 | 1 | 138 | 1 |
| 19 | 0 | 43 | 0 | 67 | 0 | 91 | 0 | 115 | 1 | 139 | 1 |
| 20 | 1 | 44 | 0 | 68 | 0 | 92 | 1 | 116 | 0 | 140 | 1 |
| 21 | 0 | 45 | 0 | 69 | 0 | 93 | 1 | 117 | 0 | 141 | 1 |
| 22 | 0 | 46 | 0 | 70 | 0 | 94 | 1 | 118 | 1 | 142 | 0 |
| 23 | 1 | 47 | 1 | 71 | 1 | 95 | 0 | 119 | 1 | 143 | 0 |

**Table G.17—Last 144 bits after scrambling**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 720 | 0 | 744 | 0 | 768 | 1 | 792 | 0 | 816 | 0 | 840 | 0 |
| 721 | 1 | 745 | 0 | 769 | 0 | 793 | 0 | 817 | 0 | 841 | 0 |
| 722 | 1 | 746 | 0 | 770 | 1 | 794 | 1 | 818 | 1 | 842 | 0 |
| 723 | 1 | 747 | 1 | 771 | 0 | 795 | 1 | 819 | 0 | 843 | 0 |
| 724 | 1 | 748 | 0 | 772 | 0 | 796 | 0 | 820 | 1 | 844 | 1 |
| 725 | 1 | 749 | 1 | 773 | 0 | 797 | 0 | 821 | 0 | 845 | 1 |
| 726 | 0 | 750 | 1 | 774 | 0 | 798 | 0 | 822 | 0 | 846 | 1 |
| 727 | 1 | 751 | 1 | 775 | 0 | 799 | 0 | 823 | 0 | 847 | 0 |
| 728 | 1 | 752 | 0 | 776 | 1 | 800 | 1 | 824 | 1 | 848 | 1 |
| 729 | 0 | 753 | 0 | 777 | 1 | 801 | 0 | 825 | 1 | 849 | 1 |
| 730 | 0 | 754 | 1 | 778 | 1 | 802 | 0 | 826 | 0 | 850 | 1 |
| 731 | 0 | 755 | 1 | 779 | 0 | 803 | 0 | 827 | 1 | 851 | 1 |
| 732 | 1 | 756 | 1 | 780 | 1 | 804 | 1 | 828 | 1 | 852 | 0 |
| 733 | 0 | 757 | 0 | 781 | 1 | 805 | 1 | 829 | 1 | 853 | 0 |
| 734 | 1 | 758 | 0 | 782 | 0 | 806 | 0 | 830 | 0 | 854 | 1 |
| 735 | 0 | 759 | 1 | 783 | 0 | 807 | 0 | 831 | 0 | 855 | 0 |
| 736 | 0 | 760 | 0 | 784 | 0 | 808 | 1 | 832 | 0 | 856 | 1 |
| 737 | 0 | 761 | 0 | 785 | 0 | 809 | 1 | 833 | 1 | 857 | 1 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.17—Last 144 bits after scrambling *(continued)***

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 738 | 1 | 762 | 0 | 786 | 1 | 810 | 0 | 834 | 1 | 858 | 0 |
| 739 | 0 | 763 | 1 | 787 | 0 | 811 | 0 | 835 | 1 | 859 | 0 |
| 740 | 0 | 764 | 0 | 788 | 1 | 812 | 1 | 836 | 1 | 860 | 1 |
| 741 | 1 | 765 | 1 | 789 | 0 | 813 | 0 | 837 | 1 | 861 | 0 |
| 742 | 1 | 766 | 0 | 790 | 0 | 814 | 1 | 838 | 1 | 862 | 0 |
| 743 | 1 | 767 | 1 | 791 | 0 | 815 | 0 | 839 | 1 | 863 | 1 |

## G.6 Generating the first DATA symbol

### G.6.1 Coding the DATA bits

The scrambled bits are coded with a rate Ω convolutional code. The first 144 scrambled bits of Table G.16 are mapped into the 192 bits of Table G.18.

**Table G.18—Coded bits of first DATA symbol**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 32 | 1 | 64 | 0 | 96 | 1 | 128 | 1 | 160 | 1 |
| 1 | 0 | 33 | 0 | 65 | 1 | 97 | 0 | 129 | 1 | 161 | 1 |
| 2 | 1 | 34 | 0 | 66 | 0 | 98 | 0 | 130 | 0 | 162 | 1 |
| 3 | 0 | 35 | 1 | 67 | 0 | 99 | 0 | 131 | 0 | 163 | 0 |
| 4 | 1 | 36 | 1 | 68 | 1 | 100 | 1 | 132 | 0 | 164 | 0 |
| 5 | 0 | 37 | 1 | 69 | 0 | 101 | 1 | 133 | 0 | 165 | 0 |
| 6 | 1 | 38 | 0 | 70 | 1 | 102 | 1 | 134 | 0 | 166 | 0 |
| 7 | 1 | 39 | 1 | 71 | 0 | 103 | 1 | 135 | 0 | 167 | 0 |
| 8 | 0 | 40 | 1 | 72 | 1 | 104 | 1 | 136 | 0 | 168 | 1 |
| 9 | 0 | 41 | 0 | 73 | 1 | 105 | 1 | 137 | 1 | 169 | 1 |
| 10 | 0 | 42 | 1 | 74 | 1 | 106 | 0 | 138 | 0 | 170 | 0 |
| 11 | 0 | 43 | 1 | 75 | 1 | 107 | 0 | 139 | 0 | 171 | 1 |
| 12 | 1 | 44 | 0 | 76 | 1 | 108 | 0 | 140 | 0 | 172 | 0 |
| 13 | 0 | 45 | 1 | 77 | 0 | 109 | 0 | 141 | 0 | 173 | 0 |
| 14 | 0 | 46 | 0 | 78 | 1 | 110 | 0 | 142 | 1 | 174 | 1 |
| 15 | 0 | 47 | 1 | 79 | 1 | 111 | 0 | 143 | 1 | 175 | 1 |
| 16 | 1 | 48 | 1 | 80 | 1 | 112 | 1 | 144 | 1 | 176 | 1 |
| 17 | 0 | 49 | 0 | 81 | 1 | 113 | 1 | 145 | 1 | 177 | 1 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.18—Coded bits of first DATA symbol *(continued)***

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 1 | 50 | 0 | 82 | 1 | 114 | 0 | 146 | 1 | 178 | 1 |
| 19 | 0 | 51 | 1 | 83 | 0 | 115 | 0 | 147 | 0 | 179 | 0 |
| 20 | 0 | 52 | 1 | 84 | 1 | 116 | 1 | 148 | 0 | 180 | 1 |
| 21 | 0 | 53 | 0 | 85 | 0 | 117 | 0 | 149 | 0 | 181 | 0 |
| 22 | 0 | 54 | 1 | 86 | 0 | 118 | 0 | 150 | 0 | 182 | 1 |
| 23 | 1 | 55 | 0 | 87 | 0 | 119 | 0 | 151 | 0 | 183 | 1 |
| 24 | 1 | 56 | 0 | 88 | 1 | 120 | 0 | 152 | 0 | 184 | 1 |
| 25 | 1 | 57 | 0 | 89 | 1 | 121 | 1 | 153 | 0 | 185 | 0 |
| 26 | 1 | 58 | 0 | 90 | 0 | 122 | 1 | 154 | 0 | 186 | 1 |
| 27 | 1 | 59 | 1 | 91 | 0 | 123 | 1 | 155 | 1 | 187 | 1 |
| 28 | 0 | 60 | 1 | 92 | 0 | 124 | 0 | 156 | 1 | 188 | 0 |
| 29 | 0 | 61 | 1 | 93 | 0 | 125 | 0 | 157 | 0 | 189 | 0 |
| 30 | 0 | 62 | 0 | 94 | 1 | 126 | 1 | 158 | 0 | 190 | 1 |
| 31 | 0 | 63 | 1 | 95 | 0 | 127 | 1 | 159 | 1 | 191 | 0 |

## G.6.2 Interleaving the DATA bits

The interleaver is defined as a two-permutation process. The index of the coded bit before the first permutation shall be denoted by *k*; *i* shall be the index after the first and before the second permutation; and *j* shall be the index after the second permutation, just prior to modulation mapping. The mapping from *k* to *i* is shown in Table G.19, and the mapping from *i* to *j* is shown in Table G.20.

As a specific example, consider the case of $k = 17$ (the $18^{th}$ bit after encoding and puncturing). It is mapped by the first permutation to $i = 13$ and by the second permutation to $j = 12$ (the $13^{th}$ bit before mapping).

**Table G.19—First permutation**

| *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 97 | 48 | 3 | 72 | 100 | 96 | 6 | 120 | 103 | 144 | 9 | 168 | 106 |
| 1 | 12 | 25 | 109 | 49 | 15 | 73 | 112 | 97 | 18 | 121 | 115 | 145 | 21 | 169 | 118 |
| 2 | 24 | 26 | 121 | 50 | 27 | 74 | 124 | 98 | 30 | 122 | 127 | 146 | 33 | 170 | 130 |
| 3 | 36 | 27 | 133 | 51 | 39 | 75 | 136 | 99 | 42 | 123 | 139 | 147 | 45 | 171 | 142 |
| 4 | 48 | 28 | 145 | 52 | 51 | 76 | 148 | 100 | 54 | 124 | 151 | 148 | 57 | 172 | 154 |
| 5 | 60 | 29 | 157 | 53 | 63 | 77 | 160 | 101 | 66 | 125 | 163 | 149 | 69 | 173 | 166 |
| 6 | 72 | 30 | 169 | 54 | 75 | 78 | 172 | 102 | 78 | 126 | 175 | 150 | 81 | 174 | 178 |
| 7 | 84 | 31 | 181 | 55 | 87 | 79 | 184 | 103 | 90 | 127 | 187 | 151 | 93 | 175 | 190 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.19—First permutation *(continued)***

| *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* | *k* | *i* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 96 | 32 | 2 | 56 | 99 | 80 | 5 | 104 | 102 | 128 | 8 | 152 | 105 | 176 | 11 |
| 9 | 108 | 33 | 14 | 57 | 111 | 81 | 17 | 105 | 114 | 129 | 20 | 153 | 117 | 177 | 23 |
| 10 | 120 | 34 | 26 | 58 | 123 | 82 | 29 | 106 | 126 | 130 | 32 | 154 | 129 | 178 | 35 |
| 11 | 132 | 35 | 38 | 59 | 135 | 83 | 41 | 107 | 138 | 131 | 44 | 155 | 141 | 179 | 47 |
| 12 | 144 | 36 | 50 | 60 | 147 | 84 | 53 | 108 | 150 | 132 | 56 | 156 | 153 | 180 | 59 |
| 13 | 156 | 37 | 62 | 61 | 159 | 85 | 65 | 109 | 162 | 133 | 68 | 157 | 165 | 181 | 71 |
| 14 | 168 | 38 | 74 | 62 | 171 | 86 | 77 | 110 | 174 | 134 | 80 | 158 | 177 | 182 | 83 |
| 15 | 180 | 39 | 86 | 63 | 183 | 87 | 89 | 111 | 186 | 135 | 92 | 159 | 189 | 183 | 95 |
| 16 | 1 | 40 | 98 | 64 | 4 | 88 | 101 | 112 | 7 | 136 | 104 | 160 | 10 | 184 | 107 |
| 17 | 13 | 41 | 110 | 65 | 16 | 89 | 113 | 113 | 19 | 137 | 116 | 161 | 22 | 185 | 119 |
| 18 | 25 | 42 | 122 | 66 | 28 | 90 | 125 | 114 | 31 | 138 | 128 | 162 | 34 | 186 | 131 |
| 19 | 37 | 43 | 134 | 67 | 40 | 91 | 137 | 115 | 43 | 139 | 140 | 163 | 46 | 187 | 143 |
| 20 | 49 | 44 | 146 | 68 | 52 | 92 | 149 | 116 | 55 | 140 | 152 | 164 | 58 | 188 | 155 |
| 21 | 61 | 45 | 158 | 69 | 64 | 93 | 161 | 117 | 67 | 141 | 164 | 165 | 70 | 189 | 167 |
| 22 | 73 | 46 | 170 | 70 | 76 | 94 | 173 | 118 | 79 | 142 | 176 | 166 | 82 | 190 | 179 |
| 23 | 85 | 47 | 182 | 71 | 88 | 95 | 185 | 119 | 91 | 143 | 188 | 167 | 94 | 191 | 191 |

**Table G.20—Second permutation**

| *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 24 | 48 | 48 | 72 | 72 | 96 | 96 | 120 | 120 | 144 | 144 | 168 | 168 |
| 1 | 1 | 25 | 25 | 49 | 49 | 73 | 73 | 97 | 97 | 121 | 121 | 145 | 145 | 169 | 169 |
| 2 | 2 | 26 | 26 | 50 | 50 | 74 | 74 | 98 | 98 | 122 | 122 | 146 | 146 | 170 | 170 |
| 3 | 3 | 27 | 27 | 51 | 51 | 75 | 75 | 99 | 99 | 123 | 123 | 147 | 147 | 171 | 171 |
| 4 | 4 | 28 | 28 | 52 | 52 | 76 | 76 | 100 | 100 | 124 | 124 | 148 | 148 | 172 | 172 |
| 5 | 5 | 29 | 29 | 53 | 53 | 77 | 77 | 101 | 101 | 125 | 125 | 149 | 149 | 173 | 173 |
| 6 | 6 | 30 | 30 | 54 | 54 | 78 | 78 | 102 | 102 | 126 | 126 | 150 | 150 | 174 | 174 |
| 7 | 7 | 31 | 31 | 55 | 55 | 79 | 79 | 103 | 103 | 127 | 127 | 151 | 151 | 175 | 175 |
| 8 | 8 | 32 | 32 | 56 | 56 | 80 | 80 | 104 | 104 | 128 | 128 | 152 | 152 | 176 | 176 |
| 9 | 9 | 33 | 33 | 57 | 57 | 81 | 81 | 105 | 105 | 129 | 129 | 153 | 153 | 177 | 177 |
| 10 | 10 | 34 | 34 | 58 | 58 | 82 | 82 | 106 | 106 | 130 | 130 | 154 | 154 | 178 | 178 |
| 11 | 11 | 35 | 35 | 59 | 59 | 83 | 83 | 107 | 107 | 131 | 131 | 155 | 155 | 179 | 179 |
| 12 | 13 | 36 | 37 | 60 | 61 | 84 | 85 | 108 | 109 | 132 | 133 | 156 | 157 | 180 | 181 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.20—Second permutation *(continued)***

| *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* | *i* | *j* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 12 | 37 | 36 | 61 | 60 | 85 | 84 | 109 | 108 | 133 | 132 | 157 | 156 | 181 | 180 |
| 14 | 15 | 38 | 39 | 62 | 63 | 86 | 87 | 110 | 111 | 134 | 135 | 158 | 159 | 182 | 183 |
| 15 | 14 | 39 | 38 | 63 | 62 | 87 | 86 | 111 | 110 | 135 | 134 | 159 | 158 | 183 | 182 |
| 16 | 17 | 40 | 41 | 64 | 65 | 88 | 89 | 112 | 113 | 136 | 137 | 160 | 161 | 184 | 185 |
| 17 | 16 | 41 | 40 | 65 | 64 | 89 | 88 | 113 | 112 | 137 | 136 | 161 | 160 | 185 | 184 |
| 18 | 19 | 42 | 43 | 66 | 67 | 90 | 91 | 114 | 115 | 138 | 139 | 162 | 163 | 186 | 187 |
| 19 | 18 | 43 | 42 | 67 | 66 | 91 | 90 | 115 | 114 | 139 | 138 | 163 | 162 | 187 | 186 |
| 20 | 21 | 44 | 45 | 68 | 69 | 92 | 93 | 116 | 117 | 140 | 141 | 164 | 165 | 188 | 189 |
| 21 | 20 | 45 | 44 | 69 | 68 | 93 | 92 | 117 | 116 | 141 | 140 | 165 | 164 | 189 | 188 |
| 22 | 23 | 46 | 47 | 70 | 71 | 94 | 95 | 118 | 119 | 142 | 143 | 166 | 167 | 190 | 191 |
| 23 | 22 | 47 | 46 | 71 | 70 | 95 | 94 | 119 | 118 | 143 | 142 | 167 | 166 | 191 | 190 |

The interleaved bits are shown in Table G.21.

**Table G.21—Interleaved bits of first DATA symbol**

| ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 32 | 0 | 64 | 0 | 96 | 0 | 128 | 0 | 160 | 0 |
| 1 | 1 | 33 | 1 | 65 | 0 | 97 | 1 | 129 | 0 | 161 | 0 |
| 2 | 1 | 34 | 1 | 66 | 0 | 98 | 1 | 130 | 0 | 162 | 0 |
| 3 | 1 | 35 | 1 | 67 | 1 | 99 | 0 | 131 | 1 | 163 | 0 |
| 4 | 0 | 36 | 0 | 68 | 0 | 100 | 1 | 132 | 1 | 164 | 0 |
| 5 | 1 | 37 | 0 | 69 | 0 | 101 | 1 | 133 | 0 | 165 | 0 |
| 6 | 1 | 38 | 1 | 70 | 0 | 102 | 1 | 134 | 1 | 166 | 0 |
| 7 | 1 | 39 | 1 | 71 | 0 | 103 | 0 | 135 | 1 | 167 | 0 |
| 8 | 1 | 40 | 0 | 72 | 1 | 104 | 0 | 136 | 0 | 168 | 0 |
| 9 | 1 | 41 | 0 | 73 | 0 | 105 | 0 | 137 | 1 | 169 | 0 |
| 10 | 1 | 42 | 0 | 74 | 0 | 106 | 1 | 138 | 1 | 170 | 0 |
| 11 | 1 | 43 | 0 | 75 | 1 | 107 | 1 | 139 | 0 | 171 | 0 |
| 12 | 0 | 44 | 0 | 76 | 1 | 108 | 1 | 140 | 1 | 172 | 1 |
| 13 | 0 | 45 | 0 | 77 | 0 | 109 | 0 | 141 | 0 | 173 | 1 |
| 14 | 0 | 46 | 0 | 78 | 1 | 110 | 0 | 142 | 1 | 174 | 0 |
| 15 | 0 | 47 | 0 | 79 | 0 | 111 | 0 | 143 | 1 | 175 | 1 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.21—Interleaved bits of first DATA symbol *(continued)***

| ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit | | ## | Bit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 1 | | 48 | 1 | | 80 | 0 | | 112 | 1 | | 144 | 1 | | 176 | 1 |
| 17 | 1 | | 49 | 0 | | 81 | 0 | | 113 | 1 | | 145 | 0 | | 177 | 0 |
| 18 | 1 | | 50 | 1 | | 82 | 0 | | 114 | 1 | | 146 | 0 | | 178 | 1 |
| 19 | 0 | | 51 | 1 | | 83 | 1 | | 115 | 1 | | 147 | 1 | | 179 | 1 |
| 20 | 1 | | 52 | 1 | | 84 | 1 | | 116 | 0 | | 148 | 1 | | 180 | 0 |
| 21 | 1 | | 53 | 1 | | 85 | 1 | | 117 | 1 | | 149 | 0 | | 181 | 0 |
| 22 | 1 | | 54 | 1 | | 86 | 0 | | 118 | 0 | | 150 | 0 | | 182 | 1 |
| 23 | 1 | | 55 | 1 | | 87 | 1 | | 119 | 1 | | 151 | 0 | | 183 | 1 |
| 24 | 1 | | 56 | 0 | | 88 | 0 | | 120 | 0 | | 152 | 0 | | 184 | 0 |
| 25 | 1 | | 57 | 0 | | 89 | 0 | | 121 | 1 | | 153 | 1 | | 185 | 1 |
| 26 | 0 | | 58 | 0 | | 90 | 0 | | 122 | 1 | | 154 | 0 | | 186 | 1 |
| 27 | 0 | | 59 | 1 | | 91 | 1 | | 123 | 0 | | 155 | 0 | | 187 | 0 |
| 28 | 0 | | 60 | 0 | | 92 | 0 | | 124 | 1 | | 156 | 0 | | 188 | 1 |
| 29 | 1 | | 61 | 0 | | 93 | 0 | | 125 | 0 | | 157 | 0 | | 189 | 1 |
| 30 | 0 | | 62 | 0 | | 94 | 1 | | 126 | 0 | | 158 | 1 | | 190 | 0 |
| 31 | 0 | | 63 | 1 | | 95 | 0 | | 127 | 1 | | 159 | 1 | | 191 | 1 |

## G.6.3 Mapping into symbols

The frequency domain symbols are generated by grouping 4 coded bits and mapping into complex 16-QAM symbols according to Table 17-9 (in 17.3.5.7). For instance, the first 4 bits (0 1 1 1) are mapped to the complex value, –0.316 + 0.316j, inserted at subcarrier #26.

Four pilot subcarriers are added by taking the values {1.0,1.0,1.0,–1.0}, multiplying them by the second element of sequence p, given in Equation (17-22) (in 17.3.5.9), and inserting them into location {–21,–7,7,21}, respectively.

The frequency domain is shown in Table G.22.

**Table G.22—Frequency domain of first DATA symbol**

| ## | Re | Im | | ## | Re | Im | | ## | Re | Im | | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| –32 | 0.000 | 0.000 | | –16 | –0.949 | 0.316 | | 0 | 0.000 | 0.000 | | 16 | –0.316 | –0.949 |
| –31 | 0.000 | 0.000 | | –15 | –0.949 | –0.949 | | 1 | –0.316 | 0.949 | | 17 | –0.949 | 0.316 |
| –30 | 0.000 | 0.000 | | –14 | –0.949 | –0.949 | | 2 | 0.316 | 0.949 | | 18 | –0.949 | –0.949 |
| –29 | 0.000 | 0.000 | | –13 | 0.949 | 0.316 | | 3 | –0.949 | 0.316 | | 19 | –0.949 | –0.949 |
| –28 | 0.000 | 0.000 | | –12 | 0.316 | 0.316 | | 4 | 0.949 | –0.949 | | 20 | –0.949 | –0.949 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.22—Frequency domain of first DATA symbol *(continued)***

| ## | Re | Im | | ## | Re | Im | | ## | Re | Im | | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| –27 | 0.000 | 0.000 | | –11 | –0.949 | –0.316 | | 5 | 0.316 | 0.316 | | 21 | –1.000 | 0.000 |
| –26 | –0.316 | 0.316 | | –10 | –0.949 | –0.316 | | 6 | –0.316 | –0.316 | | 22 | 0.316 | –0.316 |
| –25 | –0.316 | 0.316 | | –9 | –0.949 | –0.316 | | 7 | 1.000 | 0.000 | | 23 | 0.949 | 0.316 |
| –24 | 0.316 | 0.316 | | –8 | –0.949 | –0.949 | | 8 | –0.316 | 0.949 | | 24 | –0.949 | 0.316 |
| –23 | –0.949 | –0.949 | | –7 | 1.000 | 0.000 | | 9 | 0.949 | –0.316 | | 25 | –0.316 | 0.949 |
| –22 | 0.316 | 0.949 | | –6 | 0.949 | –0.316 | | 10 | –0.949 | –0.316 | | 26 | 0.316 | –0.316 |
| –21 | 1.000 | 0.000 | | –5 | 0.949 | 0.949 | | 11 | 0.949 | 0.316 | | 27 | 0.000 | 0.000 |
| –20 | 0.316 | 0.316 | | –4 | –0.949 | –0.316 | | 12 | –0.316 | 0.949 | | 28 | 0.000 | 0.000 |
| –19 | 0.316 | –0.949 | | –3 | 0.316 | –0.316 | | 13 | 0.949 | 0.316 | | 29 | 0.000 | 0.000 |
| –18 | –0.316 | –0.949 | | –2 | –0.949 | –0.316 | | 14 | 0.949 | –0.316 | | 30 | 0.000 | 0.000 |
| –17 | –0.316 | 0.316 | | –1 | –0.949 | 0.949 | | 15 | 0.949 | –0.949 | | 31 | 0.000 | 0.000 |

The time domain samples are produced by performing IFFT, cyclically extending, and multiplying with the window function in the same manner as described in G.4.5. The time domain samples are appended with one sample overlap to the SIGNAL field symbol.

## G.7 Generating the additional DATA symbols

The generation of the additional five data symbols follows the same procedure as described in Clause 5. Special attention should be paid to the scrambling of the pilot subcarriers. Table G.23 lists the polarity of the pilot subcarriers and the elements of the sequence $p_{0...126}$ for the DATA symbols. For completeness, the pilots in the SIGNAL are included as well. The symbols are appended one after the other with a one-sample overlap.

**Table G.23—Polarity of the pilot subcarriers**

| i | OFDM symbol | Element of $p_i$ | Pilot at #-21 | Pilot at #-7 | Pilot at #7 | Pilot at #21 |
|---|---|---|---|---|---|---|
| 0 | SIGNAL | 1 | 1.0 +0 j | 1.0 +0 j | 1.0 +0 j | –1.0 +0 j |
| 1 | DATA 1 | 1 | 1.0 +0 j | 1.0 +0 j | 1.0 +0 j | –1.0 +0 j |
| 2 | DATA 2 | 1 | 1.0 +0 j | 1.0 +0 j | 1.0 +0 j | –1.0 +0 j |
| 3 | DATA 3 | 1 | 1.0 +0 j | 1.0 +0 j | 1.0 +0 j | –1.0 +0 j |
| 4 | DATA 4 | –1 | –1.0 +0 j | –1.0 +0 j | –1.0 +0 j | 1.0 +0 j |
| 5 | DATA 5 | –1 | –1.0 +0 j | –1.0 +0 j | –1.0 +0 j | 1.0 +0 j |
| 6 | DATA 6 | –1 | –1.0 +0 j | –1.0 +0 j | –1.0 +0 j | 1.0 +0 j |

Copyright © 2007 IEEE. All rights reserved.

## G.8 The entire packet

The packet in its entirety is shown in Table G.24. The short sequences section, the long sequences section, the SIGNAL field, and the DATA symbols are separated by double lines.

**Table G.24—The entire packet**

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.023 | 0.023 | 1 | –0.132 | 0.002 | 2 | –0.013 | –0.079 | 3 | 0.143 | –0.013 |
| 4 | 0.092 | 0.000 | 5 | 0.143 | –0.013 | 6 | –0.013 | –0.079 | 7 | –0.132 | 0.002 |
| 8 | 0.046 | 0.046 | 9 | 0.002 | –0.132 | 10 | –0.079 | –0.013 | 11 | –0.013 | 0.143 |
| 12 | 0.000 | 0.092 | 13 | –0.013 | 0.143 | 14 | –0.079 | –0.013 | 15 | 0.002 | –0.132 |
| 16 | 0.046 | 0.046 | 17 | –0.132 | 0.002 | 18 | –0.013 | –0.079 | 19 | 0.143 | –0.013 |
| 20 | 0.092 | 0.000 | 21 | 0.143 | –0.013 | 22 | –0.013 | –0.079 | 23 | –0.132 | 0.002 |
| 24 | 0.046 | 0.046 | 25 | 0.002 | –0.132 | 26 | –0.079 | –0.013 | 27 | –0.013 | 0.143 |
| 28 | 0.000 | 0.092 | 29 | –0.013 | 0.143 | 30 | –0.079 | –0.013 | 31 | 0.002 | –0.132 |
| 32 | 0.046 | 0.046 | 33 | –0.132 | 0.002 | 34 | –0.013 | –0.079 | 35 | 0.143 | –0.013 |
| 36 | 0.092 | 0.000 | 37 | 0.143 | –0.013 | 38 | –0.013 | –0.079 | 39 | –0.132 | 0.002 |
| 40 | 0.046 | 0.046 | 41 | 0.002 | –0.132 | 42 | –0.079 | –0.013 | 43 | –0.013 | 0.143 |
| 44 | 0.000 | 0.092 | 45 | –0.013 | 0.143 | 46 | –0.079 | –0.013 | 47 | 0.002 | –0.132 |
| 48 | 0.046 | 0.046 | 49 | –0.132 | 0.002 | 50 | –0.013 | –0.079 | 51 | 0.143 | –0.013 |
| 52 | 0.092 | 0.000 | 53 | 0.143 | –0.013 | 54 | –0.013 | –0.079 | 55 | –0.132 | 0.002 |
| 56 | 0.046 | 0.046 | 57 | 0.002 | –0.132 | 58 | –0.079 | –0.013 | 59 | –0.013 | 0.143 |
| 60 | 0.000 | 0.092 | 61 | –0.013 | 0.143 | 62 | –0.079 | –0.013 | 63 | 0.002 | –0.132 |
| 64 | 0.046 | 0.046 | 65 | –0.132 | 0.002 | 66 | –0.013 | –0.079 | 67 | 0.143 | –0.013 |
| 68 | 0.092 | 0.000 | 69 | 0.143 | –0.013 | 70 | –0.013 | –0.079 | 71 | –0.132 | 0.002 |
| 72 | 0.046 | 0.046 | 73 | 0.002 | –0.132 | 74 | –0.079 | –0.013 | 75 | –0.013 | 0.143 |
| 76 | 0.000 | 0.092 | 77 | –0.013 | 0.143 | 78 | –0.079 | –0.013 | 79 | 0.002 | –0.132 |
| 80 | 0.046 | 0.046 | 81 | –0.132 | 0.002 | 82 | –0.013 | –0.079 | 83 | 0.143 | –0.013 |
| 84 | 0.092 | 0.000 | 85 | 0.143 | –0.013 | 86 | –0.013 | –0.079 | 87 | –0.132 | 0.002 |
| 88 | 0.046 | 0.046 | 89 | 0.002 | –0.132 | 90 | –0.079 | –0.013 | 91 | –0.013 | 0.143 |
| 92 | 0.000 | 0.092 | 93 | –0.013 | 0.143 | 94 | –0.079 | –0.013 | 95 | 0.002 | –0.132 |
| 96 | 0.046 | 0.046 | 97 | –0.132 | 0.002 | 98 | –0.013 | –0.079 | 99 | 0.143 | –0.013 |
| 100 | 0.092 | 0.000 | 101 | 0.143 | –0.013 | 102 | –0.013 | –0.079 | 103 | –0.132 | 0.002 |
| 104 | 0.046 | 0.046 | 105 | 0.002 | –0.132 | 106 | –0.079 | –0.013 | 107 | –0.013 | 0.143 |
| 108 | 0.000 | 0.092 | 109 | –0.013 | 0.143 | 110 | –0.079 | –0.013 | 111 | 0.002 | –0.132 |
| 112 | 0.046 | 0.046 | 113 | –0.132 | 0.002 | 114 | –0.013 | –0.079 | 115 | 0.143 | –0.013 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 116 | 0.092 | 0.000 | 117 | 0.143 | –0.013 | 118 | –0.013 | –0.079 | 119 | –0.132 | 0.002 |
| 120 | 0.046 | 0.046 | 121 | 0.002 | –0.132 | 122 | –0.079 | –0.013 | 123 | –0.013 | 0.143 |
| 124 | 0.000 | 0.092 | 125 | –0.013 | 0.143 | 126 | –0.079 | –0.013 | 127 | 0.002 | –0.132 |
| 128 | 0.046 | 0.046 | 129 | –0.132 | 0.002 | 130 | –0.013 | –0.079 | 131 | 0.143 | –0.013 |
| 132 | 0.092 | 0.000 | 133 | 0.143 | –0.013 | 134 | –0.013 | –0.079 | 135 | –0.132 | 0.002 |
| 136 | 0.046 | 0.046 | 137 | 0.002 | –0.132 | 138 | –0.079 | –0.013 | 139 | –0.013 | 0.143 |
| 140 | 0.000 | 0.092 | 141 | –0.013 | 0.143 | 142 | –0.079 | –0.013 | 143 | 0.002 | –0.132 |
| 144 | 0.046 | 0.046 | 145 | –0.132 | 0.002 | 146 | –0.013 | –0.079 | 147 | 0.143 | –0.013 |
| 148 | 0.092 | 0.000 | 149 | 0.143 | –0.013 | 150 | –0.013 | –0.079 | 151 | –0.132 | 0.002 |
| 152 | 0.046 | 0.046 | 153 | 0.002 | –0.132 | 154 | –0.079 | –0.013 | 155 | –0.013 | 0.143 |
| 156 | 0.000 | 0.092 | 157 | –0.013 | 0.143 | 158 | –0.079 | –0.013 | 159 | 0.002 | –0.132 |
| 160 | –0.055 | 0.023 | 161 | 0.012 | –0.098 | 162 | 0.092 | –0.106 | 163 | –0.092 | –0.115 |
| 164 | –0.003 | –0.054 | 165 | 0.075 | 0.074 | 166 | –0.127 | 0.021 | 167 | –0.122 | 0.017 |
| 168 | –0.035 | 0.151 | 169 | –0.056 | 0.022 | 170 | –0.060 | –0.081 | 171 | 0.070 | –0.014 |
| 172 | 0.082 | –0.092 | 173 | –0.131 | –0.065 | 174 | –0.057 | –0.039 | 175 | 0.037 | –0.098 |
| 176 | 0.062 | 0.062 | 177 | 0.119 | 0.004 | 178 | –0.022 | –0.161 | 179 | 0.059 | 0.015 |
| 180 | 0.024 | 0.059 | 181 | –0.137 | 0.047 | 182 | 0.001 | 0.115 | 183 | 0.053 | –0.004 |
| 184 | 0.098 | 0.026 | 185 | –0.038 | 0.106 | 186 | –0.115 | 0.055 | 187 | 0.060 | 0.088 |
| 188 | 0.021 | –0.028 | 189 | 0.097 | –0.083 | 190 | 0.040 | 0.111 | 191 | –0.005 | 0.120 |
| 192 | 0.156 | 0.000 | 193 | –0.005 | –0.120 | 194 | 0.040 | –0.111 | 195 | 0.097 | 0.083 |
| 196 | 0.021 | 0.028 | 197 | 0.060 | –0.088 | 198 | –0.115 | –0.055 | 199 | –0.038 | –0.106 |
| 200 | 0.098 | –0.026 | 201 | 0.053 | 0.004 | 202 | 0.001 | –0.115 | 203 | –0.137 | –0.047 |
| 204 | 0.024 | –0.059 | 205 | 0.059 | –0.015 | 206 | –0.022 | 0.161 | 207 | 0.119 | –0.004 |
| 208 | 0.062 | –0.062 | 209 | 0.037 | 0.098 | 210 | –0.057 | 0.039 | 211 | –0.131 | 0.065 |
| 212 | 0.082 | 0.092 | 213 | 0.070 | 0.014 | 214 | –0.060 | 0.081 | 215 | –0.056 | –0.022 |
| 216 | –0.035 | –0.151 | 217 | –0.122 | –0.017 | 218 | –0.127 | –0.021 | 219 | 0.075 | –0.074 |
| 220 | –0.003 | 0.054 | 221 | –0.092 | 0.115 | 222 | 0.092 | 0.106 | 223 | 0.012 | 0.098 |
| 224 | –0.156 | 0.000 | 225 | 0.012 | –0.098 | 226 | 0.092 | –0.106 | 227 | –0.092 | –0.115 |
| 228 | –0.003 | –0.054 | 229 | 0.075 | 0.074 | 230 | –0.127 | 0.021 | 231 | –0.122 | 0.017 |
| 232 | –0.035 | 0.151 | 233 | –0.056 | 0.022 | 234 | –0.060 | –0.081 | 235 | 0.070 | –0.014 |
| 236 | 0.082 | –0.092 | 237 | –0.131 | –0.065 | 238 | –0.057 | –0.039 | 239 | 0.037 | –0.098 |
| 240 | 0.062 | 0.062 | 241 | 0.119 | 0.004 | 242 | –0.022 | –0.161 | 243 | 0.059 | 0.015 |
| 244 | 0.024 | 0.059 | 245 | –0.137 | 0.047 | 246 | 0.001 | 0.115 | 247 | 0.053 | –0.004 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 248 | 0.098 | 0.026 | 249 | –0.038 | 0.106 | 250 | –0.115 | 0.055 | 251 | 0.060 | 0.088 |
| 252 | 0.021 | –0.028 | 253 | 0.097 | –0.083 | 254 | 0.040 | 0.111 | 255 | –0.005 | 0.120 |
| 256 | 0.156 | 0.000 | 257 | –0.005 | –0.120 | 258 | 0.040 | –0.111 | 259 | 0.097 | 0.083 |
| 260 | 0.021 | 0.028 | 261 | 0.060 | –0.088 | 262 | –0.115 | –0.055 | 263 | –0.038 | –0.106 |
| 264 | 0.098 | –0.026 | 265 | 0.053 | 0.004 | 266 | 0.001 | –0.115 | 267 | –0.137 | –0.047 |
| 268 | 0.024 | –0.059 | 269 | 0.059 | –0.015 | 270 | –0.022 | 0.161 | 271 | 0.119 | –0.004 |
| 272 | 0.062 | –0.062 | 273 | 0.037 | 0.098 | 274 | –0.057 | 0.039 | 275 | –0.131 | 0.065 |
| 276 | 0.082 | 0.092 | 277 | 0.070 | 0.014 | 278 | –0.060 | 0.081 | 279 | –0.056 | –0.022 |
| 280 | –0.035 | –0.151 | 281 | –0.122 | –0.017 | 282 | –0.127 | –0.021 | 283 | 0.075 | –0.074 |
| 284 | –0.003 | 0.054 | 285 | –0.092 | 0.115 | 286 | 0.092 | 0.106 | 287 | 0.012 | 0.098 |
| 288 | –0.156 | 0.000 | 289 | 0.012 | –0.098 | 290 | 0.092 | –0.106 | 291 | –0.092 | –0.115 |
| 292 | –0.003 | –0.054 | 293 | 0.075 | 0.074 | 294 | –0.127 | 0.021 | 295 | –0.122 | 0.017 |
| 296 | –0.035 | 0.151 | 297 | –0.056 | 0.022 | 298 | –0.060 | –0.081 | 299 | 0.070 | –0.014 |
| 300 | 0.082 | –0.092 | 301 | –0.131 | –0.065 | 302 | –0.057 | –0.039 | 303 | 0.037 | –0.098 |
| 304 | 0.062 | 0.062 | 305 | 0.119 | 0.004 | 306 | –0.022 | –0.161 | 307 | 0.059 | 0.015 |
| 308 | 0.024 | 0.059 | 309 | –0.137 | 0.047 | 310 | 0.001 | 0.115 | 311 | 0.053 | –0.004 |
| 312 | 0.098 | 0.026 | 313 | –0.038 | 0.106 | 314 | –0.115 | 0.055 | 315 | 0.060 | 0.088 |
| 316 | 0.021 | –0.028 | 317 | 0.097 | –0.083 | 318 | 0.040 | 0.111 | 319 | –0.005 | 0.120 |
| 320 | 0.109 | 0.000 | 321 | 0.033 | –0.044 | 322 | –0.002 | –0.038 | 323 | –0.081 | 0.084 |
| 324 | 0.007 | –0.100 | 325 | –0.001 | –0.113 | 326 | –0.021 | –0.005 | 327 | 0.136 | –0.105 |
| 328 | 0.098 | –0.044 | 329 | 0.011 | –0.002 | 330 | –0.033 | 0.044 | 331 | –0.060 | 0.124 |
| 332 | 0.010 | 0.097 | 333 | 0.000 | –0.008 | 334 | 0.018 | –0.083 | 335 | –0.069 | 0.027 |
| 336 | –0.219 | 0.000 | 337 | –0.069 | –0.027 | 338 | 0.018 | 0.083 | 339 | 0.000 | 0.008 |
| 340 | 0.010 | –0.097 | 341 | –0.060 | –0.124 | 342 | –0.033 | –0.044 | 343 | 0.011 | 0.002 |
| 344 | 0.098 | 0.044 | 345 | 0.136 | 0.105 | 346 | –0.021 | 0.005 | 347 | –0.001 | 0.113 |
| 348 | 0.007 | 0.100 | 349 | –0.081 | –0.084 | 350 | –0.002 | 0.038 | 351 | 0.033 | 0.044 |
| 352 | 0.062 | 0.000 | 353 | 0.057 | 0.052 | 354 | 0.016 | 0.174 | 355 | 0.035 | 0.116 |
| 356 | –0.051 | –0.202 | 357 | 0.011 | 0.036 | 358 | 0.089 | 0.209 | 359 | –0.049 | –0.008 |
| 360 | –0.035 | 0.044 | 361 | 0.017 | –0.059 | 362 | 0.053 | –0.017 | 363 | 0.099 | 0.100 |
| 364 | 0.034 | –0.148 | 365 | –0.003 | –0.094 | 366 | –0.120 | 0.042 | 367 | –0.136 | –0.070 |
| 368 | –0.031 | 0.000 | 369 | –0.136 | 0.070 | 370 | –0.120 | –0.042 | 371 | –0.003 | 0.094 |
| 372 | 0.034 | 0.148 | 373 | 0.099 | –0.100 | 374 | 0.053 | 0.017 | 375 | 0.017 | 0.059 |
| 376 | –0.035 | –0.044 | 377 | –0.049 | 0.008 | 378 | 0.089 | –0.209 | 379 | 0.011 | –0.036 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 380 | –0.051 | 0.202 | 381 | 0.035 | –0.116 | 382 | 0.016 | –0.174 | 383 | 0.057 | –0.052 |
| 384 | 0.062 | 0.000 | 385 | 0.033 | –0.044 | 386 | –0.002 | –0.038 | 387 | –0.081 | 0.084 |
| 388 | 0.007 | –0.100 | 389 | –0.001 | –0.113 | 390 | –0.021 | –0.005 | 391 | 0.136 | –0.105 |
| 392 | 0.098 | –0.044 | 393 | 0.011 | –0.002 | 394 | –0.033 | 0.044 | 395 | –0.060 | 0.124 |
| 396 | 0.010 | 0.097 | 397 | 0.000 | –0.008 | 398 | 0.018 | –0.083 | 399 | –0.069 | 0.027 |
| 400 | –0.139 | 0.050 | 401 | 0.004 | 0.014 | 402 | 0.011 | –0.100 | 403 | –0.097 | –0.020 |
| 404 | 0.062 | 0.081 | 405 | 0.124 | 0.139 | 406 | 0.104 | –0.015 | 407 | 0.173 | –0.140 |
| 408 | –0.040 | 0.006 | 409 | –0.133 | 0.009 | 410 | –0.002 | –0.043 | 411 | –0.047 | 0.092 |
| 412 | –0.109 | 0.082 | 413 | –0.024 | 0.010 | 414 | 0.096 | 0.019 | 415 | 0.019 | –0.023 |
| 416 | –0.087 | –0.049 | 417 | 0.002 | 0.058 | 418 | –0.021 | 0.228 | 419 | –0.103 | 0.023 |
| 420 | –0.019 | –0.175 | 421 | 0.018 | 0.132 | 422 | –0.071 | 0.160 | 423 | –0.153 | –0.062 |
| 424 | –0.107 | 0.028 | 425 | 0.055 | 0.140 | 426 | 0.070 | 0.103 | 427 | –0.056 | 0.025 |
| 428 | –0.043 | 0.002 | 429 | 0.016 | –0.118 | 430 | 0.026 | –0.071 | 431 | 0.033 | 0.177 |
| 432 | 0.020 | –0.021 | 433 | 0.035 | –0.088 | 434 | –0.008 | 0.101 | 435 | –0.035 | –0.010 |
| 436 | 0.065 | 0.030 | 437 | 0.092 | –0.034 | 438 | 0.032 | –0.123 | 439 | –0.018 | 0.092 |
| 440 | 0.000 | –0.006 | 441 | –0.006 | –0.056 | 442 | –0.019 | 0.040 | 443 | 0.053 | –0.131 |
| 444 | 0.022 | –0.133 | 445 | 0.104 | –0.032 | 446 | 0.163 | –0.045 | 447 | –0.105 | –0.030 |
| 448 | –0.110 | –0.069 | 449 | –0.008 | –0.092 | 450 | –0.049 | –0.043 | 451 | 0.085 | –0.017 |
| 452 | 0.090 | 0.063 | 453 | 0.015 | 0.153 | 454 | 0.049 | 0.094 | 455 | 0.011 | 0.034 |
| 456 | –0.012 | 0.012 | 457 | –0.015 | –0.017 | 458 | –0.061 | 0.031 | 459 | –0.070 | –0.040 |
| 460 | 0.011 | –0.109 | 461 | 0.037 | –0.060 | 462 | –0.003 | –0.178 | 463 | –0.007 | –0.128 |
| 464 | –0.059 | 0.100 | 465 | 0.004 | 0.014 | 466 | 0.011 | –0.100 | 467 | –0.097 | –0.020 |
| 468 | 0.062 | 0.081 | 469 | 0.124 | 0.139 | 470 | 0.104 | –0.015 | 471 | 0.173 | –0.140 |
| 472 | –0.040 | 0.006 | 473 | –0.133 | 0.009 | 474 | –0.002 | –0.043 | 475 | –0.047 | 0.092 |
| 476 | –0.109 | 0.082 | 477 | –0.024 | 0.010 | 478 | 0.096 | 0.019 | 479 | 0.019 | –0.023 |
| 480 | –0.058 | 0.016 | 481 | –0.096 | –0.045 | 482 | –0.110 | 0.003 | 483 | –0.070 | 0.216 |
| 484 | –0.040 | 0.059 | 485 | 0.010 | –0.056 | 486 | 0.034 | 0.065 | 487 | 0.117 | 0.033 |
| 488 | 0.078 | –0.133 | 489 | –0.043 | –0.146 | 490 | 0.158 | –0.071 | 491 | 0.254 | –0.021 |
| 492 | 0.068 | 0.117 | 493 | –0.044 | 0.114 | 494 | –0.035 | 0.041 | 495 | 0.085 | 0.070 |
| 496 | 0.120 | 0.010 | 497 | 0.057 | 0.055 | 498 | 0.063 | 0.188 | 499 | 0.091 | 0.149 |
| 500 | –0.017 | –0.039 | 501 | –0.078 | –0.075 | 502 | 0.049 | 0.079 | 503 | –0.014 | –0.007 |
| 504 | 0.030 | –0.027 | 505 | 0.080 | 0.054 | 506 | –0.186 | –0.067 | 507 | –0.039 | –0.027 |
| 508 | 0.043 | –0.072 | 509 | –0.092 | –0.089 | 510 | 0.029 | 0.105 | 511 | –0.144 | 0.003 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 512 | –0.069 | –0.041 | 513 | 0.132 | 0.057 | 514 | –0.126 | 0.070 | 515 | –0.031 | 0.109 |
| 516 | 0.161 | –0.009 | 517 | 0.056 | –0.046 | 518 | –0.004 | 0.028 | 519 | –0.049 | 0.000 |
| 520 | –0.078 | –0.005 | 521 | 0.015 | –0.087 | 522 | 0.149 | –0.104 | 523 | –0.021 | –0.051 |
| 524 | –0.154 | –0.106 | 525 | 0.024 | 0.030 | 526 | 0.046 | 0.123 | 527 | –0.004 | –0.098 |
| 528 | –0.061 | –0.128 | 529 | –0.024 | –0.038 | 530 | 0.066 | –0.048 | 531 | –0.067 | 0.027 |
| 532 | 0.054 | –0.050 | 533 | 0.171 | –0.049 | 534 | –0.108 | 0.132 | 535 | –0.161 | –0.019 |
| 536 | –0.070 | –0.072 | 537 | –0.177 | 0.049 | 538 | –0.172 | –0.050 | 539 | 0.051 | –0.075 |
| 540 | 0.122 | –0.057 | 541 | 0.009 | –0.044 | 542 | –0.012 | –0.021 | 543 | 0.004 | 0.009 |
| 544 | –0.030 | 0.081 | 545 | –0.096 | –0.045 | 546 | –0.110 | 0.003 | 547 | –0.070 | 0.216 |
| 548 | –0.040 | 0.059 | 549 | 0.010 | –0.056 | 550 | 0.034 | 0.065 | 551 | 0.117 | 0.033 |
| 552 | 0.078 | –0.133 | 553 | –0.043 | –0.146 | 554 | 0.158 | –0.071 | 555 | 0.254 | –0.021 |
| 556 | 0.068 | 0.117 | 557 | –0.044 | 0.114 | 558 | –0.035 | 0.041 | 559 | 0.085 | 0.070 |
| 560 | 0.001 | 0.011 | 561 | –0.099 | –0.048 | 562 | 0.054 | –0.196 | 563 | 0.124 | 0.035 |
| 564 | 0.092 | 0.045 | 565 | –0.037 | –0.066 | 566 | –0.021 | –0.004 | 567 | 0.042 | –0.065 |
| 568 | 0.061 | 0.048 | 569 | 0.046 | 0.004 | 570 | –0.063 | –0.045 | 571 | –0.102 | 0.152 |
| 572 | –0.039 | –0.019 | 573 | –0.005 | –0.106 | 574 | 0.083 | 0.031 | 575 | 0.226 | 0.028 |
| 576 | 0.140 | –0.010 | 577 | –0.132 | –0.033 | 578 | –0.116 | 0.088 | 579 | 0.023 | 0.052 |
| 580 | –0.171 | –0.080 | 581 | –0.246 | –0.025 | 582 | –0.062 | –0.038 | 583 | –0.055 | –0.062 |
| 584 | –0.004 | –0.060 | 585 | 0.034 | 0.000 | 586 | –0.030 | 0.021 | 587 | 0.075 | –0.122 |
| 588 | 0.043 | –0.080 | 589 | –0.022 | 0.041 | 590 | 0.026 | 0.013 | 591 | –0.031 | –0.018 |
| 592 | 0.059 | 0.008 | 593 | 0.109 | 0.078 | 594 | 0.002 | 0.101 | 595 | –0.016 | 0.054 |
| 596 | –0.059 | 0.070 | 597 | 0.017 | 0.114 | 598 | 0.104 | –0.034 | 599 | –0.024 | –0.059 |
| 600 | –0.081 | 0.051 | 601 | –0.040 | –0.069 | 602 | –0.069 | 0.058 | 603 | –0.067 | 0.117 |
| 604 | 0.007 | –0.131 | 605 | 0.009 | 0.028 | 606 | 0.075 | 0.117 | 607 | 0.118 | 0.030 |
| 608 | –0.041 | 0.148 | 609 | 0.005 | 0.098 | 610 | 0.026 | 0.002 | 611 | –0.116 | 0.045 |
| 612 | –0.020 | 0.084 | 613 | 0.101 | 0.006 | 614 | 0.205 | –0.064 | 615 | 0.073 | –0.063 |
| 616 | –0.174 | –0.118 | 617 | –0.024 | 0.026 | 618 | –0.041 | 0.129 | 619 | –0.042 | –0.053 |
| 620 | 0.148 | –0.126 | 621 | –0.030 | –0.049 | 622 | –0.015 | –0.021 | 623 | 0.089 | –0.069 |
| 624 | –0.119 | 0.011 | 625 | –0.099 | –0.048 | 626 | 0.054 | –0.196 | 627 | 0.124 | 0.035 |
| 628 | 0.092 | 0.045 | 629 | –0.037 | –0.066 | 630 | –0.021 | –0.004 | 631 | 0.042 | –0.065 |
| 632 | 0.061 | 0.048 | 633 | 0.046 | 0.004 | 634 | –0.063 | –0.045 | 635 | –0.102 | 0.152 |
| 636 | –0.039 | –0.019 | 637 | –0.005 | –0.106 | 638 | 0.083 | 0.031 | 639 | 0.226 | 0.028 |
| 640 | 0.085 | –0.065 | 641 | 0.034 | –0.142 | 642 | 0.004 | –0.012 | 643 | 0.126 | –0.043 |


Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 644 | 0.055 | 0.068 | 645 | –0.020 | 0.077 | 646 | 0.008 | –0.056 | 647 | –0.034 | 0.046 |
| 648 | –0.040 | –0.134 | 649 | –0.056 | –0.131 | 650 | 0.014 | 0.097 | 651 | 0.045 | –0.009 |
| 652 | –0.113 | –0.170 | 653 | –0.065 | –0.230 | 654 | 0.065 | –0.011 | 655 | 0.011 | 0.048 |
| 656 | –0.091 | –0.059 | 657 | –0.110 | 0.024 | 658 | 0.074 | –0.034 | 659 | 0.124 | 0.022 |
| 660 | –0.037 | 0.071 | 661 | 0.015 | 0.002 | 662 | 0.028 | 0.099 | 663 | –0.062 | 0.068 |
| 664 | 0.064 | 0.016 | 665 | 0.078 | 0.156 | 666 | 0.009 | 0.219 | 667 | 0.147 | 0.024 |
| 668 | 0.106 | 0.030 | 669 | –0.080 | 0.143 | 670 | –0.049 | –0.100 | 671 | –0.036 | –0.082 |
| 672 | –0.089 | 0.021 | 673 | –0.070 | –0.029 | 674 | –0.086 | 0.048 | 675 | –0.066 | –0.015 |
| 676 | –0.024 | 0.002 | 677 | –0.030 | –0.023 | 678 | –0.032 | 0.020 | 679 | –0.002 | 0.212 |
| 680 | 0.158 | –0.024 | 681 | 0.141 | –0.119 | 682 | –0.146 | 0.058 | 683 | –0.155 | 0.083 |
| 684 | –0.002 | –0.030 | 685 | 0.018 | –0.129 | 686 | 0.012 | –0.018 | 687 | –0.008 | –0.037 |
| 688 | 0.031 | 0.040 | 689 | 0.023 | 0.097 | 690 | 0.014 | –0.039 | 691 | 0.050 | 0.019 |
| 692 | –0.072 | –0.141 | 693 | –0.023 | –0.051 | 694 | 0.024 | 0.099 | 695 | –0.127 | –0.116 |
| 696 | 0.094 | 0.102 | 697 | 0.183 | 0.098 | 698 | –0.040 | –0.020 | 699 | 0.065 | 0.077 |
| 700 | 0.088 | –0.147 | 701 | –0.039 | –0.059 | 702 | –0.057 | 0.124 | 703 | –0.077 | 0.020 |
| 704 | 0.030 | –0.120 | 705 | 0.034 | –0.142 | 706 | 0.004 | –0.012 | 707 | 0.126 | –0.043 |
| 708 | 0.055 | 0.068 | 709 | –0.020 | 0.077 | 710 | 0.008 | –0.056 | 711 | –0.034 | 0.046 |
| 712 | –0.040 | –0.134 | 713 | –0.056 | –0.131 | 714 | 0.014 | 0.097 | 715 | 0.045 | –0.009 |
| 716 | –0.113 | –0.170 | 717 | –0.065 | –0.230 | 718 | 0.065 | –0.011 | 719 | 0.011 | 0.048 |
| 720 | –0.026 | –0.021 | 721 | –0.002 | 0.041 | 722 | 0.001 | 0.071 | 723 | –0.037 | –0.117 |
| 724 | –0.106 | –0.062 | 725 | 0.002 | 0.057 | 726 | –0.008 | –0.011 | 727 | 0.019 | 0.072 |
| 728 | 0.016 | 0.059 | 729 | –0.065 | –0.077 | 730 | 0.142 | –0.062 | 731 | 0.087 | 0.025 |
| 732 | –0.003 | –0.103 | 733 | 0.107 | –0.152 | 734 | –0.054 | 0.036 | 735 | –0.030 | –0.003 |
| 736 | 0.058 | –0.020 | 737 | –0.028 | 0.007 | 738 | –0.027 | –0.099 | 739 | 0.049 | –0.075 |
| 740 | 0.174 | 0.031 | 741 | 0.134 | 0.156 | 742 | 0.060 | 0.077 | 743 | –0.010 | –0.022 |
| 744 | –0.084 | 0.040 | 745 | –0.074 | 0.011 | 746 | –0.163 | 0.054 | 747 | –0.052 | –0.008 |
| 748 | 0.076 | –0.042 | 749 | 0.043 | 0.101 | 750 | 0.058 | –0.018 | 751 | 0.003 | –0.090 |
| 752 | 0.059 | –0.018 | 753 | 0.023 | –0.031 | 754 | 0.007 | –0.017 | 755 | 0.066 | –0.017 |
| 756 | –0.135 | –0.098 | 757 | –0.056 | –0.081 | 758 | 0.089 | 0.154 | 759 | 0.120 | 0.122 |
| 760 | 0.102 | 0.001 | 761 | –0.141 | 0.102 | 762 | 0.006 | –0.011 | 763 | 0.057 | –0.039 |
| 764 | –0.059 | 0.066 | 765 | 0.132 | 0.111 | 766 | 0.012 | 0.114 | 767 | 0.047 | –0.106 |
| 768 | 0.160 | –0.099 | 769 | –0.076 | 0.084 | 770 | –0.049 | 0.073 | 771 | 0.005 | –0.086 |
| 772 | –0.052 | –0.108 | 773 | –0.073 | 0.129 | 774 | –0.129 | –0.034 | 775 | –0.153 | –0.111 |

Copyright © 2007 IEEE. All rights reserved.

**Table G.24—The entire packet *(continued)***

| ## | Re | Im | ## | Re | Im | ## | Re | Im | ## | Re | Im |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 776 | –0.193 | 0.098 | 777 | –0.107 | –0.068 | 778 | 0.004 | –0.009 | 779 | –0.039 | 0.024 |
| 780 | –0.054 | –0.079 | 781 | 0.024 | 0.084 | 782 | 0.052 | –0.002 | 783 | 0.028 | –0.044 |
| 784 | 0.040 | 0.018 | 785 | –0.002 | 0.041 | 786 | 0.001 | 0.071 | 787 | –0.037 | –0.117 |
| 788 | –0.106 | –0.062 | 789 | 0.002 | 0.057 | 790 | –0.008 | –0.011 | 791 | 0.019 | 0.072 |
| 792 | 0.016 | 0.059 | 793 | –0.065 | –0.077 | 794 | 0.142 | –0.062 | 795 | 0.087 | 0.025 |
| 796 | –0.003 | –0.103 | 797 | 0.107 | –0.152 | 798 | –0.054 | 0.036 | 799 | –0.030 | –0.003 |
| 800 | 0.039 | –0.090 | 801 | 0.029 | 0.025 | 802 | 0.086 | –0.029 | 803 | 0.087 | –0.082 |
| 804 | 0.003 | –0.036 | 805 | –0.096 | –0.089 | 806 | –0.073 | –0.046 | 807 | 0.105 | –0.020 |
| 808 | 0.193 | 0.018 | 809 | –0.053 | –0.073 | 810 | –0.118 | –0.149 | 811 | 0.019 | –0.019 |
| 812 | –0.042 | 0.026 | 813 | 0.041 | 0.009 | 814 | 0.028 | –0.076 | 815 | –0.038 | –0.068 |
| 816 | –0.011 | 0.010 | 817 | –0.134 | –0.064 | 818 | 0.069 | –0.067 | 819 | 0.057 | 0.006 |
| 820 | –0.134 | 0.098 | 821 | 0.152 | 0.036 | 822 | 0.041 | –0.085 | 823 | –0.099 | –0.049 |
| 824 | 0.089 | –0.099 | 825 | –0.046 | 0.018 | 826 | –0.112 | 0.135 | 827 | –0.064 | 0.018 |
| 828 | –0.022 | 0.053 | 829 | 0.041 | 0.077 | 830 | –0.021 | 0.145 | 831 | 0.007 | 0.179 |
| 832 | 0.059 | 0.041 | 833 | 0.023 | 0.064 | 834 | 0.062 | 0.022 | 835 | 0.110 | –0.081 |
| 836 | –0.016 | –0.054 | 837 | –0.014 | –0.017 | 838 | 0.171 | 0.008 | 839 | 0.070 | –0.027 |
| 840 | –0.015 | 0.002 | 841 | –0.012 | 0.053 | 842 | –0.125 | 0.009 | 843 | –0.040 | 0.012 |
| 844 | 0.036 | 0.114 | 845 | 0.007 | 0.090 | 846 | –0.016 | –0.082 | 847 | –0.008 | –0.013 |
| 848 | 0.091 | 0.030 | 849 | 0.072 | –0.068 | 850 | 0.051 | 0.063 | 851 | –0.004 | 0.049 |
| 852 | –0.130 | –0.048 | 853 | –0.121 | 0.061 | 854 | –0.095 | 0.078 | 855 | 0.011 | 0.005 |
| 856 | 0.049 | 0.001 | 857 | –0.014 | –0.011 | 858 | 0.009 | –0.063 | 859 | –0.031 | 0.040 |
| 860 | –0.011 | 0.004 | 861 | –0.033 | –0.111 | 862 | –0.115 | 0.137 | 863 | –0.025 | 0.049 |
| 864 | 0.020 | –0.160 | 865 | 0.029 | 0.025 | 866 | 0.086 | –0.029 | 867 | 0.087 | –0.082 |
| 868 | 0.003 | –0.036 | 869 | –0.096 | –0.089 | 870 | –0.073 | –0.046 | 871 | 0.105 | –0.020 |
| 872 | 0.193 | 0.018 | 873 | –0.053 | –0.073 | 874 | –0.118 | –0.149 | 875 | 0.019 | –0.019 |
| 876 | –0.042 | 0.026 | 877 | 0.041 | 0.009 | 878 | 0.028 | –0.076 | 879 | –0.038 | –0.068 |
| 880 | –0.006 | 0.005 | | | | | | | | | |


Copyright © 2007 IEEE. All rights reserved.

# Annex H

(informative)

# RSNA reference implementations and test vectors

## H.1 TKIP temporal key mixing function reference implementation and test vector

This clause provides a C-language reference implementation of the temporal key mixing function.

```
/*********************************************************************

   Contents:    Generate IEEE 802.11 per-frame RC4 key hash test vectors
   Date:        April 19, 2002
   Notes:
   This code is written for pedagogical purposes, NOT for performance.

*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <time.h>

typedef unsigned char  byte;   /*  8-bit byte (octet)  */
typedef unsigned short u16b;   /* 16-bit unsigned word */
typedef unsigned long  u32b;   /* 32-bit unsigned word */

/* macros for extraction/creation of byte/u16b values  */
#define RotR1(v16)   ((((v16) >> 1) & 0x7FFF) ^ (((v16) & 1) << 15))
#define   Lo8(v16)   ((byte)( (v16)       & 0x00FF))
#define   Hi8(v16)   ((byte)(((v16) >> 8) & 0x00FF))
#define  Lo16(v32)   ((u16b)( (v32)       & 0xFFFF))
#define  Hi16(v32)   ((u16b)(((v32) >>16) & 0xFFFF))
#define  Mk16(hi,lo) ((lo) ^ (((u16b)(hi)) << 8))

/* select the Nth 16-bit word of the Temporal Key byte array TK[]   */
#define  TK16(N)     Mk16(TK[2*(N)+1],TK[2*(N)])

/* S-box lookup: 16 bits --> 16 bits */
#define _S_(v16)     (Sbox[0][Lo8(v16)] ^ Sbox[1][Hi8(v16)])

/* fixed algorithm "parameters" */
#define PHASE1_LOOP_CNT   8    /* this needs to be "big enough"     */
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1155 of 1232

```
#define TA_SIZE           6    /*  48-bit transmitter address       */
#define TK_SIZE          16    /* 128-bit Temporal Key              */
#define P1K_SIZE         10    /*  80-bit Phase1 key                */
#define RC4_KEY_SIZE     16    /* 128-bit RC4KEY (104 bits unknown) */

/* configuration settings */
#define DO_SANITY_CHECK   1    /* validate properties of S-box?     */

/* 2-byte by 2-byte subset of the full AES S-box table */
const u16b Sbox[2][256]=       /* Sbox for hash (can be in ROM)     */
{ {
   0xC6A5,0xF884,0xEE99,0xF68D,0xFF0D,0xD6BD,0xDEB1,0x9154,
   0x6050,0x0203,0xCEA9,0x567D,0xE719,0xB562,0x4DE6,0xEC9A,
   0x8F45,0x1F9D,0x8940,0xFA87,0xEF15,0xB2EB,0x8EC9,0xFB0B,
   0x41EC,0xB367,0x5FFD,0x45EA,0x23BF,0x53F7,0xE496,0x9B5B,
   0x75C2,0xE11C,0x3DAE,0x4C6A,0x6C5A,0x7E41,0xF502,0x834F,
   0x685C,0x51F4,0xD134,0xF908,0xE293,0xAB73,0x6253,0x2A3F,
   0x080C,0x9552,0x4665,0x9D5E,0x3028,0x37A1,0x0A0F,0x2FB5,
   0x0E09,0x2436,0x1B9B,0xDF3D,0xCD26,0x4E69,0x7FCD,0xEA9F,
   0x121B,0x1D9E,0x5874,0x342E,0x362D,0xDCB2,0xB4EE,0x5BFB,
   0xA4F6,0x764D,0xB761,0x7DCE,0x527B,0xDD3E,0x5E71,0x1397,
   0xA6F5,0xB968,0x0000,0xC12C,0x4060,0xE31F,0x79C8,0xB6ED,
   0xD4BE,0x8D46,0x67D9,0x724B,0x94DE,0x98D4,0xB0E8,0x854A,
   0xBB6B,0xC52A,0x4FE5,0xED16,0x86C5,0x9AD7,0x6655,0x1194,
   0x8ACF,0xE910,0x0406,0xFE81,0xA0F0,0x7844,0x25BA,0x4BE3,
   0xA2F3,0x5DFE,0x80C0,0x058A,0x3FAD,0x21BC,0x7048,0xF104,
   0x63DF,0x77C1,0xAF75,0x4263,0x2030,0xE51A,0xFD0E,0xBF6D,
   0x814C,0x1814,0x2635,0xC32F,0xBEE1,0x35A2,0x88CC,0x2E39,
   0x9357,0x55F2,0xFC82,0x7A47,0xC8AC,0xBAE7,0x322B,0xE695,
   0xC0A0,0x1998,0x9ED1,0xA37F,0x4466,0x547E,0x3BAB,0x0B83,
   0x8CCA,0xC729,0x6BD3,0x283C,0xA779,0xBCE2,0x161D,0xAD76,
   0xDB3B,0x6456,0x744E,0x141E,0x92DB,0x0C0A,0x486C,0xB8E4,
   0x9F5D,0xBD6E,0x43EF,0xC4A6,0x39A8,0x31A4,0xD337,0xF28B,
   0xD532,0x8B43,0x6E59,0xDAB7,0x018C,0xB164,0x9CD2,0x49E0,
   0xD8B4,0xACFA,0xF307,0xCF25,0xCAAF,0xF48E,0x47E9,0x1018,
   0x6FD5,0xF088,0x4A6F,0x5C72,0x3824,0x57F1,0x73C7,0x9751,
   0xCB23,0xA17C,0xE89C,0x3E21,0x96DD,0x61DC,0x0D86,0x0F85,
   0xE090,0x7C42,0x71C4,0xCCAA,0x90D8,0x0605,0xF701,0x1C12,
   0xC2A3,0x6A5F,0xAEF9,0x69D0,0x1791,0x9958,0x3A27,0x27B9,
   0xD938,0xEB13,0x2BB3,0x2233,0xD2BB,0xA970,0x0789,0x33A7,
   0x2DB6,0x3C22,0x1592,0xC920,0x8749,0xAAFF,0x5078,0xA57A,
   0x038F,0x59F8,0x0980,0x1A17,0x65DA,0xD731,0x84C6,0xD0B8,
   0x82C3,0x29B0,0x5A77,0x1E11,0x7BCB,0xA8FC,0x6DD6,0x2C3A,
  },

  {  /* second half of table is byte-reversed version of first! */
```


Copyright © 2007 IEEE. All rights reserved.

```
    0xA5C6,0x84F8,0x99EE,0x8DF6,0x0DFF,0xBDD6,0xB1DE,0x5491,
    0x5060,0x0302,0xA9CE,0x7D56,0x19E7,0x62B5,0xE64D,0x9AEC,
    0x458F,0x9D1F,0x4089,0x87FA,0x15EF,0xEBB2,0xC98E,0x0BFB,
    0xEC41,0x67B3,0xFD5F,0xEA45,0xBF23,0xF753,0x96E4,0x5B9B,
    0xC275,0x1CE1,0xAE3D,0x6A4C,0x5A6C,0x417E,0x02F5,0x4F83,
    0x5C68,0xF451,0x34D1,0x08F9,0x93E2,0x73AB,0x5362,0x3F2A,
    0x0C08,0x5295,0x6546,0x5E9D,0x2830,0xA137,0x0F0A,0xB52F,
    0x090E,0x3624,0x9B1B,0x3DDF,0x26CD,0x694E,0xCD7F,0x9FEA,
    0x1B12,0x9E1D,0x7458,0x2E34,0x2D36,0xB2DC,0xEEB4,0xFB5B,
    0xF6A4,0x4D76,0x61B7,0xCE7D,0x7B52,0x3EDD,0x715E,0x9713,
    0xF5A6,0x68B9,0x0000,0x2CC1,0x6040,0x1FE3,0xC879,0xEDB6,
    0xBED4,0x468D,0xD967,0x4B72,0xDE94,0xD498,0xE8B0,0x4A85,
    0x6BBB,0x2AC5,0xE54F,0x16ED,0xC586,0xD79A,0x5566,0x9411,
    0xCF8A,0x10E9,0x0604,0x81FE,0xF0A0,0x4478,0xBA25,0xE34B,
    0xF3A2,0xFE5D,0xC080,0x8A05,0xAD3F,0xBC21,0x4870,0x04F1,
    0xDF63,0xC177,0x75AF,0x6342,0x3020,0x1AE5,0x0EFD,0x6DBF,
    0x4C81,0x1418,0x3526,0x2FC3,0xE1BE,0xA235,0xCC88,0x392E,
    0x5793,0xF255,0x82FC,0x477A,0xACC8,0xE7BA,0x2B32,0x95E6,
    0xA0C0,0x9819,0xD19E,0x7FA3,0x6644,0x7E54,0xAB3B,0x830B,
    0xCA8C,0x29C7,0xD36B,0x3C28,0x79A7,0xE2BC,0x1D16,0x76AD,
    0x3BDB,0x5664,0x4E74,0x1E14,0xDB92,0x0A0C,0x6C48,0xE4B8,
    0x5D9F,0x6EBD,0xEF43,0xA6C4,0xA839,0xA431,0x37D3,0x8BF2,
    0x32D5,0x438B,0x596E,0xB7DA,0x8C01,0x64B1,0xD29C,0xE049,
    0xB4D8,0xFAAC,0x07F3,0x25CF,0xAFCA,0x8EF4,0xE947,0x1810,
    0xD56F,0x88F0,0x6F4A,0x725C,0x2438,0xF157,0xC773,0x5197,
    0x23CB,0x7CA1,0x9CE8,0x213E,0xDD96,0xDC61,0x860D,0x850F,
    0x90E0,0x427C,0xC471,0xAACC,0xD890,0x0506,0x01F7,0x121C,
    0xA3C2,0x5F6A,0xF9AE,0xD069,0x9117,0x5899,0x273A,0xB927,
    0x38D9,0x13EB,0xB32B,0x3322,0xBBD2,0x70A9,0x8907,0xA733,
    0xB62D,0x223C,0x9215,0x20C9,0x4987,0xFFAA,0x7850,0x7AA5,
    0x8F03,0xF859,0x8009,0x171A,0xDA65,0x31D7,0xC684,0xB8D0,
    0xC382,0xB029,0x775A,0x111E,0xCB7B,0xFCA8,0xD66D,0x3A2C,
  }
};

#if DO_SANITY_CHECK
/*
*********************************************************************
* Routine: SanityCheckTable -- verify Sbox properties
*
* Inputs:  Sbox
* Output:  None, but an assertion fails if the tables are wrong
* Notes:
*   The purpose of this routine is solely to illustrate and
*   verify the following properties of the Sbox table:
*      - the Sbox is a "2x2" subset of the AES table:
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1157 of 1232

```
*            Sbox + affine transform + MDS.
*      - the Sbox table can be easily designed to fit in a
*            512-byte table, using a byte swap
*      - the Sbox table can be easily designed to fit in a
*            256-byte table, using some shifts and a byte swap
*****************************************************************
*/
void SanityCheckTable(void)
    {
    const static int  M_x = 0x11B;   /* AES irreducible polynomial */
    const static byte Sbox8[256] = { /* AES 8-bit Sbox */
        0x63,0x7c,0x77,0x7b,0xf2,0x6b,0x6f,0xc5,
        0x30,0x01,0x67,0x2b,0xfe,0xd7,0xab,0x76,
        0xca,0x82,0xc9,0x7d,0xfa,0x59,0x47,0xf0,
        0xad,0xd4,0xa2,0xaf,0x9c,0xa4,0x72,0xc0,
        0xb7,0xfd,0x93,0x26,0x36,0x3f,0xf7,0xcc,
        0x34,0xa5,0xe5,0xf1,0x71,0xd8,0x31,0x15,
        0x04,0xc7,0x23,0xc3,0x18,0x96,0x05,0x9a,
        0x07,0x12,0x80,0xe2,0xeb,0x27,0xb2,0x75,
        0x09,0x83,0x2c,0x1a,0x1b,0x6e,0x5a,0xa0,
        0x52,0x3b,0xd6,0xb3,0x29,0xe3,0x2f,0x84,
        0x53,0xd1,0x00,0xed,0x20,0xfc,0xb1,0x5b,
        0x6a,0xcb,0xbe,0x39,0x4a,0x4c,0x58,0xcf,
        0xd0,0xef,0xaa,0xfb,0x43,0x4d,0x33,0x85,
        0x45,0xf9,0x02,0x7f,0x50,0x3c,0x9f,0xa8,
        0x51,0xa3,0x40,0x8f,0x92,0x9d,0x38,0xf5,
        0xbc,0xb6,0xda,0x21,0x10,0xff,0xf3,0xd2,
        0xcd,0x0c,0x13,0xec,0x5f,0x97,0x44,0x17,
        0xc4,0xa7,0x7e,0x3d,0x64,0x5d,0x19,0x73,
        0x60,0x81,0x4f,0xdc,0x22,0x2a,0x90,0x88,
        0x46,0xee,0xb8,0x14,0xde,0x5e,0x0b,0xdb,
        0xe0,0x32,0x3a,0x0a,0x49,0x06,0x24,0x5c,
        0xc2,0xd3,0xac,0x62,0x91,0x95,0xe4,0x79,
        0xe7,0xc8,0x37,0x6d,0x8d,0xd5,0x4e,0xa9,
        0x6c,0x56,0xf4,0xea,0x65,0x7a,0xae,0x08,
        0xba,0x78,0x25,0x2e,0x1c,0xa6,0xb4,0xc6,
        0xe8,0xdd,0x74,0x1f,0x4b,0xbd,0x8b,0x8a,
        0x70,0x3e,0xb5,0x66,0x48,0x03,0xf6,0x0e,
        0x61,0x35,0x57,0xb9,0x86,0xc1,0x1d,0x9e,
        0xe1,0xf8,0x98,0x11,0x69,0xd9,0x8e,0x94,
        0x9b,0x1e,0x87,0xe9,0xce,0x55,0x28,0xdf,
        0x8c,0xa1,0x89,0x0d,0xbf,0xe6,0x42,0x68,
        0x41,0x99,0x2d,0x0f,0xb0,0x54,0xbb,0x16 };

    int i,k,k2,k3;
    byte bitmap[0x2000];
```


Copyright © 2007 IEEE. All rights reserved.

```
    /* show that smaller tables can be used, if desired */
    for (i=0;i<256;i++)
        {
        k  = Sbox8[i];
        k2 = (k << 1) ^ ((k & 0x80) ? M_x : 0);
        k3 =  k ^ k2;
        assert(Sbox[0][i] == ((k2 << 8) ^ k3));
        assert(Sbox[1][i] == ((k3 << 8) ^ k2));
        }

    /* now make sure that it's a 16-bit permutation */
    memset(bitmap,0,sizeof(bitmap));
    for (i=0;i<0x10000;i++)
        {
        k = _S_(i); /* do an S-box lookup: 16 --> 16 bits */
        assert(k < (1 << 16));
        assert((bitmap[k >> 3] & (1 << (k & 7))) == 0);
        bitmap[k >> 3] |= 1 << (k & 7);
        }
    for (i=0;i<sizeof(bitmap);i++)
        assert(bitmap[i] == 0xFF);

    /* if we reach here, the 16-bit Sbox is ok */
    printf("Table sanity check successful\n");
    }
#endif

/*
**********************************************************************
* Routine: Phase 1 -- generate P1K, given TA, TK, IV32
*
* Inputs:
*     TK[]      = Temporal Key                        [128 bits]
*     TA[]      = transmitter's MAC address           [ 48 bits]
*     IV32      = upper 32 bits of IV                 [ 32 bits]
* Output:
*     P1K[]     = Phase 1 key                         [ 80 bits]
*
* Note:
*     This function only needs to be called every 2**16 frames,
*     although in theory it could be called every frame.
*
**********************************************************************
*/
void Phase1(u16b *P1K,const byte *TK,const byte *TA,u32b IV32)
```

Copyright © 2007 IEEE. All rights reserved.

```
    {
    int  i;

    /* Initialize the 80 bits of P1K[] from IV32 and TA[0..5]     */
    P1K[0]      = Lo16(IV32);
    P1K[1]      = Hi16(IV32);
    P1K[2]      = Mk16(TA[1],TA[0]); /* use TA[] as little-endian */
    P1K[3]      = Mk16(TA[3],TA[2]);
    P1K[4]      = Mk16(TA[5],TA[4]);

    /* Now compute an unbalanced Feistel cipher with 80-bit block */
    /* size on the 80-bit block P1K[], using the 128-bit key TK[] */
    for (i=0; i < PHASE1_LOOP_CNT ;i++)
        {                /* Each add operation here is mod 2**16 */
        P1K[0] += _S_(P1K[4] ^ TK16((i&1)+0));
        P1K[1] += _S_(P1K[0] ^ TK16((i&1)+2));
        P1K[2] += _S_(P1K[1] ^ TK16((i&1)+4));
        P1K[3] += _S_(P1K[2] ^ TK16((i&1)+6));
        P1K[4] += _S_(P1K[3] ^ TK16((i&1)+0));
        P1K[4] +=  i;                   /* avoid "slide attacks" */
        }
    }

/*
***********************************************************************
* Routine: Phase 2 -- generate RC4KEY, given TK, P1K, IV16
*
* Inputs:
*     TK[]      = Temporal Key                        [128 bits]
*     P1K[]     = Phase 1 output key                  [ 80 bits]
*     IV16      = low 16 bits of IV counter           [ 16 bits]
* Output:
*     RC4KEY[]  = the key used to encrypt the frame   [128 bits]
*
* Note:
*     The value {TA,IV32,IV16} for Phase1/Phase2 must be unique
*     across all frames using the same key TK value. Then, for a
*     given value of TK[], this TKIP48 construction guarantees that
*     the final RC4KEY value is unique across all frames.
*
* Suggested implementation optimization: if PPK[] is "overlaid"
*     appropriately on RC4KEY[], there is no need for the final
*     for loop below that copies the PPK[] result into RC4KEY[].
*
***********************************************************************
*/
```


Copyright © 2007 IEEE. All rights reserved.

```
void Phase2(byte *RC4KEY,const byte *TK,const u16b *P1K,u16b IV16)
    {
    int  i;
    u16b PPK[6];                        /* temporary key for mixing    */

    /* all adds in the PPK[] equations below are mod 2**16         */
    for (i=0;i<5;i++) PPK[i]=P1K[i];    /* first, copy P1K to PPK      */
    PPK[5]  =  P1K[4] + IV16;           /* next,  add in IV16          */

    /* Bijective non-linear mixing of the 96 bits of PPK[0..5]         */
    PPK[0] +=    _S_(PPK[5] ^ TK16(0)); /* Mix key in each "round"     */
    PPK[1] +=    _S_(PPK[0] ^ TK16(1));
    PPK[2] +=    _S_(PPK[1] ^ TK16(2));
    PPK[3] +=    _S_(PPK[2] ^ TK16(3));
    PPK[4] +=    _S_(PPK[3] ^ TK16(4));
    PPK[5] +=    _S_(PPK[4] ^ TK16(5)); /* Total # S-box lookups == 6  */

    /* Final sweep: bijective, linear. Rotates kill LSB correlations   */
    PPK[0] +=  RotR1(PPK[5] ^ TK16(6));
    PPK[1] +=  RotR1(PPK[0] ^ TK16(7)); /* Use all of TK[] in Phase2   */
    PPK[2] +=  RotR1(PPK[1]);
    PPK[3] +=  RotR1(PPK[2]);
    PPK[4] +=  RotR1(PPK[3]);
    PPK[5] +=  RotR1(PPK[4]);
    /* At this point, for a given key TK[0..15], the 96-bit output */
    /*     value PPK[0..5] is guaranteed to be unique, as a function   */
    /*     of the 96-bit "input" value   {TA,IV32,IV16}. That is, P1K  */
    /*     is now a keyed permutation of {TA,IV32,IV16}.               */

    /* Set RC4KEY[0..3], which includes cleartext portion of RC4 key   */
    RC4KEY[0] = Hi8(IV16);              /* RC4KEY[0..2] is the WEP IV  */
    RC4KEY[1] =(Hi8(IV16) | 0x20) & 0x7F; /* Help avoid FMS weak keys  */
    RC4KEY[2] = Lo8(IV16);
    RC4KEY[3] = Lo8((PPK[5] ^ TK16(0)) >> 1);

    /* Copy 96 bits of PPK[0..5] to RC4KEY[4..15]  (little-endian)      */
    for (i=0;i<6;i++)
        {
        RC4KEY[4+2*i] = Lo8(PPK[i]);
        RC4KEY[5+2*i] = Hi8(PPK[i]);
        }
    }

/*
*********************************************************************
* Routine: doTestCase -- execute a test case, and print results
```

Copyright © 2007 IEEE. All rights reserved.

```
*********************************************************************
*/
void DoTestCase(byte *RC4KEY,u32b IV32,u16b IV16,const byte *TA,const byte
*TK)
    {
    int  i;
    u16b P1K[P1K_SIZE/2];  /* "temp" copy of phase1 key */

    printf("\nTK    =");
    for (i=0;i<TK_SIZE;i++) printf(" %02X",TK[i]);
    printf("\nTA    =");
    for (i=0;i<TA_SIZE;i++) printf(" %02X",TA[i]);
    printf("\nIV32  = %08X   [transmitted as",IV32);  /* show byte order */
    for (i=0;i<4;i++) printf(" %02X",(IV32 >> (24-8*i)) & 0xFF);
    printf("]");
    printf("\nIV16  = %04X",IV16);

    Phase1(P1K,TK,TA,IV32);

    printf("\nP1K   =");
    for (i=0;i<P1K_SIZE/2;i++) printf(" %04X ",P1K[i] & 0xFFFF);

    Phase2(RC4KEY,TK,P1K,IV16);

    printf("\nRC4KEY= ");
    for (i=0;i<RC4_KEY_SIZE;i++) printf("%02X ",RC4KEY[i]);
    }

/*
*********************************************************************
* Static (Repeatable) Test Cases
*********************************************************************
*/
void DoStaticTestCases(int testCnt)
    {
    int  i,j;
    byte TA[TA_SIZE],TK[TK_SIZE],RC4KEY[RC4_KEY_SIZE];
    u16b IV16=0;
    u32b IV32=0;

    /* set a fixed starting point */
    for (i=0;i<TK_SIZE;i++) TK[i]=i;
    for (i=0;i<TA_SIZE;i++) TA[i]=(i+1)*17;
    TA[0] = TA[0] & 0xFC;             /* Clear I/G and U/L bits in OUI */

    /* now generate tests, feeding results back into new tests */
```


Copyright © 2007 IEEE. All rights reserved.

```
    for (i=0; i<testCnt/2; i++)
        {
        printf("\n\nTest vector #%d:",2*i+1);
        DoTestCase(RC4KEY,IV32,IV16,TA,TK);
        IV16++;                     /* emulate per-frame "increment" */
        if (IV16 == 0) IV32++;
        printf("\n\nTest vector #%d:",2*i+2);
        DoTestCase(RC4KEY,IV32,IV16,TA,TK);
        /* feed results back to seed the next test input values    */
        IV16 = (i) ? Mk16(RC4KEY[15],RC4KEY[4]) : 0xFFFF;/* force wrap */
        IV32 =       Mk16(RC4KEY[14],RC4KEY[5]);
        IV32 =       Mk16(RC4KEY[13],RC4KEY[7]) + (IV32 << 16);
        for (j=0;j<TA_SIZE;j++) TA[j]^=RC4KEY[12-j];
        for (j=0;j<TK_SIZE;j++) TK[j]^=RC4KEY[(j+i+1) % RC4_KEY_SIZE] ^
                                    RC4KEY[(j+i+7) % RC4_KEY_SIZE] ;
        TA[0] = TA[0] & 0xFC;         /* Clear I/G and U/L bits in OUI */
        }
    /* comparing the final output is a good check of correctness   */
    printf("\n");
    }

/*
**********************************************************************
* Test Cases Generated at Random
**********************************************************************
*/
void DoRandomTestCases(int testCnt)
    {
    int  i,j;
    u16b IV16;
    u32b IV32;
    byte TA[TA_SIZE],RC4KEY[RC4_KEY_SIZE],TK[TK_SIZE];

    printf("Random tests:\n");
    /* now generate tests "recursively" */
    for (i=0; i<testCnt; i++)
        {
        IV16 = rand() & 0xFFFF;
        IV32 = rand() + (rand() << 16);
        for (j=0;j<TK_SIZE;j++) TK[j]=rand() & 0xFF;
        for (j=0;j<TA_SIZE;j++) TA[j]=rand() & 0xFF;
        TA[0] = TA[0] & 0xFC;         /* Clear I/G and U/L bits in OUI */
        printf("\n\nRandom test vector #%d:",i+1);
        DoTestCase(RC4KEY,IV32,IV16,TA,TK);
        }
    printf("\n");
```

Copyright © 2007 IEEE. All rights reserved.

```
    }

/*
**********************************************************************
* Usage text
**********************************************************************
*/
#define NUM_TEST_CNT  8
void Usage(void)
    {
    printf(
        "Usage:   TKIP48 [options]\n"
        "Purpose: Generate test vectors for IEEE 802.11 TKIP48\n"
        "Options  -?   -- output this usage text\n"
        "         -r   -- generate test vectors at random\n"
        "         -sN  -- init random seed to N\n"
        "         -tN  -- generate N tests (default = %d)\n",
        NUM_TEST_CNT
        );
    exit(0);
    }

/*
**********************************************************************
* Main
**********************************************************************
*/
int main(int argc, char **argv)
    {
    char *parg;
    int   i,doRand = 0;
    int   testCnt  = NUM_TEST_CNT;
    u32b  seed     = (u32b) time(NULL);

#if DO_SANITY_CHECK
    SanityCheckTable();
#endif

    for (i=1; i<argc; i++)
        {
        parg = argv[i];
        switch (parg[0])
            {
            case '-':
                switch (parg[1])
                    {
```


Copyright © 2007 IEEE. All rights reserved.

```
                        case '?':
                        case 'H':
                        case 'h':
                            Usage();
                            return 0;
                        case 'R':
                        case 'r':  /* generate some random test vectors */
                            doRand  = 1;
                            break;
                        case 'S':
                        case 's':
                            seed    = atoi(parg+2);
                            break;
                        case 'T':
                        case 't':
                            testCnt = atoi(parg+2);
                            break;
                        default:
                            break;
                        }
                    break;
                case '?':
                    Usage();
                    return 0;
                default:
                    printf("Invalid argument: \"%s\"\n", parg);
                    return 1;
                }
            }
        srand(seed);
        if (doRand) printf("Seed = %u\n",seed);

        /* generate some test vectors */
        if (doRand) DoRandomTestCases(testCnt);
        else        DoStaticTestCases(testCnt);

        return 0;
        }
```

### H.1.1 Test vectors

The following output is provided to test implementations of the temporal key hash algorithm. All input and output values are shown in hexadecimal.

```
Test vector #1:
TK    = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F [LSB on left, MSB on right]
TA    = 10-22-33-44-55-66
PN    = 000000000000
```

Copyright © 2007 IEEE. All rights reserved.

```
IV32  = 00000000
IV16  = 0000
P1K   = 3DD2  016E  76F4  8697  B2E8
RC4KEY= 00 20 00 33 EA 8D 2F 60 CA 6D 13 74 23 4A 66 0B

Test vector #2:
TK    = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F [LSB on left, MSB on right]
TA    = 10-22-33-44-55-66
PN    = 000000000001
IV32  = 00000000
IV16  = 0001
P1K   = 3DD2  016E  76F4  8697  B2E8
RC4KEY= 00 20 01 90 FF DC 31 43 89 A9 D9 D0 74 FD 20 AA

Test vector #3:
TK    = 63 89 3B 25 08 40 B8 AE 0B D0 FA 7E 61 D2 78 3E [LSB on left, MSB on right]
TA    = 64-F2-EA-ED-DC-25
PN    = 20DCFD43FFFF
IV32  = 20DCFD43
IV16  = FFFF
P1K   = 7C67  49D7  9724  B5E9  B4F1
RC4KEY= FF 7F FF 93 81 0F C6 E5 8F 5D D3 26 25 15 44 CE

Test vector #4:
TK    = 63 89 3B 25 08 40 B8 AE 0B D0 FA 7E 61 D2 78 3E [LSB on left, MSB on right]
TA    = 64-F2-EA-ED-DC-25
PN    = 20DCFD440000
IV32  = 20DCFD44
IV16  = 0000
P1K   = 5A5D  73A8  A859  2EC1  DC8B
RC4KEY= 00 20 00 49 8C A4 71 FC FB FA A1 6E 36 10 F0 05

Test vector #5:
TK    = 98 3A 16 EF 4F AC B3 51 AA 9E CC 27 1D 73 09 E2 [LSB on left, MSB on right]
TA    = 50-9C-4B-17-27-D9
PN    = F0A410FC058C
IV32  = F0A410FC
IV16  = 058C
P1K   = F2DF  EBB1  88D3  5923  A07C
RC4KEY= 05 25 8C F4 D8 51 52 F4 D9 AF 1A 64 F1 D0 70 21

Test vector #6:
TK    = 98 3A 16 EF 4F AC B3 51 AA 9E CC 27 1D 73 09 E2 [LSB on left, MSB on right]
TA    = 50-9C-4B-17-27-D9
PN    = F0A410FC058D
IV32  = F0A410FC
IV16  = 058D
P1K   = F2DF  EBB1  88D3  5923  A07C
RC4KEY= 05 25 8D 09 F8 15 43 B7 6A 59 6F C2 C6 73 8B 30

Test vector #7:
TK    = C8 AD C1 6A 8B 4D DA 3B 4D D5 B6 54 38 35 9B 05 [LSB on left, MSB on right]
TA    = 94-5E-24-4E-4D-6E
PN    = 8B1573B730F8
IV32  = 8B1573B7
IV16  = 30F8
P1K   = EFF1  3F38  A364  60A9  76F3
RC4KEY= 30 30 F8 65 0D A0 73 EA 61 4E A8 F4 74 EE 03 19

Test vector #8:
TK    = C8 AD C1 6A 8B 4D DA 3B 4D D5 B6 54 38 35 9B 05 [LSB on left, MSB on right]
TA    = 94-5E-24-4E-4D-6E
PN    = 8B1573B730F9
IV32  = 8B1573B7
```


Copyright © 2007 IEEE. All rights reserved.

```
IV16  = 30F9
P1K   = EFF1  3F38  A364  60A9  76F3
RC4KEY= 30 30 F9 31 55 CE 29 34 37 CC 76 71 27 16 AB 8F
```

## H.2 Michael reference implementation and test vectors

### H.2.1 Michael test vectors

To ensure correct implementation of Michael, here are some test vectors. These test vectors still have to be confirmed by an independent implementation.

#### H.2.1.1 Block function

Table H.1 gives some test vectors for the block function.

**Table H.1—Test vectors for block function**

| Input | # times | Output |
|---|---|---|
| (00000000, 00000000) | 1 | (00000000, 00000000) |
| (00000000, 00000001) | 1 | (c00015a8, c0000b95) |
| (00000001, 00000000) | 1 | (6b519593, 572b8b8a) |
| (01234567, 83659326) | 1 | (441492c2, 1d8427ed) |
| (00000001, 00000000) | 1000 | (9f04c4ad, 2ec6c2bf) |

The first four rows give test vectors for a single application of the block function *b*. The last row gives a test vector for 1000 repeated applications of the block function. Together these should provide adequate test coverage.

#### H.2.1.2 Michael

Table H.2 gives some test vectors for Michael.

**Table H.2—Test vectors for Michael**

| Key | Message | Output |
|---|---|---|
| 0000000000000000 | "" | 82925c1ca1d130b8 |
| 82925c1ca1d130b8 | "M” | 434721ca40639b3f |
| 434721ca40639b3f | "Mi “ | e8f9becae97e5d29 |
| E8f9becae97e5d29 | "Mic" | 90038fc6cf13c1db |
| 90038fc6cf13c1db | "Mich" | D55e100510128986 |
| D55e100510128986 | "Michael" | 0a942b124ecaa546 |

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1167 of 1232

Note that each key is the result of the previous line, which makes it easy to construct a single test out of all of these test cases.

### H.2.2 Sample code for Michael

```
//
// Michael.h    Reference implementation for Michael

//
// A Michael object implements the computation of the MIC.
//
// Conceptually, the object stores the message to be authenticated.
// At construction the message is empty.
// The append() method appends bytes to the message.
// The getMic() method computes the MIC over the message and returns the
// result.
// As a side-effect it also resets the stored message
// to the empty message so that the object can be re-used
// for another MIC computation.

class Michael
{

public:
    // Constructor requires a pointer to 8 bytes of key
    Michael( Byte * key );

    // Destructor
    ~Michael();

    // Clear the internal message,
    // resets the object to the state just after construction.
    void clear();

    // Set the key to a new value
    void setKey( Byte * key );

    // Append bytes to the message to be MICed
    void append( Byte * src, int nBytes );

    // Get the MIC result. Destination should accept 8 bytes of result.
    // This also resets the message to empty.
    void getMIC( Byte * dst );

    // Run the test plan to verify proper operations
```


Copyright © 2007 IEEE. All rights reserved.

```
    static void runTestPlan();

private:
    // Copy constructor declared but not defined,
    //avoids compiler-generated version.
    Michael( const Michael & );
    // Assignment operator declared but not defined,
    //avoids compiler-generated version.
    void operator=( const Michael & );

    // A bunch of internal functions

    // Get UInt32 from 4 bytes LSByte first
    static UInt32 getUInt32( Byte * p );

    // Put UInt32 into 4 bytes LSByte first
    static void putUInt32( Byte * p, UInt32 val );

    // Add a single byte to the internal message
    void appendByte( Byte b );

    // Conversion of hex string to binary string
    static void hexToBin( char *src, Byte * dst );

    // More conversion of hex string to binary string
    static void hexToBin( char *src, int nChars, Byte * dst );

    // Helper function for hex conversion
    static Byte hexToBinNibble( char c );

    // Run a single test case
    static void runSingleTest( char * cKey, char * cMsg, char * cResult );

    UInt32  K0, K1;         // Key
    UInt32  L, R;           // Current state
    UInt32  M;              // Message accumulator (single word)
    int     nBytesInM;      // # bytes in M
};

//
// Michael.cpp  Reference implementation for Michael
//

// Adapt these typedefs to your local platform
typedef unsigned long UInt32;
typedef unsigned char Byte;
```

Copyright © 2007 IEEE. All rights reserved.

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "Michael.h"

// Rotation functions on 32 bit values
#define ROL32( A, n ) \
 ( ((A) << (n)) | ( ((A)>>(32-(n)))  & ( (1UL << (n)) - 1 ) ) )
#define ROR32( A, n ) ROL32( (A), 32-(n) )

UInt32 Michael::getUInt32( Byte * p )
// Convert from Byte[] to UInt32 in a portable way
{
    UInt32 res = 0;
    for( int i=0; i<4; i++ )
    {
        res |= (*p++) << (8*i);
    }
    return res;
}

void Michael::putUInt32( Byte * p, UInt32 val )
// Convert from UInt32 to Byte[] in a portable way
{
    for( int i=0; i<4; i++ )
    {
        *p++ = (Byte) (val & 0xff);
        val >>= 8;
    }
}

void Michael::clear()
{
    // Reset the state to the empty message.
    L = K0;
    R = K1;
    nBytesInM = 0;
    M = 0;
}

void Michael::setKey( Byte * key )
{
    // Set the key
```


Copyright © 2007 IEEE. All rights reserved.

```
    K0 = getUInt32( key );
    K1 = getUInt32( key + 4 );
    // and reset the message
    clear();
}

Michael::Michael( Byte * key )
{
    setKey( key );
}

Michael::~Michael()
{
    // Wipe the key material
    K0 = 0;
    K1 = 0;

    // And the other fields as well.
    //Note that this sets (L,R) to (K0,K1) which is just fine.
    clear();
}

void Michael::appendByte( Byte b )
{
    // Append the byte to our word-sized buffer
    M |= b << (8*nBytesInM);
    nBytesInM++;
    // Process the word if it is full.
    if( nBytesInM >= 4 )
    {
        L ^= M;
        R ^= ROL32( L, 17 );
        L += R;
        R ^= ((L & 0xff00ff00) >> 8) | ((L & 0x00ff00ff) << 8);
        L += R;
        R ^= ROL32( L, 3 );
        L += R;
        R ^= ROR32( L, 2 );
        L += R;
        // Clear the buffer
        M = 0;
        nBytesInM = 0;
    }
}

void Michael::append( Byte * src, int nBytes )
```

Copyright © 2007 IEEE. All rights reserved.

```
{
    // This is simple
    while( nBytes > 0 )
    {
        appendByte( *src++ );
        nBytes--;
    }
}

void Michael::getMIC( Byte * dst )
{
    // Append the minimum padding
    appendByte( 0x5a );
    appendByte( 0 );
    appendByte( 0 );
    appendByte( 0 );
    appendByte( 0 );
    // and then zeroes until the length is a multiple of 4
    while( nBytesInM != 0 )
    {
        appendByte( 0 );
    }
    // The appendByte function has already computed the result.
    putUInt32( dst, L );
    putUInt32( dst+4, R );
    // Reset to the empty message.
    clear();
}

void Michael::hexToBin( char *src, Byte * dst )
{
    // Simple wrapper
    hexToBin( src, strlen( src ), dst );
}

void Michael::hexToBin( char *src, int nChars, Byte * dst )
{
    assert( (nChars & 1) == 0 );
    int nBytes = nChars/2;

    // Straightforward conversion
    for( int i=0; i<nBytes; i++ )
    {
        dst[i] = (Byte)((hexToBinNibble( src[0] ) << 4) |
            hexToBinNibble( src[1] ));
        src += 2;
```


Copyright © 2007 IEEE. All rights reserved.

```
    }
}

Byte Michael::hexToBinNibble( char c )
{
    if( '0' <= c && c <= '9' )
    {
        return (Byte)(c - '0');
    }
    // Make it upper case
    c &= ~('a'-'A');

    assert( 'A' <= c && c <= 'F' );
    return (Byte)(c - 'A' + 10);
}

void Michael::runSingleTest( char * cKey, char * cMsg, char * cResult )
{
    Byte key[ 8 ];
    Byte result[ 8 ];
    Byte res[ 8 ];

    // Convert key and result to binary form
    hexToBin( cKey, key );
    hexToBin( cResult, result );

    // Compute the MIC value
    Michael mic( key );
    mic.append( (Byte *)cMsg, strlen( cMsg) );
    mic.getMIC( res );

    // Check that it matches
    assert( memcmp( res, result, 8 ) == 0 );
}

void Michael::runTestPlan()
// As usual, test plans can be quite tedious but this should
// ensure that the implementation runs as expected.
{
    Byte key[8] ;
    Byte msg[12];
    int i;

    // First we test the test vectors for the block function

    // The case (0,0)
```

Copyright © 2007 IEEE. All rights reserved.

```
putUInt32( key, 0 );
putUInt32( key+4, 0 );
putUInt32( msg, 0 );

Michael mic( key );
mic.append( msg, 4 );

assert( mic.L == 0 && mic.R == 0 );

// The case (0,1)
putUInt32( key, 0 );
putUInt32( key+4, 1 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0xc00015a8 && mic.R == 0xc0000b95 );

// The case (1,0)
putUInt32( key, 1 );
putUInt32( key+4, 0 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0x6b519593 && mic.R == 0x572b8b8a );

// The case (01234567, 83659326)
putUInt32( key, 0x01234567 );
putUInt32( key+4, 0x83659326 );
mic.setKey( key );
mic.append( msg, 4 );

assert( mic.L == 0x441492c2 && mic.R == 0x1d8427ed );

// The repeated case
putUInt32( key, 1 );
putUInt32( key+4,0 );
mic.setKey( key );

for( i=0; i<1000; i++ )
{
    mic.append( msg, 4 );
}

assert( mic.L == 0x9f04c4ad && mic.R == 0x2ec6c2bf );

// And now for the real test cases
```


Copyright © 2007 IEEE. All rights reserved.

```
    runSingleTest( "0000000000000000", ""         , "82925c1ca1d130b8" );
    runSingleTest( "82925c1ca1d130b8", "M"        , "434721ca40639b3f" );
    runSingleTest( "434721ca40639b3f", "Mi"       , "e8f9becae97e5d29" );
    runSingleTest( "e8f9becae97e5d29", "Mic"      , "90038fc6cf13c1db" );
    runSingleTest( "90038fc6cf13c1db", "Mich"     , "d55e100510128986" );
    runSingleTest( "d55e100510128986", "Michael"  , "0a942b124ecaa546" );
}
```

## H.3 PRF reference implementation and test vectors

### H.3.1 PRF reference code

```
/*
 * PRF -- Length of output is in octets rather than bits
 *     since length is always a multiple of 8 output array is
 *     organized so first N octets starting from 0 contains PRF output
 *
 *     supported inputs are 16, 24, 32, 48, 64
 *     output array must be 80 octets to allow for sha1 overflow
 */
void PRF(
    unsigned char *key, int key_len,
    unsigned char *prefix, int prefix_len,
    unsigned char *data, int data_len,
    unsigned char *output, int len)
{
    int i;
    unsigned char input[1024]; /* concatenated input */
    int currentindex = 0;
    int total_len;

    memcpy(input, prefix, prefix_len);
    input[prefix_len] = 0; /* single octet 0 */
    memcpy(&input[prefix_len+1], data, data_len);
    total_len = prefix_len + 1 + data_len;
    input[total_len] = 0; /* single octet count, starts at 0 */
    total_len++;
    for (i = 0; i < (len+19)/20; i++) {
        hmac_sha1(input, total_len, key, key_len,
            &output[currentindex]);
        currentindex += 20;/* next concatenation location */
        input[total_len-1]++; /* increment octet count */
    }
}
```

Copyright © 2007 IEEE. All rights reserved.

### H.3.2 PRF test vectors

```
Test case 1
Key             0x0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b
Key length      20
Prefix          “prefix”
Prefix length   6
Data            "Hi There"
Data length     8
PRF-512         0xbcd4c650b30b9684951829e0d75f9d54
                0xb862175ed9f00606e17d8da35402ffee
                0x75df78c3d31e0f889f012120c0862beb
                0x67753e7439ae242edb8373698356cf5a

Test case 2
Key             "Jefe"
Key length      4
Prefix          “prefix”
Prefix length   6
Data            "what do ya want for nothing?"
Data length     28
PRF-512         0x51f4de5b33f249adf81aeb713a3c20f4
                0xfe631446fabdfa58244759ae58ef9009
                0xa99abf4eac2ca5fa87e692c440eb4002
                0x3e7babb206d61de7b92f41529092b8fc

Test case 3
Key             0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Key length      20
Prefix          “prefix”
Prefix length   6
Data            0xdd repeated 50 times
Data length     50
PRF-512         0xe1ac546ec4cb636f9976487be5c86be1
                0x7a0252ca5d8d8df12cfb0473525249ce
                0x9dd8d177ead710bc9b590547239107ae
                0xf7b4abd43d87f0a68f1cbd9e2b6f7607
```

## H.4 Suggested pass-phrase-to-PSK mapping

### H.4.1 Introduction

The RSNA PSK consists of 256 bits, or 64 octets when represented in hex. It is difficult for a user to correctly enter 64 hex characters. Most users, however, are familiar with passwords and pass-phrases and feel more comfortable entering them than entering keys. A user is more likely to be able to enter an ASCII password or pass-phrase, even though doing so limits the set of possible keys. This suggests that the best that can be done is to introduce a pass-phrase to PSK mapping.

This clause defines a pass-phrase–to–PSK mapping that is the recommended practice for use with RSNAs. This pass-phrase mapping was introduced to encourage users unfamiliar with cryptographic concepts to enable the security features of their WLAN.

Keys derived from the pass phrase provide relatively low levels of security, especially with keys generated form short passwords, since they are subject to dictionary attack. Use of the key hash is recommended only where it is impractical to make use of a stronger form of user authentication. A key generated from a pass-phrase of less than about 20 characters is unlikely to deter attacks.

The pass-phrase mapping defined in this clause uses the PBKDF2 method from PKCS #5 v2.0 [B30].


Copyright © 2007 IEEE. All rights reserved.

*PSK* = PBKDF2(*PassPhrase*, *ssid*, *ssidLength*, 4096, 256)

Here, the following assumptions apply:

— A pass-phrase is a sequence of between 8 and 63 ASCII-encoded characters. The limit of 63 comes from the desire to distinguish between a pass-phrase and a PSK displayed as 64 hexadecimal characters.

— Each character in the pass-phrase must have an encoding in the range of 32 to 126 (decimal), inclusive.

— *ssid* is the SSID of the ESS or IBSS where this pass-phrase is in use, encoded as an octet string used in the Beacon and Probe Response frames for the ESS or IBSS.

— *ssidLength* is the number of octets of the *ssid*.

— 4096 is the number of times the pass-phrase is hashed.

— 256 is the number of bits output by the pass-phrase mapping.

### H.4.2 Reference implementation

```
/*
 * F(P, S, c, i) = U1 xor U2 xor ... Uc
 * U1 = PRF(P, S || Int(i))
 * U2 = PRF(P, U1)
 * Uc = PRF(P, Uc-1)
 */

void F(
    char *password,
    unsigned char *ssid,
    int ssidlength,
    int iterations,
    int count,
    unsigned char *output)
{
    unsigned char digest[36], digest1[A_SHA_DIGEST_LEN];
    int i, j;

    for (i = 0; i < strlen(password); i++) {
        assert((password[i] >= 32) && (password[i] <= 126));
    }

    /* U1 = PRF(P, S || int(i)) */
    memcpy(digest, ssid, ssidlength);
    digest[ssidlength] = (unsigned char)((count>>24) & 0xff);
    digest[ssidlength+1] = (unsigned char)((count>>16) & 0xff);
    digest[ssidlength+2] = (unsigned char)((count>>8) & 0xff);
    digest[ssidlength+3] = (unsigned char)(count & 0xff);
    hmac_sha1(digest, ssidlength+4, (unsigned char*) password,
        (int) strlen(password), digest, digest1);
```

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1177 of 1232

```
        /* output = U1 */
        memcpy(output, digest1, A_SHA_DIGEST_LEN);

        for (i = 1; i < iterations; i++) {
            /* Un = PRF(P, Un-1) */
            hmac_sha1(digest1, A_SHA_DIGEST_LEN, (unsigned char*) password,
                (int) strlen(password), digest);
            memcpy(digest1, digest, A_SHA_DIGEST_LEN);

            /* output = output xor Un */
            for (j = 0; j < A_SHA_DIGEST_LEN; j++) {
                output[j] ^= digest[j];
            }
        }
    }

    /*
     * password - ascii string up to 63 characters in length
     * ssid - octet string up to 32 octets
     * ssidlength - length of ssid in octets
     * output must be 40 octets in length and outputs 256 bits of key
     */
    int PasswordHash (
        char *password,
        unsigned char *ssid,
        int ssidlength,
        unsigned char *output)
    {
        if ((strlen(password) > 63) || (ssidlength > 32))
            return 0;

        F(password, ssid, ssidlength, 4096, 1, output);
        F(password, ssid, ssidlength, 4096, 2,
            &output[A_SHA_DIGEST_LEN]);
        return 1;
    }
```

### H.4.3 Test vectors

```
Test case 1
Passphrase = “password”
SSID = { ‘I’, ‘E’, ‘E’, ‘E’ }
SSIDLength = 4
PSK = f42c6fc52df0ebef9ebb4b90b38a5f90 2e83fe1b135a70e23aed762e9710a12e

Test case 2
Passphrase = “ThisIsAPassword”
SSID = { ‘T’, ‘h’, ‘i’, ‘s’, ‘I’, ‘s’, ‘A’, ‘S’, ‘S’, ‘I’, ‘D’ }
SSIDLength = 11
```


Copyright © 2007 IEEE. All rights reserved.

```
PSK = 0dc0d6eb90555ed6419756b9a15ec3e3 209b63df707dd508d14581f8982721af

Test case 3
Password = “aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa”
SSID = {‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’,
             ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’, ‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’,‘Z’}
SSIDLength = 32
PSK = becb93866bb8c3832cb777c2f559807c 8c59afcb6eae734885001300a981cc62
```

## H.5 Suggestions for random number generation

In order to properly implement cryptographic protocols, every platform needs the ability to generate cryptographic-quality random numbers. IETF RFC 4086 explains the notion of cryptographic-quality random numbers and provides advice on ways to harvest suitable randomness. It recommends sampling multiple sources, each of which contains some randomness, and by passing the complete set of samples through a PRF. By following this advice, an implementation can usually collect enough randomness to distill into a seed for a PRNG whose output will be unpredictable.

This clause suggests two sample techniques that can be combined with the other recommendations of IETF RFC 4086 to harvest randomness. The first method is a software solution that can be implemented on most hardware; the second is a hardware-assisted solution. These solutions are expository only, to demonstrate that it is feasible to harvest randomness on any IEEE 802.11 platform. They are not mutually exclusive, and they do not preclude the use of other sources of randomness when available, such as a noisy diode in a power circuit; in this case, the more the merrier. As many sources of randomness as possible should be gathered into a buffer, and then hashed, to obtain a seed for the PRNG.

### H.5.1 Software sampling

Due to the nature of clock circuits in modern electronics, there will be some lack of correlation between two clocks in two different pieces of equipment, even when high-quality crystals are used—crystal clocks are subject to jitter, noise, drift, and frequency mismatch. This randomness may be as little as the placement of the clock waveform edges. Even if one entity were to attempt to synchronize itself to another entity’s clock, the correlation cannot be perfect due to noise and uncertainties of the synchronization.

Two clock circuits in the same piece of equipment may synchronize in frequency, but again the correlation will not be perfect due to the noise and jitter of the circuits.

The randomness between the two clocks may not be much per sample—a tenth of a bit or less—but enough samples may be collected to gather enough randomness to form a seed.

A device can use software methods to take advantage of this lack of synchronization, to collect randomness from different sources. As an example, an AP might measure the frame arrival times on Ethernet wireless ports. There is always some amount of traffic on modern Ethernets: ARPs, DHCP requests, NetBIOS advertisements, etc. The sample algorithm in this subclause uses this traffic. In the example, an AP obtains randomness from the available traffic. If Ethernet traffic is available, the AP utilizes that for a source of randomness. Otherwise, it waits for the first association and creates traffic from which it can obtain randomness.

The clocks used to time the frames should be the highest resolution available, preferably 1 ms resolution or better. The clock used to time frame arrival should not be related to the clock used for frame serialization.

Initialize result to empty array
LoopCounter = 0

Copyright © 2007 IEEE. All rights reserved.

```
Wait until Ethernet traffic or association
Repeat until global key counter "random enough" or 32 times {
    result = PRF-256(0, "Init Counter",
    Local Mac Address || Time || result || LoopCounter)
    LoopCounter++
    Repeat 32 times {
        If Ethernet traffic available then {
            Take least significant octet of the timestamp when Ethernet packet is received
            Concatenate this octet onto result
        } else {
            Start 4-Way Handshake; after receipt of Message 2, deauthenticate
            Take least significant octet of the timestamp of when Message 1 is sent
            Take least significant octet of the timestamp of when Message 2 is received
            Take the least significant octet of RSSI from Message 2
            Take SNonce from Message 2
            Concatenate the sent time, received time, RSSI and SNonce octets onto result
        }
    }
}
Global key counter = result = PRF-256(0, "Init Counter",
Local Mac Address || Time || result || LoopCounter)
```

NOTE—The Time may be 0 if it is not available.

### H.5.2 Hardware-assisted solution

The sample implementation in this subclause uses hardware ring oscillators to generate randomness, as depicted in Figure H.1.

The circuit in Figure H.1 generates randomness. The clock input should be about the same frequency as the ring oscillator's natural frequencies. The LFSR should be chosen to be one that is maximal length. Sample LFSRs can be found in Arazi [B2].

The three ring oscillators should be isolated from each other as much as possible to avoid harmonic locking between them. In addition, the three ring oscillators should not be near any other clock circuitry within the system to avoid these ring oscillators' locking to system clocks. The oscillators should be tested to ensure that their output is not correlated.

The output of the LFSR is read by software and concatenated until enough randomness is collected. As a rule of thumb, reading from the LFSR 8 to 16 times the number of bits as the desired number of random bits is sufficient.

```
Initialize result to empty array
Repeat 1024 times {
    Read LFSR
    result = result | LFSR
    Wait a time period
}
Global key counter = PRF-256(0, "Init Counter", result)
```


Copyright © 2007 IEEE. All rights reserved.

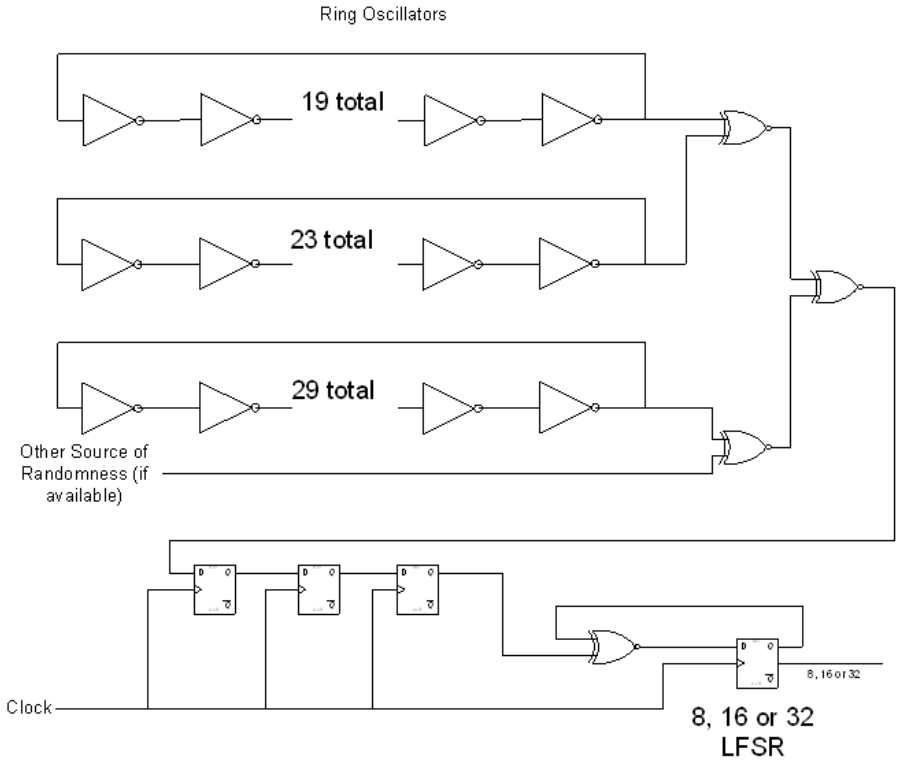



**Figure H.1—Randomness generating circuit**

## H.6 Additional test vectors

### H.6.1 Notation

In the examples that follow in H.6, frames are represented as a stream of octets, each octet in hex notation, sometimes with text annotation. The order of transmission for octets is left to right, top to bottom. For example, consider the following representation of a frame in Table H.3.

**Table H.3—Notation example**

| | |
|---|---|
| Description #1 | 00 01 02 03 04 05 |
| Description #2 | 06 07 08 |

The frame consists of 9 octets, represented in hex notation as “00”, “01”, ..., “08”. The octet represented by “00” is transmitted first, and the octet represented by “08” is transmitted last. Similar tables are used for other purposes, such as describing a cryptographic operation.

In the text discussion outside of tables, integer values are represented in either hex notation using a “0x” prefix or in decimal notation using no prefix. For example, the hex notation 0x12345 and the decimal notation 74565 represent the same integer value.

Copyright © 2007 IEEE. All rights reserved.

### H.6.2 WEP cryptographic encapsulation

The discussion in this subclause represents an ARC4 encryption using a table that shows the key, plaintext input, and cipher text output. The MPDU data, prior to WEP cryptographic encapsulation, is shown in Table H.4.

**Table H.4—Sample plaintext MPDU**

| | |
|---|---|
| MPDU data | aa aa 03 00 00 00 08 00 45 00 00 4e 66 1a 00 00 80 11 be 64 0a 00 01 22 0a ff ff ff 00 89 00 89 00 3a 00 00 80 a6 01 10 00 01 00 00 00 00 00 00 20 45 43 45 4a 45 48 45 43 46 43 45 50 46 45 45 49 45 46 46 43 43 41 43 41 43 41 43 41 43 41 41 41 00 00 20 00 01 |

ARC4 encryption is performed as shown in Table H.5.

**Table H.5—ARC4 encryption**

| | |
|---|---|
| Key | fb 02 9e 30 31 32 33 34 |
| Plaintext | aa aa 03 00 00 00 08 00 45 00 00 4e 66 1a 00 00 80 11 be 64 0a 00 01 22 0a ff ff ff 00 89 00 89 00 3a 00 00 80 a6 01 10 00 01 00 00 00 00 00 00 20 45 43 45 4a 45 48 45 43 46 43 45 50 46 45 45 49 45 46 46 43 43 41 43 41 43 41 43 41 43 41 41 41 00 00 20 00 01 1b d0 b6 04 |
| Ciphertext | f6 9c 58 06 bd 6c e8 46 26 bc be fb 94 74 65 0a ad 1f 79 09 b0 f6 4d 5f 58 a5 03 a2 58 b7 ed 22 eb 0e a6 49 30 d3 a0 56 a5 57 42 fc ce 14 1d 48 5f 8a a8 36 de a1 8d f4 2c 53 80 80 5a d0 c6 1a 5d 6f 58 f4 10 40 b2 4b 7d 1a 69 38 56 ed 0d 43 98 e7 ae e3 bf 0e 2a 2c a8 f7 |

The plaintext consists of the MPDU data, followed by a 4-octet CRC-32 calculated over the MPDU data.

The expanded MPDU, after WEP cryptographic encapsulation, is shown in Table H.6.

**Table H.6—Expanded MPDU after WEP encapsulation**

| | |
|---|---|
| IV | fb 02 9e 80 |
| MPDU data | f6 9c 58 06 bd 6c e8 46 26 bc be fb 94 74 65 0a ad 1f 79 09 b0 f6 4d 5f 58 a5 03 a2 58 b7 ed 22 eb 0e a6 49 30 d3 a0 56 a5 57 42 fc ce 14 1d 48 5f 8a a8 36 de a1 8d f4 2c 53 80 80 5a d0 c6 1a 5d 6f 58 f4 10 40 b2 4b 7d 1a 69 38 56 ed 0d 43 98 e7 ae e3 bf 0e |
| ICV | 2a 2c a8 f7 |


Copyright © 2007 IEEE. All rights reserved.

The IV consists of the first 3 octets of the ARC4 key, followed by an octet containing the Key ID value in the upper 2 bits. In this example, the Key ID value is 2. The MPDU data consists of the cipher text, excluding the last 4 octets. The ICV consists of the last 4 octets of the cipher text, which is the encrypted CRC-32 value.

### H.6.3 TKIP test vector

An example of a TKIP MSDU is provided in Table H.7 and Table H.8. The key and PN are used to create the IV, Phase1, and Phase2 keys.

**Table H.7—Sample TKIP parameters**

| | |
|---|---|
| Source MAC Accress | 02 03 04 05 06 07 |
| Destination MAC Address | 02 03 04 05 06 08 |
| Key | 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12<br>34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 |
| PN | 0x000000000001 |
| IV | 00 20 01 20 00 00 00 00 |
| Phase1 | bb 58 07 1f 9e 93 b4 38 25 4b |
| Phase2 | 00 20 01 4c fe 67 be d2 7c 86 7b 1b f8 02 8b 1c |

**Table H.8—Sample plaintext and ciphertext MPDUs, using parameter from Table H.7**

| | |
|---|---|
| Plaintext MPDU with TKIP MIC | 08 42 2c 00 02 03 04 05 06 08 02 03 04 05 06 07<br>02 03 04 05 06 07 d0 02 00 20 01 20 00 00 00 00<br>aa aa 03 00 00 00 08 00 45 00 00 54 00 00 40 00<br>40 01 a5 55 c0 a8 0a 02 c0 a8 0a 01 08 00 3a b0<br>00 00 00 00 cd 4c 05 00 00 00 00 00 08 09 0a 0b<br>0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b<br>1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b<br>2c 2d 2e 2f 30 31 32 33 34 35 36 37 68 81 a3 f3<br>d6 48 d0 3c |
| Encrypted MPDU with MIC and ICV | 08 42 2c 00 02 03 04 05 06 08 02 03 04 05 06 07<br>02 03 04 05 06 07 d0 02 00 20 01 20 00 00 00 00<br>c0 0e 14 fc e7 cf ab c7 75 47 e6 66 e5 7c 0d ac<br>70 4a 1e 35 8a 88 c1 1c 8e 2e 28 2e 38 01 02 7a<br>46 56 05 5e e9 3e 9c 25 47 02 e9 73 58 05 dd b5<br>76 9b a7 3f 1e bb 56 e8 44 ef 91 22 85 d3 dd 6e<br>54 1e 82 38 73 55 8a db a0 79 06 8a bd 7f 7f 50<br>95 96 75 ac c4 b4 de 9a a9 9c 05 f2 89 a7 c5 2f<br>ee 5b fc 14 f6 f8 e5 f8 |

Copyright © 2007 IEEE. All rights reserved.

### H.6.4 CCMP test vector

```
==== CCMP test mpdu ====

-- MPDU Fields

Version  = 0
Type     = 2   SubType  = 0  Data
ToDS     = 0   FromDS   = 0
MoreFrag = 0   Retry    = 1
PwrMgt   = 0   moreData = 0
Encrypt  = 1
Order    = 0
Duration = 11459
A1 = 0f-d2-e1-28-a5-7c    DA
A2 = 50-30-f1-84-44-08    SA
A3 = ab-ae-a5-b8-fc-ba    BSSID
SC = 0x3380
seqNum = 824 (0x0338)  fraqNum = 0 (0x00)
Algorithm = AES_CCM
Key ID = 0

TK = c9 7c 1f 67 ce 37 11 85  51 4a 8a 19 f2 bd d5 2f

PN = 199027030681356  (0xB5039776E70C)

802.11 Header = 08 48 c3 2c 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba
              80 33

Muted 802.11 Header = 08 40 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba
              00 00

CCMP Header = 0c e7 00 20 76 97 03 b5

CCM Nonce = 00 50 30 f1 84 44 08 b5  03 97 76 e7 0c

Plaintext Data = f8 ba 1a 55 d0 2f 85 ae 96 7b b6 2f b6 cd a8 eb 7e 78 a0 50

CCM MIC = 78 45 ce 0b 16 f9 76 23

-- Encrypted MPDU with FCS

08 48 c3 2c 0f d2 e1 28 a5 7c 50 30 f1 84 44 08 ab ae a5 b8 fc ba 80 33 0c e7 00
              20 76 97 03 b5 f3 d0 a2 fe 9a 3d bf 23 42 a6 43 e4 32 46 e8 0c 3c 04
              d0 19 78 45 ce 0b 16 f9 76 23 1d 99 f0 66
```

### H.6.5 PRF test vectors

A set of test vectors are provided for each size of PRF function used in this subclause. See Table H.9 through Table H.12. The inputs to the PRF function are strings for key, prefix, and data. The length can be any multiple of 8, but the values 192, 256, 384, and 512 are used in this subclause. The test vectors were taken from IETF RFC 2202-1997 [B17] with additional vectors added to test larger key and data sizes.

**Table H.9—RSN PRF Test Vector 1**

| | |
|---|---|
| Test_case | 1 |
| Key | 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b |


Copyright © 2007 IEEE. All rights reserved.

**Table H.9—RSN PRF Test Vector 1 *(continued)***

| | |
|---|---|
| Prefix | "prefix" |
| Data | "Hi There" |
| Length | 192 |
| PRF-192 | bc d4 c6 50 b3 0b 96 84  95 18 29 e0 d7 5f 9d 54<br>b8 62 17 5e d9 f0 06 06 |

**Table H.10—RSN PRF Test Vector 2**

| | |
|---|---|
| Test_case | 2 |
| Key | 'Jefe' |
| Prefix | "prefix-2" |
| Data | "what do ya want for nothing?" |
| Length | 256 |
| PRF-256 | 47 c4 90 8e 30 c9 47 52  1a d2 0b e9 05 34 50 ec<br>be a2 3d 3a a6 04 b7 73  26 d8 b3 82 5f f7 47 5c |

**Table H.11—RSN PRF Test Vector 3**

| | |
|---|---|
| Test_case | 3 |
| Key | aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa<br>aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa<br>aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa<br>aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa |
| Prefix | "prefix-3" |
| Data | "Test Using Larger Than Block-Size Key - Hash Key First" |
| Length | 384 |
| PRF-384 | 0a b6 c3 3c cf 70 d0 d7  36 f4 b0 4c 8a 73 73 25<br>55 11 ab c5 07 37 13 16  3b d0 b8 c9 ee b7 e1 95<br>6f a0 66 82 0a 73 dd ee  3f 6d 3b d4 07 e0 68 2a |

**Table H.12—RSN PRF Test Vector 4**

| | |
|---|---|
| Test_case | 4 |
| Key | 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b |
| Prefix | "prefix-4" |

Copyright © 2007 IEEE. All rights reserved.

**Table H.12—RSN PRF Test Vector 4 *(continued)***

| | |
|---|---|
| Data | "Hi There Again" |
| Length | 512 |
| PRF-512 | 24 8c fb c5 32 ab 38 ff  a4 83 c8 a2 e4 0b f1 70<br>eb 54 2a 2e 09 16 d7 bf  6d 97 da 2c 4c 5c a8 77<br>73 6c 53 a6 5b 03 fa 4b  37 45 ce 76 13 f6 ad 68<br>e0 e4 a7 98 b7 cf 69 1c  96 17 6f d6 34 a5 9a 49 |

## H.7 Key hierarchy test vectors

The test vectors in H.7.1 provide an example of PTK derivation for both CCMP and TKIP.

### H.7.1 Pairwise key derivation

Pairwise keys are derived from the PMK, AA, SPA, SNonce, and ANonce. The values in Table H.13 are used as input to the pairwise key derivation test vectors.

**Table H.13—Sample values for pairwise key derivations**

| | |
|---|---|
| PMK | 0d c0 d6 eb 90 55 5e d6  41 97 56 b9 a1 5e c3 e3<br>20 9b 63 df 70 7d d5 08  d1 45 81 f8 98 27 21 af |
| AA | a0 a1 a1 a3 a4 a5 |
| SPA | b0 b1 b2 b3 b4 b5 |
| SNonce | c0 c1 c2 c3 c4 c5 c6 c7  c8 c9 d0 d1 d2 d3 d4 d5<br>d6 d7 d8 d9 da db dc dd  de df e0 e1 e2 e3 e4 e5 |
| ANonce | e0 e1 e2 e3 e4 e5 e6 e7  e8 e9 f0 f1 f2 f3 f4 f5<br>f6 f7 f8 f9 fa fb fc fd  fe ff 00 01 02 03 04 05 |

#### H.7.1.1 CCMP pairwise key derivation

Using the values from Table H.13 for PMK, AA, SPA, SNonce, and ANonce, the key derivation process for CCMP generates a temporal key as shown in Table H.14.

**Table H.14—Sample derived CCMP temporal key (TK)**

| | |
|---|---|
| TK | b2 36 0c 79 e9 71 0f dd  58 be a9 3d ea f0 65 99 |


Copyright © 2007 IEEE. All rights reserved.

### H.7.1.2 TKIP pairwise key derivation

Using the values from Table H.13 for PMK, AA, SPA, SNonce, and ANonce, the key derivation process for TKIP generates the values shown in Table H.15.

**Table H.15—Sample derived PTK**

| | |
|---|---|
| KCK | 37 9f 98 52 d0 19 92 36  b9 4e 40 7c e4 c0 0e c8 |
| KEK | 47 c9 ed c0 1c 2c 6e 5b  49 10 ca dd fb 3e 51 a7 |
| TK | b2 36 0c 79 e9 71 0f dd  58 be a9 3d ea f0 65 99<br>db 98 0a fb c2 9c 15 28  55 74 0a 6c e5 ae 38 27 |
| Authenticator Tx MIC_key | db 98 0a fb c2 9c 15 28 |
| Supplicant Tx MIC_key | 55 74 0a 6c e5 ae 38 27 |

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex I

(informative)

# Regulatory classes

## I.1 External regulatory references

This annex and Annex J provide information and specifications for operation in many regulatory domains.

WLANs implemented in accordance with this standard and the specifications and definitions referenced in it are subject to equipment certification and operating requirements established by regional and national regulatory administrations. The specification establishes minimum technical requirements for interoperability, based upon established regulations at the time this standard was issued. These regional and national regulations are subject to revision or may be superseded. Regulatory requirements that do not affect inter-operability are not addressed in this standard. Implementers are referred to the regulatory sources in Table I.1 for further information. Operation in countries within defined regulatory domains may be subject to additional or alternative national regulations.

The documents listed in Table I.1 specify the current regulatory requirements for various geographic areas at the time this standard was developed. They are provided for information only and are subject to change or revision at any time.

**Table I.1—Regulatory requirement list**

| Geographic area | Approval standards | Documents | Approval authority |
|---|---|---|---|
| Japan | Ministry of Internal Affairs and Communications (MIC) | MIC Equipment Ordinance (EO) for Regulating Radio Equipment Articles 7, 49.20, 49.21[a] | MIC |
| United States | Federal Communications Commission (FCC) | FCC CFR47 [B8], Part 15, Sections 15.205, 15.209, and 15.247; and Subpart E, Sections 15.401–15.407, Section 90.210, Section 90.1201–90.1217 | FCC |
| Europe | European Conference of Postal and Telecommunications (CEPT) Administrations and its Electronic Communications Committee (ECC). Also, European Radiocommunications Office, European Telecommunications Standards Institute | ECC DEC (04) 08, ETSI EN301 893 | CEPT |

[a]Frequency planning for licensed STAs in Japan is performed by the regulatory authority and the licensees, addressing the coexistence among STAs operating with a variety of air propagation times and the coexistence between STAs using 20 MHz channel spacing, STAs operating with 10 MHz channel spacing, and STAs operating with 5 MHz channel spacing. Note also the CCA mechanism is preserved in licensed operation.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1189 of 1232

Emissions limits sets are listed in Table I.2, and behavior limits sets are specified in Table I.3. The external regulatory references in Table I.2 and Table I.3 are informative, as they do not specify technical parameters.

**Table I.2—Emissions limits sets**

| Emissions limits set | United States | Europe | Japan |
|---|---|---|---|
| 0 | Not specified | Not specified | Not specified |
| 1 nomadic use | FCC CFR47 [B8], Section 15.407 | ETSI EN 301 389-1 | MIC EO Articles 49.20, 49.21 |
| 2 interference-limited areas[a] | Reserved | Reserved | MIC EO Article 49.21 |
| 3 other interference areas[a] | Reserved | Reserved | MIC EO Article 49.21 |
| 4 | FCC CFR47 [B8], Section 15.247 | Reserved | Reserved |
| 5 public safety | FCC CFR47 [B8], Section 90.210 | Reserved | Reserved |
| 6–255 | Reserved | Reserved | Reserved |

[a]The deployment in Japan of licensed radios in dense urban areas (i.e., interference-limited areas) and other areas (i.e., other interference areas) are specified to allow more radios to operate in dense urban areas, while permitting what can be less expensive radio designs to be used elsewhere.

**Table I.3—Behavior limits sets**

| Behavior limits set | United States | Europe | Japan |
|---|---|---|---|
| 0 | Not specified | Not specified | Not specified |
| 1 nomadic use | FCC CFR47 [B8], Section 15.407 | ETSI EN 301 389-1 | MIC EO Articles 49.20, 49.21 |
| 2 indoor only use | FCC CFR47 [B8], Section 15.407(e) | ETSI EN 301 389-1 | MIC EO Article 49.20 |
| 3 transmit power control | Reserved | ETSI EN 301 389-1 | Reserved |
| 4 dynamic frequency selection | Reserved | ETSI EN 301 389-1 | Reserved |
| 5 IBSS prohibited | Reserved | Reserved | MIC EO Article 49.21 |
| 6 4 ms CS[a] | Reserved | Reserved | MIC EO Articles 49.20, 49.21 |
| 7 licensed base STA | Reserved | Reserved | MIC EO Article 49.21 |
| 8 mobile STA | Reserved | Reserved | MIC EO Articles 49.20, 49.21 |
| 9 public safety | FCC CFR47 [B8], Section 90.1209 | Reserved | Reserved |
| 10–255 | Reserved | Reserved | Reserved |

[a]The Japanese 4 ms CS rule says no STA can transmit for more than 4 ms without carrier sensing, whether transmitting fragments or frames, unless it is controlled by another STA.


Copyright © 2007 IEEE. All rights reserved.

## I.2 Radio performance specifications

### I.2.1 Transmit and receive in-band and out-of-band spurious emissions

Spurious transmissions from compliant devices shall conform to national regulations. For operation in the United States, refer to FCC CFR47 [B8], Section 15.407. For operation in Europe, refer to ETSI EN 301 389-1. For operation in Japan, refer to MIC EO Article 49.20 and Article 49.21, Section 1.

### I.2.2 Transmit power levels

The maximum allowable output power by regulatory domain (except in Japan) is shown in Table I.4. The maximum allowable output power by regulatory domain for the U.S. 4.9 GHz public safety band is shown in Table I.5.

**Table I.4—Transmit power level by regulatory domain**

| Frequency band (GHz) | United States (Maximum output power with up to 6 dBi antenna gain) (mW) | Europe (EIRP) |
|---|---|---|
| 5.15–5.25 | 40 (2.5 mW/MHz) | 200 mW |
| 5.25–5.35 | 200 (12.5 mW/MHz) | 200 mW |
| 5.470–5.725 | — | 1 W |
| 5.725–5.825 | 800 (50 mW/MHz) | — |

**Table I.5—U.S. public safety transmit power levels by regulatory domain**

| Frequency band (GHz) | U.S. public safety (mW) | | |
|---|---|---|---|
| | 20 MHz channels | 10 MHz channels | 5 MHz channels |
| 4.94–4.99 low power | 100 | 50 | 25 |
| 4.94–4.99 high power | 2000 | 1000 | 500 |

The maximum allowable output power by Japanese regulatory domain is shown in Table I.6. For the Japanese 4.9 GHz and 5.0 GHz bands, the maximum deviation of designated or rated equivalent isotropically radiated power (EIRP) shall be nominal +20% and nominal –80%.

Copyright © 2007 IEEE. All rights reserved.

**Table I.6—Japanese transmit power levels by regulatory domain**

| Frequency band (GHz) | Regulatory type | Japan |
|---|---|---|
| 4.9–5.091 | Fixed wireless access, licensed | < 250 mW EIRP, and<br>< 50 mW/MHz EIRP for licensed access |
| 4.9–5.091 | Nomadic access, unlicensed | < 10 mW/MHz EIRP |
| 5.15–5.25 | Unlicensed | < 10 mW/MHz EIRP |

### I.2.3 Transmit spectrum mask

For operation using 20 MHz channel spacing, the transmitted spectrum shall have a 0 dBr (decibel relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset, and –40 dBr at 30 MHz frequency offset and above. For operation using 10 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 9 MHz, –20 dBr at 5.5 MHz frequency offset, –28 dBr at 10 MHz frequency offset, and –40 dBr at 15 MHz frequency offset and above. The transmitted spectral density of the transmitted signal shall fall within the spectral mask, as shown in Figure I.1. The measurements shall be made using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

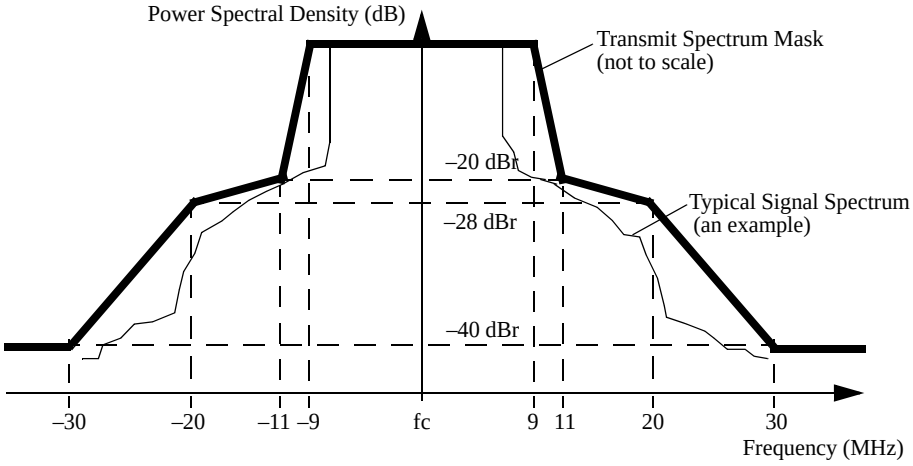



**Figure I.1—Transmit spectrum mask**

For operation in Japan, the average power emitted in adjacent and alternate adjacent channels of the same channel width shall be less than –3 dBr and –18 dBr, respectively, for both the 4.9 GHz and 5.0 GHz bands, measured relative to the EIRP in the 100 kHz at the channel edges and the band edges. For operation in Japan in the 5.15–5.25 GHz band, the average power emitted in adjacent and alternate adjacent channels shall be lower than average on-channel power by 25 dB and 40 dB, respectively.


Copyright © 2007 IEEE. All rights reserved.

For operation in the 4.94–4.99 GHz low-power U.S. public safety band, FCC CFR47 [B8], Section 90.210 Mask L, the transmitted spectrum shall be as follows:

— For operation using 20 MHz channel spacing, the transmitted spectrum shall have a 0 dBr (decibel relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –10 dBr at 10 MHz frequency offset, –20 dBr at 11 MHz frequency offset, –28 dBr at 20 MHz frequency offset, –40 dBr at 30 MHz frequency offset, and –50 dBr above.

— For operation using 10 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 9 MHz, –10 dBr at 5 MHz frequency offset, –20 dBr at 5.5 MHz frequency offset, –28 dBr at 10 MHz frequency offset, –40 dBr at 15 MHz frequency offset, and –50 dBr above.

— For operation using 5 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 4.5 MHz, –10 dBr at 2.5 MHz frequency offset, –20 dBr at 2.75 MHz frequency offset, –28 dBr at 5 MHz frequency offset, –40 dBr at 7.5 MHz frequency offset, and –50 dBr above.

For operation in the 4.94–4.99 GHz high-power U.S. public safety band, FCC CFR47 [B8], Section 90.210 Mask M, the transmitted spectrum shall be as follows:

— For operation using 20 MHz channel spacing, the transmitted spectrum shall have a 0 dBr (decibel relative to the maximum spectral density of the signal) bandwidth not exceeding 18 MHz, –26 dBr at 10 MHz frequency offset, –32 dBr at 11 MHz frequency offset, –40 dBr at 20 MHz frequency offset, –50 dBr at 30 MHz frequency offset and above.

— For operation using 10 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 9 MHz, –26 dBr at 5 MHz frequency offset, –32 dBr at 5.5 MHz frequency offset, –40 dBr at 10 MHz frequency offset, –50 dBr at 15 MHz frequency offset and above.

— For operation using 5 MHz channel spacing, the transmitted spectrum shall have a 0 dBr bandwidth not exceeding 4.5 MHz, –26 dBr at 2.5 MHz frequency offset, –32 dBr at 2.75 MHz frequency offset, –40 dBr at 5 MHz frequency offset, –50 dBr at 7.5 MHz frequency offset and above.

Figure I.2 shows the spectral masks for the U.S. 4.9 GHz public safety band.

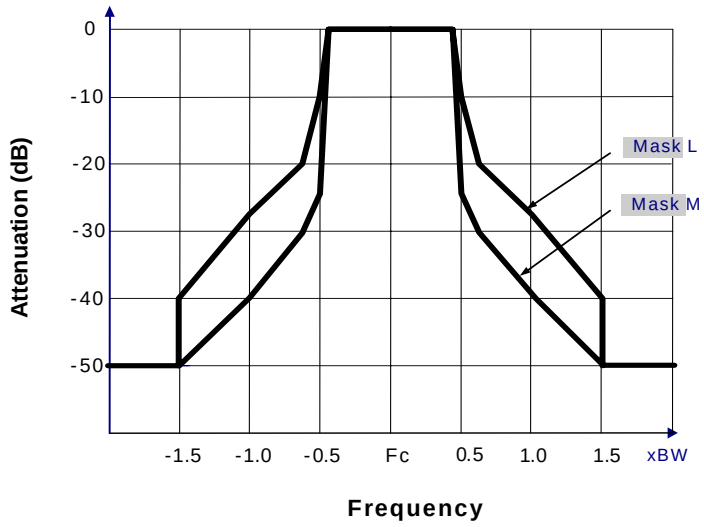



**Figure I.2—Transmit spectrum masks for the U.S. 4.9 GHz public safety band**

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex J

(normative)

## Country information element and regulatory classes

The Country information element (see 7.3.2.9) allows a STA to configure its PHY and MAC for operation when the regulatory triplet of Regulatory Extension Identifier, Regulatory Class, and Coverage Class fields is present. The regulatory triplet indicates both PHY and MAC configuration characteristics and operational characteristics. The First Channel Number field of subsequent subband triplet(s) is based on the dot11ChannelStartingFactor that is indicated by the Regulatory Class field.

The regulatory class for the OFDM PHY is an index into a set of values for radio equipment sets of rules.

The channel starting frequency variable is a frequency, used together with a channel number, to calculate a channel center frequency.

Channel spacing is the frequency difference between nonoverlapping adjacent channel center frequencies.

The channel set shall be the list of integer channel numbers that are legal for a regulatory domain and class.

The transmit power limit shall be the maximum transmit power that is legal for a regulatory domain and class.

An emissions limits set is an enumerated list of spectral masks and emissions limits that are legal for a regulatory domain and are listed in Table I.1.

Specific transmit restrictions and limits are listed in I.2.

A behavior limits set is an enumerated list of behaviors that are legal for a regulatory domain and are specified in Table I.3.

The regulatory classes specified for 4.9 GHz and 5 GHz operation in the United States are enumerated in Table J.1.

The regulatory classes specified for 5 GHz operation in Europe are enumerated in Table J.2.

The regulatory classes specified for 4.9 GHz and 5 GHz operation in Japan are enumerated in Table J.3.

Copyright © 2007 IEEE. All rights reserved.

**Table J.1—Regulatory classes for 4.9 GHz and 5 GHz bands in the United States**

| Regulatory class | Channel starting frequency (GHz) | Channel spacing (MHz) | Channel set | Transmit power limit (mW) | Emissions limits set | Behavior limits set |
|---|---|---|---|---|---|---|
| 1 | 5 | 20 | 36, 40, 44, 48 | 40 | 1 | 1, 2 |
| 2 | 5 | 20 | 52, 56, 60, 64 | 200 | 1 | 1 |
| 3 | 5 | 20 | 149, 153, 157, 161 | 800 | 1 | 1 |
| 4 | 5 | 20 | 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140 | 200 | 1 | 1 |
| 5 | 5 | 20 | 165 | 1000 | 4 | 1 |
| 6 | 4.9375 | 5 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 | 25 | 5 | 9 |
| 7 | 4.9375 | 5 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 | 500 | 5 | 9 |
| 8 | 4.89 | 10 | 11, 13, 15, 17, 19 | 50 | 5 | 9 |
| 9 | 4.89 | 10 | 11, 13, 15, 17, 19 | 1000 | 5 | 9 |
| 10 | 4.85 | 20 | 21, 25 | 100 | 5 | 9 |
| 11 | 4.85 | 20 | 21, 25 | 2000 | 5 | 9 |
| 12–255 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |

**Table J.2—Regulatory classes for 5 GHz bands in Europe**

| Regulatory class | Channel starting frequency (GHz) | Channel spacing (MHz) | Channel set | Transmit power limit (EIRP) | Emissions limits set | Behavior limits set |
|---|---|---|---|---|---|---|
| 1 | 5 | 20 | 36, 40, 44, 48 | 200 | 1 | 2, 3 |
| 2 | 5 | 20 | 52, 56, 60, 64 | 200 | 1 | 1, 3, 4 |
| 3 | 5 | 20 | 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140 | 1 W | 1 | 1, 3, 4 |
| 4–255 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |


Copyright © 2007 IEEE. All rights reserved.

**Table J.3—Regulatory classes for 4.9 GHz and 5 GHz bands in Japan**

| Regulatory class | Channel starting frequency (GHz) | Channel spacing (MHz) | Channel set | Transmit power limit (dBm) | Emissions limits set | Behavior limits set |
|---|---|---|---|---|---|---|
| 1 | 5 | 20 | 34, 38, 42, 46 | 22 | 1 | 1, 2, 6 |
| 2 | 5 | 20 | 8, 12, 16 | 24 | 2 | 5, 6, 7 |
| 3 | 5 | 20 | 8, 12, 16 | 24 | 2 | 5, 6, 8 |
| 4 | 5 | 20 | 8, 12, 16 | 24 | 3 | 5, 6, 7 |
| 5 | 5 | 20 | 8, 12, 16 | 24 | 3 | 5, 6, 8 |
| 6 | 5 | 20 | 8, 12, 16 | 22 | 1 | 5, 6, 8 |
| 7 | 4 | 20 | 184, 188, 192, 196 | 24 | 2 | 5, 6, 7 |
| 8 | 4 | 20 | 184, 188, 192, 196 | 24 | 2 | 5, 6, 8 |
| 9 | 4 | 20 | 184, 188, 192, 196 | 24 | 3 | 5, 6, 7 |
| 10 | 4 | 20 | 184, 188, 192, 196 | 24 | 3 | 5, 6, 8 |
| 11 | 4 | 20 | 184, 188, 192, 196 | 22 | 1 | 5, 6, 8 |
| 12 | 5 | 10 | 7, 8, 9, 11 | 24 | 2 | 5, 6, 7 |
| 13 | 5 | 10 | 7, 8, 9, 11 | 24 | 2 | 5, 6, 8 |
| 14 | 5 | 10 | 7, 8, 9, 11 | 24 | 3 | 5, 6, 7 |
| 15 | 5 | 10 | 7, 8, 9, 11 | 24 | 3 | 5, 6, 8 |
| 16 | 4 | 10 | 183, 184, 185, 187, 188, 189 | 24 | 2 | 5, 6, 7 |
| 17 | 4 | 10 | 183, 184, 185, 187, 188, 189 | 24 | 2 | 5, 6, 8 |
| 18 | 4 | 10 | 183, 184, 185, 187, 188, 189 | 24 | 3 | 5, 6, 7 |
| 19 | 4 | 10 | 183, 184, 185, 187, 188, 189 | 24 | 3 | 5, 6, 8 |
| 20 | 4 | 10 | 183, 184, 185, 187, 188, 189 | 17 | 1 | 5, 6, 8 |
| 21 | 5.0025 | 5 | 6, 7, 8, 9, 10, 11 | 24 | 2 | 5, 6, 7 |
| 22 | 5.0025 | 5 | 6, 7, 8, 9, 10, 11 | 24 | 2 | 5, 6, 8 |
| 23 | 5.0025 | 5 | 6, 7, 8, 9, 10, 11 | 24 | 3 | 5, 6, 7 |
| 24 | 5.0025 | 5 | 6, 7, 8, 9, 10, 11 | 24 | 3 | 5, 6, 8 |
| 25 | 4.0025 | 5 | 182, 183, 184, 185, 186, 187, 188, 189 | 24 | 2 | 5, 6, 7 |

Copyright © 2007 IEEE. All rights reserved.

**Table J.3—Regulatory classes for 4.9 GHz and 5 GHz bands in Japan *(continued)***

| Regulatory class | Channel starting frequency (GHz) | Channel spacing (MHz) | Channel set | Transmit power limit (dBm) | Emissions limits set | Behavior limits set |
|---|---|---|---|---|---|---|
| 26 | 4.0025 | 5 | 182, 183, 184, 185, 186, 187, 188, 189 | 24 | 2 | 5, 6, 8 |
| 27 | 4.0025 | 5 | 182, 183, 184, 185, 186, 187, 188, 189 | 24 | 3 | 5, 6, 7 |
| 28 | 4.0025 | 5 | 182, 183, 184, 185, 186, 187, 188, 189 | 24 | 3 | 5, 6, 8 |
| 29 | 4.0025 | 5 | 182, 183, 184, 185, 186, 187, 188, 189 | 17 | 1 | 5, 6, 8 |
| 30–255 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |


Copyright © 2007 IEEE. All rights reserved.

# 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 |
| Minimum PHY Rate | Always specified | | | | |

Copyright © 2007 IEEE. All rights reserved.

**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) |
|---|---|---|---|---|---|
| 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 STA; 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 AP should account for unadmitted traffic in these ACs.

### K.2.2 Deriving medium time

It is recommended that the AP 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


Copyright © 2007 IEEE. All rights reserved.

## 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 STA to derive the aggregate minimum SI contained in the non-AP STA'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 STA'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 STA's admitted TSs. The aggregate burst size is equal to the sum of the burst size of all of the non-AP STA'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 STA is assigned the TXOPs, not any particular stream.

Copyright © 2007 IEEE. All rights reserved.

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 STA 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 $N_{\text{excess}}$ in order to avoid dropping packets with some fixed probability (denoted as $p_{\text{drop}}$). For example, if $p_{\text{drop}} = 10^{-8}$, then the number of retries $N_{\text{excess}}$ 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


Copyright © 2007 IEEE. All rights reserved.

Surplus bandwidth allowance = $\frac{100 + N_{excess}}{100}$, 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 $1.6 \times 10^{-15}$.

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 $N_{allocated}$ 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 $\frac{N_{allocated}}{N_{payload}}$.

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 STA 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.

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1203 of 1232

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 STAs with admitted streams. See Figure K.1.

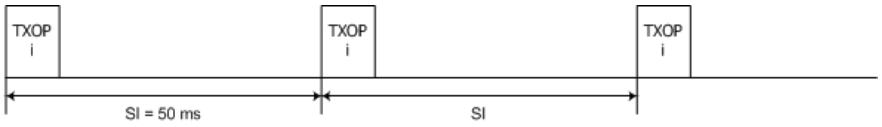



**Figure K.1—Schedule for stream from STA *i***

For the calculation of the TXOP duration for an admitted stream, the scheduler uses the following parameters: Mean Data Rate ($\rho$) 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 octets ($M$), 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 STA *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 STAs *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.


Copyright © 2007 IEEE. All rights reserved.

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 STAs following the STA dropped to use the unused time. An example is shown in Figure K.3, when the stream for STA *j* is removed. However, this option might require the announcement of a new schedule to all STAs.




**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 STAs, 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.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex L

(informative)

# An example of encoding a TIM virtual bit map

## L.1 Introduction

The purpose of this annex is to show an example of encoding a Partial Virtual Bit Map field of the TIM information element of the MAC, as described in 7.3.2.6.

## L.2 Examples

The following examples help clarify the use of TIM values.

The first example is one in which there are no broadcast or multicast MSDUs buffered in the AP but there is traffic for two STAs queued in the AP. STAs with AID 2 and AID 7 have data buffered in the AP. Figure L.1 shows the values of the Bit Map Control and Partial Virtual Bit Map fields that would be part of the TIM information element for this example.

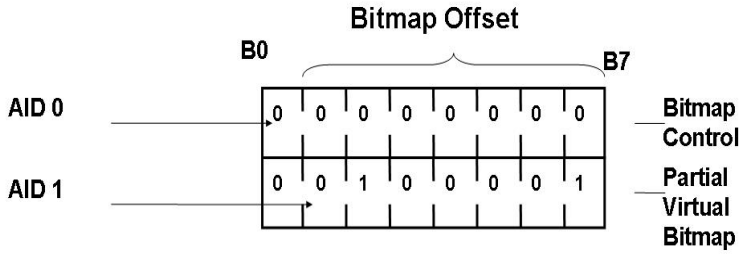



**Figure L.1—Virtual bitmap example #1**

The next example is one in which broadcast or multicast MSDUs are buffered in the AP as well as traffic for STAs. The DTIM Count field in the TIM IE equals zero. STAs with AID 2, AID 7, AID 22 and AID 24 have data buffered in the AP. Figure L.2 shows the values of the Bit Map Control and Partial Virtual Bit Map fields.

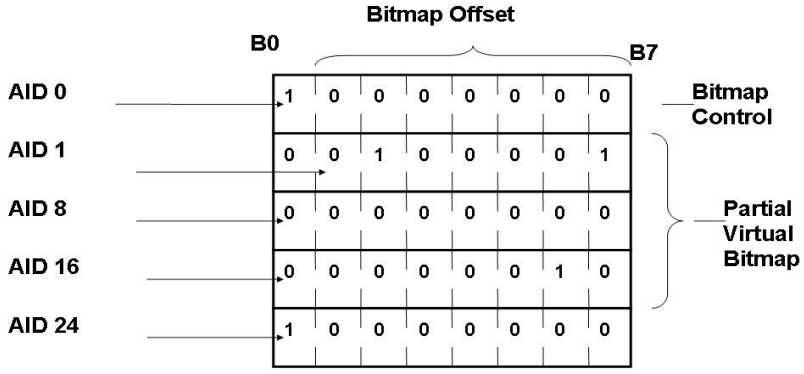



**Figure L.2—Virtual bitmap example #2**

Copyright © 2007 IEEE. All rights reserved.

Another example is one in which broadcast or multicast MSDUs are buffered in the AP as well as traffic for STAs. The DTIM Count field in the TIM IE equals zero. Only the node with AID 24 has data buffered in the AP. In this example, the Bit Map Offset is used to start the Partial Virtual Bit Map at the third byte. Figure L.3 shows the values of the Bit Map Control and Partial Virtual Bit Map fields.

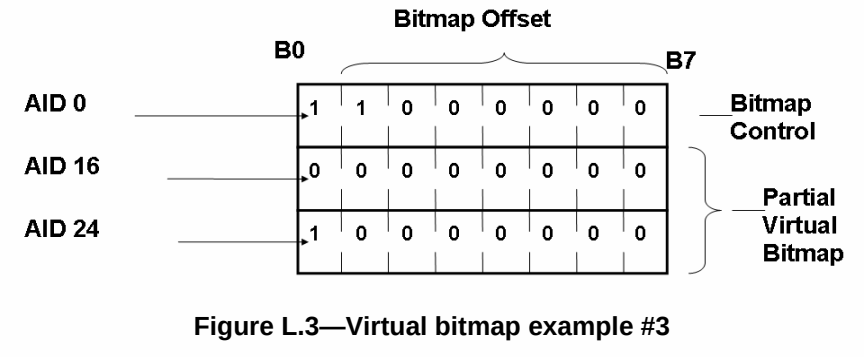



**Figure L.3—Virtual bitmap example #3**

## L.3 Sample C code

The following C source code illustrates how to construct the TIM Virtual Bit Map. Because this is an illustration, no efficiency or appropriateness for actual implementation is implied.

```
#include <stdio.h>

#define ADD_TIM_BIT     0
#define REMOVE_TIM_BIT  1

#define TIM_ELEMENT_ID  5
#define TIM_BASE_SIZE   3       /* size of TIM fields */

#define AID_SIZE        2008    /* valid AIDs are 1 thru 2007 */
#define VBM_SIZE        251     /* size of VBM array = 2008/8 = 251 */

typedef unsigned char UINT8;
typedef unsigned short int UINT16;


struct  _tim
   {
   UINT8   Element_id;
   UINT8   IELength;
   UINT8   DtimCount;
   UINT8   DtimPeriod;
   UINT8   BitMapControl;
   UINT8   PartialVirtualBitMap [VBM_SIZE];
   };


UINT8   virtualBitMap [VBM_SIZE];
UINT8   mcast_pending = 0;
UINT8   dtimCount = 0;
UINT8   dtimPeriod = 5;
```


Copyright © 2007 IEEE. All rights reserved.

```
void
Build_TIM (struct _tim * Tim)
   {
   UINT8 octetIndex;
   UINT8 offset = 0;
   UINT8 lengthOfPartialVirtualBitMap = 0;

   /* Find first nonzero octet in the virtual bit map */
   for (octetIndex = 0; ((virtualBitMap [octetIndex] == 0) && (octetIndex < VBM_SIZE)); octetIndex++)
      /* empty */;

   if (octetIndex < VBM_SIZE)
      /* Clear the lsb as it is reserved for the broadcast/ multicast indication bit */
      offset = octetIndex & 0xFE;

   /* Find last nonzero octet in the virtual bit map */
   for (octetIndex = (VBM_SIZE-1); ((virtualBitMap [octetIndex] == 0) && (octetIndex > 0)); octetIndex--)
      /* empty */;

   lengthOfPartialVirtualBitMap = octetIndex - offset + 1;

   Tim->Element_id = TIM_ELEMENT_ID;
   Tim->IELength = lengthOfPartialVirtualBitMap + TIM_BASE_SIZE;
   Tim->DtimCount = dtimCount;
   Tim->DtimPeriod = dtimPeriod;
   Tim->BitMapControl = offset;

   /* Update broadcast/ multicast indication bit if necessary */
   if ((Tim->DtimCount == 0) && mcast_pending)
      Tim->BitMapControl |= 0x01;

   /* Copy the virtual bit map octets that are nonzero */
   /* Note: A NULL virtualBitMap will still add a single octet of zero */
   for (octetIndex = 0;  octetIndex < lengthOfPartialVirtualBitMap;  octetIndex++)
      Tim->PartialVirtualBitMap [octetIndex] = virtualBitMap [offset + octetIndex];
   }


void
Update_VirtualBitMap (UINT16 station_id, UINT8 Action)
   {
   UINT16  aid = station_id;
   UINT8   aid_octet;
   UINT8   aid_bit;

   if ((aid > 0) && (aid < AID_SIZE))
      {
      /* Get aid position in Virtual Bit Map. */
      aid_octet = (UINT8)(aid >> 3);
      aid_bit   = (UINT8)(0x01 << (aid & 0x07));

      if (Action == REMOVE_TIM_BIT)
```

Copyright © 2007 IEEE. All rights reserved.

```
        virtualBitMap [aid_octet] &= ~aid_bit;
    else
        virtualBitMap [aid_octet] |= aid_bit;
    }
}


void main (void)

{
struct _tim    Tim;
UINT8          ExampleCase;

ExampleCase = 1;
switch (ExampleCase)
    {
    case 1:
        mcast_pending = 0;
        Update_VirtualBitMap (2, ADD_TIM_BIT);
        Update_VirtualBitMap (7, ADD_TIM_BIT);
        break;
    case 2:
        mcast_pending = 1;
        Update_VirtualBitMap (2, ADD_TIM_BIT);
        Update_VirtualBitMap (7, ADD_TIM_BIT);
        Update_VirtualBitMap (22, ADD_TIM_BIT);
        Update_VirtualBitMap (24, ADD_TIM_BIT);
        break;
    case 3:
        mcast_pending = 1;
        Update_VirtualBitMap (24, ADD_TIM_BIT);
        break;
    case 4:
        mcast_pending = 0;
        Update_VirtualBitMap (3, ADD_TIM_BIT);
        Update_VirtualBitMap (37, ADD_TIM_BIT);
        Update_VirtualBitMap (43, ADD_TIM_BIT);
        break;
    case 5:
        mcast_pending = 0;
        Update_VirtualBitMap (35, ADD_TIM_BIT);
        break;
    case 6:
        mcast_pending = 0;
        Update_VirtualBitMap (43, ADD_TIM_BIT);
        break;
    case 7:
        mcast_pending = 0;
        Update_VirtualBitMap (35, ADD_TIM_BIT);
        Update_VirtualBitMap (35, REMOVE_TIM_BIT);
        break;
    case 8:
        mcast_pending = 1;
        Update_VirtualBitMap (13, ADD_TIM_BIT);
```


Copyright © 2007 IEEE. All rights reserved.

```
            Update_VirtualBitMap (43, ADD_TIM_BIT);
            Update_VirtualBitMap (63, ADD_TIM_BIT);
            Update_VirtualBitMap (73, ADD_TIM_BIT);
            break;
         case 9:
            mcast_pending = 1;
            Update_VirtualBitMap (2007, ADD_TIM_BIT);
            break;
         default:
            break;
         }

      Build_TIM (&Tim);

      printf ("Element_id = %d.\n", Tim.Element_id);
      printf ("IELength = %d.\n", Tim.IELength);
      printf ("DtimCount = %d.\n", Tim.DtimCount);
      printf ("DtimPeriod = %d.\n", Tim.DtimPeriod);
      printf ("BitMapControl = 0x%02X\n", Tim.BitMapControl);
      if (Tim.IELength - TIM_BASE_SIZE > 0)
         {
         int octetIndex;

         for (octetIndex = 0;  octetIndex < Tim.IELength - TIM_BASE_SIZE;  octetIndex++)
                  printf ("PartialVirtualBitMap [%d] = 0x%02X\n", octetIndex, Tim.PartialVirtualBitMap
[octetIndex]);
         }
      }


/* The End. */


}
```

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex M

(informative)

# Integration function

## M.1 Introduction

The purpose of this annex is to guide the implementor of a WLAN system that includes a portal that integrates the WLAN system with a wired LAN.

## M.2 Ethernet V2.0/IEEE 802.3 LAN integration function

It is recommended that any WLAN system that logically incorporates a portal that integrates the WLAN system with an Ethernet V2.0/IEEE 802.3 LAN use the procedures defined in ISO/IEC Technical Report 11802-5:1997(E) (previously known as IEEE Std 802.1H-1997), with the two-entry selective translation table (STT) shown in Table M.1, to perform the integration service. Note that the majority of such IEEE 802.11 implementations currently use this STT, rather than the single-entry STT recommended in Annex A of IEEE Std 802.1H-1997 [B13].

**Table M.1—IEEE 802.11 integration service STT**

| Protocol use | Ethernet type value | Encoding in Type field | |
|---|---|---|---|
| | | First octet | Second octet |
| AppleTalk Address Resolution Protocol | 0x80F3 | 80 | F3 |
| Novell NetWare Internetwork Packet exchange (IPX) | 0x8137 | 81 | 37 |

## M.3 Example

In order to illustrate the translations performed by the integration service using the encapsulation/ decapsulation procedures defined in ISO/IEC Technical Report 11802-5:1997(E) with the IEEE 802.11 integration service STT, the following tables show how the octets in an IEEE 802.11 MSDU correspond to the octets in the Ethernet/IEEE 802.3 MSDU that represents the same LLC SDU on the integrated Ethernet/ IEEE 802.3 LAN. Table M.2 shows the encapsulation example, and Table M.3 shows the decapsulation example.

Note that examples in both tables showing a Type/Length field value of 81-00 represents bridging between an Ethernet/IEEE 802.3 LAN and an IEEE 802.11 LAN, both of which are carrying VLAN-tagged MSDUs (User Priority=4, CFI-0, VLAN ID=1893).

Copyright © 2007 IEEE. All rights reserved.

**Table M.2—Ethernet/IEEE 802.3 to IEEE 802.11 translation**

| Protocol | Type / Length | LLC header | IEEE 802.11 LLC header |
|---|---|---|---|
| IP | 08-00 | — | AA-AA-03-00-00-00-08-00 |
| IP 802.3[a] | length | AA-AA-03-00-00-00-08-00 | AA-AA-03-00-00-00-08-00 |
| IP ARP | 08-06 | — | AA-AA-03-00-00-00-08-06 |
| AppleTalk (1) | 80-9B | — | AA-AA-03-00-00-00-80-9B |
| AppleTalk (2) | length | AA-AA-03-08-00-07-80-9B | AA-AA-03-08-00-07-80-9B |
| AppleTalk AARP (1) | 80-F3 | — | AA-AA-03-00-00-F8-80-F3 |
| AppleTalk AARP (2) | length | AA-AA-03-00-00-00-80-F3 | AA-AA-03-00-00-00-80-F3 |
| IPX Ethernet II | 81-37 | — | AA-AA-03-00-00-F8-81-37 |
| IPX SNAP | length | AA-AA-03-00-00-00-81-37 | AA-AA-03-00-00-00-81-37 |
| IPX 802.2 | length | E0-E0-03 | E0-E0-03 |
| IPX 802.3[b] | length | FF-FF | FF-FF |
| VLAN-tagged IP | 81-00 | 87-65-08-00 | AA-AA-03-00-00-00-81-00-87-65-AA-AA-03-00-00-00-08-00[c] |

[a]This format of IP packet over IEEE Std 802.3 is denigrated, and the change to the canonical Ethernet IP format is not considered harmful.

[b]The use of this nonstandard format happens to work with these rules, even though the FF-FF is not actually a valid LLC header value. (The broadcast LSAP is not valid as a source SAP in LLC. See IEEE Std 802.2.)

[c]The sequence of octets AA-AA-03-00-00-00-81-00-87-65 represents the SNAP-encoded VLAN header. The sequence of octets AA-AA-03-00-00-00-08-00 represents the IEEE 802.1H-translated Type/Length field, using the same translation as the untagged IP MSDU on the first line of Table M.2.

**Table M.3—IEEE 802.11 to Ethernet/IEEE 802.3 translation**

| Protocol | IEEE 802.11 LLC header | Type / Length | LLC header |
|---|---|---|---|
| IP | AA-AA-03-00-00-00-08-00 | 08-00 | — |
| IP 802.3[a] | AA-AA-03-00-00-00-08-00 | length | — |
| IP ARP | AA-AA-03-00-00-00-08-06 | 08-06 | — |
| AppleTalk (1) | AA-AA-03-00-00-00-80-9B | 80-9B | — |
| AppleTalk (2) | AA-AA-03-08-00-07-80-9B | length | AA-AA-03-08-00-07-80-9B |
| AppleTalk AARP (1) | AA-AA-03-00-00-F8-80-F3 | 80-F3 | — |
| AppleTalk AARP (2) | AA-AA-03-00-00-00-80-F3 | length | AA-AA-03-00-00-00-80-F3 |
| IPX Ethernet II | AA-AA-03-00-00-F8-81-37 | 81-37 | — |
| IPX SNAP | AA-AA-03-00-00-00-81-37 | length | AA-AA-03-00-00-00-81-37 |
| IPX 802.2 | E0-E0-03 | length | E0-E0-03 |


Copyright © 2007 IEEE. All rights reserved.

**Table M.3—IEEE 802.11 to Ethernet/IEEE 802.3 translation *(continued)***

| Protocol | IEEE 802.11 LLC header | Type / Length | LLC header |
|---|---|---|---|
| IPX 802.3 | FF-FF | length | FF-FF |
| VLAN-tagged IP ARP | AA-AA-03-00-00-00-81-00-87-65-AA-AA-03-00-00-00-08-06 | 81-00 | 87-65-08-06 |

[a]This format of IP packet does not survive the trip across the non-IEEE-802.3 LAN intact.

## M.4 Integration service versus bridging

There are a number of differences between the IEEE 802.11 integration service and the service provided by an IEEE 802.1 bridge. In the IEEE 802.11 architecture, a portal provides the minimum connectivity between an IEEE 802.11 WLAN system and a non-IEEE-802.11 LAN. Requiring an IEEE 802.1D bridge in order to be compliant with IEEE Std 802.11 would unnecessarily render some implementations noncompliant.

The most important distinction is that a portal has only one “port” (in the sense of IEEE Std 802.1D, for example) through which it accesses the DS. This renders it unnecessary to update bridging tables inside a portal each time a STA changes its association status. In other words, the details of distributing MSDUs inside the IEEE 802.11 WLAN need not be exposed to the portal.

Another difference is that the DS is not an IEEE 802 LAN (although it carries IEEE 802 LLC SDUs). Requiring that the DS implements all behaviors of an IEEE 802 LAN places an undue burden on the architecture.

Finally, it is an explicit intent of this standard to permit transparent integration of an IEEE 802.11 WLAN into another non-IEEE-802.11 LAN, including passing bridge PDUs through a portal. While an implementer may wish to attach an IEEE 802.1D bridge to the portal (note that the non-IEEE-802.11 LAN interface on the bridge need not be any particular type of LAN), it is not an architectural requirement of this standard to do so.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex N

(informative)

# AP functional description

## N.1 Introduction

This informative annex seeks to clarify the AP functional description. At times there is some confusion surrounding the term “AP” and the relation of that term to the AP functions and common implementations of AP devices. The core IEEE 802.11 conceptual definitions that surround the AP (refer to Clause 5) are abstract (and can sometimes cause confusion), but Clause 5 definitions are crafted to be flexible and hence serve to allow the adaptation and extension of this standard in a wide variety of ways.

## N.2 Terminology

An enhanced description of these access entities begins with clarification of several terms.

This standard defines an entity called a STA. STAs can operate in different modes. The possible operational modes of a STA are

a) Infrastructure mobile STAs
b) Ad hoc mobile STAs
c) Access control mode STAs

The mobile STAs are the STA entities that are ordinarily moving around, but may also be in a fixed location. The mobile adjective prefix often helps in visualizing the type of STA under discussion.

Infrastructure mobile STAs operate in infrastructure BSS mode, i.e., they are the users of an AP. Devices that incorporate an infrastructure mobile STA are referred to in this annex by the term *mobile unit* (MU). An MU device may consist of just a mobile STA implementation, but also likely includes an SME and a client. The exact configuration of the MU is not relevant to the descriptions in this annex.

Ad hoc mobile STAs operate in IBSS mode. Ad hoc mobile STAs form autonomous networks that do not require an AP.

A STA can also form an integral part of an AP. To do so, the STA must operate in access control mode. This type of STA is called an *access control mode STA* (ACM_STA).

The primary function of an AP is to provide the MUs with access to the DS, as shown in the Unified Modeling Language (UML) use case diagram in Figure N.1. Complete UML specifications can be found in Schneier [B32]. The UML diagram shows a system boundary box containing a single use case (ellipse). Entities that are outside the system boundary box are shown as stick people. These external entities represent the actors (formal term), or users, of the system. Since the actors are outside the system boundary, their internal behavior is not described (i.e., it is out of scope for the current view). Instead, references to the external entities are limited to descriptions of their interactions with, and expectations of, the system, which is accomplished by describing the use cases and scenarios (i.e., functions) of the system and later (in Figure N.4) a decomposition of the entities (objects and behaviors) within the system that provide that functionality. The use case diagram employs connecting lines to indicate relationships between the various

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1217 of 1232

artifacts present in the diagram. Relationship lines lacking arrows on both ends indicate that there is a bidirectional relationship between the artifacts.

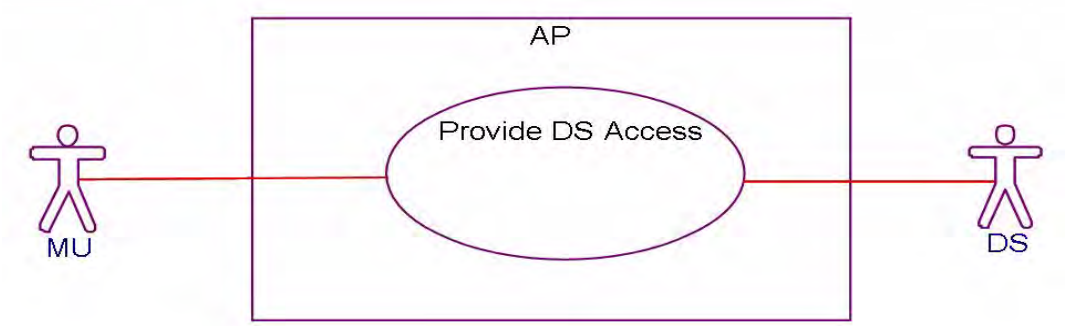



**Figure N.1—Very high level UML use case diagram for the AP**

The DS enables communication between MUs and the construction of collections of APs. To enable communication between MUs and a non-IEEE-802.11-LAN entity requires the presence of a (logical) portal from the DS to the non-IEEE-802.11 LAN.

Often the functions of an AP, which includes an ACM_STA, a DS, and a portal, are combined into a single device, referred to in this annex as an *access unit* (AU). While reference to that basic implementation is commonplace, it is helpful to discuss the abstract case: a WLAN system. The WLAN system includes the DS, AP, the AP's STA, and portal entities. It is also the logical location of distribution and integration service functions of an ESS. An infrastructure WLAN system contains one or more APs and zero or more portals in addition to the DS.

The primary function of a WLAN system is to provide the MUs with access to the non-IEEE-802.11 LAN, as shown in Figure N.2. A secondary function is to provide the MUs with access to each other.

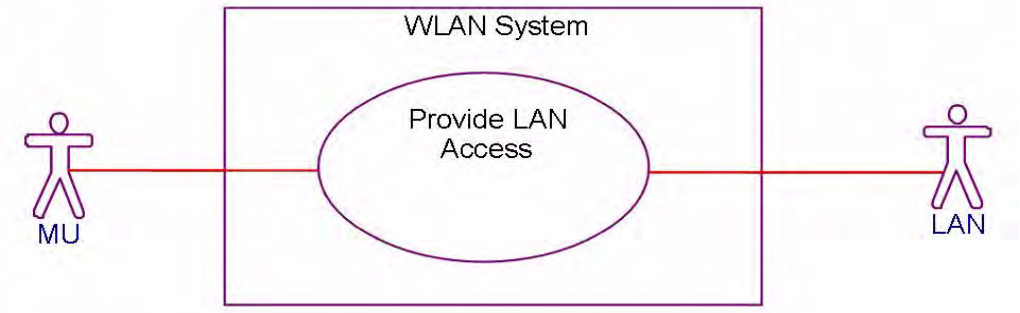



**Figure N.2—Very high level UML use case diagram for the WLAN system**

The primary functions of the WLAN system can be further characterized as follows:

a) Provide non-IEEE-802.11-LAN access.
   1) Includes MU validation.
   2) Includes moving data between the MUs and the non-IEEE-802.11 LAN.
      i) Uses a special data movement function called *filtering data*.

b) Configure the system.


Copyright © 2007 IEEE. All rights reserved.

Those high-level use cases of the WLAN system are shown in the UML use case diagram in Figure N.3. The UML diagram shows multiple use cases with two types of relationships between those use cases: include and generalization. The include relationship "includes" functionality in one use case that is described by another use case. The "Provide LAN Access" use case includes the functionality of the "Validate MU" use case. The generalization relationship indicates that one use case is a more general form of another use case. The relationship line with a hollow triangle at the end indicates that the "Move Data" use case is a more generalized form of the "Filter Data" use case. Or, equivalently, "Filter Data" is a more specialized form of "Move Data." The constraint artifact (in the lower left corner of the diagram) requires the WLAN system and the MUs to be set to the same SSID.

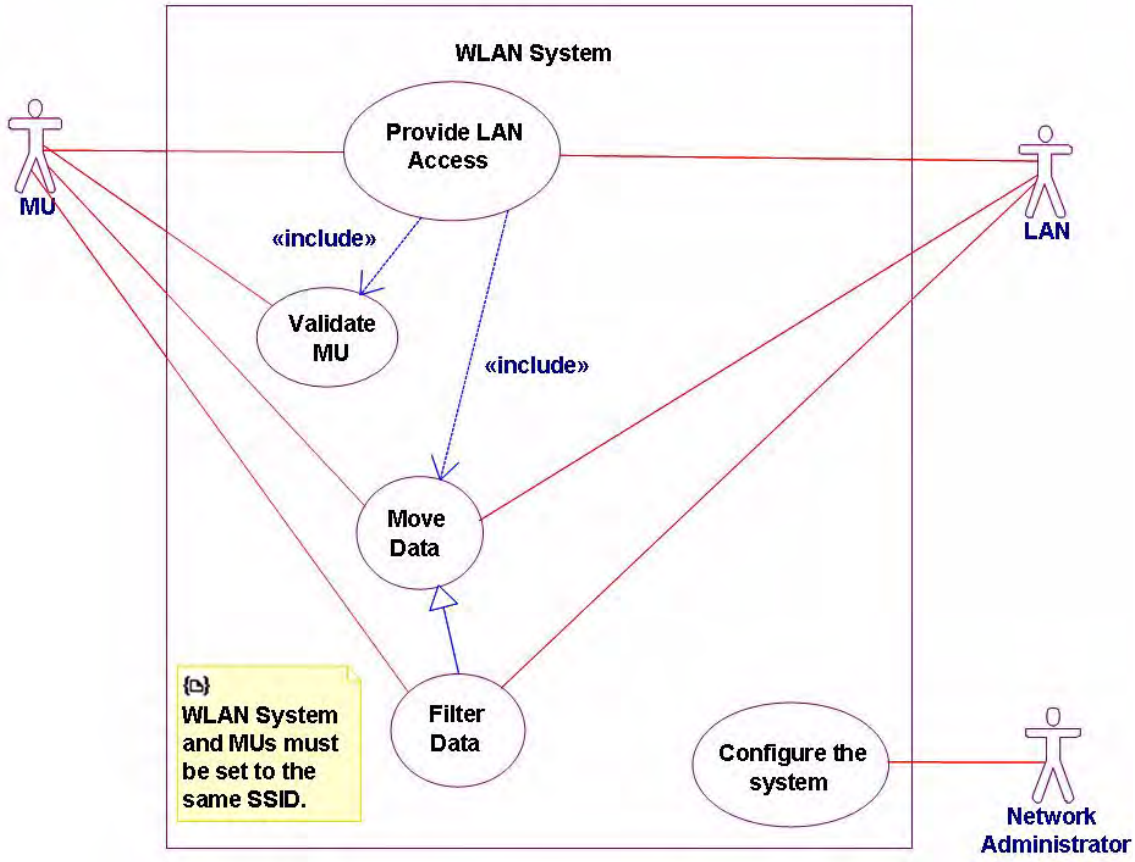



**Figure N.3—High-level UML use case diagram for the WLAN system**

The primary functions of the WLAN system, depicted in the UML object model diagram in Figure N.4, are provided by the ACM_STA, AP, and DS entities, the latter via the DSSs. The portal is merely a conceptual link from the DS to the non-IEEE-802.11 LAN. The UML object model diagram shows a package containing a set of object classes. References to the actors (external entities) are limited to descriptions of their interactions with, and expectations of, the object classes, which is accomplished by describing the attributes and behaviors of the class entities. The object model diagram uses connecting lines to indicate interfaces (called *associations* in UML terminology) between the various artifacts present in the diagram. UML association lines lacking arrows on both ends indicate that there is a bidirectional interface between the artifacts. The UML association lines are annotated with multiplicity counts to indicate the how many entities may fill the position at that end of the association.

Copyright © 2007 IEEE. All rights reserved.

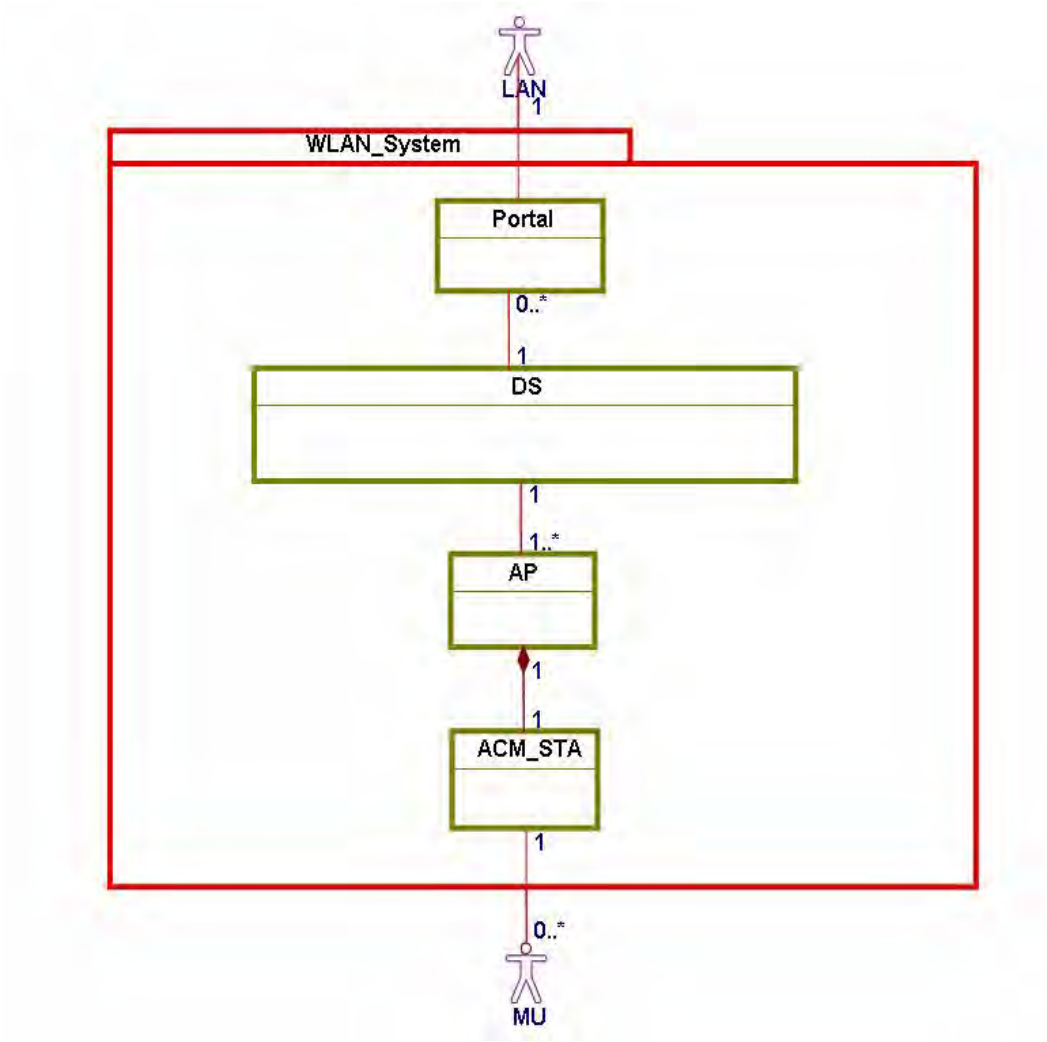



**Figure N.4—High-level UML entity diagram for the WLAN system**

Figure N.4 also shows the relationships between the WLAN system entities. There exists a bidirectional association between zero or more [0..*] MUs and a single (given) ACM_STA. The solid diamond terminated line indicates that there is a composition relationship between the ACM_STA and the AP, i.e., the AP is composed of (or always has an) ACM_STA. Hence there is a one-to-one mapping between APs and ACM_STAs. This composition and one-to-one relationship can also be drawn as shown in Figure N.5. These two forms are equivalent. There are one or more [1..*] APs connected to a single DS. The DS in turn connects to zero or more [0..*] portals. Each portal connects to a single non-IEEE-802.11 LAN.


Copyright © 2007 IEEE. All rights reserved.

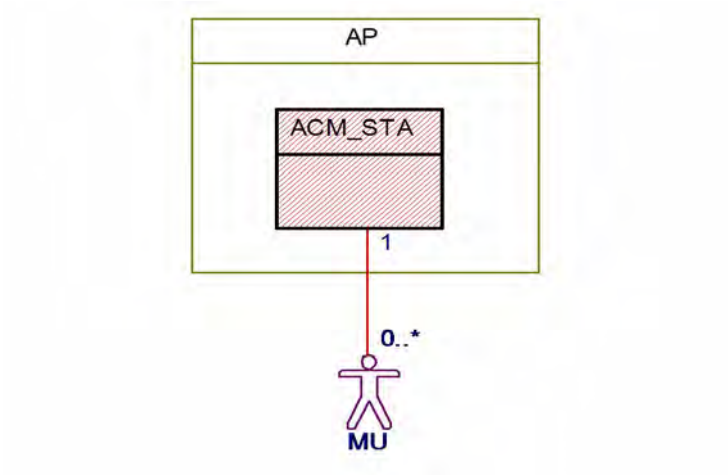



**Figure N.5—AP UML composition diagram (alternate syntax)**

## N.3 Primary ACM_STA functions

The primary functions of an ACM_STA are as follows:

a) Instantiate the infrastructure BSS.

b) Move data between the MUs and the AP.

Instantiating the infrastructure BSS consists of advertising the BSS and defining timing for the entire BSS (i.e., infrastructure mode TSF). Advertising the BSS includes creating an infrastructure mode Beacon frame, transmitting that Beacon frame, and replying to MU probe requests with corresponding probe response transmissions. Beacon frames and probe responses provide a way for the MUs to find, join with the ACM_STA, and (subsequently) associate with the AP. This includes, for example, the AP providing channel, regulatory, and country information to the MUs.

Moving data between the MUs and the AP consists of translating between MSDUs and MPDUs, buffering data, and transmitting/receiving MPDUs via an IEEE 802.11 PHY.

## N.4 Primary AP functions

The primary functions of an AP are as follows:

a) Provide DS access for the MUs.
   1) Includes MU validation and extends (in some cases) to notifying the DS.
   2) Includes moving data between the MUs and the DS.
      i) Uses a special data movement function called *data filtering*.

b) Configure the AP (both the AP itself and the included ACM_STA).

Those high-level use cases of the AP are shown in the UML use case diagram in Figure N.6. The UML extend relationship “extends” a use case with optional functions provided by another use case that are applied only in some scenarios. In Figure N.6, the “Notify the DS” use case extends the “Validate MU” use case in some scenarios.

Copyright © 2007 IEEE. All rights reserved.

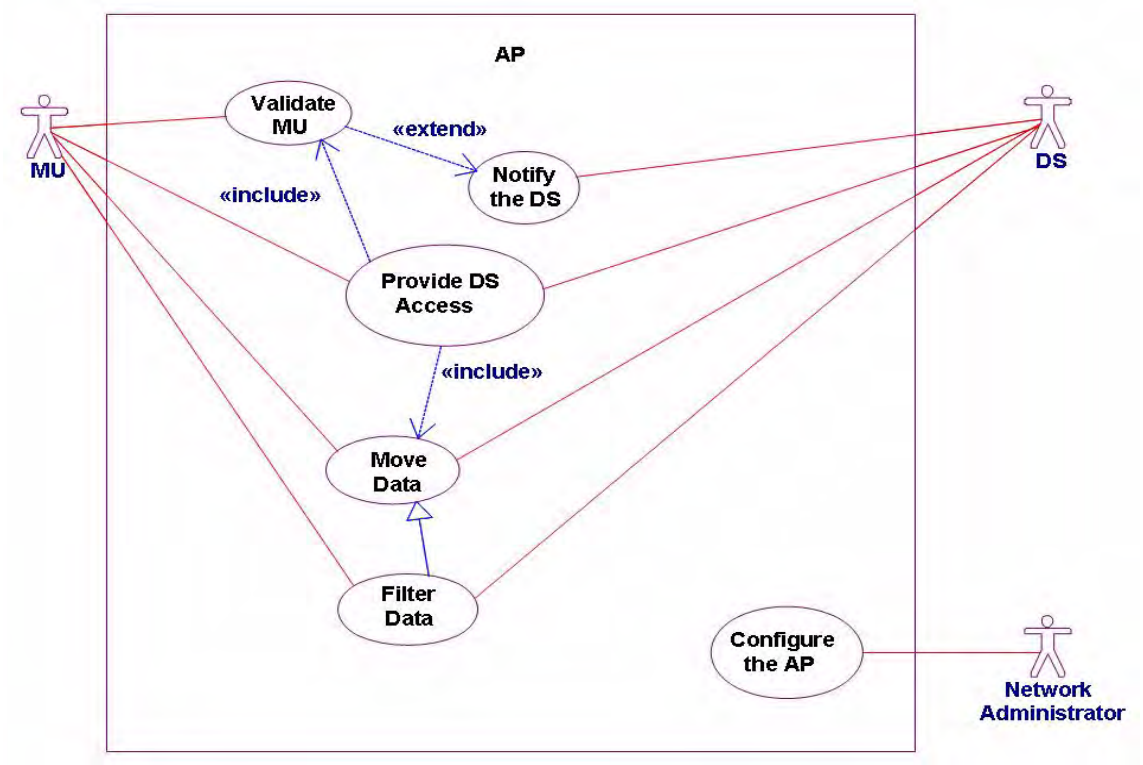



**Figure N.6—High-level UML use case diagram for the AP**

The AP provides DS access for the MUs, including validating the MUs (e.g., via STA and/or client authentication) and providing access and admission control (e.g., via the association process). An AP is literally a point of access to the DS (and, by extension, to the non-IEEE-802.11 LAN beyond). Upon validating an MU, the AP updates the DS mapping of the MU to the AP. DS mapping updates can, for example, be based on association and reassociation requests (received from the ACM_STA) or aging of inactive links (based on session timers). An AP may also receive access control updates directly from other APs, via a protocol outside the scope of this standard, in the form of inter-AP notifications of MU association events and transitions. In this way, MU validation and subsequent changes in MU access control lead to adjustments in how data are allowed to move through the AP (i.e., between the MUs and the DS).

Providing DS access for the MUs also includes moving data between the MUs and the DS, which is accomplished by moving MSDUs between the ACM_STA and the DS (bidirectionally).

The moving data function includes “filtering data.” The filtering data function controls which MSDUs (if any) are moved between the DS and the MUs. For example, filtering of data to and from a particular MU is adjusted based on the various stages of validation of that MU.

The AP also provides a function for configuration. Configuring the AP includes configuring the AP itself as well as the included ACM_STA. For example, configuration may include setting the local values of the SSID, PHY channel, beacon interval, and so on.


Copyright © 2007 IEEE. All rights reserved.

## N.5 Primary DS functions

The primary functions of the DS (i.e., the DSSs) are as follows:

a) Map MU to AP (for DS-to-MU traffic delivery).

b) Move data.

The DS's MU-to-AP map determines which AP is to be used for a given MU's data delivery. This function includes mapping update adjustments, which are based on notification from the APs, of changes in MU access control.

Moving data consists of moving encapsulated MSDUs among the APs (including returning MSDUs to the source AP for MU-to-MU communications) and between the APs and the portal(s).

## N.6 Primary portal function

The primary function of a portal is as follows:

a) Move data with a special data movement function called *data transformation*.

Moving data consists of moving MSDUs between the DS and the external non-IEEE-802.11 LAN. Moving data through the portal transforms the MSDUs using the integration function. The integration function translates external non-IEEE-802.11-LAN MSDUs to and from IEEE 802.11 MSDUs using, for example, the procedures defined in Annex M.

## N.7 AU example

Now that the functions of a WLAN system have been described, here is the example of an AU implementation. Quite simply, an AU is an instantiation of a WLAN system as described in this annex.

Since transiting from a DS through a portal onto an integrated non-IEEE-802.11 LAN and then subsequently via another portal onto its DS is transparent, it is possible to define a DS in terms of not only the portals that are directly connected to a particular DS, but also in terms of all the integrated MAC endpoints. Therefore, a collection of AUs connected to an integrated non-IEEE-802.11 LAN can define a DS that consists of the union of all the DSs inside the AUs. The union of such a set of AUs would itself constitute a WLAN system.

Copyright © 2007 IEEE. All rights reserved.

Copyright © 2007 IEEE. All rights reserved.

# Annex O

(informative)

# DS SAP specification

## O.1 Introduction

The purpose of this informative annex is to describe and clarify the DS SAP. The DS SAP is the interface between the DS SAP service users and the DS SAP service provider. The DS SAP service users are the connected APs and the portals. The DS SAP service provider is the DS. Figure O.1 shows the location of the DS SAP in the IEEE 802.11 architecture.

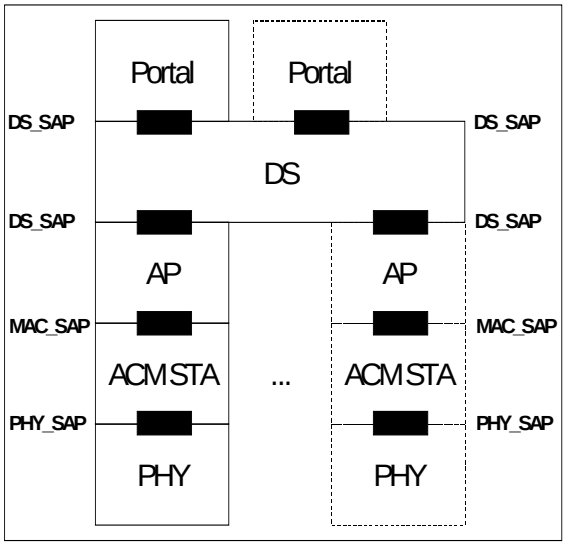



**Figure O.1—Location of the DS SAP**

The DS SAP interface specification describes the primitives required to get MSDUs in and out of the DS and update the DS's mapping of STAs to APs. Describing the DS itself or the functions thereof is out of scope of this annex.

The DS SAP actions are as follows:

a) Accept MSDUs from APs and portals.

b) Deliver MSDUs to either APs or portals, or both.

c) Accept STA-to-AP mapping updates from the APs.

When the DS delivers the MSDUs to an AP, the AP then determines when and how to deliver the MSDUs to the AP's MAC (via the MAC SAP).

Copyright © 2007 IEEE. All rights reserved.

## O.2 SAP primitives

The DS SAP service interface primitives are as follows:

a) DS-UNITDATA.request
b) DS-UNITDATA.indication
c) DS-STA-NOTIFY.request

### O.2.1 MSDU transfer

The DS-UNITDATA primitives accept and deliver IEEE 802.11 MSDUs, including all the parameters and data as defined in 6.2.1.1.2. These tuples are called *distribution system service data units* (DSSDUs).

#### O.2.1.1 DS-UNITDATA.request

##### O.2.1.1.1 Function

This primitive requests distribution of a DSSDU across the DS.

##### O.2.1.1.2 Semantics of the service primitive

The primitive parameters are as follows:

DS-UNITDATA.request(
Dssdu,
SourceType
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Dssdu | IEEE 802.11 MSDU | Any valid data unit according to 6.2.1.1.2, including data and all parameters | Specifies the DSSDU to be distributed via the DS. |
| SourceType | Enumeration | SRC_AP, SRC_PORTAL | Specifies the type of entity that is requesting distribution of a DSSDU. |

##### O.2.1.1.3 When generated

This primitive is generated by an AP or a portal to submit a DSSDU to the DS for distribution.

##### O.2.1.1.4 Effect of receipt

This primitive initiates distribution of the DSSDU through the DS. An individually addressed DSSDU from an AP or a portal is distributed through the DS to the corresponding AP or portal. A broadcast or multicast DSSDU from an AP is distributed to all APs and all portals, including the originating AP. A broadcast or multicast DSSDU from a portal is distributed to all APs and all portals, except the originating portal.


Copyright © 2007 IEEE. All rights reserved.

### O.2.1.2 DS-UNITDATA.indication

#### O.2.1.2.1 Function

This primitive indicates delivery of a DSSDU from the DS to either an AP or a portal.

#### O.2.1.2.2 Semantics of the service primitive

The primitive parameters are as follows:

DS-UNITDATA.indication(
Dssdu,
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| Dssdu | IEEE 802.11 MSDU | Any valid data unit according to 6.2.1.1.2, including data and all parameters | Specifies the DSSDU that is being delivered by the DS. |

#### O.2.1.2.3 When generated

This primitive is generated by the DS to deliver a DSSDU to an AP or a portal.

#### O.2.1.2.4 Effect of receipt

This primitive delivers a DSSDU to an AP or a portal.

Copyright © 2007 IEEE. All rights reserved.

## O.2.2 Mapping updates

The DS-STA-NOTIFY primitive is used to maintain the STA-to-AP mapping data of the DS.

### O.2.2.1 DS-STA-NOTIFY.request

#### O.2.2.1.1 Function

This primitive requests an update to the DS's STA-to-AP map.

#### O.2.2.1.2 Semantics of the service primitive

The primitive parameters are as follows:

DS-STA-NOTIFY.request(
STAAddress,
UpdateType
)

| Name | Type | Valid range | Description |
|---|---|---|---|
| STAAddress | MACAddress | Any valid individual MAC address | Specifies the address of the STA whose association status with the AP has changed. |
| UpdateType | Enumeration | ADD, MOVE, DELETE | Specifies the DS mapping update operation to be performed. |

#### O.2.2.1.3 When generated

This primitive is generated by an AP to update the DS's STA-to-AP map.

#### O.2.2.1.4 Effect of receipt

This primitive updates the DS's STA-to-AP map, which controls to which AP the DS delivers DSSDUs that are destined for a given STA.

There are many mechanisms to implement this mapping update for the cases of ADD and MOVE. One example mechanism, in the case where the DS is an IEEE 802 LAN, is to use an IEEE 802.2 XID null frame.


Copyright © 2007 IEEE. All rights reserved.

# Annex P

# (informative)

# Bibliography

## P.1 General

[B1] ANSI Z136.1-1993, American National Standard for the Safe Use of Lasers.[33]

[B2] Arazi, E. G., *A Commonsense Approach to the Theory of Error Correcting Codes*, MIT Press, 1988.

[B3] ARIB RCR STD-33 (5.0), Low Power Data Communication/Wireless LAN System, ARIB, February 1999.[34]

[B4] Engwer, D., and Zweig, J., "Algorithmically Derived Hop Sequences," submission 99/195 to the IEEE P802.11 Working Group, Sept. 1999.

[B5] "Ethernet, a local area network: Data link layer and physical layer specifications version 1.0." Digital Equipment Corporation, Intel Corporation, Xerox Corporation, September 1980.

[B6] ETS 300-328/A1 ed.2 (1997-07), Radio Equipment and Systems (RES); Wideband transmission systems; Technical characteristics and test conditions for data transmission equipment operating in the 2,4 GHz ISM band and using spread spectrum modulation techniques.[35]

[B7] ETS 300-339 V1.1.1 (1998-06), Electromagnetic compatibility and Radio spectrum Matters (ERM); General ElectroMagnetic Compatibility (EMC) for radio communications equipment.

[B8] FCC CFR47, Title 47 of the Code of Federal Regulations, Federal Communication Commission, October 2001.[36]

[B9] IC GL-36, Technical Requirements for Low Power Devices in the 2400–2483.5 MHz Band, Industry Canada (IC), August 1999.[37]

[B10] IEC 60825-1:1993, Safety of laser products—Part 1: Equipment classification, requirements and user's guide.[38]

[33]ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[34]ARIB publications are available from the Association of Radio Industries and Businesses (www.arib.or.jp)

[35]ETSI publications are available from the European Telecommunications Standards Institute (www.etsi.org).

[36]FCC publications are available from the Government Printing Office (GPO, www.gpo.gov). Updates are published in the FCC Register (also published by the GPO) and are available online (www.fcc.gov).

[37]IC publications are available from Industry Canada (www.strategic.ic.gc.ca).

[38]IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3, rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iec.ch/). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA.

Copyright © 2007 IEEE. All rights reserved.

[B11] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.[39]

[B12] IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networds—Media Access Control (MAC) Bridges.[40]

[B13] IEEE Std 802.1H™-1995, IEEE Standards for Local and Metropolitan Area Networks: Recommended Practice for Media Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks [now known as ISO/IEC Technical Report 11802-5:1997(E); see Clause 2].

[B14] IEEE Std 802.1Q™, 2003 Edition, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.

[B15] IEEE Std 802.10™-1998, IEEE Standards for Local and Metropolitan Area Networks: Interoperable LAN/MAN Security (SILS).

[B16] IETF RFC 1305-1992, Network Time Protocol (Version 3) Specification, Implementation and Analysis.

[B17] IETF RFC 2202-1997, Test Cases for HMAC-MD5 and HMAC-SHA-1, P. Cheng, R. Glenn, September 1997 (status: informational).

[B18] IETF RFC 2205-1997, Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification.[41]

[B19] IETF RFC 2212-1997, Specification of the Guaranteed Quality of Service.

[B20] IETF RFC 2215-1997, General Characterization Parameters for Integrated Service Network Elements.

[B21] IETF RFC 2474-1998, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.

[B22] IETF RFC 2548-1999, Microsoft Vendor-specific RADIUS Attributes.

[B23] IETF RFC 2865-2000, Remote Authentication Dial in User Service (RADIUS).

[B24] IETF RFC 3290-2002, An Informal Management Model for Diffserv Routers.

[B25] IETF RFC 3588-2003, Diameter Base Protocol.

[B26] IETF RFC 3748-2004, Extensible Authentication Protocol (EAP).

[B27] IETF RFC 4086-2005, Randomness Recommendations for Security, D. Eastlake, 3rd, J. Schiller, S. Crocker.

[39]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[40]The IEEE standards or products referred to inthis annex are trademarks owned by the Institute of Electrical and Electronics Engineers, Inc.

[41]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).


Copyright © 2007 IEEE. All rights reserved.

[B28] ITU-T Recommendation X.210 (11/93), Information technology—Open systems interconnection—Basic Reference Model: Conventions for the definition of OSI services (common text with ISO/IEC 10731).[42]

[B29] Maric, S. V., and Titlebaum, E. L., "A Class of Frequency Hop Codes with Nearly Ideal Characteristics for Use in Multiple-Access Spread-Spectrum Communications and Radar and Sonar Systems," *IEEE Transactions on Communications*, vol. 40, no. 9, pp. 1442-1447, Sept. 1992.

[B30] PKCS #5 v2.0, "Password-Based Cryptography Standard," http://www.rsasecurity.com/rsalabs/node.asp?id=2127.

[B31] Rumbaugh, J., Jacobson, I., & Booch, G., *The Unified Modeling Language (UML) Reference Manual (Second Edition)*, Reading, MA: Addison-Wesley Longman, 2004.

[B32] Schneier, Bruce, *Applied Cryptography, Protocols, Algorithms and Source Code in C.* New York: Wiley, 1994.

## P.2 Specification and description language (SDL) documentation

[B33] Belina, F., Hogrefe, D., and Sarma, A., *SDL with Applications from Protocol Specification.* Hertfordshire, UK: Prentice Hall Europe, 1991.[43]

[B34] Ellsberger, J., Hogrefe, D., and Sarma, A., *SDL, Formal Object-Oriented Language for Communicating Systems*. Hertfordshire, UK: Prentice Hall Europe, 1997.[44]

[B35] Faergemand, O., and Olsen, A., "New Features in SDL-92," *SDL Newsletter* (ISSN 1023-7151), no. 16, May 1993, pp. 10–29. Also available online at http://www.tdr.dk/public/SDL/SDL.html.[45]

[B36] Olsen, A., Faergemand, O., Moller-Pedersen, B., Reed, R., and Smith, T. R. W., *Systems Engineering Using SDL-92.* Amsterdam, the Netherlands: Elsevier Science B.V., 1994.[46]

[42]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

[43]An introductory text on SDL, also useful as a language reference (for SDL-88).

[44]A recently published book, which appears to be the most comprehensive single-volume introduction and reference for SDL-92, including its object-oriented extensions.

[45]This provides a summary of the changes from SDL-88 to SDL-92.

[46]A detailed guide to using SDL-92, including a thorough explanation of abstract data type mechanism and SDL combined with ASN.1 (ITU-T Recommendation Z.105).

Copyright © 2007 IEEE. All rights reserved.

Samsung Ex. 1044, Page 1231 of 1232

Copyright © 2007 IEEE. All rights reserved.
**Samsung Ex. 1044, Page 1232 of 1232**