IEEE Standards

802.15.3™

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

Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

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

29 September 2003

Print: SH95136
PDF: SS95136

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

Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 12 June 2003

IEEE-SA Standards Board

Abstract: The protocol and compatible interconnection of data and multimedia communication equipment via 2.4 GHz radio transmissions in a Wireless Personal Area Network (WPAN) using low power and multiple modulation formats to support scalable data rates is defined in this standard. The Medium Access Control (MAC) sublayer protocol supports both isochronous and asynchronous data types.

Keywords: ad hoc network, mobility, PAN, Personal Area Network, radio frequency, Wireless, WPAN

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

Print: ISBN 0-7381-3704-9 SH95136
PDF: ISBN 0-7381-3705-7 SS95136

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.

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
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. 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 applicants 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 Standards Department.

Introduction

This introduction is not part of IEEE Std 802.15.3-2003, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs).

IEEE Std 802.15.3-2003 was designed to enable wireless connectivity of high-speed, low-power, low-cost, multimedia-capable portable consumer electronic devices. This standard provides data rates from 11 to 55 Mb/s at distances of greater than 70 m while maintaining quality of service (QoS) for the data streams. In addition, this standard is designed to provide simple, ad-hoc connectivity that allows the devices to automatically form networks and exchange information without the direct intervention of the user. Privacy and integrity are provided for data and commands with 128-bit AES encryption used in CCM mode. This standard has also provided a variety of techniques that can be used to enhance the coexistence of 802.15.3 piconets with other wireless networks.

The idea of a high-rate addition to the IEEE 802.15 family of standards was first proposed in November 1999 at the IEEE Plenary meeting in Kaua'i, HI. The 802.15.3 task group began its official work at the March 2000 IEEE Plenary meeting in Albuquerque, NM, creating a criteria document and evaluation method. The down-selection of MAC and PHY proposals was completed at the November 2000 IEEE 802 Plenary meeting in Tampa, FL, and the writing of the draft began in December 2000. After working on the draft for one year, the document was ready for the task group ballot process in December 2001. The draft received final working group approval at the November 2002 IEEE Plenary meeting in Kaua'i, HI, and began the sponsor ballot process following the meeting. The draft went through one sponsor ballot and two recirculations before it was submitted to the IEEE Standards Association Standards Board (IEEE-SASB) for approval. The IEEE-SASB approved 802.15.3 as an IEEE standard in June 2003.

Interpretations and errata

Interpretations and errata associated with this standard may be found at one of the following Internet locations:

— http://standards.ieee.org/reading/ieee/interp/

— http://standards.ieee.org/reading/ieee/updates/errata

Conformance test methodology

An additional standards series, identified by the number 1802™, has been established to identify the conformance test methodology documents for the IEEE 802® family of standards. Thus the conformance test documents for IEEE 802.3™ are numbered 1802.3™, the conformance test documents for IEEE 802.5™ will be 1802.5™, and so on. Similarly, ISO will use 18802 to number conformance test standards for 8802 standards.

Participants

At the time this standard was completed, the 802.15 working group had the following membership:

Working group 802.15
Robert F. Heile, *Chair*
James D. Allen, *Vice-Chair*
Patrick Kinney, *Secretary*

John R. Barr, *802.15.3 Chair*
James D. Allen, *802.15.3 Vice Chair*
James P. K. Gilb, *802.15.3 Technical Editor, PHY Committee Chair*

Allen Heberling, *802.15.3 MAC Committee Chair, MAC Assistant Editor*
Richard Roberts, *802.15.3 Systems Committee Chair, Layer Management Assistant Editor*

Jay Bain, *802.15.3 MAC Assistant Editor*
Jeyhan Karaoguz, *802.15.3 PHY Assistant Editor*
John Sarallo, *802.15.3 Layer Management Assistant Editor, MAC Contributing Editor*
Ari Singer, *802.15.3 Security Assistant Editor*

Dan Bailey, *802.15.3 Security Contributing Editor*
Rajugopal Gubbi, *802.15.3 MAC Contributing Editor*
Knut Odman, *802.15.3 MAC Contributing Editor*
Mark Schrader, *802.15.3 MAC Contributing Editor*
Bill Shvodian, *802.15.3 MAC Contributing Editor*

Roberto Aiello
Masaaki Akahane
Richard Alfvin
Arun Arunachalam
Naiel Askar
Venkat Bahl
Anuj Batra
Timothy J. Blaney
Stan Bottoms
Monique Bourgeois
Chuck Brabenac
Ed Callaway
Soo-Young Chang
Hung Kun Chen
Aik Chindapol
David E. Cypher
Michael Derby
Mary DuVal
Michael Dydyk
Jason Ellis
Jeff Foerster
Pierre Gandolfo
Nada Golmie
Paul Gorday
Jose Gutierrez
Yasuo Harada
Barry Herold
Bob Huang
Laura L. Huckabee
Eran Igler
Katsumi Ishii
Phil Jamieson
Park Jong-Hun
Joy H. Kelly
Stuart J. Kerry
Yongsuk Kim
Gunter Kleindl
Bruce P. Kraemer
Jim Lansford
David G. Leeper
Liang Li
Jie Liang
Shawn T. Liu
Yeong-Chang Maa
Ralph Mason
Michael D. McInnis
Jim Meyer
Leonard Miller
Akira Miura
Tony Morelli
Said Moridi
Marco Naeve
Chiu Y. Ngo
Erwin R. Noble
Jack Pardee
Marcus Pendergrass
Robert D. Poor
Gregg Rasor
Ivan Reede
Jim Richards
William Roberts
Chris Rogers
Philippe Rouzet
Chandos Rypinski
John Santhoff
Tom Schuster
Erik Schylander
Michael Seals
Stephen J. Shellhammer
Nick Shepherd
Gadi Shor
Thomas Siep
Kazimierz Siwiak
Carl Stevenson
Rene Struik
Shigeru Sugaya
Kazuhisa Takamura
Katsumi Takaoka
Teik-Kheong Tan
Larry Taylor
Wim van Houtum
Hans van Leeuwen
Ritesh Vishwakarma
Thierry Walrant
Fujio Watanabe
Matthew Welborn
Richard Wilson
Stephen Wood
Edward G. Woodrow
Hirohisa Yamaguchi
Song-Lin Young

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

Roberto Aiello
Masaaki Akahane
Richard Alfvin
James Allen
Arun Arunachalam
Naiel Askar
Venkat Bahl
Daniel Bailey
Jay Bain
James Baker
Jaiganesh Balakrishnan
John Barr
Anuj Batra
Timothy Blaney
Kenneth Boehike
Stan Bottoms
Monique Bourgeois
Mark V. Bowles
Chuck Brabenac
Ed Callaway
Soo-Young Chang
Francois Po-Shin Chin
Aik Chindapol
Craig Conkling
David Cypher
Anand Dabak
Kai Dombrowski
Mary DuVal
Michael Dydyk
Jason L. Ellis
Mark W. Fidler
Jeff R. Foerster
David S. Furuno
Pierre Gandolfo
Atul Garg
Ian C. Gifford
James P. K. Gilb
Nada Golmie
Paul Gorday
Jose Gutierrez
Yasuo Harada
Allen Heberling
Robert F. Heile
Barry Herold
Robert Y. Huang
Eran Igler
Katsumi Ishii
Phil Jamieson
Jeyhan Karaoguz
Masami Katagiri
Joy H. Kelly
Stuart J. Kerry
Yongsuk Kim
Young Hwan Kim
Patrick Kinney
Günter Kleindl
Bruce P. Kraemer
DoHoon Kwon
Jim Lansford
David Leeper
Liang Li
Yeong-Chang Maa
Steven March
Ralph Mason
Michael D. McInnis
Jim Meyer
Leonard E. Miller
Akira Miura
Andreas Molisch
Antonio Mondragon
Tony Morelli
Said Moridi
Marco Naeve
Yves-paul Nakache
Chiu Ngo
Kei Obara
Knut Odman
Yuen Oren
John B. Pardee
Jonghun Park
Dave Patton
Marcus Pendergrass
Robert D. Poor
Gregg Rasor
Ivan Reede
Jim Richards
William Roberts
Glyn Roberts
Richard Roberts
Chris Rogers
Philippe Rouzet
Chandos Rypinski
John H. Santhoff
Mark Schrader
Tom Schuster
Erik Schylander
Michael Seals
Stephen J. Shellhammer
Nick Shepherd
Gadi Shor
Bill Shvodian
Thomas Siep
Kazimierz Siwiak
Carl Stevenson
Rene Struik
Shigeru Sugaya
Kazuhisa Takamura
Katsumi Takaoka
Teik-Kheong Tan
Larry Taylor
Stephen E. Turner
Hans van Leeuwen
Ritesh Vishwakarma
Thierry Walrant
Jing Wang
Fujio Watanabe
Mathew Welborn
Richard Wilson
Stephen Wood
Edward G. Woodrow
Hirohisa Yamaguchi
Amos Young
Song-Lin Young
Jim Zyren

When the IEEE-SA Standards Board approved this recommended practice on 12 June 2003, it had the following membership:

Don Wright, *Chair*
Howard M. Frazier, *Vice Chair*
Judith Gorman, *Secretary*

H. Stephen Berger
Joe Bruder
Bob Davis
Richard DeBlasio
Julian Forster*
Toshio Fukuda
Arnold M. Greenspan
Raymond Hapeman
Donald M. Heirman
Laura Hitchcock
Richard H. Hulett
Anant Jain
Lowell G. Johnson
Joseph L. Koepfinger*
Tom McGean
Steve Mills
Daleep C. Mohla
William J. Moylan
Paul Nikolich
Gary Robinson
Malcolm V. Thaden
Geoffrey O. Thompson
Doug Topping
Howard L. Wolfman

*Member Emeritus

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

Alan Cookson, *NIST Representative*
Satish K. Aggarwal, *NRC Representative*

Andy Ickowicz
IEEE Standards Project Editor

Contents

1. Overview 1

1.1 Scope 1
1.2 Purpose 2

2. References 2

3. Definitions 3

4. Acronyms and abbreviations 5

5. General description 8

5.1 What is a piconet? 8
5.2 Components of an 802.15.3 piconet 8
5.3 Overview of MAC functionality 8
5.4 Characteristics of the 2.4 GHz PHY 15

6. Layer management 16

6.1 Overview of management model 16
6.2 Generic management primitives 18
6.3 MLME SAP interface 21
6.4 PLME SAP interface 81
6.5 MAC management 84
6.6 MAC SAP 87
6.7 Physical layer (PHY) service specification 91

7. MAC frame formats 100

7.1 Frame format conventions 101
7.2 General frame format 102
7.3 Format of individual frame types 108
7.4 Information elements 116
7.5 MAC command types 126

8. MAC functional description 149

8.1 Introduction 149
8.2 Starting, maintaining and stopping piconets 150
8.3 Association and disassociation with a piconet 164
8.4 Channel access 169
8.5 Channel time management 179
8.6 Synchronization 191
8.7 Fragmentation and defragmentation 193
8.8 Acknowledgement and retransmission 194
8.9 Peer discovery 196
8.10 Changing piconet parameters 202
8.11 Interference mitigation 204
8.12 Multi-rate support 207
8.13 Power management 208

8.14 ASIE operation 219
8.15 MAC sublayer parameters 220

9. Security 221

9.1 Security mechanisms 221
9.2 Security modes 222
9.3 Security support 222
9.4 Protocol details 229

10. Security specifications 233

10.1 Modes for security 233
10.2 Symmetric cryptography building blocks 233
10.3 Symmetric cryptography implementation 234
10.4 CCM mode 236

11. PHY specification for the 2.4 GHz band 242

11.1 Overview of the 2.4 GHz PHY 242
11.2 General requirements 243
11.3 Modulation and coding 248
11.4 PHY frame format 258
11.5 Transmitter specifications 267
11.6 Receiver specifications 271
11.7 PHY management 273

Annex A (normative) Frame convergence sublayer 277

Annex B (informative) Security considerations 285

Annex C (informative) Coexistence, interoperability, and interference 289

Annex D (normative) Protocol implementation conformance statement (PICS) proforma 303

Annex E (informative) Bibliography 314

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

Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs)

1. Overview

Wireless personal area networks (WPANs) are used to convey information over relatively short distances among a relatively few participants. Unlike wireless local area networks (WLANs), connections effected via WPANs involve little or no infrastructure. This allows small, power efficient, inexpensive solutions to be implemented for a wide range of devices.

The term WPAN in this document refers specifically to a wireless personal area network as defined by this document. The terms “wireless personal area network,” “WPAN,” and “802.15.3 WPAN” in this document are synonymous.

1.1 Scope

This standard defines the PHY and MAC specifications for high data rate wireless connectivity with fixed, portable and moving devices within or entering a personal operating space. A goal of this standard will be to achieve a level of interoperability or coexistence with other 802.15™ standards. It is also the intent of this standard to work toward a level of coexistence with other wireless devices in conjunction with coexistence task groups such as 802.15.2™.

Based on the previous calls for applications collected for 802.15, there remained a significant group of applications that could not be addressed by 802.15.1™. High data rates are required for time dependent and large file transfer applications such as video or digital still imaging without sacrificing the requirements of low complexity, low cost and low power consumption. 20 Mb/s is proposed to be the lowest rate for these types of data.

It is possible, for example, that several data rates would be supported for different consumer applications. Consequently, the notions of cost, frequency band, performance, power and data rate scalability were addressed in the development of this standard.

A personal operating space is a space about a person or object that typically extends up to 10 m in all directions and envelops the person whether stationary or in motion. Personal operating space use models permit more freedom over the design of the radio than in medical or enterprise LAN applications where the primary

goal is link robustness at long range. In an area covered by a WLAN, it is expected that a robust link would be established anywhere within the coverage area without any special action on the part of the user. Link robustness is equally important for a WPAN but it is acceptable to take an action like moving closer to establish it. Consequently, WPAN standards are able to focus on other priorities, such as cost, size, power consumption and data rate.

It is not the intent of this standard to be an extension of 802.15.1, because the MAC needs are different. It is, however, in the best interest of users and the industry to strive for compatibility, or at least coexistence with other wireless systems, especially those in similar market spaces such as 802.15.1. Compatibility and coexistence criteria were included in the proposal evaluations.

1.2 Purpose

The purpose of this standard is to provide for low complexity, low cost, low power consumption (comparable to the goals of 802.15.1) and high data rate wireless connectivity among devices within or entering the personal operating space. The data rate is high enough, 20 Mb/s or more, to satisfy a set of consumer multimedia industry needs for WPAN communications. This standard also addresses the quality of service capabilities required to support multimedia data types.

2. References

This standard shall be used in conjunction with the following publications. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

ANSI X3.66-1979: Advanced data communication control procedures (ADCCP).[1]

IEEE Std 802®-2001, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[2, 3]

ISO/IEC 646:1991, Information Technology—ISO 7-bit coded character set for information interchange.[4]

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

ISO/IEC 8802-2:1994, Information technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 2: Logical Link Control.

ISO/IEC 10039:1991, Information Technology—Open Systems Interconnection—Local Area Networks—Medium Access Control (MAC) Service Definition.

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.

[1]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org).

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

[3]The IEEE standards referred to in Clause 2 are trademarks belonging to the Institute of Electrical and Electronics Engineers, Inc.

[4]ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO/IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

NIST FIPS Pub 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T., November 26, 2001.[5]

3. Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of Standards Terms,* Seventh Edition [B2][6], should be referenced for terms not defined in this clause.

3.1 ad hoc network: A network typically created in a spontaneous manner. The principal characteristic of an ad hoc network is its limited temporal and spatial extent.

3.2 alternate coordinator: A member of the piconet that is capable of being the coordinator but is not currently functioning as the coordinator.

3.3 association: The service used to assign a device identifier to a device to enable communications in a piconet.

3.4 channel time allocation: A contiguous period of time in the superframe allocated by the piconet coordinator for communication between specified source and destination.

3.5 child piconet: A piconet that exists entirely within a channel time allocation of another piconet, the parent piconet, and is controlled by a device that is a member of the parent piconet. The area of overlapping coverage between the two piconets may vary between congruency with the parent coverage area to mostly non-overlapping.

3.6 coexistence: The ability of one system to perform a task in a given shared environment where other systems have an ability to perform their tasks and may or may not be using the same set of rules.

3.7 coverage area: The area where two 802.15.3™ devices are able to exchange messages with acceptable quality and performance.

3.8 data authentication: Authentication of the sender of the data and provision of data integrity.

3.9 data integrity: The assurance that the data has not been modified from its original form.

3.10 dependent picoent: A piconet that exists entirely within a channel time allocation of another piconet, the parent piconet. Child and neighbor piconets are both types of dependent piconets.

3.11 device: An entity that implements an IEEE Std. 802.15.3™-2003 conformant media access control and physical layer interface to the wireless medium.

3.12 device address: The 64-bit IEEE 802® address of a device in an 802.15.3™ piconet.

3.13 device-host: The equipment that incorporates an 802.15.3™ device. The device-host may have more than one device incorporated in it as well as other networking connections, both wired and wireless.

3.14 disassociation: The service which removes an existing association.

3.15 extended beacon: A beacon followed by one or more broadcasted Announce commands from the piconet controller.

[5]NIST FIPS publications are available from the National Institute for Standards and Technology, 100 Bureau Drive, Stop 8900, Gaithersburg, MD 20899-8900 (http://www.nist.gov).

[6]The numbers in brackets correspond to those of the bibliography in Annex E.

3.16 frame: Format of aggregated bits that are transmitted together in time.

3.17 interoperable: The ability of two systems to perform a given task using a single set of rules.

3.18 integrity code: A data string generated using a symmetric key that is typically appended to data in order to provide data integrity and source authentication similar to a digital signature.

3.19 medium access control command protocol data unit: The unit of data exchanged between two peer medium access control entities using the services of the physical layer to implement the medium access control management protocol.

3.20 medium access control protocol data unit: The unit of data exchanged between two peer medium access control entities using the services of the physical layer.

3.21 medium access control service data unit: Information that is delivered as a unit between medium access control service access points.

3.22 mobile device: A device that uses network communications while in motion.

3.23 neighbor piconet: A piconet that exist entirely within a channel time allocation of another piconet, the parent piconet, and is controlled by a device that is not a member of the parent piconet. The area of overlapping coverage between the two piconets may vary between congruency with the parent coverage area to mostly non-overlapping.

3.24 parent piconet: A piconet which allocates guaranteed time slots for another piconet (child or neighbor types) operating in the same channel.

3.25 payload protection: The generic term for providing security services on the contents of a data message, including confidentiality, integrity and authentication.

3.26 piconet: A collection of one or more logically associated devices that share a single identifier with a common coordinator.

3.27 piconet coordinator: An entity that has device functionality and also provides coordination and other services, e.g. quality of service, synchronization, association, via the wireless medium for associated devices.

3.28 pseudo-random number generation: The process of generating a deterministic sequence of bits from a given seed that has the statistical properties of a random sequence of bits when the seed is not known.

3.29 quality of service: A collective measure of the level of service delivered between devices. Quality of service is characterized by several basic performance criteria, including availability (low downtime), error performance, response time and throughput, lost calls or transmissions, connection set-up time, and speed of fault detection and correction.

3.30 random number generator: A device that provides a sequence of bits that is unpredictable.

3.31 secure frame: A command or data frame in which cryptographic techniques are applied to provide encryption or integrity.

3.32 secure piconet: A piconet in which cryptographic techniques are implemented to provide security services.

3.33 seed: Data that is used as input to an algorithm to produce additional data.

3.34 stream: A unidirectional, logical data connection between two devices that may or may not have quality of service requirements associated with it.

3.35 sub-rate allocation: A channel time allocation that occurs only once every *n* superframes (n > 1).

3.36 super-rate allocation: A channel time allocation that occurs at least once in every superframe.

3.37 superframe: The basic time division of an 802.15.3™ piconet containing a beacon, the channel time allocation period and optionally the contention access period.

3.38 symmetric key: A secret key shared between two or more parties that may be used for encryption/decryption or integrity protection/integrity verification.

3.39 time token: A sequence number that is transmitted in the beacon to indicate the current relative time of the piconet.

3.40 wake beacon: The beacon to which the synchronous power save device will listen. For other beacons, the synchronous power save mode device is presumed to be unavailable for communications.

3.41 wake superframe: A superframe when the synchronous power save device will listen to the beacon and based on beacon information also be available for sending or receiving operations.

3.42 wireless medium: The medium used to implement the transfer of protocol data units between peer physical layer entities of a wireless personal area network.

4. Acronyms and abbreviations

ACK	acknowledgment
ACTIVE	active mode
AES	advanced encryption standard
ASIE	application specific information element
AssocID	association identifier
ATP	association timeout period
AWAKE	awake state
AWGN	additive white Gaussian noise
BcstID	broadcast identifier
BER	bit error rate
BIFS	backoff interframe space
BSID	beacon source identifier
CAP	contention access period
CBC	cipher block chaining
CBC-MAC	cipher block chaining-message authentication code
CCA	clear channel assessment
CCM	counter mode encryption and cipher block chaining message authentication code
CPS	common part sublayer
CRC	cyclic redundancy check
CSMA/CA	carrier sense multiple access with collision avoidance
CTA	channel time allocation
CTAP	channel time allocation period

CTRq	channel time request
CTRqB	channel time request block
CWB	continued wake beacon
DCS	dynamic channel selection
DestID	destination identifier
DEV	device
DEV-host	device-host
DEVID	device identifier
Dly-ACK	delayed acknowledgment
DME	device management entity
DQPSK	differential quadrature phase-shift keying
DSPS	device synchronized power save
FCS	frame check sequence
FCSL	frame convergence sublayer
FEC	forward error correction
FER	frame error rate
HCS	header check sequence
ID	identifier
IE	information element
IFS	interframe space
Imm-ACK	immediate acknowledgment
IP	internet protocol
ISM	industrial scientific medical
KO	key originator
LAN	local area network
LFSR	linear feedback shift register
LLC	logical link control
LME	layer management entity
LQI	link quality indication
lsb	least significant bit
MAC	medium access control
MAN	metropolitan area network
MCDU	MAC command data unit
McstID	multicast identifier
MCTA	management channel time allocation
MIFS	minimum interframe space
MIC	message integrity code
MLME	MAC layer management entity
MPDU	MAC protocol data unit
msb	most significant bit
MSC	message sequence chart
MSDU	MAC service data unit
NbrID	neighbor identifier
OID	object identifier
OrigID	originator identifier
OUI	organizationally unique identifier

PAN	personal area network
PCTM	pending channel time map
PDU	protocol data unit
PHY	physical layer
PIB	PAN information base
PLME	PHY layer management entity
PN	pseudo noise
PNC	piconet coordinator
PNCID	piconet coordinator identifier
PNID	piconet ID
PPDU	PHY protocol data unit
ppm	parts per million
PRNG	pseudo-random number generator
PS	power save
PSPS	piconet synchronized power save
PSRC	power source
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quadrature phase-shift keying
RAC	registration authority committee
RF	radio frequency
RIFS	retransmission interframe space
RNG	random number generator
RSSI	received signal strength indication
RX	receive or receiver
SAP	service access point
SDU	service data unit
SEC	security
SECID	security identifier
SFC	secure frame counter
SIFS	short interframe space
SNR	signal to noise ratio
SPS	synchronous power save
SrcID	source identifier
TCM	trellis coded modulation
TrgtID	target identifier
TPC	transmit power control
TU	time unit
TX	transmit or transmitter
WAN	wide area network
WM	wireless medium
WLAN	wireless local area network
WPAN	wireless personal area network

5. General description

5.1 What is a piconet?

A piconet is a wireless ad hoc data communications system which allows a number of independent data devices (DEVs) to communicate with each other. A piconet is distinguished from other types of data networks in that communications are normally confined to a small area around person or object that typically covers at least 10 m in all directions and envelops the person or a thing whether stationary or in motion.

This is in contrast to local area network (LAN), metropolitan area network (MAN), and wide area network (WAN), each of which covers a successively larger geographic area, such as a single building or a campus or that would interconnect facilities in different parts of a country or of the world.

5.2 Components of an 802.15.3 piconet

An 802.15.3 piconet consists of several components, as shown in Figure 1. The basic component is the DEV. One DEV is required to assume the role of the piconet coordinator (PNC) of the piconet. The PNC provides the basic timing for the piconet with the beacon. Additionally, the PNC manages the quality of service (QoS) requirements, power save modes and access control to the piconet.

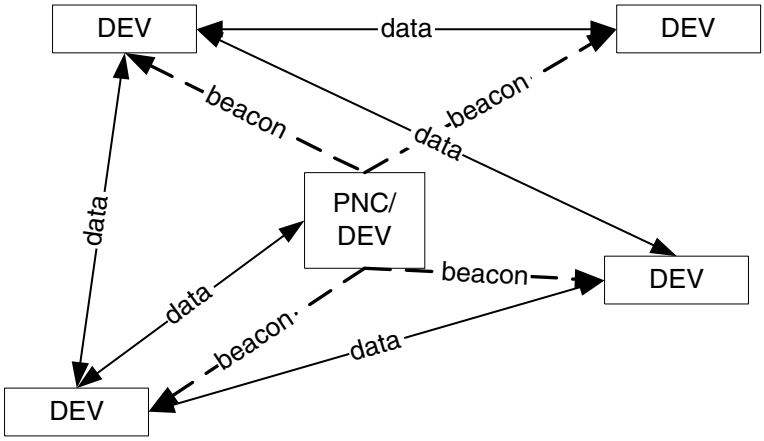

Figure 1—802.15.3 piconet elements

Because 802.15.3 piconets form without pre-planning and for only as long as the piconet is needed, this type of operation is referred to as an ad hoc network.

This standard allows a DEV to request the formation of a subsidiary piconet. The original piconet is referred to as the parent piconet. The subsidiary piconet is referred to as either a child or neighbor piconet, depending on the method the DEV used to associate with the parent PNC. Child and neighbor piconets are also referred to as dependent piconets since they rely on the parent PNC to allocate channel time for the operation of the dependent piconet. An indepedent piconet is a piconet that does not have any dependent piconets.

5.3 Overview of MAC functionality

IEEE 803.15.3 MAC is designed to support the following goals:

— Fast connection time
— Ad hoc networks

— Data transport with quality of service (QoS)
— Security
— Dynamic membership
— Efficient data transfer

5.3.1 Coordination

A piconet is formed when an 802.15.3 DEV that is capable of acting as the PNC begins transmitting beacons. Thus, even if there are no associated DEVs, the PNC sending the beacon is considered to be a piconet. One of the primary functions of the PNC is to transmit a beacon with appropriate information about the piconet.

5.3.1.1 Starting a piconet

To start a piconet, a DEV that is capable of acting as the PNC scans the available channels to find one that is not being used, as described in 8.2.1. If it finds one that is clear, it starts the piconet by simply sending the beacon after making sure that the channel has remained empty for a specified period of time, as described in 8.2.2. If no channels are available, the DEV has the option of attempting to start a dependent piconet, as described in 8.2.5 and 8.2.6 and summarized in 5.3.1.3 and 5.3.1.4. While the process of starting a piconet does not ensure that the “most capable” PNC is initially selected based on the criteria in 8.2.3, the association and handover process does allow the “most capable” DEV to eventually become the PNC of the piconet.

While the PNC is allowed to handover to a dependent PNC, this does not imply that the dependent PNC will merge the two piconets. The 802.15.3 standard does not provide a process for merging two piconets into a single piconet.

5.3.1.2 Handing over control of the piconet

When a DEV associates with an existing piconet,as described in 8.3.1 and 5.3.3, the PNC checks the capabilities of the new DEV to see if it is more capable to be the PNC of the piconet based on the criteria defined in 8.2.3. If the new DEV is more capable and the current security policies allow it, then the PNC has the option of handing over control of the piconet to the DEV that has just joined. This handover process, as described in 8.2.3, maintains all existing time allocations so that there is no interruption in the delivery of data in the piconet. If the PNC is shutting down or wants to leave the piconet, it also uses the handover process to give control to another DEV in the piconet. The handover process also supports the handing over of a dependent PNC, as described in 8.2.4, which is somewhat more complex than the handover of an independent or parent PNC.

5.3.1.3 Creating a child piconet

A child piconet is one that is formed under an established piconet. The established piconet then becomes the parent piconet. The child piconet functionality is useful for either extending the area of coverage of the piconet or shifting some computational or memory requirements to another PNC capable DEV. It is possible for the parent piconet to have more than one child piconet. In addition, it is also possible for a child PNC to allow a child piconet as a part of its own piconet.

The child piconet uses a distinct piconet ID (PNID) and acts as an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association and security membership for the child piconet are handled within the child piconet and do not involve the parent PNC.

The child PNC is a member of the parent piconet and thus is able to exchange data with any DEV in the parent piconet. The child PNC is also a member of the child piconet and thus is able to exchange data with any DEV in the child piconet. The use of the child piconet is described in 8.2.5.

5.3.1.4 Creating a neighbor piconet

A neighbor piconet is formed under an established piconet. The established piconet then becomes the parent piconet. The neighbor piconet functionality is a mechanism for sharing the frequency spectrum between different piconets when there are no vacant PHY channels. It is possible for a single piconet to have more than one neighbor piconet or to have both child and neighbor piconets as a part of the parent piconet. In addition, it is possible for the neighbor PNC to allocate a child or neighbor piconets within its own piconet.

The neighbor piconet uses a distinct PNID and is an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association and security membership for the neighbor piconet are handled within the neighbor piconet and do not involve the parent PNC.

The neighbor PNC is not a member of the parent piconet and thus does not exchange information with any DEV in the parent piconet. The neighbor piconet mechanism is available to other wireless DEVs as a means of sharing the frequency spectrum. Any entity capable of initiating (i.e. requesting status as) an 802.15.3 neighbor piconet would also be capable of using this as a coexistence method. The use of the neighbor piconet is described in 8.2.6.

5.3.2 Ending a piconet

If the PNC is going to stop operation and there are no other PNC capable DEVs in the piconet, the PNC places the PNC Shutdown information element (IE), as described in 7.4.5, into the beacon as described in 8.2.7.1 to notify the members of the piconet.

In the case that the PNC abruptly leaves the piconet without handing over control to another PNC capable DEV in the piconet, the piconet stops operation. After the association timeout period (ATP) expires, a PNC capable DEV from the old piconet will be able to start a new piconet using the normal process, as described in 8.2.2.

In the case of dependent piconets, the parent PNC is able to end the dependent piconet via the Disassociation Request command, described in 7.5.1.3, for neighbor piconets, or by using the stream termination procedure, described in 8.5.1.3, for child piconets, as described in 8.2.7.2.

5.3.2.1 Ending a piconet with a dependent piconet involved

If the parent piconet ends operation, the parent PNC will indicate in the PNC Shutdown IE, as described in 7.4.5, a dependent piconet that will be able to continue to operate. All other dependent piconets will cease operation when the parent piconet ends operation. The dependent PNC that was selected to remain will remove the Parent Piconet IE, as described in 7.4.3, from its beacon frame, signifying that it is no longer a dependent piconet. In the case where the parent piconet was temporarily disrupted, the parent PNC is able to attempt to join the dependent piconet and potentially receive a transfer of control via PNC handover.

A child piconet ends its piconet with the shutdown procedure and then uses the Channel Time Request command, as described in 7.5.6.1, to terminate the stream and release the resources in the parent piconet. When the child PNC shuts down its piconet, it is not required to leave the parent piconet.

The neighbor piconet uses the Disassociation Request command, as described in 7.5.1.3, to end its relationship with the parent PNC.

When a dependent piconet ends operation it has no affect on the parent piconet except to release resources.

5.3.3 Association and disassociation

In order to participate in a piconet, a DEV needs to join the piconet using the association process, as described in 8.3.1. Associating with the piconet provides the DEV with a unique identifier, the DEVID, for that piconet, as described in 7.2.3. The DEVID, one octet in length, is used instead of the DEV's address, 8 octets in length, to save overhead in the system. The association process optionally provides information about the services available in the piconet as well as the services provided by the DEV, as described in 8.3.2. The association process also provides the PNC with the capabilities of the new DEV to enable the PNC to decide if it wants to hand over control of the piconet to the new DEV, as described in 8.2.3 and 5.3.1.2.

When a new DEV joins the piconet, the PNC broadcasts the information about all of the DEVs in the piconet, as described in 8.3.3, and places information in the beacon about the new DEV. This allows other DEVs in the piconet to become aware of the new DEV as well as giving information to the new DEV about the members of the piconet.

When a DEV wants to leave the piconet or if the PNC wants to remove a DEV from the piconet, the disassociation process, as described in 8.3.4, is used. The DEVID of the disassociated DEV is no longer valid, until reissued by the PNC. However, the PNC is not allowed to reissue the DEVID until a waiting period has expired, as described in 8.3.1.

5.3.4 Security overview

Security for the piconet is one of two modes, as described in 9.2:

a) Mode 0—Open: Security membership is not required and payload protection (either data integrity or data encryption) is not used by the MAC. The PNC is allowed to use a list of DEV addresses to admit or deny entry to the piconet.
b) Mode 1—Secure membership and payload protection: DEVs establish secure membership with the PNC before they have access to the piconet's resources. Data sent in the piconet is allowed to use payload protection (data integrity and/or data encryption). Data integrity is required for most of the commands that are sent in the piconet.

When security is enabled, i.e. the piconet is using security mode 1, DEVs that wish to join the piconet are required to establish secure membership with the PNC. The DEVs are also allowed to establish a secure relationship with other DEVs for secure communications. A DEV has established a secure membership or a secure relationship when it gets a managment key for the security relationship. The process of establishing secure membership or a secure relationship is outside of the scope of this standard. The PNC or DEV that generates and distributes the key is called the key originator.

The payload protection protocol, as described in 10.2.2, uses a symmetric key that is generated by the key originator and is securely distributed to DEVs that have established secure membership or a secure relationship with the key originator, as described in 9.4.2.

5.3.5 The 802.15.3 superframe

Timing in the 802.15.3 piconet is based on the superframe, which is illustrated in Figure 2. The superframe is composed of three parts:

— The beacon, as described in 7.3.1, which is used to set the timing allocations and to communicate management information for the piconet. The beacon consists of the beacon frame, as described in 7.3.1, as well as any Announce commands sent by the PNC as a beacon extension, as described in 8.6.3.
— The contention access period (CAP), as described in 8.4.2, which is used to communicate commands and/or asynchronous data if it is present in the superframe.

— The channel time allocation period (CTAP), 8.4.3, which is composed of channel time allocations (CTAs), including management CTAs (MCTAs). CTAs are used for commands, isochronous streams and asynchronous data connections.

In Figure 2 the MCTAs are shown first, but the PNC is allowed to place any number of them at any position in the superframe.

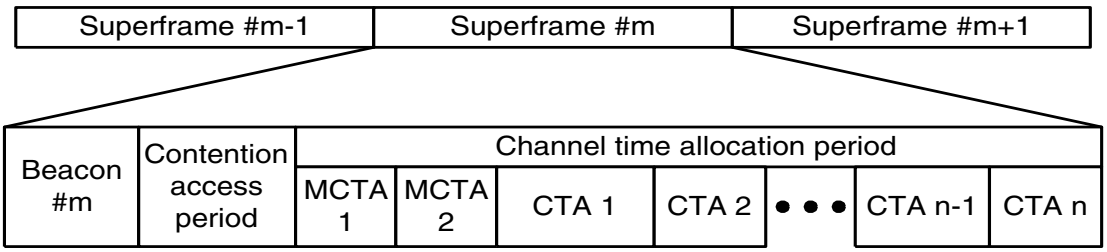

Figure 2—802.15.3 piconet superframe

The length of the CAP is determined by the PNC and communicated to the DEVs in the piconet via the beacon. However, the PNC is able to replace the functionality provided in the CAP with management CTAs (MCTAs), except in the case of the 2.4 GHz PHY where the PNC is required to allow DEVs to use the CAP, as described in 11.2.10. MCTAs are a type of CTA that is used for communications between the DEVs and the PNC.

The CAP uses CSMA/CA for the medium access, as described in 8.4.2. The CTAP, on the other hand, uses a standard TDMA protocol where the DEVs have specified time windows, as described in 8.4.3.1. MCTAs, as described in 8.4.3.3, are either assigned to a specific source/destination pair and use TDMA for access or they are shared CTAs that are accessed using the slotted aloha protocol, as described in 8.4.3.4.

5.3.6 Channel time management

All data in the 802.15.3 piconet is exchanged in a peer-to-peer manner. There are three methods for communicating data between DEVs in the piconet:

a) Sending asynchronous data in the CAP, if present, as described in 8.4.2.
b) Allocating channel time for isochronous streams in the CTAP, as described in 8.5.1.
c) Allocating asynchronous channel time in the CTAP, as described in 8.5.2.

If the CAP is present in the superframe and the PNC allows data in the CAP, DEVs in the piconet are able to use the CAP to send small amounts of data without having to allocate channel time.

If the DEV needs channel time on a regular basis, it makes a request from the PNC for isochronous channel time, as described in 8.5.1.1. If the resources are available, the PNC allocates time in a CTA for the DEV. If the requirements for the data change, then the DEV is able to request a change to the allocation, as described in 8.5.1.2. The source DEV, destination DEV or the PNC are all allowed to terminate the stream, as described in 8.5.1.3.

For regular CTAs, the PNC is able to change their position within the superframe every superframe. If a DEV misses a beacon, it is unable to use the allocation for a regular CTA. To avoid lost throughput due to missed beacons, DEVs are allowed to request a special type of CTA called a pseudo-static CTA, as described in 8.4.3.1. If the DEV is allocated a pseudo-static CTA, it is allowed to use the CTA for up to mMaxLostBeacons missed beacons. The PNC is allowed to move the locations of these CTAs, but needs to maintain

the time for the old allocation for mMaxLostBeacons superframes to avoid collisions, as described in 8.4.3.1.

Asynchronous allocation is slightly different. Rather than requesting recurring channel time, an asynchronous channel time request, as described in 8.5.2.1, is a request for a total amount of time to be used to transfer its data. The PNC is then able to schedule time for this request when available based on the channel time requirements. Unlike an isochronous allocation, only the source DEV or PNC are all allowed to terminate an asynchronous allocation, as described in 8.5.2.2.

5.3.7 Data communications between DEVs

In order to handle large data frames from layers above the MAC sublayer, this standard supports the fragmentation and defragmentation of these data frames, as described in 8.7. The ability to fragment data frames is also useful to reduce the frame error rate (FER) of a marginal link by decreasing the frame size. The fragments are numbered with a sequence number for the upper layer frame as well as a sequence number for the fragment itself. The total number of fragments of that data frame is also sent to enable the receiving DEV to allocate the correct amount of internal memory to hold the incoming frame.

If the source DEV wishes to verify the delivery of a frame, then one of the acknowledgement (ACK) policies is used, as described in 8.8. This standard provides for three types of ACKs to enable different applications. The no-ACK policy, as described in 8.8.1, is appropriate for frames that do not require guaranteed delivery, where the retransmitted frame would arrive too late or where an upper layer protocol is handling the ACK and retransmission protocol. The immediate-ACK (Imm-ACK) policy, as described in 8.8.2, provides an ACK process in which each frame is individually ACKed following the reception of the frame. The delayed-ACK (Dly-ACK) policy, as described in 8.8.3 lets the source send multiple frames without the intervening ACKs. Instead, the ACKs of the individual frames are grouped into a single response frame that is sent when requested by the source DEV. The Dly-ACK process decreases the overhead in the Imm-ACK process while allowing the source DEV to verify the delivery of frames to the destination.

If the source DEV does not receive the requested ACK, then it has the option of retransmitting the frame, as described in 8.8.4, or dropping the frame. The decision to retransmit or drop the frame depends on the type of data or command that is being sent, the number of times that the source DEV has attempted sending the frame, the length of time it has spent attempting to send the frame, or other implementation dependent factors.

5.3.8 Information discovery in the piconet

Since 802.15.3 piconets are ad hoc in nature, it is important for the DEVs in the piconet to be able to find out information about the services and capabilities of the other DEVs in the piconet at any instant in time. This standard supports four methods for discovering information about other DEVs in the piconet: the PNC Information Request command, as described in 7.5.4.1, the Probe Request command, as described in 7.5.4.5, the Announce command, as described in 7.5.5.2, and the Piconet Services command, as described in 7.5.5.1. In addition, the PNC is able to ask a DEV in the piconet to evaluate the channel conditions in either the current channel or in an alternate channel with the Remote Scan Request command, as described in 7.5.7.3. Any DEV in the piconet is able to ask another DEV about the status of the current channel with the Channel Status Request command, as described in 7.5.7.1.

The PNC Information Request command, as described in 8.9.1, is used to obtain information from the PNC about either a specific DEV in the piconet or all of the DEVs in the piconet. The PNC responds to the request with the PNC Information command, as described in 7.5.4.2, which contains information about the DEV or DEVs that was requested by the originator of the command.

A DEV uses the Probe Request command, 8.9.2, to find out more detailed information about other DEVs in the piconet. This command allows the originating DEV to retrieve many of the valid IEs, 7.4, from a target DEV in the piconet.

One of the goals of connecting DEVs in a piconet is to enable them to share services. However, to do this a DEV needs to be able to discover the services are available in the piconet as well as to advertise its services to other DEVs in the piconet. This standard enables this with the Piconet Services IE, as described in 7.4.16, which optionally is exchanged in the association process, as described in 8.3.1. Both the DEV and the PNC have the option of not sending the element if either entity is concerned about the security aspects involved in advertising this information.

The PNC is responsible for deciding in which channel the piconet will operate. In addition to the information that the PNC is able to determine about the quality of the channels at its location, it is also able to request information about the channel from other DEVs in the piconet using the Remote Scan Request command, as described in 8.9.5. Since the DEVs in the piconet in general are at different geographic locations, they provide additional information about interference or other piconets in the area. In response to the PNC's request, the DEV is able to either scan the channels as requested or reject the request from the PNC if it is not going to perform the scan.

Any DEV in the piconet is able to request information about the quality of the link between itself and another DEV with the Channel Status Request command, as described in 7.5.7.1. This command is used for two purposes. The first is to allow the DEVs to change transmit power, the data rate or the requested channel time based on the quality of the data connection between the DEVs. The other use is for the PNC to determine if the DEVs in the piconet are having trouble with the channel. This information along with PNC's scanning of the channel and optional remote scan requests assists the PNC in determining if it needs to change the channel that the piconet is currently using.

5.3.9 Dynamic channel selection

The piconet operates in a dynamic environment and under unlicensed operation rules. Thus, it is subject to interference from licensed users, other 802.15.3 piconets as well as other unlicensed wireless entities in its channels. To enable the piconet to continue operation in this type of environment, the PNC has the capability to dynamically change the channel that the piconet is using without requiring either user intervention or the disruption of services in the piconet. To evaluate the status of the current channel as well as other channels, the PNC is able to use many methods including:

— Gathering information about the current channel from other DEVs in the piconet using the Channel Status Request command, as described in 8.9.4.
— Performing a passive scan of the channels, as described in 8.11.1.
— Requesting other DEVs to perform a channel scan using the Remote Scan Request command, as described in 8.9.5.

If the PNC determines that the current channel is unsuitable, it uses the dynamic channel selection procedure, as described in 8.11.1, to move the piconet to the new channel. The configuration of the piconet and the channel time allocations do not change with a channel change so that the services provided by the piconet are not interrupted by the change.

5.3.10 Power management

An important goal of the 802.15.3 standard is to enable long operation time for battery powered DEVs. The best method for extending battery life is to enable DEVs to turn off completely or reduce power for long periods of time, where a long period is relative to the superframe duration. This standard provides three techniques to enable DEVs to turn off for one or more superframes: device synchronized power save (DSPS) mode, piconet-synchronized power save (PSPS) mode and asynchronous power save (APS) mode. In the

piconet, DEVs operate in one of four power management (PM) modes; ACTIVE mode, DSPS mode, PSPS mode or APS mode.

PSPS mode, as described in 8.13.1, allows DEVs to sleep at intervals defined by the PNC. The DEV sends a request to the PNC when it wants to enter the PSPS mode. The PNC informs the piconet by setting the DEV's bit in its PS Status IE in the beacon. The PNC then selects beacons that will be the system wake beacons and indicates the next one in the PS Status IE for the PSPS set. All DEVs in PSPS mode are required to listen to the system wake beacons.

DSPS mode, as described in 8.13.2, is designed to enable groups of DEVs to sleep for multiple superframes but still be able to wake up during the same superframe. DEVs synchronize their sleep patterns by joining a DSPS set which specifies the interval between wake periods for the DEVs and the next time the DEVs will be awake. Besides allowing the DEVs to wake up and exchange traffic at the same time, the use of DSPS sets makes it easy for other DEVs in the piconet to determine exactly when a DSPS DEV will be available to receive traffic.

APS mode, as described in 8.13.3, allows a DEV to conserve power for extended periods until the DEV chooses to listen for a beacon. The only responsibility of a DEV in APS mode is to communicate with the PNC before the end of its ATP in order to preserve its membership in the piconet.

The PNC allocates asychronous CTAs to a destination DEV that is in either PSPS mode or DSPS mode in the wake superframes for that DEV.

Regardless of the DEV's power management mode, every DEV in the piconet is allowed to power down during parts of the superframe when the DEV is not scheduled to transmit or receive data.

5.3.11 Controlling transmit power in the piconet

The ability to control transmit power in the piconet enables DEVs to minimize interference with other wireless networks that share the same channel as well as to decrease the power usage in some PHY implementations. Two methods are provided by this standard for controlling transmitter power. The first method allows the PNC to set a maximum transmit power for the CAP, beacon, and MCTAs, excluding association MCTAs, as described in 8.11.2.1. Since the link between the PNC and the DEV defines the size of the piconet, controlling the power during these times allows the PNC to reduce transmit power without adversely affecting operation of the piconet.

The second method allows DEVs using a CTA to request either an increase or a decrease in the transmit power of the remote DEV, as described in 8.11.2.2. Thus if two DEVs have a "good" link in a CTA, they are able to reduce their transmitter power to decrease the power usage in some PHY implementations, and to reduce interference to other networks.

5.4 Characteristics of the 2.4 GHz PHY

5.4.1 General characteristics

The 2.4 GHz PHY, Clause 11, uses the 2.4 to 2.4835 GHz band that is available for unlicensed use in much of the world, as described in 11.1. Two channel plans are defined, one with 4 channels for high density applications and one with 3 channels to enable better coexistence with IEEE Std 802.11b-1999™[B3] networks, as described in 11.2.3. The PHY also supports 5 data rates, ranging from 11 to 55 Mb/s. The base rate of 22 Mb/s is uncoded, while the 11, 33, 44 and 55 Mb/s use trellis coded modulation, 11.3.

For efficiency, the PHY calculates the header check sequence over both the MAC and PHY headers and appends this to the MAC header. The header for all frames is sent at the base rate, 22 Mb/s, to allow all DEVs in the piconet to detect traffic. In the 11 Mb/s mode, the entire MAC and PHY header is repeated at the lower modulation rate to increase the probability of receiving the entire header correctly.

The PHY uses a constant-amplitude, zero-autocorrelation (CAZAC) sequence for the preamble, as described in 11.4.2. This sequence has good properties for obtaining synchronization, timing information and frequency offset.

The on-air bandwidth is limited to 15 MHz in order to allow more channels as well as to decrease the interference to other systems and to decrease the susceptibility to interference from other systems. The transmitter power is constrained by the limitations of the appropriate regulatory bodies.

The receiver of a compliant system reports both the signal level and, if the higher order modulations are used, an indication of the signal quality. This allows a DEV to determine if errors in the channel are due to poor signal quality or due to interference from other systems.

5.4.2 Coexistence and interoperability

While the 802.15.3 standard does not require interoperability with other standards, there were choices made with the 2.4 GHz PHY specification that make it easier to design dual-mode radios. The commonalities that allow interoperable radios are discussed in C.1.

Because the 2.4 GHz PHY operates as an unlicensed system, it needs to share the medium with both licensed and other unlicensed users in the band. The 802.15.3 MAC and the 2.4 GHz PHY offer a variety of techniques to enhance the coexistence with other users in the band. The methods provided by this standard include:

— passive scanning
— dynamic channel selection
— the ability to request channel quality information
— link quality and RSSI
— a channel plan that minimizes channel overlap
— lower transmit power
— transmit power control
— neighbor piconet capability

The use of these methods to improve coexistence is described in C.2.

6. Layer management

6.1 Overview of management model

Both MAC and PHY layers conceptually include management entities, called the MAC sublayer management entity and PHY layer management entity (MLME and PLME, respectively). These entities provide the layer management service interfaces for the layer management functions.

In order to provide correct MAC operation, a device management entity (DME) should be present within each DEV. The DME is a layer-independent entity that may be viewed as residing in a separate management plane or as residing “off to the side.” The exact functionality of the DME is not specified in this standard, but in general this entity may be viewed as being responsible for such functions as the gathering of layer-dependent status from the various layer management entities, and similarly setting the value of layer-specific

parameters. The DME typically performs such functions on behalf of the general system management entities and implements standard management protocols. Figure 3 depicts the relationship among the management entities.

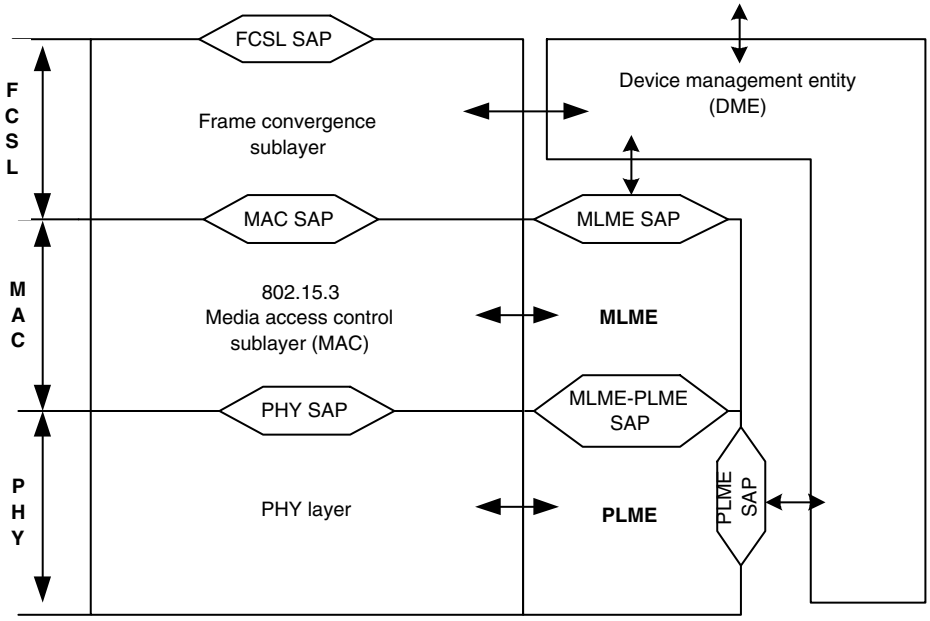

Figure 3—The reference model used in this standard

The various entities within this model interact in various ways. Certain of these interactions are defined explicitly within this standard, via a service access point (SAP) across which defined primitives are exchanged. Other interactions are not defined explicitly within this standard, such as the interface between the MAC and the MLME or the interface between the PHY and the PLME. The specific manner in which these MAC and PHY interfaces are integrated into the overall MAC and PHY layers are not specified within this standard.

The various SAPs within this model are the following:

a) FCSL SAP
b) MAC SAP
c) PHY SAP
d) MLME SAP
e) PLME SAP
f) MLME-PLME SAP

The latter two SAPs support identical primitives, and in fact may be viewed as a single SAP (called simply the PLME SAP) that may be used either directly by the MLME or by the DME. In this fashion, the model reflects the approach that is anticipated to be a common implementation in which the PLME functions are controlled by the MLME (on behalf of the DME). In particular, PHY implementations are not required to have separate interfaces defined other than their interfaces with the MAC and the MLME.

The MAC SAP is described further in Annex A.

If the SAP interfaces are not exposed in an 802.15.3 DEV, then these interfaces do not have to be implemented as described here. If the interfaces are exposed, then they should support the primitives described in this clause.

The split in functionality between the MLME and DME in this standard is intended to facilitate the formal verification of the protocol. It is not intended to be an architectural description of a particular implementation.

6.2 Generic management primitives

The management information specific to each layer is represented as a personal area network (PAN) information base (PIB) for that layer. In a LAN/MAN the corresponding information is in the management information base (MIB) and is often associated with a management protocol such as SNMP (simple network management protocol) (see [B12]). However, piconets are not intended to be managed across a network but rather use the management information to ascertain the characteristics of the layer or sublayer.

The MLME and PLME are viewed as "containing" the PIB for that layer or sublayer. The generic model of PIB-related management primitives exchanged across the management SAPs is to allow the SAP user entity to either "GET" the value of a PIB attribute, or to "SET" the value of a PIB attribute. The invocation of a SET.request primitive may require the layer entity to perform certain defined actions.

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 or PHY layer management SAP is involved. The DME uses the services provided by the MLME through the MLME SAP. The primitives are summarized in Table 1. In Table 1 and in the subclauses that describe the primitives, XX denotes either MLME or PLME.

Table 1—Summary of generic management primitives

Name	Request	Confirm
XX-GET	6.2.1	6.2.2
XX-SET	6.2.3	6.2.4

The parameters used for these primitives are defined in Table 2.

Table 2—MLME and PLME generic management primitive parameters

Name	Type	Valid range	Description
PIBattribute	Octet string	Any PIB attribute as defined in 6.5 or 11.7	The name of the PIB attribute.
PIBvalue	Variable	As defined in 6.5 or 11.7	The PIB value.
ResultCode	Enumeration	SUCCESS, INVALID_PIB_ATTRIBUTE_NAME, INVALID_PIB_ATTRIBUTE_VALUE, READ_ONLY_PIB_ATTRIBUTE, WRITE_ONLY_PIB_ATTRIBUTE	Indicates the result of the MLME or PLME request.

Other SAP-specific primitives are identified in 6.3 and 6.4.

6.2.1 MLME-GET.request and PLME-GET.request

This primitive requests information about a given MAC or PHY PIB attribute. The semantics of this primitive are:

```
XX-GET.request (
               PIBattribute
               )
```

The primitive parameter is defined in Table 2.

6.2.1.1 When generated

This primitive is generated by either the DME or the MLME (in the case of PLME-GET.request) to obtain information from either the MAC or PHY PIB.

6.2.1.2 Effect of receipt

The appropriate management entity attempts to retrieve the requested PIB attribute from its database and responds with XX-GET.confirm that gives the result.

6.2.2 MLME-GET.confirm and PLME-GET.confirm

This primitive reports the results of an information request about either the MAC or PHY PIB. The semantics of this primitive are:

```
XX-GET.confirm (
               ResultCode,
               PIBattribute,
               PIBvalue
               )
```

The primitive parameters are defined in Table 2.

6.2.2.1 When generated

This primitive is generated in response to an XX-GET.request by either the DME or the MLME (in the case of PLME-GET.confirm).

6.2.2.2 Effect of receipt

The primitive returns the appropriate PIB attribute value if the ResultCode is SUCCESS, otherwise it returns an error indication in the ResultCode. Possible values of the ResultCode that would indicate an error are INVALID_PIB_ATTRIBUTE_NAME and WRITE_ONLY_PIB_ATTRIBUTE.

6.2.3 MLME-SET.request and PLME-SET.request

This primitive attempts to set the indicated MAC or PHY PIB attribute to the given value. The semantics of this primitive are:

```
XX-SET.request        (
                      PIBattribute,
                      PIBvalue
                      )
```

The primitive parameters are defined in Table 2.

6.2.3.1 When generated

This primitive is generated by either the DME or the MLME (in the case of PLME-SET.request) to set the indicated MAC or PHY PIB attribute.

6.2.3.2 Effect of receipt

The appropriate management entity attempts to set the requested PIB attribute in its database. If this PIB attribute implies a specific action, then this requests that the action be performed. The management entity that receives this primitive responds with XX-SET.confirm that gives the result.

6.2.4 MLME-SET.confirm and PLME-SET.confirm

This primitive reports the results of an attempt to set the value of an attribute in either the MAC or PHY PIB. The semantics of this primitive are:

```
XX-SET.confirm        (
                      ResultCode,
                      PIBattribute
                      )
```

The primitive parameters are defined in Table 2.

6.2.4.1 When generated

This primitive is generated in response to an XX-SET.request by either the DME or the MLME (in the case of PLME-SET.confirm).

6.2.4.2 Effect of receipt

If the ResultCode is SUCCESS, this confirms that the indicated PIB attribute was set to the requested value, otherwise it returns an error condition in the ResultCode. If this PIBattribute implies a specific action, then this confirms that the action was performed. Possible ResultCodes for an error are:

— INVALID_PIB_ATTRIBUTE_NAME
— INVALID_PIB_ATTRIBUTE_VALUE
— READ_ONLY_PIB_ATTRIBUTE.

6.3 MLME SAP interface

The services provided by the MLME to the DME 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. An ACTION.indication provides information to the DME that originated from another DEV. The DEV optionally responds to the indication by issuing an ACTION.response. The DME uses the services provided by the MLME through the MLME SAP. The primitives are summarized in Table 3.

Table 3—Summary of MLME primitives

Name	Request	Indication	Response	Confirm
MLME-RESET	6.3.1.1	–	–	–
MLME-SCAN	6.3.2.1	–	–	6.3.2.2
MLME-START	6.3.3.1	–	–	6.3.3.2
MLME-START-DEPENDENT	6.3.3.3	–	–	6.3.3.4
MLME-SYNCH	6.3.4.1	–	–	6.3.4.2
MLME-ATP-EXPIRED	–	6.3.4.3	–	–
MLME-ASSOCIATE	6.3.5.1	6.3.5.2	6.3.5.3	6.3.5.4
MLME-DEV-ASSOCIATION-INFO	–	6.3.5.5	–	–
MLME-DISASSOCIATE	6.3.6.1	6.3.6.2	–	6.3.6.3
MLME-REQUEST-KEY	6.3.7.1	6.3.7.2	6.3.7.3	6.3.7.4
MLME-DISTRIBUTE-KEY	6.3.8.1	6.3.8.2	6.3.8.3	6.3.8.4
MLME-MEMBERSHIP-UPDATE	6.3.9.1	–	–	–
MLME-SECURITY-ERROR	–	6.3.9.2	–	–
MLME-SECURITY-MESSAGE	6.3.9.3	6.3.9.4		6.3.9.5
MLME-PNC-HANDOVER	6.3.10.1	6.3.10.2	6.3.10.3	6.3.10.4
MLME-NEW-PNC	–	6.3.10.5	–	–
MLME-PNC-INFO	6.3.11.1	6.3.11.2	6.3.11.3	6.3.11.4
MLME-SECURITY-INFO	6.3.12.1	6.3.12.2	6.3.12.3	6.3.12.4
MLME-CREATE-ASIE	6.3.13.1	–	–	6.3.13.2
MLME-RECEIVE-ASIE	–	6.3.13.3	–	–
MLME-PROBE	6.3.14.1	6.3.14.2	6.3.14.3	6.3.14.4
MLME-ANNOUNCE	6.3.15.1	6.3.15.2	–	6.3.15.3
MLME-PICONET-SERVICES	–	6.3.16.1	6.3.16.2	6.3.16.3
MLME-CREATE-STREAM	6.3.17.1	–	–	6.3.17.2
MLME-MODIFY-STREAM	6.3.17.3	–	–	6.3.17.4

Table 3—Summary of MLME primitives (*Continued*)

Name	Request	Indication	Response	Confirm
MLME-TERMINATE-STREAM	6.3.17.5	6.3.17.6	–	6.3.17.7
MLME-MULTICAST-RX-SETUP	6.3.17.8	–	–	–
MLME-CHANNEL-STATUS	6.3.18.1	6.3.18.2	6.3.18.3	6.3.18.4
MLME-REMOTE-SCAN	6.3.19.1	6.3.19.2	6.3.19.3	6.3.19.4
MLME-PICONET-PARM-CHANGE	6.3.20.1	–	–	6.3.20.2
MLME-TX-POWER-CHANGE	6.3.21.1	6.3.21.2	–	6.3.21.3
MLME-PS-SET-INFORMATION	6.3.22.1	–	–	6.3.22.2
MLME-SPS-CONFIGURE	6.3.22.3	–	–	6.3.22.4
MLME-PM-MODE-CHANGE	6.3.22.5	–	–	6.3.22.6
MLME-PM-MODE-ACTIVE	–	6.3.22.7	–	–

6.3.1 Reset

This mechanism supports the process of resetting the MAC. The parameters used for these primitives are defined in Table 4.

Table 4—MLME-RESET primitive parameters

Name	Type	Valid range	Description
SetDefaultPIB	Boolean	TRUE, FALSE	If TRUE, all PIB attributes are set to their default values. The default values are implementation dependent. If FALSE, the MAC is reset, but all PIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive.
ResetTimeout	Duration	0–65535	The time in milliseconds allowed to complete the reset procedure.

6.3.1.1 MLME-RESET.request

This primitive requests that the MAC entity be reset. The semantics of this primitive are:

```
MLME-RESET.request          (
                            SetDefaultPIB,
                            ResetTimeout
                            )
```

The primitive parameters are defined in Table 4.

6.3.1.1.1 When generated

This primitive is sent by the DME to its MLME to reset the MAC to its initial conditions.

6.3.1.1.2 Effect of receipt

If the DEV is currently associated with a piconet, the DEV MLME, upon receiving this primitive, sends a Disassociation Request command, as described in 7.5.1.3, to the PNC. In all cases, the DEV MLME sets the MAC to its initial conditions and sets all of its internal variables to their default values but still consistent with the SetDefaultPIB parameter of Table 4.

The PNC MLME, upon receiving this primitive, behaves the same as the DEV MLME with the exception that it performs either a PNC handover, as described in 8.2.3, followed by disassociation or just performs the piconet shutdown operation, as described in 8.2.7.1. The PNC MLME decides which action to take.

If the ResetTimeout expires while the DEV (or PNC) MLME is still performing disassociation, handover or shutdown, the MLME resets the MAC and as a result, all pending operations are cancelled.

6.3.2 Scan

This mechanism supports the process of determining the presence or absence of piconets in a communications channel. The parameters used for these primitives are defined in Table 5.

A PiconetDescriptionSet is an array of PiconetDescriptions. Each PiconetDescription consists of the elements shown in Table 6.

In Table 5, the ChannelRatingList is an array of N integer values, where N equals the number of channel numbers provided in the ChannelList of the received MLME-SCAN.request. The elements of the array are channel numbers and they are ordered from best (least interference) at the lowest array index to worst (most interference) at the highest array index.

Any security features of an existing piconet are ignored during the scan process.

6.3.2.1 MLME-SCAN.request

This primitive is used to initiate a passive scan for either a specific PNID/BSID or any PNID/BSID in each of the channels listed in the ChannelList parameter. The semantics of this primitive are:

```
MLME-SCAN.request       (
                        OpenScan,
                        BSID,
                        PNID,
                        ChannelList,
                        ChannelScanDuration
                        )
```

The primitive parameters are defined in Table 5.

6.3.2.1.1 When generated

This primitive is sent from the DME to its MLME to initiate a passive scan for either a specific BSID, PNID, or for any BSID and/or PNID.

Table 5—MLME-SCAN primitive parameters

Name	Type	Valid range	Description
OpenScan	Boolean	TRUE, FALSE	Indicates whether scan is an open scan or not. Open scan is defined in 8.2.1.
PNID	Integer	0–65535	The ID of a specific piconet for which to scan.
BSID	Octet string	As defined in 7.4.2	The text string of a specific piconet for which to scan. This parameter is not used if open scan, 8.2.1, is requested.
ChannelList	Ordered set of integers	0 to the maximum PHY channel ID as defined in 11.2.3	Specifies a list of channels to be examined when scanning for either a specific PNID/BSID or any PNID/BSID.
ChannelScanDuration	Duration	0–65535	The length of time in milliseconds that the DEV is to spend scanning a channel to find either a specific PNID/BSID, or any PNID/BSID.
NumberOfPiconets	Integer	0–255	The number of piconets found during the scanning process.
PiconetDescriptionSet	Set of piconet descriptions as defined in Table 6	A set containing zero or more instances of a PiconetDescription	The PiconetDescriptionSet is returned to indicate the results of the scan request.
NumberOfChannels	Integer	0–n PHY dependent channels defined in 11.2.3	Indicates the number of channels scanned.
ChannelRatingList	Ordered list of integers.	0 to the maximum number of PHY dependent channels defined in 11.2.3	Specifies a list of found channels ordered from the best to the worst in terms of interference.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS	Indicates the result of the MLME request.

6.3.2.1.2 Effect of receipt

When the MLME receives this primitive from its DME, it initiates a passive scan in each of the channels specified in the ChannelList parameter as described in 8.2.1. Upon completion of the scan procedure, the MLME will send an MLME-SCAN.confirm to its DME to report the results of the scan.

Table 6—Elements of PiconetDescription

Name	Type	Valid range	Description
BSID	Octet string	As defined in 7.4.2	The text string identifier of a discovered piconet.
PNCDEVAddress	MAC address	Any valid individual MAC address	The MAC address of the PNC of the piconet that was found.
PNID	Integer	0–65535	The PNID of a discovered piconet.
PiconetType	Enumeration	DEPENDENT, NON-DEPENDENT	The type of a discovered piconet.
ParentPiconetBSID	Octet string	As defined in 7.4.3	The BSID of the parent piconet if a beacon of a dependent piconet was found.
ParentPNCAddress	MAC address	Any valid individual MAC address.	The MAC address of the parent PNC of the piconet that was found.
ScannedFrameType	Enumeration	BEACON, NON-BEACON	Indicates the type of frame that was found.
ChannelIndex	Integer	0-255	A PHY dependent channel number.
SuperframeDuration	Duration	0-65535	As defined in 7.3.1.
CAPEndTime	Integer	0-65535	As defined in 7.3.1.
SECID	2 octets	As defined in 7.2.7.2.	As defined in 7.2.7.2.
CAPData	Boolean	TRUE, FALSE	As defined in 7.3.1.
CAPCommands	Boolean	TRUE, FALSE	As defined in 7.3.1.
CAPAssociation	Boolean	TRUE, FALSE	As defined in 7.3.1.
SECmode	Enumeration	MODE_0, MODE_1	As defined in 7.3.1.
MCTAAllocationRate	Integer	0-15	As defined in 7.3.1.

6.3.2.2 MLME-SCAN.confirm

This primitive and its parameters, which were collected during the scan process, are returned upon completion of the scan process. The semantics of this primitive are:

```
MLME-SCAN.confirm       (
                        NumberOfPiconets,
                        PiconetDescriptionSet,
                        NumberOfChannels,
                        ChannelRatingList,
                        ResultCode
                        )
```

The primitive parameters are defined in Table 5.

6.3.2.2.1 When generated

This primitive is sent by the MLME to its DME when the passive scan of all the channels listed in the ChannelList parameter is complete or when it determines that the parameters of the MLME-SCAN.request are invalid.

6.3.2.2.2 Effect of receipt

The DME is notified of the results of the scan procedure.

6.3.3 Start

This mechanism supports the process of creating a new piconet. The parameters used for these primitives are defined in Table 7.

Table 7—MLME-START and MLME-START-DEPENDENT primitive parameters

Name	Type	Valid range	Description
PNID	Integer	0-65535	The PNID of the new piconet.
BSID	Octet string	As defined in 7.4.2	The BSID of the new piconet.
ChannelIndex	Integer	0-255	Indicates the PHY dependent channel, 11.2.3, in which to start a piconet.
StreamIndex	Integer	As defined in 7.2.5	The stream index that was assigned in the channel time allocation process for the dependent piconet.
SuperframeDuration	Duration	0-65535	As defined in 7.3.1.
CAPEndTime	Integer	0-65535	As defined in 7.3.1.
SECID	2 octets	As defined in 7.2.7.2.	As defined in 7.2.7.2.
CAPData	Boolean	TRUE, FALSE	As defined in 7.3.1.
CAPCommands	Boolean	TRUE, FALSE	As defined in 7.3.1.
CAPAssociation	Boolean	TRUE, FALSE	As defined in 7.3.1.
SECMode	Enumeration	MODE_0, MODE_1	As defined in 7.3.1.
MaxTXPowerLevel	As defined in 7.3.1	As defined in 7.3.1	Maximum TX power allowed in the piconet.
MCTAUsed	Enumeration	TRUE, FALSE	As defined in 7.3.1.
MCTAAllocationRate	Integer	0-15	As defined in 7.3.1.
ParentPiconetIE	Information element	As defined in 7.4.3.	Provides the DEV address and BSID of the parent piconet.
ResultCode	Enumeration	SUCCESS, ALREADY_STARTED, CHANNEL_INTERFERENCE, PICONET_DETECTED, INVALID_PARAMETERS	Indicates the result of the requested action.

6.3.3.1 MLME-START.request

This primitive requests that the MAC entity start a new piconet. The semantics of this primitive are:

```
MLME-START.request      (
                        PNID,
                        BSID,
                        ChannelIndex,
                        SuperframeDuration,
                        CAPEndTime,
                        SECID,
                        CAPData,
                        CAPCommands,
                        CAPAssociation,
                        SECMode,
                        MaxTXPowerLevel,
                        MCTAUsed,
                        MCTAAllocationRate
                        )
```

The primitive parameters are defined in Table 7.

6.3.3.1.1 When generated

This primitive is generated by the DME to start a piconet with the DEV acting as the PNC.

6.3.3.1.2 Effect of receipt

This primitive initiates the piconet initialization procedure defined in 8.2.2. The MLME subsequently issues an MLME-START.confirm that reflects the results of the creation procedure.

6.3.3.2 MLME-START.confirm

This primitive reports the results of a piconet creation procedure. The semantics of this primitive are:

```
MLME-START.confirm      (
                        ResultCode
                        )
```

The primitive parameter is defined in Table 7.

6.3.3.2.1 When generated

This primitive is generated by the MLME as a result of an MLME-START.request.

6.3.3.2.2 Effect of receipt

The DME is notified of the results of the piconet creation procedure. A ResultCode of SUCCESS indicates that the DEV is the PNC. If the requested channel is occupied by other 802.15.3 piconets, the ResultCode is set to PICONET_DETECTED. If the piconet is already started the ResultCode is set to ALREADY_STARTED. If the requested channel for starting the piconet has unacceptable interference, then the ResultCode is set to CHANNEL_INTERFERENCE. If any of the parameters are in error the ResultCode is INVALID_PARAMETERS.

6.3.3.3 MLME-START-DEPENDENT.request

This primitive requests that the MAC entity start operations of a dependent piconet as PNC. The semantics of this primitive are:

```
MLME-START-DEPENDENT.request (
                             PNID,
                             BSID,
                             StreamIndex,
                             SuperframeDuration,
                             CAPEndTime,
                             SECID,
                             CAPData,
                             CAPCommands,
                             CAPAssociation,
                             SECMode,
                             MaxTXPowerLevel,
                             MCTAUsed,
                             MCTAAllocationRate,
                             ParentPiconetIE
                             )
```

The primitive parameters are defined in Table 7.

6.3.3.3.1 When generated

This primitive is generated by the DME to convey operating parameters to the PNC of a dependent piconet.

6.3.3.3.2 Effect of receipt

This primitive initiates the piconet initialization procedure, as described in 8.2.5 or 8.2.6. The MLME subsequently issues an MLME-START-DEPENDENT.confirm that reflects the results of the start procedure.

6.3.3.4 MLME-START-DEPENDENT.confirm

This primitive reports the results of a dependent piconet start procedure. The semantics of this primitive are:

```
MLME-START-DEPENDENT.confirm (
                             ResultCode
                             )
```

The primitive parameter is defined in Table 7.

6.3.3.4.1 When generated

This primitive is generated by the MLME as a result of an MLME-START-DEPENDENT.request.

6.3.3.4.2 Effect of receipt

The DME is notified of the results of the dependent PNC start procedure. A ResultCode of SUCCESS indicates that the DEV has started dependent PNC operations. If the dependent PNC is already established, the ResultCode is ALREADY_STARTED.

6.3.4 Synchronization

The synchronization procedure is a preliminary step for a DEV associating with a particular piconet. A primitive is also provided to inform the DME when the DEV loses synchronization. The parameters used for these primitives are defined in Table 8.

Table 8—MLME-SYNCH primitive parameters

Name	Type	Valid range	Description
PNID	Integer	0–65535	The piconet identifier of the piconet with which to synchronize.
BSID	Octet string	As defined in 7.4.2	The beacon source identifier of the piconet with which to synchronize.
ChannelIndex	Integer	0–255	The PHY channel to be used to find the piconet.
ChannelScanDuration	Duration	0–65535	The time duration to be spent searching for the piconet.
ResultCode	Enumeration	SUCCESS, TIMEOUT INVALID_PARAMETERS	Indicates the result of the MLME-SYNCH.request.

6.3.4.1 MLME-SYNCH.request

This primitive is used to initiate synchronization with a specific piconet beacon. The semantics of this primitive are:

```
MLME-SYNCH.request      (
                        PNID,
                        BSID,
                        ChannelIndex,
                        ChannelScanDuration
                        )
```

The primitive parameters are defined in Table 8.

6.3.4.1.1 When generated

This primitive is generated by the DME to establish synchronization with a particular piconet beacon.

6.3.4.1.2 Effect of receipt

When the MLME receives this primitive from its DME, the MLME scans the specified channel until either the desired beacon is detected or the ChannelScanDuration interval is exceeded. In the case where the desired beacon is detected, the MLME sends to its DME an MLME-SYNCH.confirm with a ResultCode of SUCCESS. In the case where the ChannelScanDuration interval is exceeded, the MLME sends an MLME-SYNCH.confirm with a ResultCode of TIMEOUT.

6.3.4.2 MLME-SYNCH.confirm

This primitive informs the originating DME whether the requested piconet synchronization is successful or unsuccessful. The semantics of this primitive are:

```
MLME-SYNCH.confirm      (
                        ResultCode
                        )
```

The primitive parameter is defined in Table 8.

6.3.4.2.1 When generated

This primitive is generated by the MLME upon completion of the requested piconet synchronization procedure.

6.3.4.2.2 Effect of receipt

The DME is notified of the results of the synchronization procedure.

6.3.4.3 MLME-ATP-EXPIRED.indication

This primitive is used to indicate that the DEV is no longer able to hear the beacon. The semantics of this primitive are:

```
MLME-ATP-EXPIRED.indication      ()
```

6.3.4.3.1 When generated

This primitive is generated by a non-PNC MLME when it does not correctly receive a beacon for greater than the association timeout period (ATP) interval of time, as described in 7.5.1.1.

6.3.4.3.2 Effect of receipt

The DME is notified that the ATP has expired.

6.3.5 Association

The following primitives support the process of a DEV associating with a PNC. The parameters used for these primitives are defined in Table 9.

Table 9—MLME-ASSOCIATE primitive parameters

Name	Type	Valid range	Description
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	The UnassocID or the DEVID that was just assigned by the PNC.
DEVAddress	MAC address	Any valid individual MAC address.	The MAC address of the DEV that is requesting association with the PNC.
PSRC	Boolean	TRUE, FALSE	If TRUE, the DEV is receiving power from alternating current mains instead of battery power.
SEC	Boolean	TRUE, FALSE	If TRUE, the DEV is capable of acting as a key originator.
PNCDesMode	Boolean	TRUE, FALSE	If TRUE, the DEV desires to be the PNC of a piconet.
PNCCapable	Boolean	TRUE, FALSE	If TRUE, the DEV is capable of being a PNC in a piconet.
MaxAssociatedDEVs	Integer	0-mMaxNumValidDEVs	The maximum number of DEVs that this DEV is able to handle as a PNC.
MaxCTRqBs	Integer	0–255	The maximum number of isochronous CTRqBs that this DEV is able to handle as a PNC.
MaxTXPower	Integer	As defined in 7.4.11	The maximum transmitter power for the DEV.
SupportedDataRates	As defined in 7.4.11.	As defined in 7.4.11.	A PHY dependent mapping of the optional data rates supported by a DEV.
PreferredFragmentSize	Integer	As defined in 7.4.11.	A PHY dependent mapping that indicates the maximum MAC frame size preferred to be received by the DEV when fragmentation is used.
ATP	Duration	As defined in 7.5.1.1.	As defined in 7.5.1.1.
PiconetServicesInquiry	Enumeration	REQUEST, NO_REQUEST	Requests that the PNC sends the services information about the piconet as described in 7.5.1.1.
NeighborPNCRequest	Boolean	TRUE, FALSE	If TRUE, indicates that the DEV is associating as a neighbor piconet and not as a member of the current piconet.
DEVID	Integer	Any valid DEVID as defined in 7.2.3.	If association is successful, the assigned ID for the DEV. Otherwise, the UnassocID, 7.2.3.
VendorSpecificIE	Octet string	As defined in 7.4.17.	As defined in 7.4.17.
AssociationStatus	Enumeration	ASSOCIATED, DISASSOCIATED	Indicates if the DEV is either newly associated or disassociated.
AssocTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.

Table 9—MLME-ASSOCIATE primitive parameters (*Continued*)

Name	Type	Valid range	Description
AlwaysAWAKE	Boolean	TRUE, FALSE	As defined in 7.4.11.
ListenToSource	Boolean	TRUE, FALSE	As defined in 7.4.11.
ListenToMulticast	Boolean	TRUE, FALSE	As defined in 7.4.11.
ReasonCode	Integer	As defined in 7.5.1.2.	Indicates the reason why the attempted association failed as indicated in the Association Response command or indicates that the association was successful.
ResultCode	Enumeration	COMPLETED, TIMEOUT	Indicates if the primitive completed successfully or timed out.

6.3.5.1 MLME-ASSOCIATE.request

This primitive is used to request an association with a specified PNC. The semantics of this primitive are:

```
MLME-ASSOCIATE.request      (
                            DEVAddress,
                            PSRC,
                            SEC,
                            PNCDesMode,
                            PNCCapable,
                            MaxAssociatedDEVs,
                            MaxCTRqBs,
                            MaxTXPower,
                            SupportedDataRates,
                            PreferredFragmentSize,
                            ATP,
                            PiconetServicesInquiry,
                            NeighborPNCRequest,
                            AlwaysAWAKE,
                            ListenToSource,
                            ListenToMulticast,
                            AssocTimeout
                            )
```

The primitive parameters are defined in Table 9.

6.3.5.1.1 When generated

This primitive is generated by the originating DME to initiate an association with a PNC.

6.3.5.1.2 Effect of receipt

When the originating DEV MLME receives this primitive from its DME via the MLME-SAP, it generates an Association Request command, as described in 7.5.1.1, which is sent to the PNC MLME.

6.3.5.2 MLME-ASSOCIATE.indication

This primitive is used to indicate a received Association Request command, as described in 7.5.1.1. The semantics of this primitive are:

```
MLME-ASSOCIATE.indication   (
                            OrigID,
                            DEVAddress,
                            PSRC,
                            SEC,
                            PNCDesMode,
                            PNCCapable,
                            MaxAssociatedDEVs,
                            MaxCTRqBs,
                            MaxTXPower,
                            SupportedDataRates,
                            PreferredFragmentSize,
                            ATP,
                            PiconetServicesInquiry,
                            NeighborPNCRequest,
                            AlwaysAWAKE,
                            ListenToSource,
                            ListenToMulticast
                            )
```

The primitive parameters are defined in Table 9.

6.3.5.2.1 When generated

This primitive is sent by the PNC MLME to its DME upon receiving an Association Request command, as described in 7.5.1.1, from an unassociated DEV.

6.3.5.2.2 Effect of receipt

When the PNC DME receives this primitive and the OrigID is the UnassocID, it will determine whether to accept or reject the unassociated DEV's request to associate, using an algorithm that is outside of the scope of this standard. The PNC DME will then send an MLME-ASSOCIATE.response, with appropriate parameter values, to its MLME via the MLME-SAP.

When the PNC DME receives this primitive and the OrigID is the DEVID just assigned to the associating DEV, the PNC DME is notified that the assigned DEVID has been received by the associating DEV. The PNC DME will not send an MLME-ASSOCIATE.response in this case.

6.3.5.3 MLME-ASSOCIATE.response

This primitive is used to initiate a response to an MLME-ASSOCIATE.indication. The semantics of this primitive are:

```
MLME-ASSOCIATE.response     (
                            DEVAddress,
                            DEVID,
                            ATP,
                            VendorSpecificIE,
                            ReasonCode
                            )
```

The primitive parameters are defined in Table 9.

6.3.5.3.1 When generated

This primitive is generated by the PNC DME upon receiving an MLME-ASSOCIATE.indication.

6.3.5.3.2 Effect of receipt

When the PNC MLME receives this primitive from its DME, it generates an Association Response command, as described in 7.5.1.2.

6.3.5.4 MLME-ASSOCIATE.confirm

This primitive is used to inform the originating DME whether its request to associate is successful or unsuccessful. The semantics of this primitive are:

```
MLME-ASSOCIATE.confirm      (
                            DEVID,
                            ATP,
                            ReasonCode,
                            VendorSpecificIE,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 9.

6.3.5.4.1 When generated

This primitive is sent from the MLME to the DME either after the completion of the association process described in 8.3.1 or the timeout has occurred.

6.3.5.4.2 Effect of receipt

The originating DEV upon receiving this primitive is notified whether its request to associate with the PNC is successful or unsuccessful. If successful, the originating DME is provided with a unique DEVID. If unsuccessful, the originating DME is not provided with a valid DEVID and consequently remains unassociated.

6.3.5.5 MLME-DEV-ASSOCIATION-INFO.indication

This primitive is used to indicate to other associated DEVs the reception of a beacon containing a DEV Association IE, as described in 7.4.4. The semantics of this primitive are:

```
MLME-DEV-ASSOCIATION-INFO.indication
                                    (
                                    DEVAddress,
                                    DEVID,
                                    SupportedDataRates,
                                    AssociationStatus
                                    )
```

The primitive parameters are defined in Table 9.

6.3.5.5.1 When generated

This primitive is sent by the MLME to its DME upon receiving from the PNC a beacon containing a DEV Association IE, as described in 7.4.4. If the DEV Association IE contains more than one DEV Association Info field, this primitive is sent once for each DEV Association Info field contained in the received DEV Association IE.

6.3.5.5.2 Effect of receipt

The DME upon receiving this primitive is provided with information about the DEV that has either just completed the association or disassociation procedure.

6.3.6 Disassociation

The following primitives are used when a DEV disassociates from a PNC and when the PNC disassociates a DEV from the piconet, as described in 8.3.4. The parameters used for these primitives are defined in Table 10.

Table 10—MLME-DISASSOCIATE primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the originator of the MLME request.
DisassocTimeout	Duration	0-65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ReasonCode	Integer	As defined in 7.5.1.3	Indicates the reason the Disassociation Request command was issued.
ResultCode	Enumeration	SUCCESS, ACK_TIMEOUT	Indicates the result of the MLME request.

6.3.6.1 MLME-DISASSOCIATE.request

This primitive is used to initiate a disassociation request. The semantics of this primitive are:

```
MLME-DISASSOCIATE.request   (
                            TrgtID,
                            ReasonCode,
                            DisassocTimeout
                            )
```

The primitive parameters are defined in Table 10.

6.3.6.1.1 When generated

This primitive is sent by the DME to its MLME to initiate the disassociation process.

6.3.6.1.2 Effect of receipt

In the case where a DEV MLME receives this primitive, the DEV MLME sends a directed Disassociation Request command, as described in 7.5.1.3, to the PNC MLME.

In the case where the PNC MLME receives this primitive, the PNC MLME sends a directed Disassociation Request command to the DEV MLME to be disassociated.

6.3.6.2 MLME-DISASSOCIATE.indication

This primitive is used to indicate the reception of a Disassociation Request command, as described in 7.5.1.3. The semantics of this primitive are:

```
MLME-DISASSOCIATE.indication   (
                               OrigID,
                               ReasonCode
                               )
```

The primitive parameters are defined in Table 10.

6.3.6.2.1 When generated

This primitive is sent by the MLME to its DME upon receiving a Disassociation Request command, as described in 7.5.1.3, from either a PNC or a DEV.

6.3.6.2.2 Effect of receipt

The target DME is notified of the reason for the disassociation request.

6.3.6.3 MLME-DISASSOCIATE.confirm

This primitive reports the results of a disassociation request. The PNC DME, when it receives the MLME-DISASSOCIATE.confirm primitive, is notified as to which DEV has been disassociated. The semantics of this primitive are:

```
MLME-DISASSOCIATE.confirm    (
                             TrgtID,
                             ResultCode
                             )
```

The primitive parameters are defined in Table 10.

6.3.6.3.1 When generated

This primitive is sent by the originating MLME to its DME after sending a Disassociation Request command, as described in 7.5.1.3, and receiving either an ACK or an ACK_TIMEOUT. The disassociation procedure is considered successful if an ACK is received by the originating MLME. The disassociation procedure is considered unsuccessful if an ACK_TIMEOUT is received by the originating MLME.

6.3.6.3.2 Effect of receipt

The originating DME, when it receives the MLME-DISASSOCIATE.confirm primitive, is notified of the result of the disassociation procedure.

6.3.7 Key request

This mechanism supports the process of a DEV requesting and receiving a key from the key originator. The parameters used for these primitives are defined in Table 11.

Table 11—MLME-REQUEST-KEY primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
SECID	2 octets	As defined in 7.2.7.2.	As defined in 7.2.7.2.
Key	Octet string	Any valid key as defined by the symmetric key security operations, 10.3	The key to be used as the current payload protection key for this security relationship. The MAC/MLME encrypts the key before it is placed in the Encrypted Key field and decrypts the field before passing the received key to the DME.
KeyRequestTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, FAILURE, TIMEOUT	Indicates the result of the MLME request.

6.3.7.1 MLME-REQUEST-KEY.request

This primitive is used by a DEV to request the transmission of a key from the key originator. The semantics of this primitive are:

```
MLME-REQUEST-KEY.request    (
                            TrgtID,
                            KeyRequestTimeout
                            )
```

The primitive parameters are defined in Table 11.

6.3.7.1.1 When generated

This primitive is generated by the DME for a DEV to obtain the designated key from the key originator.

6.3.7.1.2 Effect of receipt

The MLME creates a Request Key command, 7.5.2.1, and sends it to the indicated DEV.

6.3.7.2 MLME-REQUEST-KEY.indication

This primitive reports the request of a key from a DEV. The semantics of this primitive are:

```
MLME-REQUEST-KEY.indication    (
                               OrigID,
                               TrgtID,
                               ResultCode
                               )
```

The primitive parameters are defined in Table 11.

6.3.7.2.1 When generated

This primitive is generated by the MLME as a result of receiving a Request Key command, as described in 7.5.2.1. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS.

6.3.7.2.2 Effect of receipt

Upon receipt of the MLME-REQUEST-KEY.indication with the ResultCode set to SUCCESS, the DME issues an MLME-REQUEST-KEY.response to the MLME.

6.3.7.3 MLME-REQUEST-KEY.response

This primitive is used by a DEV to respond to a key request from a DEV with the requested key. The semantics of this primitive are:

```
MLME-REQUEST-KEY.response   (
                            OrigID,
                            SECID,
                            Key
                            )
```

The primitive parameters are defined in Table 11.

6.3.7.3.1 When generated

This primitive is generated by the DME as a result of the receipt of an MLME-REQUEST-KEY.indication primitive with ResultCode equal to SUCCESS where the OrigID corresponds to a DEV that has established secure membership or a secure relationship with the key originator.

6.3.7.3.2 Effect of receipt

The MLME generates a Request Key Response command, as described in 7.5.2.2, and sends it to the specified DEV. The MLME encrypts the key before transmission.

6.3.7.4 MLME-REQUEST-KEY.confirm

This primitive reports the results of a key request and, if the response was received, the requested key to the DME. The semantics of this primitive are:

```
MLME-REQUEST-KEY.confirm    (
                            TrgtID,
                            SECID,
                            Key,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 11.

6.3.7.4.1 When generated

This primitive is generated as a result of the MLME receiving a Request Key Response command, as described in 7.5.2.2, from the key originator of this relationship or due to a timeout. If there is no response from the key originator within KeyRequestTimeout, the ResultCode is set to TIMEOUT. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS and the MLME decrypts the key.

6.3.7.4.2 Effect of receipt

The DME is informed of the results of a previously issued key request and, if successful, obtains the requested key.

6.3.8 Key distribution

This mechanism supports a DEV acting as key originator sending a key to another DEV. The parameters used for these primitives are defined in Table 12.

Table 12—MLME-DISTRIBUTE-KEY primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
SECID	2 octets	As defined in 7.2.7.2.	As defined in 7.2.7.2.
Key	Octet string	Any valid key as defined by the symmetric key security operations, 10.3	The key to be used as the current payload protection key for this security relationship. The MAC/MLME encrypts the key before it is placed in the Encrypted Key field and decrypts the field before passing the received key to the DME.
DistributeKeyTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, FAILURE, TIMEOUT	Indicates the result of the MLME request.

6.3.8.1 MLME-DISTRIBUTE-KEY.request

This primitive is used by a DEV to distribute a key to another DEV. The semantics of this primitive are:

```
MLME-DISTRIBUTE-KEY.request     (
                                TrgtID,
                                SECID,
                                Key,
                                DistributeKeyTimeout
                                )
```

The primitive parameters are defined in Table 12.

6.3.8.1.1 When generated

This primitive is generated by the DME for a DEV to distribute a key to a DEV that has established secure membership or a secure relationship with the key originator.

6.3.8.1.2 Effect of receipt

The MLME creates a Distribute Key Request command, as described in 7.5.2.3, and sends it to the indicated DEV. The MLME encrypts the key before transmission.

6.3.8.2 MLME-DISTRIBUTE-KEY.indication

This primitive reports the reception of a key from a DEV in a key originator role. The semantics of this primitive are:

```
MLME-DISTRIBUTE-KEY.indication  (
                                OrigID,
                                SECID,
                                Key,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 12.

6.3.8.2.1 When generated

This primitive is generated by the MLME as a result of receiving a Distribute Key Request command, as described in 7.5.2.3. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS and the MLME decrypts the key.

6.3.8.2.2 Effect of receipt

If the ResultCode is SUCCESS, the DME will store the designated key and generate an MLME-DISTRIBUTE-KEY.response primitive to acknowledge successful receipt of the key.

6.3.8.3 MLME-DISTRIBUTE-KEY.response

This primitive is used by a DEV to respond to a key distribution from a DEV in the key originator role with an acknowledgement of successful receipt of the key. The semantics of this primitive are:

```
MLME-DISTRIBUTE-KEY.response  (
                              OrigID,
                              SECID
                              )
```

The primitive parameters are defined in Table 12.

6.3.8.3.1 When generated

This primitive is generated by the DME as a result of the receipt of an MLME-DISTRIBUTE-KEY.indication primitive from a peer DEV. It is not generated by the DME as a result of the receipt of an MLME-DISTRIBUTE-KEY.indcation primitive from the PNC.

6.3.8.3.2 Effect of receipt

The MLME generates a Distribute Key Response command, as described in 7.5.2.4, and sends it to the specified DEV.

6.3.8.4 MLME-DISTRIBUTE-KEY.confirm

This primitive reports the results of a distribute key process with a DEV. The semantics of this primitive are:

```
MLME-DISTRIBUTE-KEY.confirm     (
                                TrgtID,
                                SECID,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 12.

6.3.8.4.1 When generated

This primitive is generated as a result of the MLME receiving a Distribute Key Response command, as described in 7.5.2.4, from another DEV. If there is no response from the DEV within DistributeKeyTimeout, the ResultCode is set to TIMEOUT. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS.

6.3.8.4.2 Effect of receipt

The DME is informed of the result of a previously issued key distribution.

6.3.9 Security management

These primitives are used to initialize, update or delete the security information as a result of a membersip or key change process or as the result of a security event. Primitives are also provided to transfer security messages. These primitives are suitable for use in an authentication process.

The parameters used for the MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitives are defined in Table 13.

The parameters used for the MLME-SECURITY-MESSAGE primitive are defined in Table 14.

6.3.9.1 MLME-MEMBERSHIP-UPDATE.request

This primitive requests that the membership status, SECID and keying information associated with a security relationship be included or updated. The semantics of the primitive are as follows:

```
MLME-MEMBERSHIP-UPDATE.request(
                                TrgtID,
                                MembershipStatus,
                                SECID,
                                KeyType,
                                KeyOriginator,
                                KeyInfoLength
                                KeyInfo
                                )
```

The primitive parameters are defined in Table 13.

Table 13—MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitive parameters

Name	Type	Valid range	Description
SECID	2 octets	As defined in 7.2.7.2.	As defined in 7.2.7.2.
KeyType	Enumeration	MANAGEMENT, DATA	Specifies the type of key that is being updated, Clause 10.
TrgtID	Integer	Any valid DEVID as defined in 7.2.3 except for the BcstID, the McstID or the UnassocID.	The DEVID of the target DEV for this relationship.
MembershipStatus	Enumeration	MEMBER, NON-MEMBER	Indicates the membership status of the TrgtID for the provided SECID. If NON-MEMBER, KeyInfoLength is 0.
KeyOriginator	Boolean	TRUE, FALSE	This DEV is the key originator for this relationship.
KeyInfoLength	Integer	0 or 16	Length of KeyInfo.
KeyInfo	Dynamic	Any valid symmetric key as defined by the symmetric key security operations, 10.3	The key used for protecting frames between this DEV and the TrgtID DEV.
ReceivedMACHeader	Octet string	Any valid MAC header, Figure 6.	The MAC header of the received frame that induced a failed security check or for which the DEV is unable to find the designated key.
ReceivedFramePayload	Octet string	Any valid Frame Payload, Figure 8	The received Frame Payload that induces a failed security check or for which the DEV is unable to find the designated key.
ReasonCode	Enumeration	UNAVAILABLE-KEY, FAILED-SECURITY-CHECK, BAD-TIME-TOKEN	The reason for the security error.

Table 14—MLME-SECURITY-MESSAGE primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the originator of the MLME request.
VendorOUI	As defined in 7.4.7.	As defined in 7.4.7.	As defined in 7.4.7.
SecurityInformation	Octet string	Any valid octet string	Security information that will be passed from one DME to another peer DME in the piconet.
SecMsgTimeout	Integer	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ReasonCode	Enumeration	COMPLETED, TIMEOUT	Indicates if the the MAC/MLME was successful in sending the security message.

6.3.9.1.1 When generated

The DME sends this request to the MLME after completing a membership change or key update process with the PNC or a DEV.

6.3.9.1.2 Effect of receipt

This primitive initiates the membership update procedure defined in 9.3.4.

6.3.9.2 MLME-SECURITY-ERROR.indication

This primitive allows the MLME of any DEV to indicate a failed security processing operation to the DME. This semantics of this primitive are:

```
MLME-SECURITY-ERROR.indication (
                                    ReceivedMACHeader,
                                    ReceivedFramePayload
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 13.

6.3.9.2.1 When generated

This primitive is issued by the MLME when it receives an MLME.request message from a higher layer that requires security to be applied to a frame, but it is unable to find an appropriate key or fails to be able to apply security to the frame. This primitive is also issued by the MLME when it receives a validly formatted frame from another DEV that a fails the security check according to the symmetric key security operations or for which the DEV is unable to find the designated key. This primitive is also issued if the time token in the beacon was not within the range of valid time tokens, 9.3.6.

6.3.9.2.2 Effect on receipt

On receipt of this primitive, the DME is notified of a security error and the reason for the security error. When ReasonCode is BAD-TIME-TOKEN, the ReceivedMACHeader is the MAC header of the beacon frame and the ReceivedFramePayload is the payload of the beacon frame.

6.3.9.3 MLME-SECURITY-MESSAGE.request

This primitive sends a security message to a DEV in the piconet. The semantics of this primitive are:

```
MLME-SECURITY-MESSAGE.request(
                                    TrgtID,
                                    VendorOUI,
                                    SecurityInformation,
                                    SecMsgTimeout
                                    )
```

The primitive parameters are defined in Table 15.

6.3.9.3.1 When generated

This primitive is generated by the DME to send security related information to another DEV in the piconet.

6.3.9.3.2 Effect of receipt

The MLME creates an Security Message command, as described in 7.5.9.1, and sends it to the appropriate DEV.

6.3.9.4 MLME-SECURITY-MESSAGE.indication

This primitive reports the reception of an Security Message command, as described in 7.5.9.1 from a DEV. The semantics of this primitive are:

```
MLME-SECURITY-MESSAGE.indication (
                                    TrgtID,
                                    OrigID,
                                    VendorOUI,
                                    SecurityInformation
                                    )
```

The primitive parameters are defined in Table 15.

6.3.9.4.1 When generated

This primitive is generated by the MLME upon receiving a valid Security Message command from a DEV.

6.3.9.4.2 Effect of receipt

The DME recieves the security related information. The use of this information is outside of the scope of this standard.

6.3.9.5 MLME-SECURITY-MESSAGE.confirm

This primitive reports the result of an attempt send security information to another DEV. The semantics of this primitive are:

```
MLME-SECURITY-MESSAGE.confirm (
                                    ResultCode
                                    )
```

The primitive parameter is defined in Table 15.

6.3.9.5.1 When generated

This primitive is generated as a result of the MLME receiving an Imm-ACK for the Security Message command or due to a timeout. If the Imm-ACK is not received within SecMsgtTimeout, the ResultCode is TIMEOUT. Otherwise, the ResultCode is COMPLETED.

6.3.9.5.2 Effect of receipt

The DME is informed of whether the message was successfully sent and ACKed or not.

6.3.10 PNC handover

These primitives are used to handover the current PNC's responsibilities to another DEV selected as the most qualified DEV to perform the duties of a PNC from a list of alternate PNC capable DEVs. The parameters used for these primitives are defined in Table 15.

Table 15—MLME-PNC-HANDOVER and MLME-NEW-PNC primitive parameters

Name	Type	Valid range	Description
NewPNCDEVID	Integer	Any valid DEVID as defined in 7.2.3.	The DEVID of the DEV being requested to assume responsibilities as PNC.
NewPNCDEVAddress	MAC address	Any valid individual MAC address.	The DEV address of the DEV assuming responsibilities as PNC.
NumberOfDEVs	Integer	As defined in 7.2.3.	The number of DEVs in the piconet.
HndOvrBeaconNumber	Integer	0–65535	The beacon number of the superframe when the new PNC will take over as PNC for the piconet.
DEVInfoSet	A set of DEV Info fields as defined in 7.5.4.2	A set containing 3 to mMaxNumValidDEVs instances of fixed length DEV Info fields.	A set of DEV Info fields for all of the DEVs currently associated in the piconet.
HandoverTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
HandoverStatus	Enumeration	STARTED, CANCELLED	Indicates if the PNC is beginning or cancelling a handover to the DEV.
ReasonCode	Octet	As defined in 7.5.3.2.	Indicates if the new PNC is ready to begin handover or if it will be unable to accept the handover request, 7.5.3.2.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

6.3.10.1 MLME-PNC-HANDOVER.request

This primitive initiates a request to handover the current PNC's responsibilities to another DEV selected as the most qualified of the PNC capable DEVs in the piconet. The semantics of this primitive are:

```
MLME-PNC-HANDOVER.request   (
                            NewPNCDEVID,
                            NumberOfDEVs,
                            HndOvrBeaconNumber,
                            DEVInfoSet,
                            HandoverStatus,
                            HandoverTimeout
                            )
```

The primitive parameters are defined in Table 15.

6.3.10.1.1 When generated

The PNC's DME sends this primitive to its MLME to initiate a PNC handover or to cancel the PNC handover process.

6.3.10.1.2 Effect of receipt

The PNC's MLME, upon receiving this primitive, sends a directed PNC Handover Request command, as described in 7.5.3.1, to the specified PNC capable DEV.

6.3.10.2 MLME-PNC-HANDOVER.indication

This primitive indicates the reception of a directed PNC Handover Request command, as described in 7.5.3.1. The semantics of this primitive are:

```
MLME-PNC-HANDOVER.indication   (
                               NumberOfDEVs,
                               HandoverStatus
                               )
```

The primitive parameters are defined in Table 15.

6.3.10.2.1 When generated

The MLME, upon receiving a directed PNC Handover Request command, as described in 7.5.3.1, from the PNC, sends this primitive to its DME. The MLME also sends this primitive after it has sent the first beacon as the new PNC of the piconet.

6.3.10.2.2 Effect of receipt

If HandoverStatus is STARTED, then the DME is informed that the MLME has started the PNC handover process. If HandoverStatus is CANCELLED, then the DME is informed that the PNC handover process has been terminated.

6.3.10.3 MLME-PNC-HANDOVER.response

This primitive is used to initiate a response to an MLME-PNC-HANDOVER.indication. The semantics of this primitive are:

```
MLME-PNC-HANDOVER.response   (
                             ReasonCode
                             )
```

6.3.10.3.1 When generated

This primitive is sent by the new PNC's DME to its MLME after receiving these two primitives in succession, MLME-PNC-HANDOVER.indication with a HandoverStatus of STARTED, MLME-PNC-INFO.confirm and the DME is ready to take over as the new PNC of the piconet.

6.3.10.3.2 Effect of receipt

When the new PNC's MLME receives this primitive from its DME, it is informed that its DME is ready to become the new PNC of the piconet.

6.3.10.4 MLME-PNC-HANDOVER.confirm

This primitive informs the originating DME its request to initiate a PNC handover is complete. The semantics of this primitive are:

```
MLME-PNC-HANDOVER.confirm    (
                             ResultCode
                             )
```

The primitive parameters are defined in Table 15.

6.3.10.4.1 When generated

The PNC MLME sends this primitive to its DME with ResultCode set to SUCCESS after it has sent its last beacon frame in the handover procedure, as described in 8.2.3. If the PNC is not able to successfully transfer the piconet data to the chosen PNC capable DEV, it sends this primitive to the DME with ResultCode set to TIMEOUT.

6.3.10.4.2 Effect of receipt

The PNC DME, upon receiving this primitive, is informed whether its MLME-PNC-HANDOVER.request was successful or unsuccessful. The ResultCode is set to SUCCESS when the PNC sends its last beacon as PNC before the HandoverTimeout interval expires. The ResultCode is set to TIMEOUT if the PNC fails to receive the PNC Handover Response command, as described in 7.5.3.2, before the HandoverTimeout interval expires.

6.3.10.5 MLME-NEW-PNC.indication

This primitive indicates the reception of a PNC Handover IE, as described in 7.4.9, in a beacon. The semantics of this primitive are:

```
MLME-NEW-PNC.indication    (
                           NewPNCDEVID,
                           NewPNCDEVAddress
                           )
```

6.3.10.5.1 When generated

The MLME sends this primitive to its DME upon receiving a beacon containing a PNC Handover IE.

6.3.10.5.2 Effect of receipt

The DME is notified that a new PNC is taking over the piconet.

6.3.11 PNC requesting information

This mechanism supports the ability for a DEV to request information from the PNC about either a specific DEV or all of the DEVs in the piconet, as described in 8.9. The parameters used for these primitives are defined in Table 16.

Table 16—MLME-PNC-INFO primitive parameters

Name	Type	Valid range	Description
QueriedDEVID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV for which information is being requested from the PNC. A value of BcstID is defined as a request for information about all associated DEVs.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
NumDevInfoSet	Integer	3-mMaxNumValidDEVs	Number of entries in the DEVInfoSet.
DEVInfoSet	A set of DEV Info fields as defined 7.5.4.2	A set containing 3 to mMaxNumValidDEVs instances of fixed length DEV Info fields.	The DEVInfoSet is returned to indicate the results of a PNC Information Request command.
PNCInfoTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

6.3.11.1 MLME-PNC-INFO.request

This primitive initiates a request to the PNC for information regarding either a single DEV or all of the DEVs in the piconet. The semantics of this primitive are:

```
MLME-PNC-INFO.request       (
                            QueriedDEVID,
                            PNCInfoTimeout
                            )
```

The primitive parameters are defined in Table 16.

6.3.11.1.1 When generated

The originating DME sends this primitive to its MLME when it wants to obtain information about either an individual DEV or all of the DEVs in the piconet.

6.3.11.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends the PNC Information Request command, as described in 7.5.4.1, to the PNC to request information managed by the PNC.

6.3.11.2 MLME-PNC-INFO.indication

This primitive indicates the reception of a request by the PNC for information it manages regarding either a specific DEV or all of the DEVs in the piconet. The semantics of this primitive are:

```
MLME-PNC-INFO.indication    (
                            QueriedDEVID,
                            OrigID
                            )
```

The primitive parameters are defined in Table 16.

6.3.11.2.1 When generated

The PNC MLME sends this primitive to its DME upon receiving a PNC Information Request command, as described in 7.5.4.1, from the requesting DEV specified by the OrigID.

6.3.11.2.2 Effect of receipt

The PNC DME upon receiving this primitive sends an MLME-PNC-INFO.response to its MLME.

6.3.11.3 MLME-PNC-INFO.response

This primitive initiates a DME response to an MLME-PNC-INFO.indication. The semantics of this primitive are:

```
MLME-PNC-INFO.response    (
                          OrigID,
                          NumDEVInfoSet,
                          DEVInfoSet
                          )
```

The primitive parameters are defined in Table 16.

6.3.11.3.1 When generated

The PNC DME sends this primitive to its MLME as a result of receiving an MLME-PNC-INFO.indication.

6.3.11.3.2 Effect of receipt

The PNC MLME upon receiving this primitive sends a PNC Information command, as described in 7.5.4.2, to the requesting DEV.

6.3.11.4 MLME-PNC-INFO.confirm

This primitive informs the DME that the MLME has received a PNC Information command, as described in 7.5.4.2. The semantics of this primitive are:

```
MLME-PNC-INFO.confirm    (
                         NumDEVInfoSet,
                         DEVInfoSet,
                         ResultCode
                         )
```

The primitive parameters are defined in Table 16.

6.3.11.4.1 When generated

The MLME sends this primitive to its DME upon receiving either a PNC Information command, as described in 7.5.4.2, or a TIMEOUT.

6.3.11.4.2 Effect of receipt

The originating DME is informed whether its request for information about either a single DEV or all of the DEVs in the piconet was successful or unsuccessful. If unsuccessful, the DME is able to resend the MLME-PNC-INFO.request. If successful, the DME will have acquired the information it requested. If the PNC Information command, as described in 7.5.4.2, was received as an unsolicited frame then the DME is informed of the current information for all of the DEVs currently a member of the piconet.

6.3.12 Security information retrieval

These primitives are used to request security information about other DEVs in the piconet. The parameters used for the MLME-SECURITY-INFO primitives are defined in Table 17.

6.3.12.1 MLME-SECURITY-INFO.request

This primitive initiates a request to the DEV for security information regarding either a single DEV or all of the DEVs in the piconet. The semantics of the primitive are as follows:

```
MLME-SECURITY-INFO.request    (
                              TrgtID,
                              QueriedDEVID,
                              SecurityInfoTimeout
                              )
```

The primitive parameters are defined in Table 17.

6.3.12.1.1 When generated

The originating DME sends this primitive to its MLME when it wants to obtain security information about either an individual DEV or all of the DEVs in the piconet.

6.3.12.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends the Security Information Request command, as described in 7.5.4.3, to the DEV specified by the TrgtID to request security information managed by the DEV of TrgtID.

Table 17—MLME-SECURITY-INFO primitive parameters

Name	Type	Valid range	Description
QueriedDEVID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV for which information is being requested. A value of BcstID is defined as a request for information for all associated DEVs.
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV for which the security information request is intended.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
NumSecurity-RecordSet	Integer	0–65535	Number of entries in the SecurityRecord-Set.
SecurityRecordSet	A set of Security Record fields as defined in 7.5.4.4.	A set containing 0 or more instances of variable length Security Record field. The maximum number of instances depends on the size of the records, pMax-FrameBodySize and the length of the secure command security fields, 7.3.3.2.	The SecurityRecordSet is returned to indicate the results of an Security Information Request command.
SecurityInfoTime-out	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

6.3.12.2 MLME-SECURITY-INFO.indication

This primitive indicates the reception of a request by a DEV for security information it manages regarding either a specific DEV or all of the DEVs in the piconet. The semantics of the primitive are as follows:

```
MLME-SECURITY-INFO.indication   (
                                QueriedDEVID,
                                OrigID
                                )
```

The primitive parameters are defined in Table 17.

6.3.12.2.1 When generated

The DEV MLME sends this primitive to its DME upon receiving an Security Information Request command, as described in 7.5.4.3, from the requesting DEV specified by the OrigID.

6.3.12.2.2 Effect of receipt

The DME upon receiving this primitive sends an MLME-SECURITY-INFO.response to its MLME.

6.3.12.3 MLME-SECURITY-INFO.response

This primitive initiates a DME response to an MLME-SECURITY-INFO.indication. The semantics of the primitive are as follows:

```
MLME-SECURITY-INFO.response     (
                                OrigID,
                                NumSecurityRecordSet,
                                SecurityRecordSet
                                )
```

The primitive parameters are defined in Table 17.

6.3.12.3.1 When generated

The DME sends this primitive to its MLME as a result of receiving an MLME-SECURITY-INFO.indication.

6.3.12.3.2 Effect of receipt

The MLME upon receiving this primitive sends an Security Information command, as described in 7.5.4.4, to the requesting DEV.

6.3.12.4 MLME-SECURITY-INFO.confirm

This primitive informs the originating DME that its request for security information from the specified DEV is complete. The semantics of the primitive are as follows:

```
MLME-SECURITY-INFO.confirm      (
                                TrgtID,
                                NumSecurityRecordSet,
                                SecurityRecordSet,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 17.

6.3.12.4.1 When generated

The MLME sends this primitive to its DME upon receiving either an Security Information command, as described in 7.5.4.4, or a TIMEOUT.

6.3.12.4.2 Effect of receipt

The originating DME is informed whether its request for information about either a single DEV or all of the DEVs in the piconet was successful or unsuccessful. If unsuccessful, the DME is allowed to resend the MLME-SECURITY-INFO.request. If successful, the DME will have acquired the information it requested.

6.3.13 ASIE management

These primitives are used to request that the PNC adds an ASIE to the beacon and to report the reception of an ASIE in a beacon. The parameters used for these primitives are defined in Table 18.

Table 18—MLME-CREATE-ASIE and MLME-RECEIVE-ASIE primitive parameters

Name	Type	Valid range	Description
Cmd	Enumeration	NEW, TERMINATE	NEW requests that a new ASIE be placed in the beacon., TERMINATE causes a currently scheduled ASIE to no longer be placed in the beacon.
NumBeacons	Octet	0–255	Specifies the number of beacons in which the ASIE will appear. If NumBeacons is 0, then the ASIE is put in beacons until the DME requests its termination with the MLME-CREATE-ASIE primitive.
VendorOUI	As defined in 7.4.7.	As defined in 7.4.7.	As defined in 7.4.7.
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	As defined in 8.14.
ASIEIndex	Integer	Application specific.	Used to uniquely identify an ASIE.
MessageLength	Integer	As defined in 8.14.	The length of the ASIE message.
ASIEMessage	Octet string	Any valid octet string of length up to MessageLength	As defined in 8.14.
CreateASIETimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.

6.3.13.1 MLME-CREATE-ASIE.request

This primitive is used to request the initialization and broadcast of an ASIE in one or more beacons or to terminate the broadcast of an ASIE in a beacon. The semantics of this primitive are:

```
MLME-CREATE-ASIE.request    (
                            Cmd,
                            NumBeacons,
                            VendorOUI,
                            TrgtID,
                            ASIEIndex,
                            MessageLength,
                            ASIEMessage,
                            CreateASIETimeout
                            )
```

The primitive parameters are defined in Table 18.

6.3.13.1.1 When generated

This primitive is sent by the PNC DME to its MLME when it wants to create or terminate the broadcast of an ASIE in beacons. If Cmd is TERMINATE, the ASIEIndex is set to the value assigned in a previous MLME-CREATE-ASIE.request.

6.3.13.1.2 Effect of receipt

If Cmd is NEW, then the PNC MLME when it receives this primitive initializes the fields of the ASIE, prepares a new beacon containing this IE, and responds to the PNC DME with an MLME-CREATE-ASIE.confirm primitive with a ResultCode of SUCCESS. If the PNC MLME is unable to put the ASIE in the beacon, it responds with an MLME-CREATE-ASIE.confirm primitive with a ResultCode of FAILURE. If Cmd is TERMINATE, then the PNC MLME no longer sends the ASIE in the beacon and responds to the PNC DME with an MLME-CREATE-ASIE.confirm primitive with a ResultCode of SUCCESS.

6.3.13.2 MLME-CREATE-ASIE.confirm

This primitive is used to inform the PNC DME that its request to initiate or terminate an ASIE broadcast has been completed. The semantics of this primitive are:

```
MLME-CREATE-ASIE.confirm    (
                            VendorOUI,
                            ASIEIndex,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 18.

6.3.13.2.1 When generated

This primitive is sent by the PNC MLME to its DME upon completion of the action requested with an MLME-CREATE-ASIE.request primitive.

6.3.13.2.2 Effect of receipt

If the PNC DME had previously sent an MLME-CREAT-ASIE.request primitive with Cmd set to NEW, the PNC DME upon receiving this primitive from its MLME is informed that the result of its request to initiate an ASIE broadcast is successful, if the ResultCode is SUCCESS, or that the PNC MLME was unable to fit the ASIE into the beacon, if the ResultCode is FAILURE. If the request fails, the PNC DME is able to send the ASIE with an Announce command, as described in 7.5.5.2. It also is allowed try to send the ASIE in another beacon.

If the PNC DME had previously sent an MLME-CREAT-ASIE.request primitive with Cmd set to TERMINATE, the PNC DME upon receiving this primitive from its MLME is informed that the request to terminate the broadcast of an ASIE is successful.

6.3.13.3 MLME-RECEIVE-ASIE.indication

This primitive is used to indicate reception of an ASIE in a beacon. The semantics of this primitive are:

```
MLME-RECEIVE-ASIE.indication    (
                                VendorOUI,
                                MessageLength,
                                ASIEMessage
                                )
```

The primitive parameters are defined in Table 18.

6.3.13.3.1 When generated

This primitive is sent by the DEV MLME to its DME upon reception of a beacon containing an ASIE containing its DEVID, as described in 8.14.

6.3.13.3.2 Effect of receipt

The DEV DME is informed of the new ASIE in the beacon.

6.3.14 Peer information retrieval

The MLME-PROBE primitives are used to request information about other DEVs in the piconet. The parameters used for the MLME-PROBE primitives are defined in Table 19.

Table 19—MLME-PROBE primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
InformationRequested	4 octets	As defined in 7.5.4.5.	Indicates the IEs that are being requested as defined in 7.5.4.5.
RequestIndex	2 octets	As defined in 7.5.4.5.	As defined in 7.5.4.5.
IEsProvided	Variable number of octets.	As defined in 7.5.4.6.	The IEs in the Probe Response command, as defined in 7.5.4.6.
ProbeTimeout	Duration	0–65535	The time in milliseconds by which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

6.3.14.1 MLME-PROBE.request

This primitive initiates a request for a list of selected IEs from a target DEV. The semantics of this primitive are:

```
MLME-PROBE.request      (
                        TrgtID,
                        InformationRequested,
                        RequestIndex,
                        ProbeTimeout
                        )
```

The primitive parameters are defined in Table 19.

6.3.14.1.1 When generated

The originating DME sends this primitive to its MLME when it wants to request information from another DEV in the piconet.

6.3.14.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends a Probe Request command, as described in 7.5.4.5, to the target DEV specified by the TrgtID. The use of the Probe Request command is described in 7.5.4.5.

6.3.14.2 MLME-PROBE.indication

This primitive indicates the reception of a request for a list of selected IEs. The semantics of this primitive are:

```
MLME-PROBE.indication   (
                        OrigID,
                        InformationRequested,
                        RequestIndex
                        )
```

The primitive parameters are defined in Table 19.

6.3.14.2.1 When generated

This primitive is sent by the MLME to its DME upon receiving a Probe Request command, as described in 7.5.4.5.

6.3.14.2.2 Effect of receipt

The DME upon receiving this primitive sends an MLME-PROBE.response to its MLME.

6.3.14.3 MLME-PROBE.response

This primitive initiates a response to an MLME-PROBE.indication. The semantics of this primitive are:

```
MLME-PROBE.response     (
                        OrigID,
                        IEsProvided
                        )
```

The primitive parameters are defined in Table 19.

6.3.14.3.1 When generated

The DME sends this primitive to its MLME in response to an MLME-PROBE.indication.

6.3.14.3.2 Effect of receipt

The MLME upon receiving this primitive sends a Probe Response command, as described in 7.5.4.6, to the requesting DEV specified by the OrigID.

6.3.14.4 MLME-PROBE.confirm

This primitive informs the originating DME that its request for DEV IEs from a target DEV is complete. The semantics of this primitive are:

```
MLME-PROBE.confirm      (
                        TrgtID,
                        IEsProvided,
                        ResultCode
                        )
```

The primitive parameters are defined in Table 19.

6.3.14.4.1 When generated

The MLME sends this primitive to its DME either upon receiving a Probe Response command, 7.5.4.6, or due to a timeout.

6.3.14.4.2 Effect of receipt

The originating DME upon receiving this primitive is informed whether the request for the list of IEs from the target DEV was successful or unsuccessful.

6.3.15 Information announcement to peers

The MLME-ANNOUNCE primitives are used by a DEV to send information about itself to other DEVs in the piconet. The parameters used for the MLME-ANNOUNCE primitives are defined in Table 20- .

Table 20—MLME-ANNOUNCE primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
IEsProvided	Variable number of octets.	As defined in 7.5.5.2.	The IEs in the Announce command, as defined in 7.5.5.2.
AnnounceTimeout	Duration	0–65535	The time in milliseconds by which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of ACK_TIMEOUT.
ResultCode	Enumeration	SUCCESS, ACK_TIMEOUT	Indicates the result of the MLME request.

6.3.15.1 MLME-ANNOUNCE.request

This primitive initiates a request to send selected IEs to a target DEV. The semantics of this primitive are:

```
MLME-ANNOUNCE.request   (
                        TrgtID,
                        IEsProvided,
                        AnnounceTimeout
                        )
```

The primitive parameters are defined in Table 20.

6.3.15.1.1 When generated

The originating DME sends this primitive to its MLME when it wants to request to send information to another DEV in the piconet.

6.3.15.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends the Announce command, as described in 7.5.5.2, to the target DEV specified by the TrgtID. The use of the Announce command is described in 7.5.5.2.

6.3.15.2 MLME-ANNOUNCE.indication

This primitive indicates the reception of selected IEs from a DEV. The semantics of this primitive are:

```
MLME-ANNOUNCE.indication   (
                           OrigID,
                           IEsProvided
                           )
```

The primitive parameters are defined in Table 20.

6.3.15.2.1 When generated

This primitive is sent by the MLME to its DME upon receiving an Announce command, as described in 7.5.5.2.

6.3.15.2.2 Effect of receipt

The DME upon receiving this primitive obtains selected information from the originating DEV.

6.3.15.3 MLME-ANNOUNCE.confirm

This primitive informs the originating DME that its request to send IEs to a target DEV is complete. The semantics of this primitive are:

```
MLME-ANNOUNCE.confirm        (
                             TrgtID,
                             ResultCode
                             )
```

The primitive parameters are defined in Table 20.

6.3.15.3.1 When generated

This primitive is sent by the originating MLME to its DME after sending an Announce command, as described in 7.5.5.2, and either receiving an ACK or an ACK_TIMEOUT. The result code is set to SUCCESS if an ACK was received, and to ACK_TIMEOUT if successful reception of the command is never acknowledged by the TrgtID.

6.3.15.3.2 Effect of receipt

The originating DME upon receiving this primitive is informed whether the request to send IEs to the target DEV was successful or unsuccessful. If unsuccessful, the DME is able to resend the MLME-ANNOUNCE.request with the same list of IEs. If successful, the DME will have successfully sent the requested IEs to the target DEV and is able initiate another MLME-ANNOUNCE.request to either the same target DEV or a different target DEV.

6.3.16 Piconet services

These primitives are used to transfer information regarding the services offered by DEVs in a piconet. The parameters used for the MLME-PICONET-SERVICES primitives are defined in Table 21.

6.3.16.1 MLME-PICONET-SERVICES.indication

This primitive is used to indicate the that a DEV has requested the Piconet Services command, as described in 7.5.5.1, in its association process, as described in 8.3.1. The semantics of this primitive are:

```
MLME-PICONET-SERVICES.indication(
                             TrgtD,
                             )
```

The primitive parameters are defined in Table 21.

Table 21—MLME-PICONET-SERVICES primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME response.
NumPiconetServicesIEs	Integer	0-mMaxNumValidDEVs	Specifies the number of Piconet Services IEs provided in the PiconetServicesIESet
PiconetServicesIESet	A set of Piconet Services IE(s), 7.4.16	0 to mMaxNumValidDEVs Piconet Services IE(s)	The set of Piconet Services IE(s) provided.

6.3.16.1.1 When generated

This PNC MLME generates this primitive when it receives an Association Request command with the SrcID set to the newly assigned DEVID and the Piconet Services Inquiry bit set, as described in 8.3.2.

6.3.16.1.2 Effect of receipt

The PNC DME receives the requested list of piconet services.

6.3.16.2 MLME-PICONET-SERVICES.response

This primitive is used to send a Piconet Services command, as described in 7.5.5.1. The semantics of this primitive are:

```
MLME-PICONET-SERVICES.response(
                                    TrgtID,
                                    NumPiconetServicesIEs,
                                    PiconetServicesIESet
                                    )
```

The primitive parameters are defined in Table 21.

6.3.16.2.1 When generated

The PNC DME will send this primitve to its MLME if the PNC supports reporting the piconet services information, as described in 8.3.2. The primitive is sent in response to a request for piconet services from an associating DEV.

6.3.16.2.2 Effect of receipt

The PNC MLME upon receiving this primitive will send a Piconet Services command to the TrgtID.

6.3.16.3 MLME-PICONET-SERVICES.confirm

This primitive is used to indicate the reception of the Piconet Services command, as described in 7.5.5.1. The semantics of this primitive are:

```
MLME-PICONET-SERVICES.confirm (
                                NumPiconetServicesIEs,
                                PiconetServicesIESet
                                )
```

The primitive parameters are defined in Table 21.

6.3.16.3.1 When generated

This DEV MLME generates this primitive when it receives a Piconet Services command.

6.3.16.3.2 Effect of receipt

The DEV DME receives the requested Piconet Services IE.

6.3.17 Stream management

This mechanism supports the creation, modification, and termination of channel time for streams as well as controlling the reception of multicast streams. The parameters used for the MLME-CREATE-STREAM, MLME-MODIFY-STREAM and MLME-TERMINATE-STREAM primitives are defined in Table 22.

6.3.17.1 MLME-CREATE-STREAM.request

This primitive is used to request the channel time. The semantics of this primitive are:

```
MLME-CREATE-STREAM.request    (
                              TrgtID,
                              DSPSSetIndex,
                              StreamRequestID,
                              StreamIndex,
                              ACKPolicy,
                              Priority,
                              PMCTRqType,
                              CTAType,
                              CTARateType,
                              CTARateFactor,
                              CTRqTU,
                              MinNumTUs,
                              DesiredNumTUs,
                              RequestTimeout
                              )
```

The primitive parameters are defined in Table 22.

6.3.17.1.1 When generated

This primitive is generated by an originating DME to initiate a channel time negotiation between a DEV and its PNC. The purpose is to establish an isochronous data stream for the originating DEV to communicate with one DEV, a multicast group, or all DEVs (broadcast) in the piconet. If a multicast or broadcast stream

Table 22—MLME-CREATE-STREAM, MLME-MODIFY-STREAM, and MLME-TERMINATE-STREAM primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
RequestTimeout	Duration	0–65535	The time in milliseconds by which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
DSPSSetIndex	Integer	As defined in 7.5.6.1	The DSPS set with which this channel time request is associated.
StreamIndex	Integer	As defined in 7.2.5	The stream index to create, modify or terminate.
ACKPolicy	Enumeration	IMM_ACK, NO_ACK, DLY_ACK	Specifies the ACK policy for the stream.
CTRqTU	Duration	0–65535	Duration in microseconds of the time unit used for the request.
CTARateType	Enumeration	SUB_RATE, SUPER_RATE	The type of allocation requested, either sub-rate or super-rate, as defined in 7.5.6.1.
CTARateFactor	Integer	0–65535	The sub-rate or super-rate value requested, as defined in 7.5.6.1.
CTAType	Enumeration	DYNAMIC, PSUEDO_STATIC	Indicates if the request is for a dynamic or pseudo-static CTA.
PMCTRqType	Enumeration	ACTIVE, DSPS	Indicates the type of request being made as defined in 7.5.6.1.
Priority	Integer	0–7	As defined in A.1.2.1.
MinNumTUs	Integer	0–255	The minimum number of TUs per CTA required by the DEV to support its stream.
DesiredNumTUs	Integer	0–255	The desired number of TUs per CTA requested by a DEV to support its stream.
AvailableNumTUs	Integer	0–255	The number of TUs per CTA available to the requesting DEV.
StreamRequestID	Integer	0–255	A unique value created by the originating DME to correlate this primitive with the response primitive it receives from the PNC MLME.
ReasonCode	Integer	As defined in 7.5.6.2.	As defined in 7.5.6.2.
ResultCode	Enumeration	COMPLETED, TIMEOUT	Indicates if the request has received a response or timed out.

was opened with any other ACK-Policy than no-ACK, the MLME will not send a Channel Time Request command, as described in 7.5.6.1, to the PNC and will respond with MLME-CREATE-STREAM.confirm with the ResultCode set to ILLEGAL_ACK_POLICY.

6.3.17.1.2 Effect of receipt

When a DEV MLME receives this primitive from its DME via the MLME-SAP, it generates a Channel Time Request command, as described in 7.5.6.1, which is sent to its corresponding PNC MLME.

6.3.17.2 MLME-CREATE-STREAM.confirm

This primitive is used to confirm the acceptance or rejection of a request to allocate channel time. The semantics of this primitive are:

```
MLME-CREATE-STREAM.confirm    (
                              StreamRequestID,
                              StreamIndex,
                              AvailableNumTUs,
                              ReasonCode,
                              ResultCode
                              )
```

The primitive parameters are defined in Table 22.

6.3.17.2.1 When generated

This primitive is sent by the originating MLME to its DME upon receiving either:

— A TIMEOUT
— A Channel Time Response command, as described in 7.5.6.2, with the request rejected as indicated in 8.5.1.1.
— A Channel Time Response command with reason code equal to success and followed by a beacon containing a CTA for the requested stream.

6.3.17.2.2 Effect of receipt

The originating DME, when it receives the MLME-CREATE-STREAM.confirm primitive, is informed whether its stream request was successful or unsuccessful.

6.3.17.3 MLME-MODIFY-STREAM.request

This primitive is used to request a modification to an existing stream. The semantics of this primitive are:

```
MLME-MODIFY-STREAM.request  (
                            StreamIndex,
                            Priority,
                            PMCTRqType,
                            CTAType,
                            CTARateType,
                            CTARateFactor,
                            CTRqTU,
                            MinNumTUs,
                            DesiredNumTUs,
                            RequestTimeout
                            )
```

The primitive parameters are defined in Table 22.

6.3.17.3.1 When generated

This primitive is generated by an originating DME to request a modification to an existing stream.

6.3.17.3.2 Effect of receipt

When a DEV MLME receives this primitive from its DME, it will generate a Channel Time Request command, as described in 7.5.6.1, which it will send to the PNC MLME.

6.3.17.4 MLME-MODIFY-STREAM.confirm

This primitive is used to inform the originating DME whether the requested stream modification was successful or unsuccessful. The semantics of this primitive are:

```
MLME-MODIFY-STREAM.confirm  (
                            StreamIndex,
                            AvailableNumTUs,
                            ReasonCode,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 22.

6.3.17.4.1 When generated

The originating MLME sends this primitive to its DME upon receiving either:

— A TIMEOUT,
— A Channel Time Response command, as described in 7.5.6.2, if the modification was not successful as defined in 8.5.1.2.
— A Channel Time Response command with reason code set to SUCCESS.

6.3.17.4.2 Effect of receipt

The originating DME, when it receives the MLME-MODIFY-STREAM.confirm primitive, is informed whether its request to modify a stream was successful or unsuccessful.

6.3.17.5 MLME-TERMINATE-STREAM.request

This primitive is used to request the termination of a specific stream. The semantics of this primitive are:

```
MLME-TERMINATE-STREAM.request(
                                    StreamIndex,
                                    RequestTimeout
                                    )
```

The primitive parameters are defined in Table 22.

6.3.17.5.1 When generated

This primitive is generated by the DME to request the termination of an existing stream.

6.3.17.5.2 Effect of receipt

When a DEV MLME receives this primitive from its DME, it will send a Channel Time Request command, as described in 7.5.6.1, to the PNC MLME with the values specified in 8.5.1.3.

6.3.17.6 MLME-TERMINATE-STREAM.indication

This primitive is used to inform the DEV DME that a stream has been terminated.

```
MLME-TERMINATE-STREAM.indication(
                                    StreamIndex
                                    )
```

The primitive parameters are defined in Table 22.

6.3.17.6.1 When generated

This primitive is sent by the DEV's MLME to its DME upon receiving a beacon which contains a null CTA with the stream index. The primitive is also be sent by the DEV's MLME to its DME if the MLME has determined that the CTA associated with that stream index has been absent from the beacon for a length of time that is implementation dependent.

6.3.17.6.2 Effect of receipt

The DME upon receiving this primitive is informed that the CTA associated with the indicated StreamIndex has been terminated.

6.3.17.7 MLME-TERMINATE-STREAM.confirm

This primitive is used to inform the originating DME whether the requested stream termination was successful or unsuccessful. The semantics of this primitive are:

```
MLME-TERMINATE-STREAM.confirm(
                                    StreamIndex,
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 22.

6.3.17.7.1 When generated

The originating DEV MLME sends this primitive to its DME either after the DEV MLME has received an ACK to its Channel Time Request command, as described in 7.5.6.1, for terminating a stream or the RequestTimeout has expired.

6.3.17.7.2 Effect of receipt

The originating DME, when it receives this primitive, is notified of the result of its stream termination request.

6.3.17.8 MLME-MULTICAST-RX-SETUP.request

This primitive allows the DME to control multicast reception and to allow filtering for a particular multicast stream.

6.3.17.8.1 Semantics of the service primitive

The semantics of this primitive are:

```
MLME-MULTICAST-RX-SETUP.request(
                                    MulticastStatus,
                                    SrcID,
                                    StreamIndex
                                    )
```

The primitive parameters are defined in Table 23.

Table 23—MLME-MULTICAST-RX-SETUP.request parameters

Name	Type	Valid range	Description
MulticastStatus	Enumeration	ENABLE, DISABLE, ALL, NONE	If ENABLE or DISABLE, indicates whether the MAC will pass multicast traffic defined by the stream index to the FCSL. If ALL, then all multicast traffic is passed to the FCSL. If NONE then no multicast traffic will be passed to the FCSL.
SrcID	Integer	Any valid DEVID, 7.2.3	The DEVID of the source of a multicast stream.
StreamIndex	Integer	As defined in 7.2.5	The stream index of a multicast stream.

6.3.17.8.2 When generated

This primitive is sent by the DME to the MLME to control the multicast receive parameters.

6.3.17.8.3 Effect on receipt

If MulticastStatus is ENABLE, the DEV will receive multicast frames with the SrcID and StreamIndex specified in the primitive parameters. If MulticastStatus is DISABLE, the DEV will disable reception of multicast frames with the SrcID and StreamIndex specified in the primitive parameters. If the MulticastStatus is ALL then the DEV will pass all multicast frames regardless of StreamIndex to the upper layers. If MulticastStatus is NONE all multicast reception is disabled.

6.3.18 Channel status request

These primitives provide a means of checking the status of a specific communications channel. The parameters used for these primitives are defined in Table 24.

Table 24—MLME-CHANNEL-STATUS primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that initiated the MLME request.
MeasurementWindowSize	Integer	0–65535	As defined in 7.5.7.2.
TXFrameCount	Integer	0–65535	As defined in 7.5.7.2.
RXFrameCount	Integer	0–65535	As defined in 7.5.7.2.
RXFrameErrorCount	Integer	0–65535	As defined in 7.5.7.2.
RXFrameLossCount	Integer	0–65535	As defined in 7.5.7.2.
ChannelStatusTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

6.3.18.1 MLME-CHANNEL-STATUS.request

This primitive initiates checking the status of the channel between two DEVs in the same piconet. The semantics of this primitive are:

```
MLME-CHANNEL-STATUS.request   (
                              TrgtID,
                              ChannelStatusTimeout
                              )
```

The primitive parameters are defined in Table 24.

6.3.18.1.1 When generated

This primitive is sent by the originating DME to its MLME to request the status of the channel between the originating DEV and the DEV specified in the TrgtID.

6.3.18.1.2 Effect of receipt

The MLME upon receiving this primitive from its DME will send a Channel Status Request command, as described in 7.5.7.1, to the DEV specified in the TrgtID.

6.3.18.2 MLME-CHANNEL-STATUS.indication

This primitive indicates the reception of a Channel Status Request command, as described in 7.5.7.1. The semantics of this primitive are:

```
MLME-CHANNEL-STATUS.indication (
                               OrigID
                               )
```

The primitive parameter is defined in Table 24.

6.3.18.2.1 When generated

The MLME sends this primitive to its DME upon receiving a Channel Status Request command, as described in 7.5.7.1, from the originating DEV.

6.3.18.2.2 Effect of receipt

The target DME upon receiving an MLME-CHANNEL-STATUS.indication, sends its MLME an MLME-CHANNEL-STATUS.response.

6.3.18.3 MLME-CHANNEL-STATUS.response

This primitive initiates the DME response to an MLME-CHANNEL-STATUS.indication. The semantics of this primitive are:

```
MLME-CHANNEL-STATUS.response (
                                    OrigID,
                                    MeasurementWindowSize,
                                    TXFrameCount,
                                    RXFrameCount,
                                    RXFrameErrorCount,
                                    RXFrameLossCount
                                    )
```

The primitive parameters are defined in Table 24.

6.3.18.3.1 When generated

This primitive is sent by the target DME to its MLME upon receiving an MLME-CHANNEL-STATUS.indication.

6.3.18.3.2 Effect of receipt

The MLME generates a Channel Status Response command, as described in 7.5.7.2, and sends it to the DEV specified in the OrigID.

6.3.18.4 MLME-CHANNEL-STATUS.confirm

This primitive informs the originating DME that its request for channel status is complete. The semantics of this primitive are:

```
MLME-CHANNEL-STATUS.confirm   (
                                    TrgtID,
                                    MeasurementWindowSize,
                                    TXFrameCount,
                                    RXFrameCount,
                                    RXFrameErrorCount,
                                    RXFrameLossCount,
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 24.

6.3.18.4.1 When generated

The MLME sends this primitive to its DME after receiving either a Channel Status Response command, as described in 7.5.7.2, or a TIMEOUT.

6.3.18.4.2 Effect of receipt

The originating DME is informed that its channel status request is either successful or unsuccessful.

6.3.19 Remote scan

These primitives are used by the PNC to request that a target DEV initiate a channel scan and to have the target DEV report the results of the channel scan to the PNC. The parameters used for these primitives are defined in Table 25.

Table 25—MLME-REMOTE-SCAN primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3	Specifies the DEVID of the target DEV.
ChannelList	As defined in Table 5	As defined in Table 5	Specifies a list of channels to be examined when scanning for piconets or interference.
RemoteScanTimeout	Duration	0–65535	The time in milliseconds during which the PNC expects to receive a response to its request.
NumberOfPiconets	Integer	0–255	The number of piconets found during the scanning process.
RemotePiconetDescriptionSet	Set of remote piconet descriptions as defined in Table 26	A set containing zero or more instances of a RemotePiconetDescription.	The PiconetDescriptionSet is returned to indicate the results of the remote scan request.
NumberOfChannels	As defined in Table 5	As defined in Table 5	The number of channels scanned.
ChannelRatingList	As defined in Table 5	As defined in Table 5	Specifies a list of channels from the best to the worst in terms of interference.
ReasonCode	Integer	As defined in 7.5.7.4	Indicates wether the scan was denied or was successful, as defined in 7.5.7.4.
ResultCode	Enumeration	COMPLETED, TIMEOUT	Indicates if the remote scan request has received a response or timed out.

6.3.19.1 MLME-REMOTE-SCAN.request

This primitive is used by the PNC to request that a DEV perform a channel scan. The semantics of this primitive are:

```
MLME-REMOTE-SCAN.request    (
                            TrgtID,
                            ChannelList,
                            RemoteScanTimeout
                            )
```

The primitive parameters are defined in Table 25.

Table 26—Elements of RemotePiconetDescription

Name	Type	Valid range	Description
BSID	Octet string	As defined in 7.4.2	The text string of a discovered piconet.
PNCDEVAddress	MAC address	Any valid individual MAC address	The MAC address of the PNC of the piconet that was found.
PNID	Integer	0-65535	The PNID of a discovered piconet.
PiconetType	Enumeration	DEPENDENT, NON-DEPENDENT	The type of a discovered piconet.
ParentBSID	Octet string	As defined in 7.4.3.	The BSID of the parent piconet if a beacon of a dependent piconet was found.
ParentPNCDEVAddress	MAC address	Any valid individual MAC address.	The MAC address of the parent PNC of the piconet that was found.
ScannedFrameType	Enumeration	BEACON, NON-BEACON	Indicates the type of frame that was found.
ChannelIndex	Integer	0-255	A PHY dependent channel as defined in 7.5.7.4.

6.3.19.1.1 When generated

The PNC DME sends this primitive to its MLME to request a remote channel scan by the DEV specified by the TrgtID.

6.3.19.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends a Remote Scan Request command, as described in 7.5.7.3, to the DEV specified by the TrgtID.

6.3.19.2 MLME-REMOTE-SCAN.indication

This primitive indicates the reception of a Remote Scan Request command, as described in 7.5.7.3, from the PNC. The semantics of this primitive are:

```
MLME-REMOTE-SCAN.indication     (
                                ChannelList,
                                )
```

The primitive parameters are defined in Table 25.

6.3.19.2.1 When generated

The target DEV MLME sends this primitive to its DME upon receiving a Remote Scan Request command, as described in 7.5.7.3, frame from the PNC.

6.3.19.2.2 Effect of receipt

The DME upon receiving this primitive sends either an MLME-SCAN.request to its MLME to initiate the requested channel scan or an MLME-REMOTE-SCAN.response with the ReasonCode indicating that the request for a remote scan was denied, as described in 7.5.7.4.

6.3.19.3 MLME-REMOTE-SCAN.response

This primitive is the DME response to an MLME-REMOTE-SCAN.indication. The semantics of this primitive are:

```
MLME-REMOTE-SCAN.response   (
                            ReasonCode,
                            NumberOfChannels,
                            ChannelRatingList,
                            NumberOfPiconets,
                            RemotePiconetDescriptionSet
                            )
```

The primitive parameters are defined in Table 25.

6.3.19.3.1 When generated

The DME sends this primitive to its MLME, after either refusing the MLME-REMOTE-SCAN.request or accepting the request by initiating an MLME-SCAN.request and receiving a subsequent MLME-SCAN.confirm.

6.3.19.3.2 Effect of receipt

The MLME upon receiving this primitive sends a Remote Scan Response command, as described in 7.5.7.4, to the PNC.

6.3.19.4 MLME-REMOTE-SCAN.confirm

This primitive informs the PNC DME that its request for the target DEV to perform a channel scan and report its results to the PNC is complete. The semantics of this primitive are:

```
MLME-REMOTE-SCAN.confirm   (
                            TrgtID,
                            ReasonCode,
                            NumberOfChannels,
                            ChannelRatingList,
                            NumberOfPiconets,
                            RemotePiconetDescriptionSet
                            ResultCode
                            )
```

The primitive parameters are defined in Table 25.

6.3.19.4.1 When generated

The MLME sends this primitive to its DME upon receiving either a Remote Scan Response command, as described in 7.5.7.4, or a TIMEOUT if the response command was not received before the RemoteScanTmeout expired.

6.3.19.4.2 Effect of receipt

The originating PNC DME is informed whether its request for a channel scan performed by a target DEV is successful or unsuccessful. If unsuccessful, the PNC DME has the choice of either resending the request to

the same target DEV when the ResultCode is TIMEOUT or to another target DEV when the ReasonCode indicates that the remote scan request was denied, as described in 7.5.7.4.

6.3.20 Piconet parameter change

This set of primitives allows the PNC to change certain characteristics of the piconet. The parameters used for these primitives are defined in Table 27.

Table 27— MLME-PICONET-PARM-CHANGE primitive parameters

Name	Type	Valid range	Description
NewChannelIndex	Integer	0–255	Index of the new PHY channel.
ChangeBeaconNumber	Integer	0–65535	The beacon number of the superframe when the new piconet parameter will take effect.
SuperframeTiming	Duration	0–65535	The change of superframe duration or beacon position in milliseconds.
ChangeType	Enumeration	CHANNEL, MOVE, SIZE, PNID, BSID, POWER	Indicates the parameter of the piconet that is changing as defined in 7.4.2.
PNID	Integer	0–65535	The ID of the piconet.
BSID	Octet string	As defined in 7.4.2.	A text string that identifies the piconet.
MaxTXPowerLevel	As defined in 7.3.1	As defined in 7.3.1	Maximum TX power allowed in the piconet.
ResultCode	Enumeration	SUCCESS, INVALID_PARMETERS	Indicates the result of the MLME request.

6.3.20.1 MLME-PICONET-PARM-CHANGE.request

This primitive initiates changing the PHY channel, the superframe duration, the beacon position, the maximum piconet TX power, the PNID, or the BSID. The semantics of this primitive are:

```
MLME-PICONET-PARM-CHANGE.request
                    (
                    ChangeType,
                    ChangeBeaconNumber,
                    NewChannelIndex,
                    SuperframeTiming,
                    PNID,
                    BSID,
                    MaxTXPowerLevel
                    )
```

The primitive parameters are defined in Table 27.

6.3.20.1.1 When generated

The PNC DME sends this primitive to its MLME after the PNC DME determines that it wishes to change one of the piconet parameters.

6.3.20.1.2 Effect of receipt

The PNC MLME, upon receiving this primitive with the ChangeType set to POWER, updates the Max TX Power Level field of the beacon's Piconet Synchronization Parameters field, as described in 7.3.1, with the MaxTXPowerLevel received. Otherwise, the PNC MLME takes the appropriate action as defined by the ChangeType parameter, as described in 7.4.6, 8.10, and 8.11.1.

6.3.20.2 MLME-PICONET-PARM-CHANGE.confirm

This primitive confirms that the MLME-PICONET-PARM-CHANGE.request has been fulfilled. The semantics of this primitive are:

```
MLME-PICONET-PARM-CHANGE.confirm
                                    (
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 27.

6.3.20.2.1 When generated

The PNC MLME sends this primitive to its DME when the PNC has broadcast the first beacon with the change in effect.

6.3.20.2.2 Effect of receipt

The PNC DME upon receiving this primitive is notified that the piconet parameter change is complete.

6.3.21 Power change

This mechanism supports the process of requesting a peer DEV to either increase or decrease its transmit power. The parameters used for these primitives are defined in Table 28.

Table 28—MLME-TX-POWER-CHANGE primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
TXPowerChange	Integer	–127 to +127	The requested power change in units of dB.
TXPowerChangeTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	Enumeration	SUCCESS, ACK_TIMEOUT	Indicates the result of the MLME request.

6.3.21.1 MLME-TX-POWER-CHANGE.request

This primitive initiates a request directed to a peer DEV to either increase or decrease its transmit power. The semantics of this primitive are:

```
MLME-TX-POWER-CHANGE.request
                    (
                    TrgtID,
                    TXPowerChange,
                    TXPowerChangeTimeout
                    )
```

The primitive parameters are defined in Table 28.

6.3.21.1.1 When generated

This primitive is sent by the originating DME to its MLME when it wants to request a change in the transmitter power of the DEV indicated in the TrgtID.

6.3.21.1.2 Effect of receipt

The MLME, upon receiving this primitive, sends a directed Transmit Power Change command, as described in 7.5.7.5, to a peer DEV with which it is currently communicating.

6.3.21.2 MLME-TX-POWER-CHANGE.indication

This primitive indicates the reception of a Transmit Power Change command, as described in 7.5.7.5. The semantics of this primitive are:

```
MLME-TX-POWER-CHANGE.indication
                    (
                    TXPowerChange
                    )
```

The primitive parameter is defined in Table 28.

6.3.21.2.1 When generated

The MLME, upon receiving a directed Transmit Power Change command, as described in 7.5.7.5, from a peer DEV, sends this primitive to its DME.

6.3.21.2.2 Effect of receipt

The DME, upon receiving this primitive, will either honor the request, ignore the request, or attempt a best effort change in its transmit power. See 8.11.2.2.

6.3.21.3 MLME-TX-POWER-CHANGE.confirm

This primitive informs the originating DME that its request for a transmit power change is complete. The semantics of this primitive are:

```
MLME-TX-POWER-CHANGE.confirm
                                    (
                                    ResultCode
                                    )
```

The primitive parameter is defined in Table 28.

6.3.21.3.1 When generated

The MLME sends this primitive to its DME after receiving either an ACK or an ACK_TIMEOUT.

6.3.21.3.2 Effect of receipt

The originating DME is informed that its request for a transmit power change is either successful or unsuccessful. If unsuccessful, i.e. the ACK was not received before the TXPowerChangeTimeout interval expired, the ResultCode is ACK_TIMEOUT and the DME is allowed to resend the MLME-TX-POWER-CHANGE.request. If successful, i.e. the ACK was received before the TXPowerChangeTimeout interval expired, the DME is notified that the TX-POWER-CHANGE.request has been executed and the ResultCode is SUCCESS.

6.3.22 Power management operation

This mechanism supports the process of establishment and maintenance of PM modes of a DEV. The parameters used for these primitives are defined in Table 29.

A PSSetStructureSet is an array of PSSetStructure. Each PSSetStructure consists of the elements shown in Table 30.

6.3.22.1 MLME-PS-SET-INFORMATION.request

This primitive requests the PS set information from the PNC. The semantics of this primitive are:

```
MLME-PS-SET-INFORMATION.request (
                                    PSRequestTimeout
                                    )
```

The primitive parameter is defined in Table 29.

6.3.22.1.1 When generated

This primitive is generated when a DME wants to determine the PS set configuration. It is usually generated before an MLME-SPS-CONFIGURE.request primitive.

6.3.22.1.2 Effect of receipt

When received, the local MLME sends a PS Set Information Request command, as described in 7.5.8.1, to the PNC.

Table 29—MLME-PS-SET-INFORMATION, MLME-SPS-CONFIGURE, MLME-PM-MODE-CHANGE, and MLME-PM-MODE-ACTIVE primitive parameters

Name	Type	Valid range	Description
PMMode	Enumeration	ACTIVE, APS, SPS	The PM mode requested by the DEV DME, as described in 7.5.8.5.
PSRequestTimeout	Duration	0–65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
OperationType	Enumeration	JOIN, LEAVE	The SPS set operation requested, 7.5.8.3.
SPSSetIndex	Integer	As defined in 7.5.8.3.	As defined in 7.5.8.3.
WakeBeaconInterval	Integer	As defined in 7.5.8.3.	As defined in 7.5.8.3 and 8.13.
NextWakeBeacon	Integer	As defined in 7.5.8.4.	The superframe number corresponding to the immediate next wake beacon of the DEVs in this PS set, 8.13.
MaxSupportedPSSets	Integer	As defined in 7.5.8.2 and 8.13	The total number of PS sets currently supported by the PNC of this piconet.
NumCurrentPSSets	Integer	As defined in 7.5.8.2 and 8.13	Indicates the number of currently active PS sets in the piconet.
PSSetStructureSet	Set of PSSetStructure	As defined in Table 30.	The PSSetStructureSet returns the information about the PS sets currently active in the PNC.
PMActiveEvent	Enumeration	DATA_PENDING, MAX_SLEEP	An event that causes the MLME to change the PM mode of operation to ACTIVE.
ReasonCode	Integer	As defined in 7.5.8.4.	As defined in 7.5.8.4.
ResultCode	Enumeration	SUCCESS, TIMEOUT	Indicates the result of the MLME request.

Table 30—Elements of PSSetStructure

Name	Type	Valid range	Description
PSSetIndex	Integer	As defined in 7.5.8.2.	As defined in 7.5.8.2.
WakeBeaconInterval	Integer	As defined in 7.5.8.2.	As defined in 7.5.8.2 and 8.13.
NextWakeBeacon	Integer	$0–(2^{32}–1)$	The superframe number corresponding to the immediate next wake beacon of the DEVs in this PS set, and 8.13.
BitmapLength	Integer	1–32	The number of octets in the DEVIDBitmap.
StartDEVID	Integer	Any valid DEVID as defined in 7.2.3.	As defined in 7.5.8.2.
DEVIDBitmap	1-32 octets	As defined in 7.5.8.2.	As defined in 7.5.8.2.

6.3.22.2 MLME-PS-SET-INFORMATION.confirm

This primitive is generated by the MLME to indicate completion of the MLME-PS-SET-INFORMATION.request. The semantics of this primitive are:

```
MLME-PS-SET-INFORMATION.confirm (
                                    MaxSupportedPSSets,
                                    NumCurrentPSSets,
                                    PSSetStructureSet,
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 29.

6.3.22.2.1 When generated

This primitive is generated by the MLME when it receives a PS Set Information Response command, as described in 7.5.8.2, from the PNC or if a timeout occurs.

6.3.22.2.2 Effect of receipt

This primitive provides a complete set of information to the DME on power management from the perspective of the PNC. This information would typically be used to determine the parameters for an MLME-SPS-CONFIGURE.request.

6.3.22.3 MLME-SPS-CONFIGURE.request

This primitive requests a change to the current SPS set information. Possible requests include; create a new DSPS set and add the current DEVID, add the current DEVID to an existing SPS set, or remove the current DEVID from an existing set, as described in 7.5.8.3. The semantics of this primitive are:

```
MLME-SPS-CONFIGURE.request
                                    (
                                    OperationType,
                                    SPSSetIndex,
                                    WakeBeaconInterval,
                                    PSRequestTimeout
                                    )
```

The primitive parameters are defined in Table 29.

6.3.22.3.1 When generated

This primitive is used to request a change to the SPS set information managed by the PNC with respect to the DEVID of the DEV making the request.

6.3.22.3.2 Effect of receipt

When the MLME receives this primitive from its DME, it sends an SPS Configuration Request command, as described in 7.5.8.3, to the PNC.

6.3.22.4 MLME-SPS-CONFIGURE.confirm

This primitive is generated by the MLME to indicate completion of the MLME-SPS-CONFIGURE.request. The semantics of this primitive are:

```
MLME-SPS-CONFIGURE.confirm
                                    (
                                    OperationType,
                                    ReasonCode,
                                    SPSSetIndex,
                                    NextWakeBeacon,
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 29.

6.3.22.4.1 When generated

This primitive is generated by the MLME when it receives an SPS Configuration Response command, as described in 7.5.8.4, from the PNC or if a timeout occurs.

6.3.22.4.2 Effect of receipt

The result of the requested SPS set configuration is reported to the DME.

6.3.22.5 MLME-PM-MODE-CHANGE.request

This primitive requests a change to the PM mode of operation. The semantics of this primitive are:

```
MLME-PM-MODE-CHANGE.request  (
                                    PMMode,
                                    PSRequestTimeout
                                    )
```

The primitive parameters are defined in Table 29.

6.3.22.5.1 When generated

This primitive is generated by the DME when it desires to change to the PM mode of operation.

6.3.22.5.2 Effect of receipt

Upon receipt of this primitive, the MLME will send a PM Mode Change command, as described in 7.5.8.5, to the PNC indicating the new PM mode. If the MAC is already in the ACTIVE state when the primitive is passed with the PMMode ACTIVE, the MLME will take no action before responding with an MLME-PM-MODE-CHANGE.confirm.

6.3.22.6 MLME-PM-MODE-CHANGE.confirm

This primitive is generated by the MLME to indicate completion of the MLME-PM-MODE-CHANGE.request. The semantics of this primitive are:

```
MLME-PM-MODE-CHANGE.confirm (
                            ResultCode
                            )
```

The primitive parameters are defined in Table 29.

6.3.22.6.1 When generated

This primitive is sent by the MLME with the ResultCode set to SUCCESS if it received an acknowledgement from the PNC on the PM Mode Change command, as described in 7.5.8.5, otherwise it is set to TIMEOUT.

6.3.22.6.2 Effect of receipt

The DME is informed about the result of the MLME-PM-MODE-CHANGE.request.

6.3.22.7 MLME-PM-MODE-ACTIVE.indication

This primitive is generated by the MLME to notify the DME that it has changed the PM mode of operation from APS or SPS to ACTIVE.

The semantics of this primitive are:

```
MLME-PM-MODE-ACTIVE.indication
                            (
                            PMActiveEvent
                            )
```

The primitive parameter is defined in Table 29.

6.3.22.7.1 When generated

This primitive is sent by the MLME when an event occurs that requires it to change the current PM mode of operation from APS or SPS to ACTIVE.

6.3.22.7.2 Effect of receipt

The DME is informed of the PM mode change to ACTIVE.

6.4 PLME SAP interface

The PHY management service interface consists of the generic PLME-GET and PLME-SET primitives operating on PHY PIB attributes, as described in 6.2, together with the primitives in Table 31.

The parameters used for these primitives are defined in Table 32. The parameters associated with these primitives are considered as recommendations and are optional in any particular implementation.

Table 31—Summary of PLME primitives

Name	Request	Confirm	Indication	Response
PLME-RESET	6.4.1	6.4.2	–	–
PLME-TESTMODE	6.4.3	6.4.4	–	–
PLME-TESTOUTPUT	6.4.5	–	–	–

Table 32—PLME primitive parameters

Name	Type	Valid range	Description
ResetResultCode	Enumeration	SUCCESS, FAILED	Indicates the result of the reset request.
TestEnable	Boolean	TRUE, FALSE	If TRUE, enables the PHY test mode according to the remaining parameters
TestMode	Integer	1–3	Selects one of three operational states: 1 = transparent receive 2 = continuous transmit 3 = 50% duty cycle
DataType	Integer	1–3	Selects one of three data patterns to be used for the transmit portions of the tests.
DataRate	Integer	PHY dependent.	PHY dependent index of the data rate.
TestResultCode	Enumeration	SUCCESS, FAILED, UNSUPPORTED_MODE	Indicates the result of the corresponding test mode request.
TestOutput	Boolean	TRUE, FALSE	If TRUE, enables the selected test signals for testing the PHY.

6.4.1 PLME-RESET.request

This primitive is a request by either the DME or the MLME to reset the PHY. The PHY is always reset to the off state to save power and to avoid accidental data transmission. The semantics of this primitive are:

PLME-RESET.request ()

There are no parameters associated with this primitive.

6.4.1.1 When generated

This primitive is generated at any time to reset the PHY.

6.4.1.2 Effect of receipt

Receipt of this primitive by the PHY sublayer will cause the PHY entity to reset both the transmit and the receive state machines and place the PHY into the off state.

6.4.2 PLME-RESET.confirm

This primitive reports the results of a reset procedure. The semantics of this primitive are:

```
PLME-RESET.confirm      (
                        ResetResultCode
                        )
```

The primitive parameter is defined in Table 32.

6.4.2.1 When generated

This primitive is generated by the PLME as a result of an PLME-RESET.request.

6.4.2.2 Effect of receipt

The DME or MLME is notified of the results of the reset procedure.

6.4.3 PLME-TESTMODE.request

This primitive requests that the PHY entity enter a test mode of operation. The semantics of this primitive are:

```
PLME-TESTMODE.request   (
                        TestEnable,
                        TestMode,
                        DataType,
                        DataRate
                        )
```

The primitive parameters are defined in Table 32.

6.4.3.1 When generated

This primitive is generated at any time to enter the PHY test mode.

6.4.3.2 Effect of receipt

Receipt of this primitive by the PHY layer will cause the PHY entity to enter the test mode of operation.

6.4.4 PLME-TESTMODE.confirm

This primitive reports the result of the PHY entering a test mode of operation. The semantics of this primitive are:

```
PLME-TESTMODE.confirm   (
                        TestResultCode
                        )
```

The primitive parameter is defined in Table 32.

6.4.4.1 When generated

This primitive is generated by the PLME as a result of an PLME-TESTMODE.request.

6.4.4.2 Effect of receipt

The DME or MLME is notified of the results of starting the test mode.

6.4.5 PLME-TESTOUTPUT.request

This optional primitive is a request by either the DME or the MLME 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. The semantics of this primitive are:

```
PLME-TESTOUTPUT.request     (
                            TestOutput
                            )
```

The primitive parameter is defined in Table 32.

6.4.5.1 When generated

This primitive is generated to enable the test outputs when in the PHY test mode.

6.4.5.2 Effect of receipt

Receipt of this primitive by the PHY layer will cause the PHY entity to enable the test outputs using the modes set by the most recent PLME-TESTMODE.request primitive.

6.5 MAC management

The MAC PIB comprises the managed objects, attributes, actions, and notifications required to manage the MAC sublayer of a DEV. The MAC PIB is divided into two groups, PNC characteristics and DEV characteristics. In the Access column of the Table 33 and Table 34, read only indicates that the parameter is only allowed to be read by the DME while read/write indicates that the DME is able to change it using the MLME-SET.request primitive.

6.5.1 MAC PIB PNC group

The MAC PIB PNC group, Table 33, describes both the DEV's PNC capabilities as well as the characteristics of the current piconet.

6.5.2 MAC PIB characteristic group

The MAC PIB characteristics group, Table 34, contains information about the capabilities and characteristics of the DEV.

Table 33—MAC PIB PNC group parameters

Managed object	Octets	Definition	Access
MACPIB_CAPEndTime	2	The end time of the CAP interval in the super-frames, 8.6.	Read only
MACPIB_SuperframeDuration	2	Duration of the superframe.	Read only
MACPIB_PNCCapable	1 bit	1 if the DEV has the capability to become the PNC, 0 otherwise.	Read only
MACPIB_PNCDesMode	1 bit	1 if it is desired that the DEV be the PNC.	Read only
MACPIB_MaxPSSets	1	The maximum number of PS sets supported by the PNC.	Read only
MACPIB_BSID	6-32	Identifies the piconet.	Read only
MACPIB_MaxAssociatedDEVs	2	As defined in 7.4.11.	Read only
MACPIB_MaxCTRqBs	2	As defined in 7.4.11.	Read only
MACPIB_SEC	1 bit	Indicates if the DEV is capable of operating a secure piocnet as the PNC.	Read only
MACPIB_PNCServicesBroadcast	1	0x00 = PNC sends information about its services 0x01 = PNC will not send information about its services	Read/write

Table 34—MAC PIB characteristic group parameters

Managed object	Octets	Definition	Access
MACPIB_DEVAddress	6	The MAC address of the DEV.	Read only
MACPIB_DEVID	1	The ID of the DEV.	Read only
MACPIB_PowerManagementMode	1	The current power management mode of the DEV. 0x00 = ACTIVE 0x01 = PSPS 0x02 = DSPS 0x03 = DSPS and PSPS 0x04 = APS	Read only
MACPIB_PSPSSupported	1	0x00 = DEV does not support PSPS mode 0x01 = DEV supports PSPS mode	Read only
MACPIB_DSPSSupported	1	0x00 = DEV does not support DSPS mode 0x01 = DEV supports DSPS mode	Read only
MACPIB_APSSupported	1	0x00 = DEV does not support APS mode 0x01 = DEV supports APS mode	Read only
MACPIB_MaxStreams	1	Maximum number of streams that the DEV is able to handle.	Read only
MACPIB_PowerSource	1	0x00 = battery power 0x01 = mains power	Read/write
MACPIB_SecurityOptionImplemented	1	0x00 = mode 0 0x01 = mode 1	Read only
MACPIB_DEVServicesBroadcast	1	0x00 = DEV sends information about its services 0x01 = DEV will not send information about its services	Read/write

6.6 MAC SAP

The MAC provides both stream and non-stream, i.e. asynchronous, service to the frame convergence sublayer (FCSL). This service enables the FCSL to map source/destination DEV addresses to SrcID/DestIDs and their associated channel time requirements to specific channel time allocations identified by stream indices. Streams, in this context, provide a mechanism for managing the channel time requirements of uplink (DEV to PNC), downlink (PNC to DEV) and peer-to-peer (DEV to DEV) streams.

All MSDUs of a given stream that do not use the Dly-ACK policy shall be transmitted in the order which they were received from the FCSL. This implies that it is possible that MSDUs from different streams will be transmitted in a different order than they were received from the FCSL. MSDUs that use the Dly-ACK policy may be transmitted out of order by the MAC.

Streams are dynamic in nature, in that they may be created, modified, and deleted. An established stream may need to be modified due to the type of service assigned to it. Asynchronous traffic is dynamic in nature as well and consequently needs the ability to reserve and terminate channel time allocations. For instance, IP services may require the channel time requirements to be modified due to the bursty nature of the service.

The MAC SAP defines the logical interface between the MAC and the FCSL above it. This logical interface description includes a list of primitives and their definitions. Although these primitives and their definitions are informative, they provide a context in which to understand the parameters which need to be passed between the MAC and the FCSL so that each sublayer may fulfill its specified functions.

The 802.15.3 MAC SAP primitives are summarized in Table 35.

Table 35—Summary of MAC SAP primitives

Name	Request	Indication	Response	Confirm
MAC-ASYNC-DATA	6.6.1	6.6.3	–	6.6.2
MAC-ISOCH-DATA	6.6.4	6.6.3	–	6.6.5

The parameters used for these primitives are defined in Table 36.

Table 36—MAC-ASYNC-DATA and MAC-ISOCH-DATA primitive parameters

Name	Type	Valid range	Description
TrgtID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the target of the MLME request.
OrigID	Integer	Any valid DEVID as defined in 7.2.3.	Specifies the DEVID of the DEV that originated the MAC request.
Priority	Integer	0–7	Specifies the priority of the data. See Table A.1 for a description of priority values.
ACKPolicy	Enumeration	IMM_ACK, NO_ACK, DLY_ACK	Specifies the ACK policy for the MSDU.
StreamIndex	Integer	As defined in 7.2.5	The stream over which the data is to be sent.

Table 36—MAC-ASYNC-DATA and MAC-ISOCH-DATA primitive parameters (*Continued*)

Name	Type	Valid range	Description
TransmissionTimeout	Duration	0–65535	The amount of time in milliseconds in which the data needs to be successfully sent.
Length	Integer	0–65535	The length of the MSDU in octets.
Data	Variable number of octets		MSDU portion of the primitive.
ResultCode	Enumeration	SUCCESS, TX_TIMEOUT, DLY_ACK_FAILED, INVALID_ACK_POLICY, INVALID_STREAM	Indicates the result of the corresponding MAC request.

6.6.1 MAC-ASYNC-DATA.request

This primitive is used to initiate the transfer of an asynchronous data from one MAC entity to another MAC entity or entities. The semantics of this primitive are:

```
MAC-ASYNC-DATA.request      (
                            TrgtID,
                            OrigID,
                            Priority,
                            ACKPolicy,
                            TransmissionTimeout,
                            Length,
                            Data
                            )
```

The primitive parameters are defined in Table 36.

6.6.1.1 When generated

This primitive is sent by the FCSL to the MAC SAP after receiving an MA-UNITDATA.request from the LLC sublayer.

6.6.1.2 Effect of receipt

The MAC upon receiving this primitive uses the received parameters to format an appropriate MPDU which is then sent to the PHY-SAP for transfer over the wireless medium to a peer MAC entity or entities. If the ACKPolicy in the request is set to DLY_ACK, the MAC will take no action except to return a MAC-ASYNC-DATA.confirm with the ResultCode set to INVALID_ACK_POLICY.

6.6.2 MAC-ASYNC-DATA.confirm

This primitive is used to inform the FCSL of a successful delivery or a failed delivery. The semantics of this primitive are:

```
MAC-ASYNC-DATA.confirm      (
                            TrgtID,
                            OrigID,
                            Priority,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 36.

6.6.2.1 When generated

This primitive is generated by the MAC upon either a successful delivery or an unsuccessful delivery. An unsuccessful delivery results either due to a TX_TIMEOUT expiration or because the maximum number of allowed retries has been exceeded without receiving an Imm-ACK (assuming the ACKPolicy parameter was set to IMM_ACK). If the ACKPolicy in the corresponding request was set to DLY_ACK, the ResultCode will be set to INVALID_ACK_POLICY.

6.6.2.2 Effect of receipt

The FCSL, upon receiving this primitive from the MAC, will send an MA-UNITDATA-STATUS.indication.

6.6.3 MAC-ASYNC-DATA.indication

This primitive is used to indicate the reception of an asynchronous MSDU. The semantics of this primitive are:

```
MAC-ASYNC-DATA.indication   (
                            TrgtID,
                            OrigID,
                            Length,
                            Data
                            )
```

The primitive parameters are defined in Table 36.

6.6.3.1 When generated

This primitive is generated by the MAC upon successfully processing a received asynchronous MSDU.

6.6.3.2 Effect of receipt

When the FCSL receives this primitive from the MAC it will generate an MA-UNITDATA.indication.

6.6.4 MAC-ISOCH-DATA.request

This primitive is used to initiate the transfer of an isochronous MSDU from one MAC entity to another MAC entity or entities. The semantics of this primitive are:

```
MAC-ISOCH-DATA.request      (
                            StreamIndex,
                            ACKPolicy
                            TransmissionTimeout,
                            Length,
                            Data
                            )
```

The primitive parameters are defined in Table 36.

6.6.4.1 When generated

This primitive is sent by the FCSL to the MAC SAP after receiving an MA-UNITDATA.request from the LLC sublayer and then assigning it an appropriate StreamIndex.

6.6.4.2 Effect of receipt

The MAC upon receiving this primitive uses the received parameters to format an appropriate MPDU which is then sent to the PHY-SAP for transfer over the wireless medium to a peer MAC entity or entities. If the StreamIndex for the request does not correspond to an existing stream with the DEV as the source, the MLME will not attempt to transmit the frame and will respond with a MAC-ISOCH-DATA.confirm with a ResultCode of INVALID_STREAM.

6.6.5 MAC-ISOCH-DATA.confirm

This primitive is used to inform the FCSL of a successful delivery or a failed delivery. The semantics of this primitive are:

```
MAC-ISOCH-DATA.confirm      (
                            StreamIndex,
                            ResultCode
                            )
```

The primitive parameters are defined in Table 36.

6.6.5.1 When generated

This primitive is generated by the MAC upon either a successful delivery, or an unsuccessful delivery. An unsuccessful delivery results either due to a TX_TIMEOUT expiration or because the maximum number of allowed retries has been exceeded without receiving an Imm-ACK (assuming the policy parameter was set to IMM_ACK). If the Dly-ACK policy was used, but the destination refused the use of Dly-ACK, the ResultCode is set to DLY_ACK_FAILED. This indicates successful transmission of the corresponding data frame.

6.6.5.2 Effect of receipt

The FCSL, upon receiving this primitive from the MAC, will send an MA-UNITDATA-STATUS.indication.

6.6.6 MAC-ISOCH-DATA.indication

This primitive is used to indicate the reception of an isochronous MSDU. The semantics of this primitive are:

```
MAC-ISOCH-DATA.indication   (
                            TrgtID,
                            OrigID,
                            StreamIndex,
                            Length,
                            Data
                            )
```

The primitive parameters are defined in Table 36.

6.6.6.1 When generated

This primitive is generated by the MAC upon successfully processing a received isochronous MSDU.

6.6.6.2 Effect of receipt

When the FCSL receives this primitive from the MAC it will generate an MA-UNITDATA.indication.

6.7 Physical layer (PHY) service specification

The PHY services provided to the 802.15.3 MAC are described in this clause. The 802.15.3 PHY is a system whose function defines the characteristics and methods of transmitting and receiving data through a wireless medium (WM) between two or more DEVs. These services are described in an abstract way and do not imply any particular implementation or exposed interface.

The protocol reference model for the 802.15.3 architecture is shown in Figure 3. The 802.15.3 PHY definition contains two functional entities: the PHY function and the physical layer management function (PLME).

The PHY service is provided to the MAC entity at the DEV through a service access point (SAP), called the PHY SAP, as shown in Figure 3.

If the PHY SAP is an exposed interface, then all of the service primitives described here are considered mandatory unless otherwise specified. If the PHY SAP interface is not exposed in an implementation, then there is no requirement to implement the service primitives described here.

Table 37 indicates the primitives used in the PHY SAP and the subclause in which the primitive is defined. The parameters used by one or more of the PHY SAP service primitives are defined in Table 38.

Table 37—Summary of PHY SAP service primitives

Primitive	Request	Indication	Confirm
PHY-DATA	6.7.1.1	6.7.1.2	6.7.1.3
PHY-TX-START	6.7.2.1	–	6.7.2.2
PHY-TX-END	6.7.2.3	–	6.7.2.4
PHY-CCA-START	6.7.3.1	–	6.7.3.2
PHY-CCA-END	6.7.3.3	–	6.7.3.4
PHY-CCA	–	6.7.3.5	–
PHY-RX-START	6.7.4.1	6.7.4.3	6.7.4.2
PHY-RX-END	6.7.4.4	6.7.4.6	6.7.4.5
PHY-PM	6.7.5.1	–	6.7.5.2

Table 38—PHY SAP service primitive parameters

Parameter	Type	Valid range	Definitions
Data	Octet	0–255	The data that is part of the frame body including the FCS.
CCAStatus	Enumeration	BUSY, IDLE	The status of the channel.
TXDataRate	Octet	PHY dependent.	Data rate to be used in transmitting the frame.
TXLength	2 octets	0-pMaxFrameBodySize	Length of the MAC frame to be transmitted, 7.2.
TXPowerLevel	Octet	PHY dependent.	The transmitter power to be used for the frame.
TXMACHead	10 octets	Any valid MAC header.	The MAC header of the frame to be transmitted. Note that the MAC header does not include the HCS, as indicated in 7.2.
TXAntSelect	Octet	0–255	The antenna to use for transmitting the data. The value 0 is always valid, other values are implementation dependent.
RXDataRate	Octet	PHY dependent.	The data rate of the received frame.
RXLength	2 octets	0-pMaxFrameBodySize	Length of the frame that was received.
RXMACHead	10 octets	Any valid MAC header.	The MAC header of the frame that was received. Note that the MAC header does not include the HCS, as indicated in 7.2.
RSSI	Octet	PHY dependent.	The power level of the received signal.
LQI	Octet	PHY dependent.	The quality of the received signal.
RXERROR	Enumeration	NO_ERROR, CARRIER_LOST, FORMAT_VIOLATION, UNSUPPORTED_RATE	The result of the receive process.
PMLevel	Integer	0-PHYPIB_NumPMLevels	Numeric value of one of the supported power management levels of the PHY. PMLevel value 0 is used by the MAC to instruct the PHY to return from a reduced power state, or off state, to a state where it is ready to receive command. Other values are implementation dependent.
PMResultCode	Enumeration	SUCCESS, FAILED, UNSUPPORTED_MODE	Indicates the result of the PHY request.

6.7.1 Transferring PHY data

The following subclause describes in detail the services provided by each PHY layer primitive.

6.7.1.1 PHY-DATA.request

This primitive defines the transfer of an octet of data from the MAC sublayer to the local PHY entity. The semantics of this primitive are:

```
PHY-DATA.request        (
                        Data
                        )
```

The primitive parameter is defined in Table 38.

6.7.1.1.1 When generated

This primitive is generated by the MAC sublayer to transfer an octet of data to the PHY entity. This primitive is only issued following a transmit initialization response (PHY-TX-START.confirm) from the PHY layer.

6.7.1.1.2 Effect of receipt

The receipt of this primitive by the PHY entity causes the transmit state machine to transmit an octet of data.

6.7.1.2 PHY-DATA.indication

This primitive indicates the transfer of data from the PHY layer to the local MAC entity. The semantics of this primitive are:

```
PHY-DATA.indication     (
                        Data
                        )
```

The primitive parameter is defined in Table 38.

6.7.1.2.1 When generated

This primitive is generated by a receiving PHY entity to transfer the received octet of data to the local MAC entity.

6.7.1.2.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.1.3 PHY-DATA.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm the transfer of data from the MAC entity to the PHY layer. The semantics of this primitive are:

```
PHY-DATA.confirm        ()
```

This primitive has no parameters.

6.7.1.3.1 When generated

The PHY layer will issue this primitive in response to every PHY-DATA.request primitive issued by the MAC sublayer.

6.7.1.3.2 Effect of receipt

The receipt of this primitive by the MAC will cause the MAC to start the next MAC entity request.

6.7.2 Controlling PHY transmission

6.7.2.1 PHY-TX-START.request

This primitive is a request by the MAC sublayer to the local PHY entity to start the transmission of an MPDU. The semantics of this primitive are:

```
PHY-TX-START.request    (
                        TXDataRate,
                        TXLength,
                        TXPowerLevel,
                        TXAntSelect,
                        TXMACHead
                        )
```

The primitive parameters are defined in Table 38.

6.7.2.1.1 When generated

This primitive is issued by the MAC sublayer to the PHY entity whenever the MAC sublayer needs to begin the transmission of an MPDU. The TXMACHead is passed to the PHY for transmission and for the PHY to calculate the HCS.

6.7.2.1.2 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to start the transmission of the frame.

6.7.2.2 PHY-TX-START.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm the start of a transmission. The semantics of this primitive are:

```
PHY-TX-START.confirm    ()
```

There are no parameters associated with this primitive.

6.7.2.2.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-TX-START.request from the MAC entity and is ready to begin receiving data octets from the MAC entity.

6.7.2.2.2 Effect of receipt

The receipt of this primitive by the MAC entity will cause the MAC to start the transfer of data octets.

6.7.2.3 PHY-TX-END.request

This primitive is a request by the MAC sublayer to the local PHY entity that the current transmission of the MPDU be completed. The semantics of this primitive are:

PHY-TX-END.request ()

There are no parameters associated with this primitive.

6.7.2.3.1 When generated

This primitive is generated whenever the MAC sublayer has received the last PHY-DATA.confirm from the local PHY entity for the current MPDU.

6.7.2.3.2 Effect of receipt

The effect of receipt of this primitive by the local PHY entity is to stop the current transmission.

6.7.2.4 PHY-TX-END.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm the completion of a transmission. The semantics of this primitive are:

PHY-TX-END.confirm ()

There are no parameters associated with this primitive.

6.7.2.4.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-TX-END.request immediately after finishing the transmission of the last data octet.

6.7.2.4.2 Effect of receipt

The effect of receipt of this primitive by the MAC entity is unspecified.

6.7.3 PHY CCA functions

6.7.3.1 PHY-CCA-START.request

This primitive is a request by the MAC sublayer to the local PHY entity to start the CCA process. The semantics of the primitives are as follows:

PHY-CCA-START.request ()

There are no parameters associated with this primitive.

6.7.3.1.1 When generated

This primitive is generated by the MAC sublayer for the local PHY entity to start the CCA process.

6.7.3.1.2 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to start the CCA process.

6.7.3.2 PHY-CCA-START.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm that the PHY has begun the CCA process. The semantics of the primitives are as follows:

PHY-CCA-START.confirm ()

There are no parameters associated with this primitive.

6.7.3.2.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-CCA-START.request.

6.7.3.2.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.3.3 PHY-CCA-END.request

This primitive is a request by the MAC sublayer to the local PHY entity to end the CCA process. The semantics of the primitives are as follows:

PHY-CCA-END.request ()

There are no parameters associated with this primitive.

6.7.3.3.1 When generated

This primitive is generated by the MAC sublayer for the local PHY entity when it desired to end the CCA process.

6.7.3.3.2 Effect of receipt

The effect of receipt of this primitive by the PHY entity is to end the CCA process.

6.7.3.4 PHY-CCA-END.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm that the PHY has ended the CCA process. The semantics of the primitives are as follows:

PHY-CCA-END.confirm ()

There are no parameters associated with this primitive.

6.7.3.4.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-CCA-END.request.

6.7.3.4.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.3.5 PHY-CCA.indication

This primitive is an indication by the PHY layer to the local MAC entity of the current state of the medium. The semantics of this primitive are:

```
PHY-CCA.indication      (
                        CCAStatus
                        )
```

The primitive parameter is defined in Table 38.

6.7.3.5.1 When generated

This primitive is generated every time 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 layer is receiving data. The PHY layer maintains the channel busy indication until the pCCADetectTime period has expired.

6.7.3.5.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.4 Controlling the PHY receiver

6.7.4.1 PHY-RX-START.request

This primitive is a request by the MAC sublayer to the local PHY entity to activate the receiver and select a particular antenna. The semantics of this primitive are:

```
PHY-RX-START.request    (
                        RXAntSelect
                        )
```

The primitive parameter is defined in Table 38.

6.7.4.1.1 When generated

This primitive is generated whenever the MAC sublayer anticipates that an MPDU addressed to this DEV is about to be transmitted.

6.7.4.1.2 Effect of receipt

The effect of receipt of this primitive by the local PHY entity is not specified.

6.7.4.2 PHY-RX-START.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm that the PHY receiver has been activated. The semantics of this primitive are:

```
PHY-RX-START.confirm    ()
```

There are no parameters associated with this primitive.

6.7.4.2.1 When generated

This primitive is issued by the PHY layer to the MAC entity when the PHY receiver has entered the active state.

6.7.4.2.2 Effect of receipt

The effect of receipt of this primitive by the MAC entity is unspecified.

6.7.4.3 PHY-RX-START.indication

This primitive is an indication by the PHY layer to the local MAC entity that the PHY has received a valid PHY and MAC header. The semantics of this primitive are:

```
PHY-RX-START.indication (
                        RXDataRate,
                        RXLength,
                        RXMACHeader,
                        RSSI
                        )
```

The primitive parameters are defined in Table 38.

6.7.4.3.1 When generated

This primitive is generated by the local PHY entity to the MAC sublayer whenever the PHY has successfully validated the header check sequence (HCS) at the start of a new PHY PDU.

6.7.4.3.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.4.4 PHY-RX-END.request

This primitive is a request by the MAC sublayer to the local PHY entity to turn off the receiver. The semantics of this primitive are:

```
PHY-RX-END.request ()
```

There are no parameters associated with this primitive.

6.7.4.4.1 When generated

This primitive is generated whenever the MAC sublayer wants to turn off the PHY receiver.

6.7.4.4.2 Effect of receipt

The effect of receipt of this primitive by the local PHY entity is not specified.

6.7.4.5 PHY-RX-END.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm that the PHY receiver has been turned off. The semantics of this primitive are:

```
PHY-RX-END.confirm          ()
```

There are no parameters associated with this primitive.

6.7.4.5.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-RX-END.request.

6.7.4.5.2 Effect of receipt

The effect of receipt of this primitive by the MAC entity is unspecified.

6.7.4.6 PHY-RX-END.indication

This primitive is an indication by the PHY layer to the local MAC entity that the MPDU currently being received is complete. The semantics of this primitive are:

```
PHY-RX-END.indication       (
                            LQI,
                            RXERROR
                            )
```

The primitive parameters are defined in Table 38.

6.7.4.6.1 When generated

This primitive is generated by the PHY layer for the local MAC entity to indicate that the receive state machine has completed a reception with or without errors. A number of error conditions may occur after the PHYs receive state machine has detected a signal that appears to be a valid preamble and the start of the frame. The value for RXERROR that is returned is set as follows:

— NO_ERROR: No error occurred during the receive process in the PHY.
— FORMAT_VIOLATION: The format of the received PHY PDU was in error.
— CARRIER_LOST: The carrier was lost during the reception of the incoming MPDU and no further processing of the MPDU is possible.
— UNSUPPORTED_RATE: An unsupported data rate was detected during the reception of the incoming PHY-PDU.

6.7.4.6.2 Effect of receipt

The effect of receipt of this primitive by the MAC is unspecified.

6.7.5 Controlling the PHY power usage

6.7.5.1 PHY-PM.request

This primitive is a request by the MAC sublayer to the local PHY to enter the specified power management state. This includes returning the PHY to a state where it is ready to receive commands from a management entity. The semantics of this primitive are:

```
PHY-PM.request          (
                        PMLevel
                        )
```

The primitive parameter is defined in Table 38.

6.7.5.1.1 When generated

This primitive is issued by the MAC sublayer to the PHY entity whenever the MAC sublayer needs to change the power management state of the PHY.

6.7.5.1.2 Effect of receipt

The effect of receipt is to transition the PHY to the desired state if possible, and then generate the PHY-PM.confirm primitive.

6.7.5.2 PHY-PM.confirm

This primitive is issued by the PHY layer to the local MAC entity to confirm that the PHY has entered the requested power management state. The semantics of this primitive are:

```
PHY-PM.confirm          (
                        PMResultCode
                        )
```

The primitive parameter is defined in Table 38.

6.7.5.2.1 When generated

This primitive is issued by the PHY layer to the MAC entity whenever the PHY has received a PHY-PM.request from the MAC entity and has entered the requested power management state.

6.7.5.2.2 Effect of receipt

The receipt of this primitive by the MAC entity is unspecified.

7. MAC frame formats

This clause specifies the format of the MAC frames. An overview of the MAC frame is presented first. This is followed by a description of the general frame formats and then a description of the individual frame types. Next is a listing of the information elements and finally the definitions of the commands.

The MAC in all DEVs shall be able to validate the error free reception of every frame received from the PHY using the frame check sequence (FCS). Note that the PHY only passes frames to the MAC that have passed the header check sequence (HCS) test. In addition, every DEV shall be able to construct a subset of

the command frames for transmission, and to decode another (potentially different) subset of the command frames upon reception. The particular subsets of these commands that a DEV shall construct and decode are determined by the functional capabilities supported by that particular DEV.

For a frame to be correctly received by the MAC it shall pass the frame check sequence validation, have a protocol revision supported by the MAC, have a DestID equal to either a DEVID, BcstID, McstID or when applicable the PNCID or UnassocID, and have a PNID equal to the PNID of the piconet with which the DEV is synchronized. The MAC shall ACK all correctly received frames with the ACK Policy field set to either Imm-ACK or Dly-ACK Request and DestID set to the DEVID of this DEV or when applicable the PNCID. If a DEV correctly receives a frame from an unassociated DEV it may ignore the frame and may choose not to respond to the frame. If secure membership is required in the piconet and a DEV correctly receives a frame from a DEV that is not a member of the piconet, it shall ignore the frame and shall not respond to the frame, except for the ACK, if the ACK Policy field is set to either Imm-ACK or Dly-ACK Request.

7.1 Frame format conventions

The MAC frames in the MAC sublayer are described as a sequence of fields in a specific order. Each figure in Clause 7 depicts the fields as they appear in the MAC frame and in the order in which they are transmitted in the wireless medium, from right to left where the right-most bit is transmitted first in time.

In the figures, all bits within fields are numbered from k (left) to 0 (right) where the length of the field is k+1 bits. The octet boundaries within a field are 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 decreasing order of significance, from highest numbered bit on the left to the lowest numbered bit on the right. The octets in fields longer than a single octet are sent to the PHY in order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits. For any text fields, the first character is in the first octet of the field with other characters following sequentially. An example of the bit and octet ordering is illustrated in Figure 4.

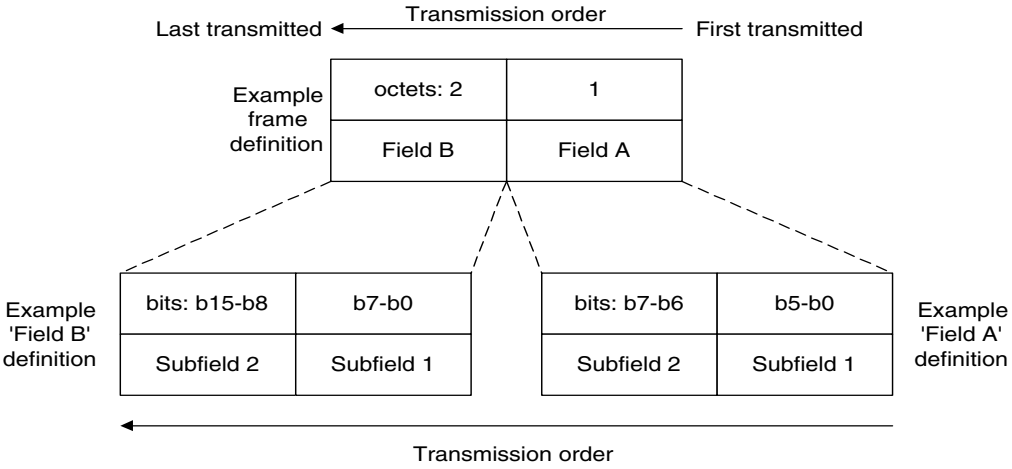

Figure 4—Example of bit and octet ordering.

The payload in the data frame is sent with the lowest numbered octet first, least significant bit first, over the air, as illustrated in Figure 5 . The FCS is an exception to this convention and is transmitted with the msb first. In the case of the HCS, the PHY determines the bit order in which the HCS is sent.

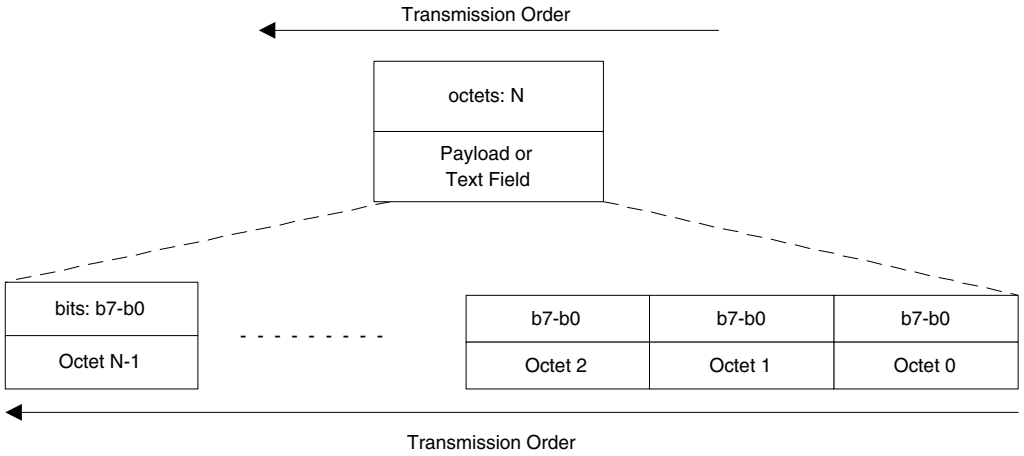

Figure 5—Example of payload transmission order.

Values specified in decimal are coded in unsigned binary unless otherwise stated.

Without further qualification, “reception” by the MAC sublayer implies that the frame contents are valid and that the protocol version is supported. However, reception implies nothing about frame addressing, nor whether the frame type or other fields in the MAC header are meaningful to the MAC entity that has received the frame.

Unless otherwise stated, any reserved field or sub-field shall be set to zero upon transmission and shall be ignored on reception.

Reserved values in non-reserved fields shall not be transmitted by conformant DEVs. However, a DEV may receive frames with values that it considers to be reserved values in non-reserved fields. These fields, along with other fields in the same frame that rely on the interpretation of these fields, shall be ignored on reception.

All DEVs shall be assigned a DEV address which is the 64-bit address as defined by IEEE Std 802®-2001.

7.2 General frame format

The MAC frame format, Figure 6, comprises a set of fields that occur in a fixed order in all frames. Each MAC frame consists of the following basic components:

a) A MAC header.
b) A MAC frame body consisting of:
   1) a variable length frame payload, and
   2) a frame check sequence (FCS).

The figures in this subclause are a representation of the MAC header and MAC frame body. The HCS is not shown since this is calculated and verified by the PHY. The MAC frame shall be formatted as illustrated in Figure 6. The maximum size of the MAC frame body, pMaxFrameBodySize, is a PHY dependent parameter

that includes the frame payload and FCS fields, but not the PHY preamble, PHY header, MAC header or MAC header validation. For the 2.4 GHz PHY, this parameter is defined in 11.2.8.1.

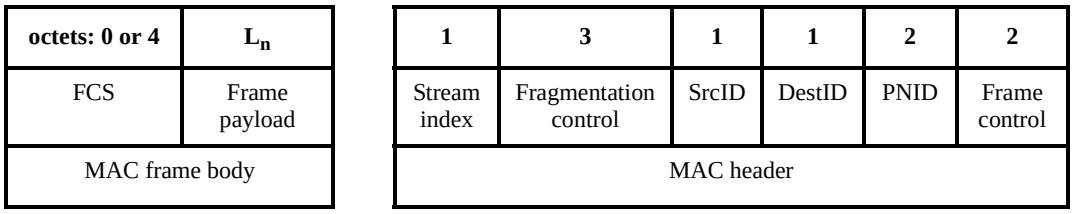

Figure 6—MAC header and frame body format

The number of octets in the MAC frame body shall range from zero to pMaxFrameBodySize, inclusive. The maximum MAC frame body length includes the length of the security fields, if present.

The non-secure MAC frame body shall be formatted as illustrated in Figure 7 when the SEC bit is set to zero in the Frame Control field.

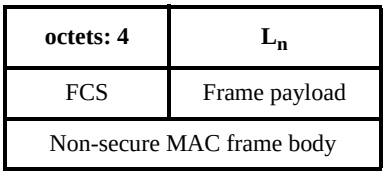

Figure 7—Non-secure MAC frame body format

The secure MAC frame body shall be formatted as illustrated in Figure 8 when the SEC bit is set to one in the Frame Control field. The Secure Payload field in the secure MAC frame body is protected as indicated in 10.2.2.

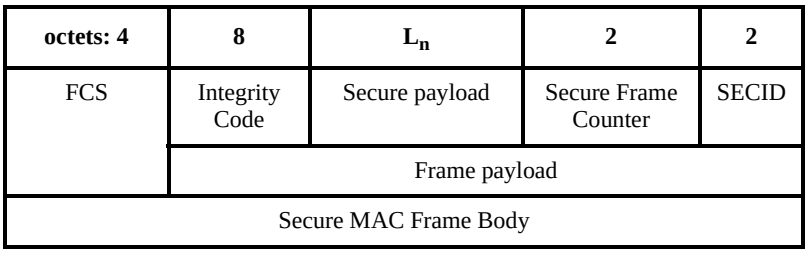

Figure 8—Secure MAC frame body format

7.2.1 Frame control

The Frame Control field shall be formatted as illustrated in Figure 9.

bits: b15–b11	b10	b9	b8–b7	b6	b5–b3	b2–b0
Reserved	More data	Retry	ACK policy	SEC	Frame type	Protocol version

Figure 9—Frame control field format

7.2.1.1 Protocol version

The Protocol Version field is invariant in size and placement across all revisions of the 802.15.3 standard. For this revision of the standard the value of the protocol version is 0b000. All other values are reserved. The revision level will be incremented only when a fundamental incompatibility exists between a new revision and the prior revision of the standard. A DEV that receives a frame with a higher revision level than it supports may discard the frame without indication to the sending DEV.

7.2.1.2 Frame type

The Frame Type field indicates the type of frame that is being sent. Table 39 lists the valid frame type values and their description. The format and the usage of each of the individual frame types is defined in 7.3.

Table 39—Valid frame type values
(numeric values in this table are shown in binary)

Type value b5 b4 b3	Frame type description	Subclause
000	Beacon frame	7.3.1
001	Immediate ACK (Imm-ACK) frame	7.3.2.1
010	Delayed ACK (Dly-ACK) frame	7.3.2.2
011	Command frame	7.3.3
100	Data frame	7.3.4
101–111	Reserved	

7.2.1.3 SEC

The SEC bit shall be set to one when the frame body is protected using the key specified by the security ID (SECID). The SEC bit shall be set to zero otherwise. Frames with the SEC bit set to one shall use the secure frame format for that frame type, as described in 7.3.

7.2.1.4 ACK policy

The ACK Policy field is used to indicate the type of acknowledgement procedure that the addressed recipient is required to perform. The use of the ACK Policy field is described in 8.8. The allowed values for the ACK Policy field are defined in Table 40.

An ACK policy of of Dly-ACK or Dly-ACK Request is valid only in the data frames of a stream that is currently employing the Dly-ACK mechanism. It is not valid for frames using the asynchronous stream index or the MCTA stream index.

7.2.1.5 Retry

The Retry bit shall be set to one in any data or command frame that is a retransmission of an earlier frame. It shall be set to zero in all other frames.

Table 40—Valid ACK policy field type values (numeric values in this table are shown in binary)

Type value b8 b7	ACK policy type	Description
00	No ACK	The recipient(s) does not acknowledge the transmission, and the sender treats the transmission as successful without regard for the actual result. The use of this policy is defined in 8.8.1.
01	Immediate ACK (Imm-ACK)	The addressed recipient returns an Imm-ACK frame after successful reception, according to the procedures defined in 8.8.2.
10	Delayed ACK (Dly-ACK)	The addressed recipient keeps track of the frames received with this policy until requested to respond with a Dly-ACK frame, according to the procedures defined in 8.8.3.
11	Dly-ACK Request	The addressed recipient returns either an Imm-ACK or a Dly-ACK frame after successful reception, according to the procedures defined in 8.8.3.

7.2.1.6 More data

The More Data bit shall be set to zero if the DEV will not use the rest of the channel time in that CTA, as described in 8.4.3.1. The More Data bit shall be set to zero in the last frame of an extended beacon and in a beacon frame that is not part of an extended beacon, as described in 8.6.2. In all other cases the More Data bit shall be set to one. This includes frames, other than the last one, that are part of an extended beacon.

7.2.2 Piconet ID (PNID)

The PNID field contains the unique identifier for the piconet, as described in 8.10.3. The PNID normally remains constant during the current instantiation of the piconet and may be persistent for multiple sequential instantiations of the piconet by the same PNC. The PNID shall be set to the current PNID for the piconet and is used to identify frames from DEVs in the piconet.

7.2.3 SrcID and DestID

There are two DEVID fields in the MAC frame format. These fields are used to indicate the source DEVID (SrcID) and destination DEVID (DestID). A DEVID for a DEV is assigned by the PNC during the association of the DEV. The DEVID is unique to an associated DEV within a piconet. The following DEVIDs are reserved.

— The DEVID value of 0x00 shall be reserved for the PNC (PNCID).
— The DEVID values of 0xED through 0xF6 shall be reserved for future use.
— The DEVID values of 0xF7, 0xF8, 0xF9, 0xFA, 0xFB or 0xFC shall be reserved for neighbor piconets (NbrID).
— The DEVID value of 0xFD shall be reserved for multicast frames (McstID).
— The DEVID value of 0xFE shall be reserved for use by all unassociated DEVs attempting to associate with a PNC (UnassocID). This value is used by an unassociated DEV until a unique DEVID is allocated by the PNC.
— The DEVID, value of 0xFF, shall be reserved for broadcast frames (BcstID).

The maximum number of valid DEVs, mMaxNumValidDEVs, is the maximum number of DEVIDs that the PNC is able to allocate in a piconet. This includes all of the regular DEVIDs, the PNCID and the NbrIDs but not the reserved IDs, the BcstID, McstID or the UnassocID.

7.2.4 Fragmentation control

The Fragmentation Control field is used to aid in the fragmentation and a reassembly of MSDUs and command frames. The Fragmentation Control field shall be formatted as illustrated in Figure 10.

bits: b23	b22–b16	b15–b9	b8–b0
Reserved	Last fragment number	Fragment number	MSDU number

Figure 10—Fragmentation control field format

7.2.4.1 MSDU number

The MSDU Number field indicates the sequence number of the current MSDU or command frame.

For data frames, each DEV shall maintain one modulo-512 counter for each of its isochronous streams, and one for its asynchronous data traffic. The MSDU numbers for all command frames shall be assigned from a single modulo-512 counter.

Each MSDU number counter shall be set to zero when the DEV is initialized.

7.2.4.2 Fragment number

The Fragment Number field indicates the order of the current fragment within the current MSDU. The Fragment Number field shall be set to zero in all unfragmented frames.

7.2.4.3 Last fragment number

The Last Fragment Number field indicates the total number of fragments within the current MSDU. The value of this field is equal to one less than the number of fragments. The Last Fragment Number field shall be the same for every fragment of a fragmented MSDU and shall be set to zero for all unfragmented MSDUs.

7.2.5 Stream index

The Stream Index field reserved values are:

— 0x00 reserved for asynchronous data.
— 0xFD reserved for MCTA traffic.
— 0xFE reserved for unassigned streams

DEVs use other values of the stream index as dynamically assigned by the PNC during the setup of the data stream, as described in 7.5.6.1. The PNC allocates a unique stream index value for each isochronous stream in the piconet.

7.2.6 MAC header validation

When the PHY receives a frame it validates the received frame’s MAC header before passing the MAC header and and its associated MAC frame body to the MAC. The protection mechanism used to validiate the MAC header is PHY dependent. In addition, the bit order and the length of the protection mechanism, pLengthHCS, are also PHY dependent. For the 2.4 GHz PHY, the MAC header protection mechanism is defined in 11.2.9.

7.2.7 MAC frame body

7.2.7.1 Frame payload

The Frame Payload field is a variable length field that carries the information that is to be transferred to a DEV or group of DEVs in the piconet. In the case of a secure frame, it also includes the required security information and the secure payload, Figure 8.

7.2.7.2 Secure session ID (SECID)

The SECID field shall be included in the frame body of all secure frames. The SECID field contains a 2-octet identifier for the key that is being used to protect the frame. The lowest order octet of the SECID for all keys except the piconet group data key shall be set to the DEVID of the key originator in the relationship. The SECID for the piconet group data key shall have the lowest order octet set to the BcstID, as described in 7.2.3. The higher order octet shall designate a unique value for the key associated with the security relationship. The SECID for a given key is selected by the key originator in a security relationship, as described in 9.3.7.

7.2.7.3 Secure frame counter (SFC)

The Secure Frame Counter field shall be included in the frame body of all secure frames. The Secure Frame Counter field contains a 2-octet counter that is used to ensure the uniqueness of the nonce in a secure frame. A DEV shall not reuse a frame counter with the same time token, as described in 7.3.1.1, and key, as described in 9.3.5. The DEV shall initialize the SFC to zero for the first frame sent and increment it for each successive secure frame sent. When the time token, as described in 7.3.1, is updated, the DEV may reset the SFC to zero if desired or allow the counter to roll over. In the case where the DEV receives a new key, the DEV shall set the SFC to zero.

7.2.7.4 Secure payload

The Secure Payload field is a variable length field that contains the information, protected by the symmetric key security operations as defined in 10.3, that is to be transferred to a DEV or group of DEVs in the piconet. As illustrated in Figure 8, the Secure Payload field is a part of the Frame Payload field and does not include the SECID, SFC or Integrity Code fields.

7.2.7.5 Integrity code

The Integrity Code field shall be included in the frame body of all secure frames. The Integrity Code field contains an 8-octet encrypted integrity code that is used to cryptographically protect the integrity of the MAC header and Frame Payload. The integrity code is computed as specified in 10.3.

7.2.7.6 FCS

The FCS field contains a 32-bit CRC. The CRC described here is equivalent to ANSI X3.66-1979. The msb of the FCS is the coefficient of the highest order term and the field is sent over the wireless medium as indicated in 7.1. The FCS shall be calculated over the entire Frame Payload field which is referred to here as the calculation field.

If the Frame Payload field has zero length (as in an immediate ACK frame) the FCS shall not be sent.

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 one's complement of the modulo 2 sum of the remainders in "a" and "b" below:

a) The remainder resulting from $((x^k*(x^{31}+x^{30}+...))$ divided (modulo 2) by G(x). The value k is the number of bits in the calculation field.
b) The remainder resulting from the calculation field contents, treated as a polynomial, is multiplied by X^{32} and then divided by G(x).

The FCS field shall be transmitted in the order specified in 7.1.

At the transmitter, the initial remainder of the division shall be preset to all ones and is then modified via division of the calculation fields by the generator polynomial G(x). The ones complement of this remainder is the FCS field.

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

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

7.3 Format of individual frame types

7.3.1 Beacon frame

7.3.1.1 Non-secure beacon frame

The Non-secure Beacon frame shall be formatted as illustrated in Figure 11.

octets: 4	**L_n**	**…**	**L_m**	**21**
FCS	Information element-n	…	Information element-1	Piconet synchronization parameters
Beacon frame body				

10
MAC header

Figure 11—Non-secure beacon frame format

The individual information elements (IEs) in the beacon frame body are listed in Table 48. These IEs are encoded in type, length, value format and are defined in 7.4. The IEs in the beacon payload may appear in any order except for the channel time allocation (CTA) IEs, which shall be the first IEs of the beacon payload following the Piconet Synchronization Parameters field.

The Piconet Synchronization Parameter field shall be formatted as illustrated in Figure 12.

octets: 8	**1**	**1**	**1**	**2**	**2**	**6**
PNC address	PNC response	Piconet mode	Max TX power Level	CAP end time	Superframe duration	Time token

Figure 12—Piconet synchronization parameters field format

The Time Token field contains a 48-bit roll-over counter which is incremented in each beacon. The beacon number is defined to be the 16 lsbs of the time token.

The Superframe Duration field contains the duration of the current superframe. The resolution of this field is 1 μs and therefore has a range of [0–65535] μs. However, the valid range of this field is [mMinSuperframeDuration, mMaxSuperframeDuration].

The CAP End Time field specifies the end of the CAP interval for the current superframe as defined in 8.6. The resolution of this field is 1 μs which gives a range of [0–65535] μs. The CAP begins a SIFS after the end of the beacon and continues until the CAP end time.

The Max TX Power Level field is used to indicate the maximum TX power level allowed in the current superframe by the PNC in the piconet as described in 8.11.2.1. The value is in dBm encoded in 2s complement format. For example, a +2 dBm TX power level is encoded as 0x02 while a –2 dBm TX power level is encoded as 0xFE. However, if the PNC does not want to limit the TX power, then it shall set the field to 0x7F.

The Piconet Mode field defines certain characteristics about the piconet and the superframe. The encoding of this octet shall be formatted as illustrated in Figure 13.

bits: b7-b5	b4	b3	b2	b1	b0
Reserved	SEC mode	MCTA used	CAP association	CAP commands	CAP data

Figure 13—Piconet mode field

If a bit is set for the CAP Data , CAP Commands or CAP Association bits, i.e. its value is 1, then that type of data or command is allowed to be sent in the CAP of the current superframe. The CAP Commands bit applies to all commands except for the Association Request command, which is covered by the CAP Association bit. Otherwise, that type of frame is not allowed to be sent in the CAP. The use of these fields is described in 8.4.2 and 11.2.10.

The MCTA Used bit shall be set to one if the PNC will be using open or association MCTAs in the superframe.

The SEC Mode field indicates the current security settings in the piconet as defined in 9.2. The field is encoded as illustrated in Table 41:

Table 41—SEC mode field encodings

Type value b3	SEC mode
0	Mode 0
1	Mode 1

The PNC Response field shall be formatted as illustrated in Figure 14.

Bits: b7–b4	b3–b0
Reserved	MCTA allocation rate

Figure 14—PNC response field format

The MCTA Allocation Rate field indicates the frequency with which the PNC will be allocating either open MCTAs or directed uplink MCTAs for each DEV. For example, if the MCTA Allocation Rate field is set to a value of 8, the PNC is indicating that it will be providing either an open MCTA or a directed uplink MCTA for each DEV in the piconet at least once out of every 8 superframes. A value of 15 means the PNC is not giving any guarantees about when it will allocate MCTAs. A value of 0 indicates that the PNC is using only the CAP to provide access to the PNC.

The PNC Address field contains the DEV address of the PNC, as described in 7.1.

The MAC header settings for a non-secure beacon frame shall be set and interpreted as described in Table 42.

Table 42—MAC header settings for a non-secure beacon frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 39	Decoded
SEC	0	Decoded
ACK policy	No-ACK value in Table 40	May be ignored
Retry	0	May be ignored
More data	As required, 8.6.2	Decoded
DestID	BcstID	Decoded
SrcID	PNCID	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

7.3.1.2 Secure beacon frame

The Secure Beacon frame shall be formatted as illustrated in Figure 15. The Secure Beacon frame format is used when the piconet is operating in a secure mode.

octets: 4	8	L_n	…	L_1	21	2	2	octets:10
FCS	Integrity code	Information element-n	…	Information element-1	Piconet synchronization parameters	SFC	SECID	MAC header
Secure Beacon frame body								

Figure 15—Secure beacon frame format

The SECID field is defined in 7.2.7.2

The SFC field is used by the DEV for this frame to ensure uniqueness of the nonce, as defined in 7.2.7.3.

The Piconet Synchronization Parameters field is defined in 7.3.1.1.

The Integrity Code is defined in 7.2.7.5.

The MAC header settings for a Secure Beacon frame shall be set and interpreted as described in Table 43.

Table 43—MAC header settings for a secure beacon frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 39	Decoded
SEC	1	Decoded
ACK policy	No-ACK value in Table 40	May be ignored
Retry	0	May be ignored
More data	As required, 8.6.2	Decoded
DestID	BcstID	Decoded
SrcID	PNCID	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

7.3.2 Acknowledgement frames

7.3.2.1 Immediate ACK (Imm-ACK) frame

The Immediate ACK frame shall be formatted as illustrated in Figure 16.

octets: 10
MAC header

Figure 16—Immediate ACK frame format

The MAC header settings for an Imm-ACK frame shall be set and interpreted as described in Table 44.

7.3.2.2 Delayed ACK (Dly-ACK) frame

The Dly-ACK frame consists of a string of MPDU identifier blocks for which the destination DEV is acknowledging reception. The frame is only used in response to an isochronous stream data frame with the ACK Policy field set to Dly-ACK Request. The MPDU-ID blocks shall be sent in the same order as the data frames were received. The Dly-ACK frame body shall be formatted as illustrated in Figure 17.

Table 44—MAC header settings of an immediate ACK frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Imm-ACK value in Table 39	Decoded
SEC	0	May be ignored
ACK policy	No-ACK value in Table 40	May be ignored
Retry	0	May be ignored
More data	0	May be ignored
DestID	SrcID of the received frame	Decoded
SrcID	DestID of the received frame	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

octets: 4	2	...	2	1	1	1
FCS	MPDU ID block-n	...	MPDU ID block-1	MPDUs ACKed	Max frames	Max burst

10
MAC header

Figure 17—Dly-ACK frame format

The Max Burst field indicates the number of frames of pMaxFrameBodySize that may be sent in one burst. A burst is the collection of the frames that are pending acknowledgement via a Dly-ACK frame.

The Max Frames field indicates the maximum number of frames, regardless of size, that may be sent before requesting a Dly-ACK from the DEV receiving the frames.

Any burst shall meet the restrictions of both theMax Frames field and the Max Burst field as described in 8.8.3.

The MPDUs ACKed field shall contain the number of MPDUs that are being ACKed with this frame. This field shall be greater than or equal to 1.

The MPDU ID block shall be formatted as illustrated in Figure 18.

bits b15–b9	b8–b0
Fragment number	MSDU number

Figure 18—MPDU ID block format

The MAC header settings for a Dly-ACK frame shall be set and interpreted as described in Table 45.

Table 45—MAC header settings of a Dly-ACK frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Dly-ACK value in Table 39	Decoded
SEC	0	May be ignored
ACK policy	No-ACK value in Table 40	May be ignored
Retry	0	May be ignored
More data	0	May be ignored
DestID	SrcID of the received frame	Decoded
SrcID	DestID of the received frame	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

7.3.3 Command frame

When sending command frames the ACK Policy field in the Frame Control field shall be set to either Imm-ACK or no-ACK. Dly-ACK and Dly-ACK Request shall not be permitted.

Only certain commands may be fragmented, as indicated in 7.5. For commands that are not allowed to be fragmented, the Fragmentation Control field shall be set to 0x000000.

7.3.3.1 Non-secure command frame

The Non-secure command frame shall be formatted as shown in Figure 19. The command types are described in 7.5.

octets: 4	(4+L)		10
FCS	Command block		MAC header

Figure 19—Non-secure command frame format

The command block shall be formatted as shown in Figure 20.

octets: L	2	2
Command payload	Length (=L)	Command type

Figure 20—Command block format

The MAC header settings for a non-secure command frame shall be set and interpreted as described in Table 46.

Table 46—MAC header settings of a non-secure command frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 39	Decoded
SEC	0	Decoded
ACK policy	As required for command protocol	Decoded
Retry	As appropriate	Decoded
More data	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
Stream index	0x00 or 0xFD	May be ignored

7.3.3.2 Secure command frame

The Secure command frame format shall be formatted as illustrated in Figure 21.

octets: 4	8	(4+L)	2	2	10
FCS	Integrity code	Command block	SFC	SECID	MAC header

Figure 21—Secure command frame format

The SECID is defined in 7.2.7.2

The SFC is defined in 7.2.7.3.

The Integrity Code field is defined in 7.2.7.5

This frame format is used when the piconet is operating in a secure mode. The command block shall be formatted as illustrated in Figure 20.

The MAC header settings for a secure command frame shall be set and interpreted as described in Table 47.

Table 47—MAC header settings of a secure command frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 39	Decoded
SEC	1	Decoded
ACK policy	As required for command protocol	Decoded
Retry	As appropriate	Decoded
More data	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
Stream index	0x00 or 0xFD	May be ignored

7.3.4 Data frame

7.3.4.1 Non-secure data frame

The Non-secure Data frame shall be formatted as shown in Figure 22.

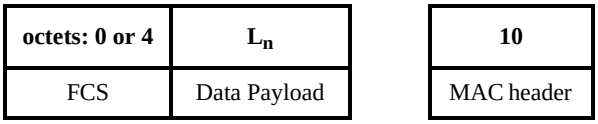

Figure 22—Non-secure data frame format

The length of the Data Payload field is limited by the maximum size allowed for the MAC frame body, as described in 7.2. Null data frames, which are non-secure data frames with a zero length Data Payload field, are allowed. For example, a null data frame may be used with Dly-ACK negotiation, as described in 8.8.3.

The FCS is not sent for zero length frames, as described in 7.2.7.6.

The frame type shall be set to the data frame value in Table 39 and the SEC bit shall be set to zero. The other fields in the MAC header take on values that are appropriate for that particular data frame. All fields in the MAC header of a Non-secure Data frame shall be decoded on reception.

7.3.4.2 Secure data frame

The Secure Data frame shall be formatted as illustrated in Figure 23.

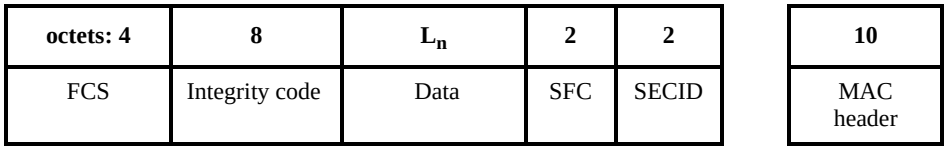

Figure 23—Secure data frame format

The SECID field is defined in 7.2.7.2

The SFC field is defined in 7.2.7.3.

The Integrity Code field is defined in 7.2.7.5

If the symmetric key security operations in use requires data encryption, the Data field shall be encrypted.

The Frame Type field shall be set to the data frame value in Table 39 and the SEC bit shall be set to one. The other fields in the MAC header take on values that are appropriate for that particular data frame. All fields in the MAC header of a Secure Data frame shall be decoded on reception.

7.4 Information elements

The information elements are listed in Table 48. Individual elements are described in the following sub-clauses.

Table 48—Information elements

Element ID hex value	Element	Subclause	Present in beacon
0x00	Channel time allocation	7.4.1	As needed
0x01	BSID	7.4.2	In every beacon
0x02	Parent piconet	7.4.3	As needed
0x03	DEV association	7.4.4	As needed
0x04	PNC shutdown	7.4.5	As needed
0x05	Piconet parameter change	7.4.6	As needed
0x06	Application specific	7.4.7	As needed
0x07	Pending channel time map (PCTM)	7.4.8	As needed
0x08	PNC handover	7.4.9	As needed
0x09	CTA status	7.4.10	As needed
0x0A	Capability	7.4.11	Non-beacon IE
0x0B	Transmit power parameters	7.4.12	Non-beacon IE
0x0C	PS status	7.4.13	As needed
0x0D	Continued wake beacon (CWB)	7.4.14	As needed
0x0E	Overlapping PNID	7.4.15	Non-beacon IE
0x0F	Piconet services	7.4.16	Non-beacon IE
0x10-0x7F	Reserved		
0x80-0xFF	Vendor specific	7.4.17	As needed

The format of an individual IE is shown in Figure 24. The first octet is the Element ID and the second octet is the Length (L_n) of the payload of the IE in octets. The following L_n octets are the payload for the IE.

Unless otherwise specified, these elements may appear in any order in the frames that are allowed to include more than one of these elements.

octets: L_n	**1**	**1**
IE payload	Length (=L_n)	Element ID

Figure 24—Information element format

7.4.1 Channel time allocation

The Channel Time Allocation (CTA) IE shall be formatted as illustrated in Figure 25. Because the length parameter supports only 255 octets of payload in an IE, the PNC may split the CTA information into more than one CTA IE entry in the beacon. The CTA blocks shall be ordered by increasing value of the CTA location with the highest value being the last.

octets: 7	⋯	**7**	**7**	**1**	**1**
CTA block-n	…	CTA block-2	CTA block-1	Length = (7*n)	Element ID

Figure 25—Channel time allocation information element format

The CTA blocks shall be formatted as illustrated in Figure 26.

octets: 2	**2**	**1**	**1**	**1**
CTA duration	CTA location	Stream index	SrcID	DestID

Figure 26—Channel time allocation block

The DestID indicates the DEV to which the source DEV may send the frames.

The SrcID indicates the DEV to which the channel time is being allocated.

If the CTA is for a child piconet, the DestID and SrcID shall both be the DEVID of the DEV that is the child piconet's PNC.

If the CTA is for a neighbor piconet, the DestID and SrcID shall both be the DEVID assigned by the PNC for the neighbor piconet and shall be one of the reserved neighbor piconet IDs, as described in 7.2.3.

The Stream Index field, as described in 7.2.5, indicates the stream corresponding to the channel time allocation.

The CTA Location field indicates the start time of the allocation. The value of this field is the time offset from the start of the beacon as described in 8.6. The resolution of this field is 1 μs, so the valid range is [0–65535] μs.

The Duration field specifies the duration of the CTA. The resolution of this field is 1 μs, so the valid range is [0–65535] μs. The end time of each allocation is the start time contained in the CTA Location field plus the CTA duration.

7.4.2 BSID

The BSID IE is used to provide a text string to identify the piconet. The BSID IE shall be formatted as illustrated in Figure 27.

octets: 6–32	**1**	**1**
Piconet BSID	Length (= 6 to 32)	Element ID

Figure 27—BSID information element format

The Piconet BSID field is a set of ISO/IEC 646:1991 encoded characters that is used to identify the piconet. The setting of the piconet BSID is described in 8.10.3.

7.4.3 Parent piconet

The Parent Piconet IE is used to provide a text string to identify the parent piconet and the DEV address of the parent PNC. The Parent Piconet IE shall be formatted as illustrated in Figure 28.

octets: 6–32	**8**	**1**	**1**
Parent piconet BSID	Parent PNC address	Length (=14 to 40)	Element ID

Figure 28—Parent piconet information element format

The Parent PNC Address field contains the DEV address, as described in 7.1, of the parent PNC for the piconet.

The Parent Piconet BSID field contains the piconet BSID from the BSID IE, as described in 7.4.2, in the parent PNC's beacon.

7.4.4 DEV association

The DEV Association IE shall be formatted as illustrated in Figure 29. This IE is used to notify current members in the piconet about one or more DEVs which have either just associated or disassociated from the piconet.

octets: 13	⋯	**13**	**13**	**1**	**1**
DEV-n association info	…	DEV-2 association info	DEV-1 association info	Length = (13*n)	Element ID

Figure 29—DEV association information element format

The DEV Association Info fields shall be formatted as illustrated in Figure 30.

octets: 3	**1**	**1**	**8**
DEV capabilities	DEV status	DEVID	DEV address

Figure 30—DEV association info fields

The DEV Address field contains the address of the DEV, as described in 7.1, that corresponds to the DEVID.

The DEVID is the identifier assigned by the PNC to a DEV.

The DEV Status field shall be formatted as illustrated in Figure 31.

bits: b7–b1	b0
Reserved	Association status

Figure 31—DEV status field format

The Association Status field shall be encoded as:

— 0 –> Disassociated
— 1 –> Associated

The DEV Capabilities field is defined in 7.4.11.

7.4.5 PNC shutdown

The PNC Shutdown IE shall be formatted as illustrated in Figure 32. This IE is used to indicate that the PNC is shutting down.

octets: 1	1	1
Remaining DEVID	Length (=1)	Element ID

Figure 32—PNC shutdown information element format

The Remaining DEVID field indicates which dependent piconet PNC is allowed to continue operation as described in 8.2.7.1. It shall be set to the PNCID if there are no dependent piconets in the current piconet.

7.4.6 Piconet parameter change

The Piconet Parameter Change IE shall be formatted as illustrated in Figure 33.

octets: 6–32	2	2	1	2	1	1	1
BSID	PNID	Superframe timing	New channel index	Change beacon number	Change type	Length (=14-40)	Element ID

Figure 33—Piconet parameter change information element format

The Change Type field indicates the parameter of the piconet that is changing and, therefore, the field that shall be interpreted by the DEV. The Change Type field value and its interpretation is given in Table 49.

The New Channel Index, Superframe Timing, PNID and BSID fields are defined in Table 49.

The Change Beacon Number field is the beacon number of the superframe when the change will take effect. The difference between the beacon number of the beacon which first includes this IE and the Change Beacon Number field is defined to be the NbrOfChangeBeacons. For a piconet without pseudo-static CTAs, NbrOfChangeBeacons shall be at least two. For a piconet that has pseudo-static CTAs, NbrOfChangeBeacons shall be at least mMaxLostBeacons. For a piconet that has child or neighbor piconets, NbrOfChangeBeacons shall be at least eight. However, a child or neighbor PNC may set the NbrOfChangeBeacons to a

Table 49—Description of field contents for change type values

Change type field value	Interpretation	Field to decode	Description of field contents
0	PNID	PNID	The new PNID, 7.2.2, that will take effect beginning with the superframe which has the beacon number equal to the Change Beacon Number field.
1	BSID	BSID	The new BSID, 7.4.2, that will take effect beginning with the superframe which has the beacon number equal to the Change Beacon Number field.
2	MOVE	Superframe timing	The offset in microseconds between the beacon's expected transmission time and the time that it will be sent by the PNC, 8.10.1. The change occurs with the beacon which has the beacon number equal to the Change Beacon Number field.
3	SIZE	Superframe timing	The new superframe duration, 8.10.2, that will be used for the superframe which has the beacon number equal to the Change Beacon Number field.
4	CHANNEL	New channel index	The channel index of the PHY channel that the piconet will begin using the beacon that has the beacon number equal to the Change Beacon Number field. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY the mapping is defined in 11.2.3.
5-255	Reserved	None	

different number based on the Change Beacon Number field in the parent PNC's beacon as defined in 8.11.1.

7.4.7 Application specific

The Application Specific IE (ASIE) shall be formatted as illustrated in Figure 34. The purpose of this IE is to allow custom information for enhanced operation that is outside of the scope of this standard.

octets: L_n	**3**	**1**	**1**
Application specific data	Vendor OUI	Length (=3+L_n)	Element ID

Figure 34—Application specific information element format

The Vendor OUI field is the OUI assigned by the IEEE standards association registration authority committee (RAC), which shall be the sole registration authority. A value of vendor OUI not understood by a receiving DEV causes the remainder of this IE to be ignored

The Application Specific Data field is provided by the PNC. Its use by an application specific capable DEV is outside of the scope of this standard.

More than one ASIE may be placed in any beacon. The negotiation of the application specific capability between the DEV and the PNC is outside of the scope of this standard.

7.4.8 Pending channel time map (PCTM)

The PCTM IE is used to request that a DSPS or APS DEV switch to ACTIVE mode. The PCTM IE shall be formatted as illustrated in Figure 35.

octets: 1-32	1	1	1
PCTM	Start DEVID	Length (=2 to 33)	Element ID

Figure 35—Pending channel time map information element format

The Start DEVID field indicates the DEVID that corresponds to the first bit in the PCTM.

The PCTM field contains a bitmap of 1 to 32 octets in length. Each bit of the PCTM field when set to one indicates the PNC is requesting that the DEV whose DEVID is equal to the start DEVID plus the bit position in the PCTM bitmap listen to the next beacon for a CTA, as described in 8.13.2. The bit position 0, i.e. the first bit or lsb of the bitmap corresponds to the start DEVID.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs and the reserved DEVIDs, as described in 7.2.3, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

7.4.9 PNC handover

The PNC Handover IE shall be formatted as ilustrated in Figure 36. This IE is included in the last beacons sent by the old PNC just prior to the old PNC relinquishing control of the piconet.

octets: 2	1	8	1	1
Handover beacon number	New PNC DEVID	New PNC address	Length (=11)	Element ID

Figure 36—PNC handover information element format

The New PNC Address field contains the DEV address, 7.1, of the DEV that will be taking over as PNC.

The New PNC DEVID field contains the current DEVID of the DEV that will be taking over as PNC

The Handover Beacon Number field contains the beacon number of the first beacon that will be sent by the new PNC. The last beacon sent by the old PNC will have a beacon number one less than the Handover Beacon Number field.

7.4.10 CTA status

The CTA Status IE is used by the PNC to inform the DEVs of certain characteristics of a CTA. The CTA Status IE shall be formatted as illustrated in Figure 37.

octets: 2	2	1	1	1	1	1	1
Start beacon number	CTA rate factor	CTRq control	Stream index	SrcID	DestID	Length (=8)	Element ID

Figure 37—CTA status information element format

The DestID field contains the DEVID of the destination for this CTA.

The SrcID field contains the DEVID of the source for this CTA.

The Stream Index field indicates the stream for which the PNC is providing information, as described in 7.5.6.1.

The CTRq Control field is defined in 7.5.6.1.

The CTA Rate Factor field is set to the number of beacons between every CTA as described in 7.5.6.1. If one or more CTAs are allocated per superframe, this value shall be set to zero.

The Start Beacon Number field is set to the beacon number, as described in 7.3.1.1, of the first beacon where the CTA of the new or modified stream will first appear.

7.4.11 Capability

The Capability IE shall be formatted as illustrated in Figure 38.

octets: 7	1	1
Overall capabilities	Length (=7)	Element ID

Figure 38—Capability information element format

The Overall Capabilities field shall be formatted as illustrated in Figure 39 .

3	4
DEV capabilities	PNC capabilities

Figure 39—Overall capabilities field format

The PNC Capabilities field shall be formatted as illustrated in Figure 41.

octets: 1	1	1	1
PNC rating	Max TX power	Max CTRqBs	Max associated DEVs

Figure 40—PNC capabilities field format

The Max Associated DEVs field indicates the number of associated DEVs this DEV is able to manage if it is PNC capable and becomes the PNC. Non PNC capable DEVs shall set this field to zero.

The Max CTRqBs field indicates the number of CTRqBs the DEV is capable of handling as a PNC. This field shall be set to zero in a non-PNC capable DEV.

The Max TX Power Level field indicates the maximum transmit power that is possible for the DEV. The power level is in dBm, encoded in 2s complement notation. For example, if a DEV was capable of 14 dBm TX power, the field would take on the value 0x0E while if the DEV was capable of –4 dBm TX power, the field would take on the value 0xFC.

The PNC Rating field shall be formatted as illustrated in Figure 41. Bits b7-b4 are arranged in order of preference for PNC selection, with the highest preference (PNC capable) corresponding to the msb.

bits: b7	b6	b5	b4	b3–b0
PNC capable	PNC Des-mode	SEC	PSRC	Reserved

Figure 41—PNC rating field format

The PSRC bit shall be set to one if the DEV is receiving power from the alternating current mains and shall be set to zero otherwise.

The SEC bit shall be set to one if the DEV is capable of acting as a key originator, 9.4. Otherwise, the SEC bit shall be set to zero.

The PNC Des-Mode bit is the desired mode of the DEV. This bit shall be set to one if it is desired that the DEV be the PNC of the piconet and the PNC Capable bit is set to one. Otherwise, this bit shall be set to zero.

The PNC Capable bit shall be set to one if the DEV is capable of being a PNC in the piconet. Otherwise, the PNC Capable bit shall be set to zero.

The DEV Capabilities field shall be formatted as illustrated in Figure 42.

bits:b23–b11	b10	b9	b8	b7–b5	b4–b0
Reserved	Listen to Multicast	Listen to Source	Always AWAKE	Preferred fragment size	Supported data rates

Figure 42—DEV capabilities field format

The Supported Data Rates field is a PHY dependent mapping that indicates the data rates that the DEV is capable of using. For the 2.4 GHz PHY, the mapping of a field value to a set of data rates is defined in Table 89.

The Preferred Fragment Size field is a PHY dependent mapping that indicates the maximum MAC frame size preferred to be received by the DEV when fragmentation is used. For the 2.4 GHz PHY, the mapping of a field value to a preferred fragment size is defined in Table 90.

The Always AWAKE bit shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all CTAs, regardless of the DestID or SrcID. Otherwise the bit shall be set to zero.

The Listen to Source bit shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all CTAs where the SrcID is equal to the DEVID of a DEV that is currently the source of a stream to that DEV regardless of the DestID of those CTAs. Otherwise, the bit shall be set to zero.

The Listen to Multicast bit shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all multicast CTAs regardless of the SrcID or the Stream Index. Otherwise the bit shall be set to zero.

The values of the bits in the PNC Capabilities field and the DEV Capabilities field shall not change while a DEV is associated in a piconet.

7.4.12 Transmit power parameters

The Transmit Power Parameters IE shall be formatted as illustrated in Figure 43. This IE is used to communicate the transmit power control (TPC) capabilities of a DEV.

octets: 1	1	1	1	1
Current TX power	TX power step size	TX power levels	Length (=3)	Element ID

Figure 43—Transmit power parameters information element format

The TX Power Levels field indicates the number of levels supported by a DEV.

The TX Step Size field iindicates the TX power level step size in 1 dB resolution, e.g. a number 4 in this field means that the DEV has nominally 4 dB steps.

If a DEV does not support TPC, it shall set the TX Power Levels and TX Power Step Size fields to 0.

The Current TX Power field is the DEV's estimate of its transmitter power measured at the antenna interface. The value is in dBm encoded in 2s complement format. For example, a +2 dBm TX power level is encoded as 0x02 while a –2 dBm TX power level is encoded as 0xFE.

7.4.13 PS status

The PS Status IE shall be formatted as illustrated in Figure 44.

octets: 1-32	1	1	2	1	1
DEVID bitmap	Start DEVID	PS set index	Next wake beacon	Length (=5-36)	Element ID

Figure 44—PS status information element format

The Next Wake Beacon field is the beacon number, as described in 7.3.1.1, of the next wake beacon for the set indicated in the PS Set Index field, as described in 8.13.1. It shall be set to zero when the PS Set Index field is zero, i.e. for the set of DEVs in APS mode.

The PS Set Index field is set to the index of the power save set as follows:

0 –> APS
1 –> PSPS
2–253 –> DSPS

The Start DEVID field indicates the DEVID that corresponds to the first bit in the DEVID bitmap.

The DEVID Bitmap field is 1 to 32 octets in length. Each bit of the DEVID Bitmap field corresponds to a DEVID that is equal to the start DEVID plus the bit position in the bitmap. The bit position zero, i.e. the first bit or lsb of the bitmap, corresponds to the start DEVID. The bit correspoding to a DEVID shall be set to one when a DEV that is a member of this PS set is in a power save mode. It shall be set to zero otherwise.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs and the reserved IDs, as described in 7.2.3, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

7.4.14 Continued wake beacon (CWB)

The Continued Wake Beacon IE shall be formatted as illustrated in Figure 45.

octets: 1–32	1	1	1
DEVID bitmap	Start DEVID	Length (=2–33)	Element ID

Figure 45—CWB information element format

The Start DEVID field indicates the DEVID that corresponds to the first bit in the DEVID bitmap.

The DEVID Bitmap field is 1 to 32 octets in length. Each bit of the DEVID bitmap corresponds to a DEVID that is equal to the start DEVID plus the bit position in the bit map. The bit position zero, i.e. the first bit or lsb of the bitmap, corresponds to the start DEVID. The bit correspoding to a DEVID shall be set to one when the PNC is requesting that the DEV listen to the next beacon for a CTA block, as described in 8.13.2.3. It shall be set to zero otherwise.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs and the reserved DEVIDs, as described in 7.2.3, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

7.4.15 Overlapping PNID

The Overlapping PNID IE is used to communicate the PNIDs that a DEV has detected either in its channel or in other channels. The Overlapping PNID IE shall be formatted as illustrated in Figure 46.

octets: 1	2	...	1	2	1	1
Channel index n	PNID n	...	Channel index 1	PNID 1	Length (=3*n)	Element ID

Figure 46—Overlapping PNID information element format

The PNID field contains the PNID from a frame that a DEV has received since the last time this IE was sent by the DEV to the PNC.

If the DEV has received a beacon from a different piconet on the current channel with the same PNID but a different PNC Address, it will add that PNID and channel index to this IE. Otherwise, the IE shall not contain the same PNID/channel index pair as for the current piconet. Thus a DEV will report piconets with the same PNID in other channels but will not erroneously report frames from the current piconet as being from an overlapping piconet.

The Channel Index field contains the channel on which the PNID was found.

7.4.16 Piconet services

The Piconet Services IE is used to provide information about the application layer capabilities of an individual DEV. The Piconet Services IE shall be formatted as illustrated in Figure 47.

octets: (0-127)	3	1	1	1
Piconet services	Vendor OUI	DEVID	Length (=2 to 131)	Element ID

Figure 47—Piconet services information element format

The DEVID field identifies the DEV correpsonding to the the Piconet Services field. If the PNC is sending the IE as the aggregate capabilities of the piconet, the DEVID field shall be set to the BcstID.

The Vendor OUI field is defined in 7.4.7.

The Piconet Services field is used to indicate the application layer capabilities of the DEV indicated by the DEVID. The content of the Piconet Services field is outside of the scope of this standard.

7.4.17 Vendor specific

The Vendor Specific IE shall be formatted as illustrated in Figure 48.

Octets: L_n	**3**	**1**	**1**
Vendor specific information	Vendor OUI	Length (=3+L_n)	Element ID

Figure 48—Vendor specific information element format

The Vendor OUI is defined in 7.4.7.

The Vendor Specific Information field is defined by the vendor identified in the Vendor OUI field. Its use by a DEV is outside of the scope of this standard.

7.5 MAC command types

The MAC command types are listed in Table 50 and are described in the following subclauses. If the column labeled “Associated” in Table 50 is marked with an “X” then that command shall only be sent by a DEV that is associated in the piconet. If the column labeled “Secure membership (if required)” in Table 50 is marked with an “X” and secure membership is required for the piconet, then that command shall only be sent by a DEV that has established secure membership with the PNC in the piconet. Because a neighbor PNC is not a member of the piconet, it sends only non-secure commands. The PNC or destination DEV shall ignore any command from a DEV that is not allowed to be sent as indicated in Table 50. The PNC or destination DEV shall transmit an Imm-ACK following reception of the frame if the ACK Policy field is set to Imm-ACK.

For peer-to-peer communications, if the DEV has established a secure relationship with a peer DEV, and the “Secure membership (if required)” column is marked with an “X”, that command shall be sent to the peer DEV with a secure command using the key specified in Table 62.

Table 50—Command types

Command type hex value b15–b0	**Command name**	**Subclause**	**Associated**	**Secure membership (if required)**
0x0000	Association request	7.5.1.1		
0x0001	Association response	7.5.1.2	X	
0x0002	Disassociation request	7.5.1.3	X	
0x0003	Request key	7.5.2.1	X	X
0x0004	Request key response	7.5.2.2	X	X
0x0005	Distribute key request	7.5.2.3	X	X

Table 50—Command types (*Continued*)

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)
0x0006	Distribute key response	7.5.2.4	X	X
0x0007	PNC handover request	7.5.3.1	X	X
0x0008	PNC handover response	7.5.3.2	X	X
0x0009	PNC handover information	7.5.3.3	X	X
0x000A	PNC information request	7.5.4.1	X	X
0x000B	PNC information	7.5.4.2	X	X
0x000C	Security information request	7.5.4.3	X	X
0x000D	Security information	7.5.4.4	X	X
0x000E	Probe request	7.5.4.5	X	
0x000F	Probe response	7.5.4.6	X	
0x0010	Piconet services	7.5.5.1	X	
0x0011	Announce	7.5.5.2	X	
0x0012	Channel time request	7.5.6.1	X	X
0x0013	Channel time response	7.5.6.2	X	X
0x0014	Channel status request	7.5.7.1	X	X
0x0015	Channel status response	7.5.7.2	X	X
0x0016	Remote scan request	7.5.7.3	X	X
0x0017	Remote scan response	7.5.7.4	X	X
0x0018	Transmit power change	7.5.7.5	X	X
0x0019	PS set information request	7.5.8.1	X	X
0x001A	PS set information response	7.5.8.2	X	X
0x001B	SPS configuration request	7.5.8.3	X	X
0x001C	SPS configuration response	7.5.8.4	X	X
0x001D	PM mode change	7.5.8.5	X	X
0x001E	Security message	7.5.9.1	X	
0x001D–0x00FF	Reserved			
0x0100–0xFFFF	Vendor specific	7.5.9.2	X	

Unless otherwise stated in the command descriptions, in all commands sent between the PNC and a DEV, the SEC bit shall be set to zero when the piconet is operating in security mode 0. When the piconet is operating in security mode 1, unless otherwise stated in the command description, the SEC bit shall be set to one and the DEV shall be a secure member of the piconet in order to send that command. The ACK Policy field shall be set to Imm-ACK for all commands unless otherwise stated in the command description.

The only commands that may be fragmented are:

— PNC information, as described in 7.5.4.2
— PNC handover information, as described in 7.5.3.3
— PS set information response, as described in 7.5.8.2

The fragmentation and defragmentation of these commands employ the same method as that used for data frames, as described in 8.7.

7.5.1 Association and disassociation commands

These commands are used by a DEV to join a piconet and by a DEV or the PNC to end a DEV's membership in the piconet.

7.5.1.1 Association request

The Association Request command shall be formatted as illustrated in Figure 49. The SEC field in the Frame Control field shall be set to zero. The DestID shall be set to the PNCID. The SrcID shall be set to either the UnassocID, as described in 7.2.3, or the DEV's newly allocated DEVID, as described in 8.3.1.

octets: 1	**2**	**7**	**8**	**2**	**2**
DEV utility	ATP	Overall capabilities	DEV address	Length (=18)	Command type

Figure 49—Association request command format

The DEV Address field is the address of the DEV, as described in 7.1, requesting association.

The Overall Capabilities field is defined in 7.4.11.

The Association Timeout Period (ATP) field is maximum amount of time in milliseconds that the association relationship will be maintained in the absence of communication between the PNC and DEV, as described in 8.3.4.

The DEV Utility field shall be formatted as illustrated in Figure 50.

bits: b7–b2	**b1**	**b0**
Reserved	Neighbor PNC	Piconet services inquiry

Figure 50—DEV utility field format

The Piconet Services Inquiry bit shall be set to one if the associating DEV is requesting that the PNC send the Piconet Services command, as described in 7.5.5.1, and shall be set to zero otherwise.

The Neighbor PNC bit shall be set to one if the DEV intends to be a neighbor PNC, as described in 8.2.6, in the current piconet and shall be set to zero otherwise.

7.5.1.2 Association response

The Association Response command shall be formatted as illustrated in Figure 51. The ACK Policy field shall be set to no-ACK. The SEC field in the Frame Control field shall be set to zero. The DestID shall be set to the UnassocID, as described in 7.2.3. The SrcID shall be set to the PNCID.

octets: 0 or L_n	**1**	**2**	**1**	**8**	**2**	**2**
Vendor specific IE	Reason code	ATP	DEVID	DEV address	Length (=12 or 12+L_n)	Command type

Figure 51—Association response command format

The DEV Address field is the address of the DEV, as described in 7.1, requesting association.

The DEVID field is the identifier allocated to the DEV if the association is successful. If this field contains the UnassocID, the DEV is not allowed to associate for the reason indicated in the reason code. For the successful association of a neighbor PNC, the DEVID shall be one of the reserved NbrIDs, as described in 7.2.3.

The ATP field contains the finalized value for the Association Timeout Period in milliseconds. This value may be different from that requested by the DEV in its Association Request command if the PNC is not able to support the value requested.

The valid values of the Reason Code are:

— 0 –> Success
— 1 –> Already serving maximum number of DEVs
— 2 –> Lack of available channel time to serve the DEV
— 3 –> Channel too severe to serve the DEV
— 4 –> PNC turning off with no PNC capable DEV in the piconet
— 5 –> Neighbor piconet not allowed
— 6 –> Channel change in progress
— 7 –> PNC handover in progress
— 8 –> Association denied
— 9–255 –> reserved

The Vendor Specific IE is defined in 7.4.17 and is an optional field.

7.5.1.3 Disassociation request

The Disassociation Request command shall be formatted as illustrated in Figure 52.

octets: 1	**2**	**2**
Reason code	Length (=1)	Command type

Figure 52—Disassociation request command format

The valid reason codes are:

— 0 –> ATP expired
— 1 –> Channel too severe to serve the DEV
— 2 –> PNC unable to service DEV
— 3 –> PNC turning off with no PNC capable DEV in the piconet

— 4 –> DEV leaving piconet

— 5–255 –> reserved

The Disassociation Request command shall use the secure command frame format if the DEV is a secure member of the piconet.

7.5.2 Security commands

This set of commands is used to establish the security and privacy functions between a DEV and the PNC and between DEVs in the piconet.

7.5.2.1 Request key

The Request Key command is used to request a payload protection key from the key originator. The SEC field in the Frame Control field shall be set to one.

This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Request Key command shall be formatted as illustrated in Figure 53.

octets: 2	**2**
Length (=0)	Command type

Figure 53—Key request command format

7.5.2.2 Request key response

The Request Key Response command is used by a key originator in a security relationship to send the requested key in an encrypted format to the requesting DEV. The SEC field in the Frame Control field shall be set to one. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The integrity code is generated using the management key that is shared between the requesting DEV and the key originator. The Request Key Response command shall be formatted as illustrated in Figure 54.

octets: L_n	**2**	**2**	**2**
Encrypted key	SECID	Length (=2+L_n)	Command type

Figure 54—Request key response command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID is used to identify the type of key and the key originator and is defined in 7.2.7.2.

The Encrypted Key field is defined in the symmetric key security operations, as described in 10.3.1.

7.5.2.3 Distribute key request

The Distribute Key Request command is used to transmit a key to another DEV. The SEC field in the Frame Control field shall be set to one. This command may have the ACK Policy field set to no-ACK only if the source ID is the PNCID. This command shall be protected using the management key that is shared between

the requesting DEV and the key originator. The Distribute Key Request command shall be formatted as illustrated in Figure 55.

octets: L_n	**2**	**2**	**2**
Encrypted key	SECID	Length (=2+L_n)	Command type

Figure 55—Distribute key request command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID is used to identify the type of key and the key originator and is defined in 7.2.7.2.

The Encrypted Key field is defined in the symmetric key security operations, as described in 10.3.1.

7.5.2.4 Distribute key response

The Distribute Key Response command is used in a distribute key protocol to inform the key originator whether or not the key was properly received. The SEC field in the Frame Control field shall be set to one. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Distribute Key Response command frame structure shall be formatted as illustrated in Figure 56.

octets: 2	**2**	**2**
SECID	Length (=2)	Command type

Figure 56—Distribute key response command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID is used to identify the type of key and the key originator and is defined in 7.2.7.2.

7.5.3 PNC handover commands

These commands are used to handover PNC responsibilities.

7.5.3.1 PNC handover request

The PNC shall use this command to hand over its responsibility to another DEV in the piconet that is capable of being a PNC. The PNC Handover Request command shall be formatted as illustrated in Figure 57.

octets: 1	**1**	**1**	**1**	**2**	**2**
Handover status	Number of PS sets	Number of CTRqBs	Number of DEVs	Length (=4)	Command type

Figure 57—PNC handover request command format

The Number Of DEVs field indicates the total number of DEVs that are currently members of the piconet. In addition, this field indicates the number of DEV Information records that will be transfered from the old PNC to the new PNC via the PNC Information command, as described in 7.5.4.2.

The Number Of CTRqBs field is the number of CTRqBs, excluding requests for asynchronous channel time, currently being served by the PNC that will be transferred from the old PNC to the new PNC via the PNC Handover Information command, as described in 7.5.3.3.

The Number Of PS sets field indicates the total number of PS sets that will be transferred from the old PNC to the new PNC via the PS Set Information Response command, as described in 7.5.8.2.

'The Handover Status field shall be set to zero when the PNC is starting the PNC handover process with the destination DEV. It shall be set to one if the PNC is cancelling the handover process with the destination DEV.

7.5.3.2 PNC handover response

The format of the PNC Handover Response command shall be as illustrated in Figure 58.

octets: 1	**2**	**2**
Reason Code	Length(=1)	CommandType

Figure 58—PNC handover response command format

The Reason Code field indicates that the new PNC is either ready to take over as the new PNC or that it will be unable to become the PNC. The valid Reason Code values are:

0x00 –> Success, ready for handover
0x01-0xEC –> Success, member of parent piconet with DEVID equal to Reason Code value
0xED-0xF6 –> Reserved
0xF7-0xFC –> Success, associated in parent piconet with NbrID equal to Reason Code value
0xFE –> Handover refused, unable to join parent piconet
0xFF –> Handover refused, unable to act as PNC for more than one piconet

7.5.3.3 PNC handover information

The PNC Handover Information command shall be formatted as illustrated in Figure 59.

octets: 2	**12**	**1**	**...**	**2**	**12**	**1**	**2**	**2**
Next beacon	CTRqB-n	DEVID	...	Next beacon	CTRqB-1	DEVID	Length (=15*n)	Command type

Figure 59—PNC handover information command format

The DEVID field contains the identifier of the source of the CTRqB that follows in the command.

The CTRqB field is defined in 7.5.6.1. Note that asynchronous CTRqBs are not passed in this command, thus the Num Targets field in the CTRqB is always one. Consequently, the CTRqBs will all be a fixed length.

The Next Beacon field indicates the the beacon number, as described in 7.3.1.1, of the next superframe when this CTRq will be allocated.

7.5.4 Information request commands

This set of commands is used to obtain information about another DEV in the piconet. The PNC Information Request and PNC Information commands are used to retrieve data about any or all of the currently associated DEVs in the piconet. The Security Information Request and Security Information commands are used to retrieve security information about any or all of the currently associated DEVs in the piconet. The Probe Request and Probe Response commands are used to retrieve IEs from a specific DEV in the piconet.

7.5.4.1 PNC information request

The DestID for the PNC Information Request command shall be the PNCID. The PNC Information Request command shall be formatted as illustrated in Figure 60.

octets: 1	2	2
Queried DEVID	Length (=1)	Command type

Figure 60—PNC information request command format

The Queried DEVID field contains the DEVID of the DEV whose information is being requested from the PNC. If the value of this field is BcstID, then the DEV is requesting information regarding the entire list of associated DEVs from the PNC.

7.5.4.2 PNC information

This command may be sent either as a response to the PNC Information Request command by a DEV or it may be sent unsolicited. In either case the SrcID shall be the PNCID. This command may be sent either in a directed command frame to a DEV or it may be sent in a broadcast command frame meant for all DEVs in the piconet. If the DestID is BcstID, then the ACK Policy field shall be no-ACK. The PNC Information command shall be formatted as illustrated in Figure 61.

octets: 20	---	20	20	2	2
DEV-m info	…	DEV-2 info	DEV-1 info	Length(=m*20)	Command type

Figure 61—PNC information command format

The DEV Info field shall be formatted as illustrated in Figure 62.

octets: 1	2	7	1	1	8
System wake beacon interval	ATP	Overall capabilities	DEV info utility	DEVID	DEV address

Figure 62—Format of a DEV info field in a PNC information command

The DEV Address field contains the address of the DEV, as described in 7.1, corresponding to the DEVID.

The DEVID field contains the ID assigned to the DEV by the PNC. This field shall not contain the the BcstID, the UnassocID, the McstID or the reserved IDs, as described in 7.2.3.

The DEV Info Utility field shall be formatted as illustrated in Figure 63.

bits: b7-b1	b0
Reserved	Membership Status

Figure 63—DEV Info Utility field format

The Membership Status bit shall be set to zero if the DEV is associated but is not a secure member of the piconet and shall be set to one if the DEV is associated and a secure member of the piconet.

The Overall Capabilities field shall be formatted as illustrated in Figure 39 and is defined in 7.4.11.

The ATP field is defined in 7.5.1.1

The System Wake Beacon Interval field, as described in 7.5.8.3, is the value that the DEV sent to the PNC via the SPS Configuration Request command, as described in 7.5.8.3

7.5.4.3 Security information request

The Security Information Request command enables a DEV to request security information regarding a single DEV or all DEVs. The Security Information Request command shall be formatted as illustrated in Figure 64.

octets: 1	**2**	**2**
Queried DEVID	Length (=1)	Command type

Figure 64—Security information request command format

The Queried DEVID field indicates the DEV whose security information is being requested. If the value of this field is the BcstID, then the DEV is requesting all of the security information maintained by the target DEV.

7.5.4.4 Security information

The Security Information command shall be formatted as illustrated in Figure 65.

octets: L_m	**---**	**L_2**	**L_1**	**1**	**1**	**2**	**2**
DEV-m security record	…	DEV-2 security record	DEV-1 security record	Sequence number	Total number of frames	Length (=2+L_1 +L_2+…+L_m)	Command type

Figure 65—Security information command

The Total Number of Frames field indicates the number of frames that will be sent to complete this request.

The Sequence Number field specifies the number of frames that have been sent prior to this frame by this DEV in response to the initial request. Thus the first frame has a Sequence Number of 0 while the last frame has a Sequence Number equal to the Total Number of Frames minus one.

A given Security Record field shall be formatted as illustrated in Figure 66.

octets: L_n	**2**	**1**	**8**	**2**
Verification info	Verification info length (=L_n)	DEVID	DEV address	Length (=10+L_m2+L_n)

Figure 66—Format of an Security Record field in an Security Information command

The DEV Address field contains the address of the DEV, as described in 7.1, corresponding to the DEVID.

The DEVID field contains the ID assigned to the DEV by the PNC. If the DEV is not currently associated in this piconet, the field shall be set to the UnassocID. This field shall not contain the broadcast or multicast DEVIDs.

The Verification Info Length field indicates the length of the verification information that is included in the Security Record field. If this length is zero, no Verification Info field shall be included.

The Verification Info field specifies the security information that may be used to verify the identity of that particular DEV.

7.5.4.5 Probe request

The Probe Request command is used either to request information about a DEV or to see if a DEV is still present in the piconet. This command may be exchanged between any two DEVs in the piconet according to the rules outlined in Table 51 and Table 52. The individual IEs used in this frame are described in 7.4. The Probe Request command shall be formatted as illustrated in Figure 67.

octets: 2	**4**	**2**	**2**
Request index	Information requested	Length (=6)	Command type

Figure 67—Probe request command format

The Information Requested field shall be formatted as illustrated in Figure 68.

bits: b31–b1	**b0**
IEs requested	IE request type

Figure 68—Information requested field format

The IE Request Type field indicates the format of the IEs requested field. This field shall be set to zero if the IEs Requested field is a bitmap and shall be set to one if the IEs Requested field is a binary encoding of the IE's ID.

If the IE Request Type field indicates that the IEs Requested field is a bitmap, then the sender shall set a value of one in a bit to request the IE that corresponds to the bit position. Otherwise the sender shall set the bit to zero. The bit position for an IE is same as the value of the element-ID for that IE. That is, the bit position of *n* in information request field corresponds with the IE whose element ID, Table 48, is *n*.

If the IE Request Type field indicates that the rest of the bits are binary coded, then the IEs Requested field contains the element ID of the IE that is being requested by the sender of this command from its intended recipient.

Both the IE Request Type field and the IEs Requested field shall be set to zero when the source DEV is not requesting any information from the destination DEV.

If the IEs Requested field indicates that the CTA Status IE, as described in 7.4.10, is being requested from the destination DEV, the first octet of the Request Index field is set to the stream index of the stream for which CTA information is requested. If the Request Index field is set to zero, the DEV is requesting information about all isochronous streams directed to the requesting DEV and to the BcstId and McstId. If the Information Requested field indicates that the CTA Status IE is not being requested from the destination DEV, the Request Index field has no meaning and shall be set to zero.

Table 51 lists the rules that shall apply to requesting IEs from another DEV based on the identity of the originator of the request.

Table 51—Rules for requesting IEs in a Probe Request command

Information element	Subclause	PNC allowed to request?	DEV allowed to request?
Channel time allocation	7.4.1	Shall not request	Shall not request
BSID	7.4.2	Shall not request	May request
Parent piconet	7.4.3	Shall not request	May request
DEV association	7.4.4	Shall not request	Shall not request
PNC shutdown	7.4.5	Shall not request	Shall not request
Piconet parameter change	7.4.6	Shall not request	Shall not request
Application specific	7.4.7	May request	May request
Pending channel time map (PCTM)	7.4.8	Shall not request	May request
PNC handover	7.4.9	Shall not request	Shall not request
CTA status	7.4.10	Shall not request	Shall not request
Capability	7.4.11	May request	May request
Transmit power parameters	7.4.12	May request	May request
PS status	7.4.13	Shall not request	Shall not request
Continued wake beacon (CWB)	7.4.14	Shall not request	Shall not request
Overlapping PNID	7.4.15	May request	Shall not request
Piconet services	7.4.16	May request	May request
Vendor specific or reserved	7.4.17	May request	May request

7.5.4.6 Probe response

The Probe Response command is used to return information about a DEV to a requesting DEV. The individual IEs used in this frame are described in 7.4. The Probe Response command shall be formatted as illustrated in Figure 69.

octets: n	2	2
IEs provided	Length (=n)	Command type

Figure 69—Probe response command format

The IEs Provided field contains the IEs, as described in 7.4, that the source DEV of this command is providing to the destination. The elements themselves may be placed in any order.

Table 52 lists the rules that shall apply to responding to a request for an IE based on the sender of the request.

Table 52—Rules for responding to requests in Probe commands

Information element	Subclause	DEV receives request from DEV	DEV receives request from PNC	PNC receives request from DEV
Channel time allocation	7.4.1	Shall ignore	Shall ignore	Shall ignore
BSID	7.4.2	Shall ignore	Shall ignore	Shall respond
Parent piconet	7.4.3	Shall ignore	Shall ignore	Shall respond
DEV association	7.4.4	Shall ignore	Shall ignore	Shall ignore
PNC shutdown	7.4.5	Shall ignore	Shall ignore	Shall ignore
Piconet parameter change	7.4.6	Shall ignore	Shall ignore	Shall ignore
Application specific	7.4.7	May respond	May respond	May respond
Pending channel time map (PCTM)	7.4.8	Shall ignore	Shall ignore	Shall respond
PNC handover	7.4.9	Shall ignore	Shall ignore	Shall ignore
CTA status	7.4.10	Shall ignore	Shall ignore	Shall ignore
Capability	7.4.11	Shall respond	Shall respond	Shall respond
Transmit power parameters	7.4.12	Shall respond	Shall respond	Shall respond
PS status	7.4.13	Shall ignore	Shall ignore	Shall ignore
Continued wake beacon (CWB)	7.4.14	Shall ignore	Shall ignore	Shall ignore
Overlapping PNID	7.4.15	Shall ignore	May respond	Shall ignore
Piconet services	7.4.16	May respond	May respond	May respond
Vendor specific or reserved	7.4.17	May respond	May respond	May respond

7.5.5 Information announcement commands

7.5.5.1 Piconet services

The Piconet Services command is sent by the PNC to provide information about the application layer capabilities of all of the DEVs in a piconet. The Piconet Services command shall be formatted as illustrated in Figure 70.

octets: L_n	…	L_2	L_1	2	2
Piconet services IE n	…	Piconet services IE-2	Piconet services IE-1	Length (=sum of L_1-L_n))	Command type

Figure 70—Piconet services command format

The Piconet Services IE is defined in 7.4.16.

7.5.5.2 Announce

The Announce command is used to send unrequested information about a DEV to one or more DEVs in the piconet. The individual IEs used in this frame are described in 7.4. The Announce command shall be formatted as illustrated in Figure 71.

octets: n	2	2
IEs Provided	Length (=n)	Command type

Figure 71—Announce command format

The IEs Provided field contains the IEs, as described in 7.4, that the source DEV of this command is providing to the destination. The elements themselves may be placed in any order.

Table 53 lists the rules that shall apply to sending an unrequested IE based on the sender of the request.

Table 53—Rules for sending IEs in an Announce command

Information element	Subclause	PNC allowed to send?	DEV allowed to send?
Channel time allocation	7.4.1	Shall not send	Shall not send
BSID	7.4.2	Shall not send	Shall not send
Parent piconet	7.4.3	Shall not send	Shall not send
DEV association	7.4.4	May send	Shall not send
PNC shutdown	7.4.5	May send	Shall not send
Piconet parameter change	7.4.6	May send	Shall not send
Application specific	7.4.7	May send	May send
Pending channel time map (PCTM)	7.4.8	May send	Shall not send
PNC handover	7.4.9	May send	Shall not send
CTA status	7.4.10	May send	Shall not send
Capability	7.4.11	May send	May send
Transmit power parameters	7.4.12	May send	May send
PS status	7.4.13	May send	Shall not send
Continued wake beacon (CWB)	7.4.14	Shall not send	Shall not send
Overlapping PNID	7.4.15	Shall not send	May send
Piconet services	7.4.16	May send	May send
Vendor specific or reserved	7.4.17	May send	May send

7.5.6 Channel time allocation request, modification, and termination commands

This group of commands is used for the request, modification, termination and grant of channel time within the CTAP.

7.5.6.1 Channel time request

The Channel Time Request (CTRq) command may be used to request, modify, or terminate CTAs corresponding with either isochronous streams or asynchronous data traffic. The Channel Time Request command structure shall be formatted as illustrated in Figure 72. The DEV that sends this command is the originator and is seeking from the PNC channel time allocations during which to communicate with a target DEV or DEVs.

octets: 12–138	**…**	**12–138**	**12–138**	**2**	**2**
CTRqB-n	…	CTRqB-2	CTRqB-1	Length (=sum of n CTRqBs)	Command type

Figure 72—Channel time request command format

Each Channel Time Request block (CTRqB) corresponds to a channel time request. If the DEV is making a request for asynchronous channel time where the destinations share CTAs, then there shall be only one asynchronous CTRqB in the command and it shall be the last CTRqB in the CTRq command. The Channel Time Request block for a given CTRq shall be formatted as illustrated in Figure 73.

octets: 1	**1**	**2**	**2**	**1**	**1**	**1**	**1**	**1–127**	**1**
Desired number of TUs	Minimum number of TUs	CTRq TU	CTA rate factor	CTRq control	Stream index	Stream request ID	DSPS set index	Target ID list	Num targets

Figure 73—Channel time request block field format

The Num Targets field indicates the number of target DEVIDs in the target ID list. For isochronous requests, i.e. stream index not equal to the asynchronous stream index, the num targets field shall be set to one. For asynchronous requests, the num targets fields shall take on values from 1 to 127.

The Target ID List field is a series of DEVIDs with which the originating DEV seeks to establish communications by requesting channel time allocations from the PNC.

The DSPS Set Index field is used to identify the DSPS set with which the CTRq corresponds, if the CTRq is for a DSPS allocation. Only valid DSPS set indices, as described in 7.5.8.3, are allowed for a DSPS allocation request. Otherwise, the field shall be set to zero and shall be ignored on reception.

The Stream Request ID field is used to uniquely identify the DEV's request before it receives a stream index from the PNC. If the channel time request is for a new isochronous stream, then the stream request ID is a non-zero identifier generated by the originating DEV that is unique among the DEV's channel time requests. The stream request ID shall remain constant during the entire frame exchange sequence for establishing a new stream. If the channel time request is to modify or terminate an existing stream or the request is for an asynchronous allocation, the stream request ID shall be set to zero and shall be ignored on reception.

The Stream Index field is defined in 7.2.5. In the case where the DEV is requesting the creation of an isochronous stream, it is set to the unassigned stream value, as described in 7.2.5, by the originating DEV. In the case where the DEV is requesting the reservation or termination of an asynchronous channel time, it is set to the asynchronous stream value, as described in 7.2.5. When the stream index is other than the unassigned stream index or asynchronous stream index value, this CTRq is a request to modify or terminate an existing CTA. In the case where the DEV is requesting a specific MCTA interval, as described in 8.4.3.3, the stream index shall be set to the MCTA stream value, as described in 7.2.5.

The CTRq Control field shall be formatted as illustrated in Figure 74.

bits: b7	b6	b5	b4	b3	b2-b0
Target ID list type	CTA rate type	CTA type	PM CTRq type	Reserved	Priority

Figure 74—CTRq control format

The Priority field is defined in Table A.1.

The PM CTRq Type field indicates the type of request. It shall be set to zero to request an ACTIVE channel time allocation and shall be set to one to request a DSPS channel time allocation. For sub-rate allocations, an ACTIVE allocation puts no restriction on the superframe of the first CTA. A DSPS allocation synchronizes all CTAs with the DSPS set awake superframes of the DSPS set specified by the DSPS index. The value of the CTA Rate Factor shall be no smaller than the DSPS set's wake beacon interval.

The CTA Type field indicates whether a pseudo-static CTA is being requested. The CTA Type field shall be set to one if the channel time request is for a pseudo-static CTA and shall be set to zero otherwise.

The CTA Rate Type field indicates whether a super-rate CTA or a sub-rate CTA is being requested. The CTA Rate Type field shall be set to one for a sub-rate CTA and zero for a super-rate CTA.

The CTA Rate Factor field in conjunction with the CTA Rate Type field specifies the frequency at which the requesting DEV would like the PNC to allocate channel time.

For instance, in the case where the CTA Rate Type field is set to zero, a value indicating a super-rate CTA request, and the CTA Rate Factor field contains a value N greater than zero, the requesting DEV is requesting super-rate CTAs from the PNC. If these super-rate CTAs, are allocated by the PNC, they will appear N times per superframe. A PNC shall support at least 8 CTAs per stream in the same superframe.

In the case where the CTA Rate Type field is set to one, a value indicating a sub-rate CTA request, and the CTA Rate Factor field contains a non-zero value N, the requesting DEV is requesting sub-rate CTAs from the PNC. If these sub-rate CTAs, are allocated by the PNC, they will appear in the beacon once every N superframes. The CTA Rate Factor in this case shall be limited to powers of 2 (i.e. 2, 4, 8, ...), up to and including the value of 65536, which shall be represented by a CTA Rate Factor equal to zero.

If the CTRqB is for an MCTA interval, only the CTA Rate Factor field and stream index shall be interpreted by the PNC. All other fields except the stream index and num targets fields shall be set to zero.

The Target ID List Type field shall be set to zero for asynchronous group channel time requests and shall be set to one for individual asynchronous channel time requests, as described in 8.5.2.1.

The CTRq Time Unit (TU) field indicates the unit of time that the DEV is using for the CTA(s) it is requesting. This allows the PNC to know the units of channel time the DEV is able to make use of so that the PNC will efficiently allocate channel time. The resolution of this field is 1 μs and therefore has a range of [0–65535] μs.

For an isochronous request, the Minimum Number of TUs field indicates the minimum number of CTRq TUs per CTA required by the originating DEV to support the stream.

For an isochronous request, the Desired Number Of TUs field indicates the number of CTRq TUs per CTA that is desired by the requesting DEV. The Desired Number Of TUs field shall be greater than or equal to the Minimum Number Of TUs field.

For isochronous requests, the Minimum Number Of TUs and the Desired Number Of TUs are the number of TUs per CTA Rate Factor requested by the DEV. In the case of a super-rate allocation, it is the number of TUs requested in each superframe. In the case of a sub-rate allocation it is the number of TUs requested in each of the superframes containing the sub-rate CTA. For example, a request for a Minimum Number Of TUs of 4 with a sub-rate CTA Rate Factor of 4 indicates that the DEV is requesting 4 TUs every fourth superframe.

For an asynchronous request, the concatenation of the Minimum Number Of TUs field and the Desired Number Of TUs field indicates the total number of TUs that are requested for this allocation, i.e. it is interpreted as a single, 2-octet field. Note that this is a request for a total amount of time rather than a recurring use of time in the superframe. The use of this field is defined in 8.5.2.

7.5.6.2 Channel time response

The Channel Time Response command shall be formatted as illustrated in Figure 75.

octets: 1	**1**	**1**	**1**	**2**	**2**
Reason code	Available number of TUs	Stream index	Stream request ID	Length (=4)	Command type

Figure 75—Channel time response command format

The Stream Request ID field is defined in 7.5.6.1.

The Stream Index field is defined in 7.5.6.1.

The Available Number Of TUs field is used by the PNC to indicate to the requesting DEV the number of TUs per CTA Rate Factor it has assigned to the requested isochronous stream. In the case of a super-rate allocation, it is the number of TUs assigned in each superframe. In the case of a sub-rate allocation it is the number of TUs assigned in each of the sub-rate superframes.

For isochronous CTRqs, if the Available Number Of TUs is greater than or equal to the Minimum Number Of TUs requested and less than or equal to the Desired Number Of TUs requested, then the requesting DEV is informed that there is channel time available. If, however, the Available Number Of TUs field is less than the Minimum Number Of TUs requested, then the requesting DEV is informed that the PNC is unable to fulfill the DEV's request for channel time. In this case, the Available Number of TUs will be set by the PNC to the number of TUs that the PNC would have been able to allocate for this request, as described in 8.5.1.1.

For asynchronous stream requests, the response frame is sent only if the PNC is unable to fulfill the request, in which case the available number of TUs field is set to zero.

The Reason Code field indicates whether a channel time request was successful or unsuccessful. The codes assignable to this field are:

0–> Success
1–> Success, DEV in PS mode
2–> Target DEV unassociated
3–> Target DEV not a member
4–> Priority unsupported
5–> Stream terminated by PNC
6–> Stream terminated by target DEV
7–> Channel time unavailable
8–> Destination DEV in power save mode
9–> Unable to allocate as pseudo-static CTA

10–> Superframe overloading
11–> Requested super-rate or sub-rate unsupported
12–> Request denied
13–255–> Reserved

7.5.7 Channel status commands

This group of commands is used to request and provide information about the remote DEV's view of the channel and to change the transmitter power based on the current channel conditions.

7.5.7.1 Channel status request

The Channel Status Request command shall be formatted as illustrated in Figure 76. This command may be sent by any DEV in the piconet to any other DEV in the piconet, including the PNC, to request the current channel condition as experienced at the target DEV. This command may also be sent by the PNC as a broadcast frame, i.e. the DestID set to the BcstID. The Channel Status Request command shall be formatted as illustrated in Figure 76.

octets: 2	**2**
Length (=0)	Command type

Figure 76—Channel status request command format

7.5.7.2 Channel status response

The Channel Status Response command shall be formatted as illustrated in Figure 77. This command is sent by the target DEV in response to the originating DEV's request to let the originating DEV know the current channel condition at the target DEV. When the DestID of this command is the PNCID, the values in the command shall correspond to all frames exchanged by the DEV with other DEVs in the piconet. When the DestID of this command is a non-PNC DEVID, the values in the command shall correspond to the frames exchanged between the requesting DEV and the target DEV. The Channel Status Response command shall be formatted as illustrated in Figure 77.

octets: 2	**2**	**2**	**2**	**2**	**2**	**2**
RX frame loss count	RX frame error count	RX frame count	TX frame count	Measurement window size	Length (=10)	Command type

Figure 77—Channel status response command format

The Measurement Window Size field is the number of superframes during which the measurements were taken. The minimum Measurement Window Size for a valid measurment for this command shall be 2 superframes. A Measurement Window Size of zero indicates that the responding DEV does not provide channel status statistics.

The TX Frame Count field contains the total number of frames, not including Imm-ACK frames, that were transmitted by the sender of this command to the destination of this command. This count includes all transmission attempts, including retransmissions of the same frame.

The RX Frame Count field contains the total number of frames, not including Imm-ACK frames, that were correctly received by the sender of this command. Only the directed frames transmitted by the destination of this command intended for the sender of this command are included.

The RX Frame Error Count field contains the total number of frames, not including Imm-ACK frames, that were received in error by the sender of this command from the destination of this command. A frame is considered to have been received in error if the header passes the HCS validation but the frame body fails the FCS validation.

The RX Frame Loss Count field contains the number of frames in streams with the ACK Policy field set to no-ACK, not including Imm-ACK frames, that were determined by the originator of the command to have been lost. The originating DEV determines this for a particular stream index by observing gaps in the Fragmentation Control field of received frames. These numbers are accumulated for all streams between the originating DEV and the target DEV.

7.5.7.3 Remote scan request

The SrcID for Remote Scan Request command shall be the PNCID. The Remote Scan Request command shall be formatted as illustrated in Figure 78.

octets: 1	**...**	**1**	**2**	**2**
Channel n	...	Channel 1	Length (=n)	Command type

Figure 78—Remote scan request command format

The Channel Number field indicates the channels that are to be scanned. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY the mapping is defined in 11.2.3.

7.5.7.4 Remote scan response

The DestID for the Remote Scan Response command shall be the PNCID. The Remote Scan Response command shall be formatted as illustrated in Figure 79.

octets: l	**n**	**1**	**m**	**1**	**1**	**2**	**2**
Optional IE	Remote piconet description set	Number of piconets	Channel rating list	Number of channels	Reason code	Length (=1 or 3+n+m+l)	Command type

Figure 79—Remote scan response command format

The allowed Reason Code field values are:

0–> Success

1–> Request denied

2–> Invalid channel requested

3–255–> Reserved

If the request is denied, then the Remote Scan Response command shall include only the Command Type, Length, and Reason Code fields.

The Number Of Channels field indicates the number of channels that were scanned by the remote DEV.

The Channel Rating List field contains a list of channel indices ordered from best (least interference) to worst (most interference). The Channel Rating List field shall be formatted as shown in Figure 80.

octets:1	**...**	**1**
Worst channel index	...	Best channel index

Figure 80—Channel rating list

The Number Of Piconets field indicates the number of piconets that were found. If the DEV did not find any piconets, then the number shall be set to zero and there shall be no remote piconet description sets in the command.

The Remote Piconet Description Set is a collection of one or more Remote Piconet Description fields. Each Remote Piconet Description field shall be formatted as shown in Figure 81.

octets: 2-40	**2-34**	**8**	**1**	**1**	**1**	**2**
Parent piconet IE	BSID IE	PNC address	Piconet Type	Channel index	Scanned frame type	PNID

Figure 81—Remote piconet description field

The PNID is the identifier of the piconet that was found by the DEV.

The Scanned Frame Type field indicates the type of frame that was received by the DEV with the piconet information. The allowed values are:

0–> The DEV found the PNID in a beacon.
1–> The DEV found the PNID only in a non-beacon frame.
2–255 –> Reserved

The Channel Index field indicates the PHY channel where the information was found. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY the mapping is defined in 11.2.3.

The Piconet Type field indicates the type of piconet that was found. The allowed values are:

— 0–> Non-dependent piconet
— 1–> Dependent piconet
— 2–255–> Reserved

The PNC Address field is defined in 7.3.1. If the DEV found a beacon, it shall put the PNC Address from the beacon into the command. Otherwise, the PNC Address field shall be set to zero.

The BSID IE is defined in 7.4.2. If the DEV found a beacon, it shall put this IE into the command. Otherwise, it shall include a BSID IE with zero length, i.e. only the element ID and length fields.

The Parent Piconet IE is defined in 7.4.3. If the DEV found a beacon from a dependent piconet, it shall put this IE into the command. Otherwise, it shall include a Parent Piconet IE with zero length, i.e. only the element ID and length fields.

The Optional IE is provided for future expansion of the standard. The IE is not defined in this revision of the standard and so the source DEV may omit the IE in the Remote Scan Response command. The IE shall be ignored upon reception by the destination DEV of this command.

7.5.7.5 Transmit power change

The Transmit Power Change command shall be formatted as illustrated in Figure 82. This command is used to request a change in the transmit power of a DEV.

octets: 1	2	2
TX power change	Length (=1)	Command type

Figure 82—Transmit power change command format

The TX Power Change field contains the requested TX power level change in dB at the destination DEV in 2s complement format. For example, a +2 db change in the TX power level is 0x02 while a –2 dB TX power level change is encoded as 0xFE.

7.5.8 Power management commands

These commands are used to enable DEVs to conserve power as well as by other DEVs that want to know when DEVs using power management will be available for communication.

7.5.8.1 PS set information request

The PS Set Information Request command is used to acquire information from the PNC regarding the number of PS sets and their structure. The PS Set Information Request command shall be formatted as illustrated in Figure 83.

2	2
Length (=0)	Command type

Figure 83—PS set information request command format

7.5.8.2 PS set information response

The PS Set Information Response command shall be formatted as illustrated in Figure 84.

octets: 8-39	...	8-39	1	1	2	2
PS set structure n	...	PS set structure 1	Number of current PS sets	Max supported PS sets	Length (=(1+ sum of length of n PS set structure)	Command type

Figure 84—PS set information response command format

The Max Supported PS Sets field indicates the number of PS sets supported by the PNC of this piconet.

The Number of Current PS Sets field is a count of the number of PS Set Structures in this command as well as the number of currently active PS sets in the piconet.

Each PS Set Structure shall be formatted as illustrated in Figure 85.

octets: 1-32	**1**	**1**	**2**	**2**	**1**
DEVID bitmap	Start DEVID	Bitmap length	Next wake beacon	Wake beacon interval	PS set index

Figure 85—PS set structure field format

When the PS Set Index field is zero, the DEVID Bitmap field lists the DEVs currently in APS mode, if any. When the PS Set Index field is one, the DEVID Bitmap field indicates the DEVs currently in PSPS mode, if any. When the PS Set Index field is any value between 0x02 and 0xFD, inclusive, the DEVID Bitmap field indicates the DEVs currently in this particular DSPS set. The PS set indices are defined as:

0x00 –> APS set
0x01 –> PSPS set
0x02–0xFD –> DSPS sets
0xFE-> Unallocated DSPS set
0xFF –> Reserved

The Wake Beacon Interval field is defined in 7.5.8.3. This field is set to the system wake beacon interval for PS sets 0 and 1. For all other PS sets it is set to the wake beacon interval of that DSPS set. Note that the wake beacon interval has no interpretation for PS set 0, as described in 8.13.3.

The Next Wake Beacon field is defined in 7.5.8.4. This field is set to the next system wake beacon for PS sets 0 and 1. For all other PS sets it is set to the next wake beacon of that DSPS set. Note that the Next Wake Beacon field has no interpretation for PS set 0, as described in 8.13.3.

The Bitmap Length field contains the number of octets in the DEVID bitmap. This field shall take on values from 1 to 32, inclusive.

The Start DEVID field indicates the DEVID corresponding to the lsb in the DEVID bitmap.

The DEVID Bitmap field is a bitmap of the DEVIDs in a specific PS set. A value of 0 in a bitmap position indicates that the DEV corresponding to that DEVID is not part of the PS set. A value of 1 in a bitmap position indicates that the DEV corresponding to that DEVID is in the PS set.

7.5.8.3 SPS configuration request

The SPS Configuration Request command is used to set up and manage SPS set memberships for SPS DEVs currently participating or requesting to participate in one or more SPS modes. The SPS Configuration Request command shall be formatted as illustrated in Figure 86.

octets: 2	**1**	**1**	**2**	**2**
Wake beacon interval	SPS set index	Operation type	Length (=4)	Command type

Figure 86—SPS configuration request command format

The Operation Type field indicates whether a DEV is requesting either to join or to leave an existing SPS set. The valid operation types are:

0–> join
1–> leave
2–255 reserved

The SPS Set Index field is used to identify the SPS set the requesting DEV wants to create/join, join, configure, or leave. The SPS Set Index field shall not be set to zero (APS) in this command.

Table 54 lists the interpretation of the fields for various combinations of the Operation Type field and SPS Set Index field.

Table 54—SPS configuration request command parameter entries

Operation type	SPS set index value	Wake beacon interval	Comments
0 (create/join)	Unallocated DSPS set (0xFE)	Any valid DSPS wake beacon interval value. This value is decoded by the PNC upon reception	The unallocated DSPS Set Index (0xFE) shall be used to request the PNC to establish a new DSPS set.
0 (join)	0x01	Any valid PSPS system wake beacon interval value. This value is decoded by the PNC upon reception	The PSPS mode is permanently associated with PS set index 0x01.
0 (join)	0x02–0xFD	Shall be set to zero and ignored by the PNC upon reception.	The DEV is requesting to join an existing DSPS set.
1 (leave)	0x01–0xFD	Shall be set to zero and ignored by the PNC upon reception	The DEV is requesting to leave the indicated SPS set.

The Wake Beacon Interval field contains the number of superframes, including the current one, between wake beacons, as described in 8.13. For example, a wake beacon interval of 8 indicates that the DEV is requesting a wake beacon every 8th beacon, Figure 137. Valid values for the wake beacon interval for either the PSPS or DSPS ranges are in powers of 2 (e.g. 2, 4, 8, ...). Furthermore, the wake beacon interval shall have a value between 2 and 256 for PSPS and between 2 and 65536 for DSPS. Because the value 65536 can not be represented with 2 octets, a wake beacon interval of 0 shall represent the interval value 65536.

7.5.8.4 SPS configuration response

The SPS Configuration Response command is sent by the PNC as a response to an SPS Configuration Request command received from a DEV. The SPS Configuration Response command shall be formatted as illustrated in Figure 87.

octets: 2	1	1	2	2
Next wake beacon	SPS set index	Reason code	Length (=4)	Command type

Figure 87—SPS configuration response command format

The Reason Code field contains the result of the SPS Configuration Request command. The valid reason codes are:

0–> Success
1–> Already member
2–> Invalid SPS set (i.e. attempting either to configure PS set 0 or to join a non-existing set.)
3–> Set creation failed
4–> Unique Wake Beacon interval required
5–255 Reserved

The Reason Code field is set to zero (Success), if the create or join operation is successful. If the SPS Set Index field has been set to zero or any value not representing an SPS set, the reason code shall be set to 'Invalid SPS Set'. If a DEV requests to join a DSPS set where it is already a member, the reason code shall be set to 'Already member'. However, a DEV is allowed to make multiple requests to join the PSPS set. This has the effect of updating the DEV's desired system wake beacon interval value. If a DSPS set creation fails for any other reason than listed above, the reason code shall be set to ‘Set creation failed.’

The SPS Set Index is defined in 7.5.8.3.

The Next Wake Beacon field indicates the beacon number, as described in 7.3.1.1, of either the next system wake beacon for the PSPS mode when the SPS set index is set to one, or the next DSPS wake beacon when the SPS set index is greater than or equal to 0x02 and less than or equal to 0xFD.

7.5.8.5 PM mode change

The Power Management (PM) Mode Change command shall be formatted as illustrated in Figure 88.

octets: 1	**2**	**2**
PM mode	Length (=1)	Command type

Figure 88—PM mode change command format

The PM Mode field shall be set as follows:

0–> ACTIVE mode

1–> APS mode

2–> SPS mode

3–255–> reserved

7.5.9 Special commands

7.5.9.1 Security message

The Security Message command is used to send security related information to another DEV in the piconet. The SEC field in the Frame Control field shall be set to zero. The Security Message command shall be formatted as illustrated in Figure 89.

Octets: L_n	**3**	**2**	**2**
Security information	Vendor OUI	Length (=3+L_n)	Command type

Figure 89—Vendor specific command format

The Vendor OUI is defined in 7.4.7.

The Secuity Information field contains security related information defined by the vendor identified in the vendor OUI field. Its use by a DEV is outside of the scope of this standard.

7.5.9.2 Vendor specific

The Vendor Specific command shall be formatted as illustrated in Figure 90.

Octets: L_n	**3**	**2**	**2**
Vendor specific data	Vendor OUI	Length (=3+L_n)	Command type

Figure 90—Vendor specific command format

The Vendor OUI is defined in 7.4.7.

The Vendor Specific Data field is defined by the vendor identified in the vendor OUI field. Its use by a DEV is outside of the scope of this standard.

8. MAC functional description

8.1 Introduction

This clause provides a description of the MAC functionality. The process of starting and maintaining a piconet is described in 8.2. Subclause 8.3 describes the method used to join and leave a piconet via the association and disassociation process. The channel access mechanisms are described in 8.4. The channel time request and allocation procedures are described in 8.5. The required synchronization for the operation of the piconet and the channel access is described in subclause 8.6.

The fragmentation and defragmentation of the MSDUs is described in 8.7. The acknowledgement and retransmission mechanisms are described in 8.8. Peer discovery is discussed in 8.9.

To overcome the problems due to overlapping piconets and interference in a given channel, the PNC moves the operations of the piconet to a new channel. The process of dynamic channel selection (DCS) is described in 8.11.1. Each DEV in the piconet chooses its transmission power based on the current channel conditions. The operation and negotiations required for transmit power control (TPC) are described in 8.11.2. Subclause 8.12 deals with the issues of multiple data rates supported by the PHY.

The DEVs in the piconet are able to employ power saving techniques to reduce their power consumption. The operation and the negotiations required for power management are described in 8.13. The use of the ASIE is described in 8.14 and a table of the MAC parameters is given in 8.15.

In this clause, unless otherwise indicated, receiving a frame means that the PHY has successfully received a frame over the medium and both the FCS and HCS calculations match their respective data as defined in 7.2.6 and 11.2.9.

Asynchronous MSDUs shall be delivered to the FCSL in the order of reception.

An example MSC is shown in Figure 91 that illustrates two MLME requests and the associated timeouts. In the first case, the request completes before the timeout expires and so the confirm returns with the ResultCode set equal to COMPLETED. In the second case, the requested action does not complete before the timeout expires and so the confirm primitive is returned with the ResultCode set equal to TIMEOUT.

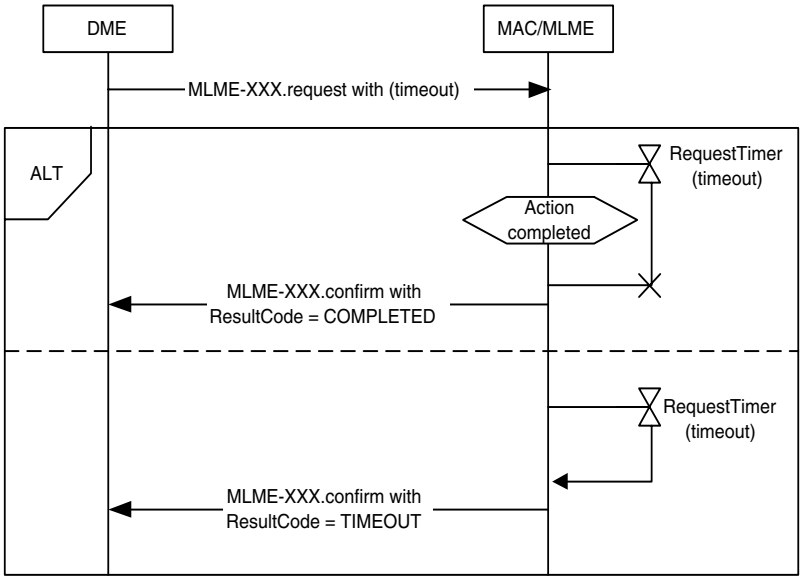

Figure 91—MSC showing examples of primitive timers

8.2 Starting, maintaining and stopping piconets

An 802.15.3 piconet begins when a PNC capable DEV takes on the responsibility of being the PNC. This subclause describes the processes involved in starting and maintaining the piconet. The types of piconets defined in this standard are:

a) Independent piconet: A piconet with no dependent piconets and no parent piconets.
b) Parent piconet: A piconet that has one or more dependent piconets.
c) Dependent piconet: A piconet that requires a time allocation in another piconet, called the parent piconet, and is synchronized with the parent piconet's timing.

There are two types of dependent piconets:

a) Child piconet: A dependent piconet where the PNC is a member of the parent piconet.
b) Neighbor piconet: A dependent piconet where the PNC is not a member of the parent piconet.

8.2.1 Scanning through channels

All DEVs shall use passive scanning to detect an active piconet. That is, DEVs shall be in receive mode for a period of time in a channel, as specified in the MLME-SCAN.request, to look for beacon frames from a PNC. If open scan is specified in the MLME-SCAN.request, the DEV searches for any beacon frame. If open scan is not specified, the DEV shall ignore all received frames not matching the PNID and BSID parameters contained in the request.

In addition, the searching DEV shall collect statistics on each channel scanned and save them in the ChannelRatingList as described in 6.3.2.

DEVs search for piconets by traversing through all the indexed channels indicated in the MLME-SCAN.request. A DEV may search the channels in any order as long as all valid channels are contained in the search pattern. The result of a scan shall include information on any parent, child, as described in 8.2.5,

or 802.15.3 neighbor, as described in 8.2.6, piconets that were detected. This provides a complete inventory of each channel.

While searching, if any frame is received, the searching DEV shall stay in the channel for a minimum of mMinChannelScan from the time of reception of first frame and look for a beacon from the PNC. If the DEV finds only a frame and no beacon it shall report it as a part of the MLME-SCAN.confirm primitive. The DEV shall scan all indicated channels to find piconets before returning the scan information via the MLME-SCAN.confirm primitive.

Figure 92 illustrates the message flow for a successful scan operation.

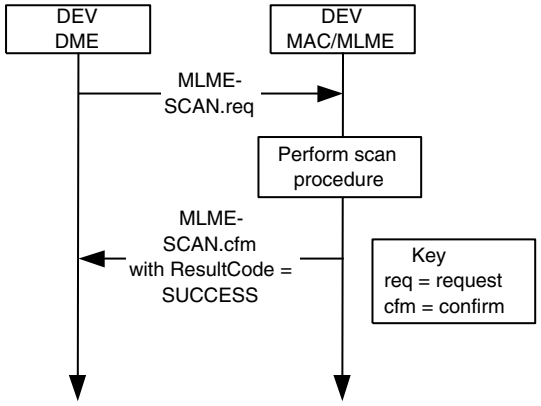

Figure 92—MSC for scan operation

8.2.2 Starting a piconet

A DEV that is instructed to start a piconet through MLME-START.request, as described in 6.3.3.1, shall try only to start its own piconet and shall not attempt to associate with an existing piconet. The DME shall have recently completed a scan procedure and will have chosen the channel in which to start the piconet.

The DME should choose the channel with the least amount of interference to start the piconet based on the ChannelRatingList returned in the MLME-SCAN.confirm primitive, as described in 6.3.2.2.

Once the DME has chosen a channel, it shall issue the MLME-START.request primitive with the chosen channel. The DEV shall listen to the channel for mMinChannelScan duration to determine if the channel is still clear. If, at the end of this listening period, the DEV determines that the channel is clear, the DEV, now the PNC, shall commence broadcasting its beacon once every superframe duration. If, however, the DEV determines that the channel is no longer clear, it shall issue an MLME-START.confirm with a ResultCode indicating a failure to start the piconet. The DME then has options that include sending another MLME-START.request with a different ChannelIndex to start a piconet in a different channel, associating as a regular DEV and requesting the formation of a dependent piconet. When the piconet starts, the PNC allocates an additional DEVID to itself for the purposes of exchanging data with other DEVs that become members of the established piconet.

Once a PNC has established a piconet, the PNC should periodically allocate channel time in the CTAP so that there is quiet time for it to scan the channel for other piconets. If the PNC detects another piconet in the same channel that is not one of its own child or neighbor piconets, it may take action to improve coexistence with the other piconet. Some of the actions the PNC may take include:

— Changing to a different channel, as described in 8.11.1.
— Becoming a child or neighbor piconet of the other piconet, as described in 8.2.5 and 8.2.6.
— Reducing the piconet's transmit power, as described in 8.11.2.

The PNC shall continue in its role for as long as it desires or until the PNC determines that an associating DEV is more capable, at which time the PNC may decide to initiate the PNC handover procedure, as described in 8.2.3.

Figure 93 illustrates the message flow for a successful start operation.

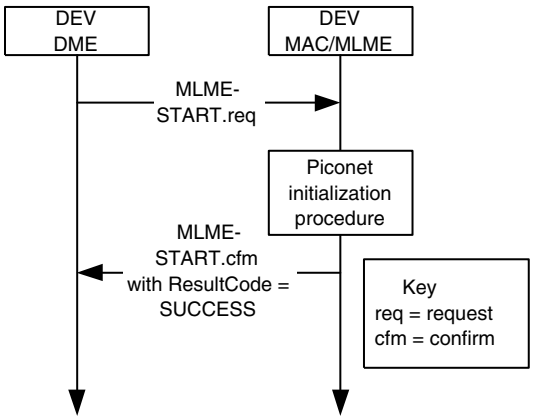

Figure 93—MSC for starting a piconet

8.2.3 PNC handover

When the PNC leaves the piconet or when it transfers its PNC functionality to another DEV, the PNC shall attempt to choose a DEV that is capable of being a PNC as its successor. The PNC Capable bit in the PNC Capabilities field, as described in 7.4.11, is used to indicate that a DEV is capable of being a PNC. The PNC shall use the information in the PNC Capabilities field of the other DEVs in the piconet with the evaluation criteria defined in Table 55 to select the most qualified PNC capable DEV that is currently a member of the piconet to be the new PNC. The PNC shall send a PNC Handover Request command, as described in 7.5.3.1, to its chosen DEV with the parameters specified in 7.5.3.1. If the piconet is not a dependent piconet, the DEV shall accept the nomination and be prepared to receive the piconet information records. If the DEV is currently the PNC of a dependent picoent, it may refuse the request by sending a PNC Handover Response command to the PNC with the Reason Code field set to 'Handover refused, unable to act as PNC for more than one piconet'. If both the current and the new PNC are members of the same dependent piconet, then the DEV shall accept the handover request unless it is unable to join the parent piconet as either a regular DEV or a neighbor PNC. In the case where the DEV is unable to join the parent piconet, the DEV sends the PNC Handover Response command to the PNC with the Reason Code field set to 'Handover refused, unable to join parent piconet.'

When the handover is initiated, the HandoverStatus is STARTED. If the handover timer expires, a PNC Handover Request command shall be sent to the DEV with a HandoverStatus of CANCELLED. In addition, if the DEV sees a PNC Shutdown IE from the PNC during the handover process, it knows that the handover was cancelled.

The PNC shall allocate channel time with the chosen PNC capable DEV as the destination for the purpose of transferring information about the DEVs in the piconet and their current CTRqBs. When the channel time has been allocated, the PNC shall first send a PNC Information command, as described in 7.5.4.2, to the chosen PNC capable DEV. In the PNC Information command, the PNC shall include all DEVs that are associ-

ated in the piconet, including any associated neighbor PNCs, the DEV personality of the PNC and an entry for the PNCID. Once the PNC has successfully sent this command it shall then begin sending all of the current channel time requests to the chosen PNC capable DEV using a PNC Handover Information command, as described in 7.5.3.3. Once the PNC has successfully sent the PNC Handover Information command, it shall send a PS Set Information Response command, as described in 7.5.8.2, to the new PNC. The PNC may fragment the PNC Information, PNC Handover Information and PS Set Information Response commands using the process described in 8.7.

The PNC Handover Information command shall not be sent if the PNC has indicated in the PNC Handover Request command that it does not have any CTRqBs to transfer. The PS Set Information Response command shall not be sent if the PNC has indicated in the PNC Handover Request command that it does not have any PS sets to transfer.

The handover procedure will transfer all information necessary for the new PNC to take over except:

— Asynchronous CTRqBs will not be transferred. All DEVs with asynchronous data to send need to send a new Channel Time Request command, as described in 7.5.6.1, to the new PNC after it has sent its first beacon.
— CTA locations are not transferred, except in the preceding beacons.

Once the chosen PNC capable DEV has received the required information from the current PNC, it shall respond to the current PNC with a PNC Handover Response command, as described in 7.5.3.2. This will signal to the current PNC that the chosen PNC capable DEV is ready to take over as the new PNC. After the PNC receives the PNC Handover Response command, it shall place a PNC Handover IE, as described in 7.4.9, in the beacon.

Meanwhile the chosen PNC capable DEV, after receiving an ACK to its PNC Handover Response command, will prepare to broadcast its first beacon as the new PNC. The current PNC shall place the PNC Handover IE in the beacon with the Handover Beacon Number field set to the beacon number of the superframe in which the new PNC will send its first beacon. After sending the last beacon, the old PNC relinquishes control of the piconet, generates an MLME-PNC-HANDOVER.confirm to its DME, and stops generating beacons. The new PNC shall broadcast its first beacon at the time the beacon would have been sent by the old PNC. This time may vary from the actual time due to clock inaccuracies of the old and new PNCs. The new PNC shall start sending beacons with the time token counter set to one more than the time token of the last beacon that will be sent by the old PNC. The new PNC shall begin using the PNCID as the SrcID for all beacon or command frames transmitted. The new PNC shall use the PNCID or its previously assigned DEVID as the SrcID for all data frames transmitted. When the PNC handover is successful, the association of the remaining DEVs with the piconet is unaffected and hence they are not required to re-associate with the new PNC.

The PNC shall ensure that the handover announcement complies with the rules for beacon announcements in 8.6.4.

Figure 94 illustrates the message sequence of a PNC handover to a PNC capable DEV which is currently a member of the piconet.

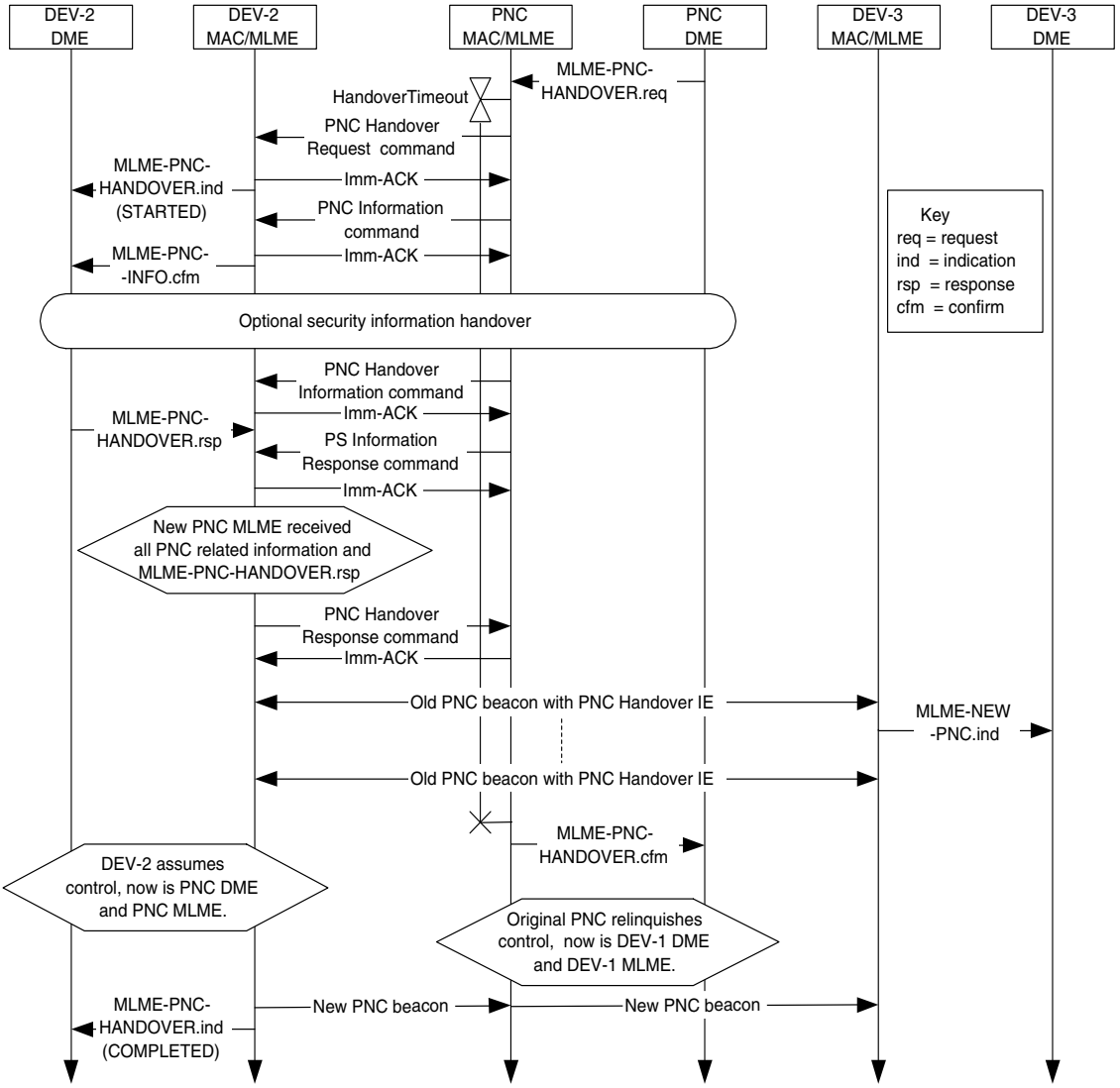

Figure 94—PNC handover MSC

The optional security information handover referenced in Figure 94 is shown in Figure 149.

In the MSC, the MLME-PNC-HANDOVER.response is sent when the DME is ready for the handover and is not tied to the arrival of the PNC Handover Information commands or PS Set Information Response commands.

Note that the PNC handover process should not stop any of the isochronous data connections. Figure 94 illustrates only the handover process and hence does not show other traffic. However, since the PNC needs to allocate sufficient channel time to transfer the DEV and CTRqB data, some of the data traffic may be affected depending on the traffic conditions within the piconet.

When a DEV joins a piconet, the PNC shall compare the PNC Capabilities field of the new DEV to its own. If the PNC Des-Mode bit is set in the new DEV but not in the current PNC, the current PNC shall perform PNC handover. If the new DEV is more qualified to be the PNC, based on the PNC selection criteria in Table 55, the PNC may perform PNC handover.

Table 55—Comparison order of fields for PNC handover

Order	Information	Note
1	PNC Des-mode bit in PNC capabilities field	PNC Des-mode=1 is preferred
2	SEC bit in PNC capabilities field	SEC=1 is preferred
3	PSRC bit in PNC capabilities field	PSRC=1 is preferred
4	Max associated DEVs	Higher value is preferred
5	Max CTRqBs	Higher value is preferred
6	Transmitter power level (PHY dependent)	Higher value is preferred
7	MAX PHY rate (PHY dependent)	Higher value is preferred
8	DEV address	Higher value is preferred

As Table 55 shows, PNC Des-Mode is the top priority field in the PNC selection criteria. Since the PNC Des-Mode is the highest priority, a DEV with this bit set is more likely to become the PNC of the piconet. Thus, this bit should be set if it is desirable for the DEV to be the PNC of the piconet. If only one DEV has the PNC Des-Mode bit set, then that DEV would become the PNC.

If the piconet is using mode 1 security, then the new PNC and DEVs need to follow the security procedures in in addition to the handover process described here.

A dependent PNC receiving a parent beacon with a PNC Handover IE may immediately insert the Piconet Parameter Change IE into its beacons with the Change Type field set to MOVE, as described in 7.4.6 and the Superframe Timing field set to zero. A member of a dependent piconet that receives this Piconet Parameter Change IE in the beacon from the dependent PNC shall not transmit after the superframe which has a beacon number equal one less than the Change Beacon Number field in the Piconet Parameter Change IE until it has correctly received a beacon from its PNC.

8.2.4 Dependent PNC handover

The dependent PNC handover process begins in the same manner as a regular PNC handover, as described in 8.2.3, with the current PNC sending a PNC Handover Request command to the target DEV that it has selected to become the new PNC, as shown in Figure 95. In this and the two subsequent figures, the identities PNC, DEV-2 and DEV-3 are all relative to the dependent piconet and not the parent piconet. If the target DEV is not a member of the parent piconet, then that DEV shall begin the association process to join the parent piconet and, if required, become a secure member of the parent piconet. The target DEV may request to associate with the parent piconet as either a neighbor PNC or a member of the piconet. While the target DEV is attempting to join the parent piconet, the current dependent PNC shall send the target DEV the information about all of the DEVs with a PNC Information command, all of the current channel time requests with a PNC Handover Information command and the power save information, if any, using a PS Set Information Response command. The target DEV may also request the transfer of any security information at this point using a Security Information Request command. Note that the transfer of this information will not interfere

with the target DEV's association because the former occurs only during the time reserved for the dependent piconet while the latter occurs only during the time reserved for the parent piconet.

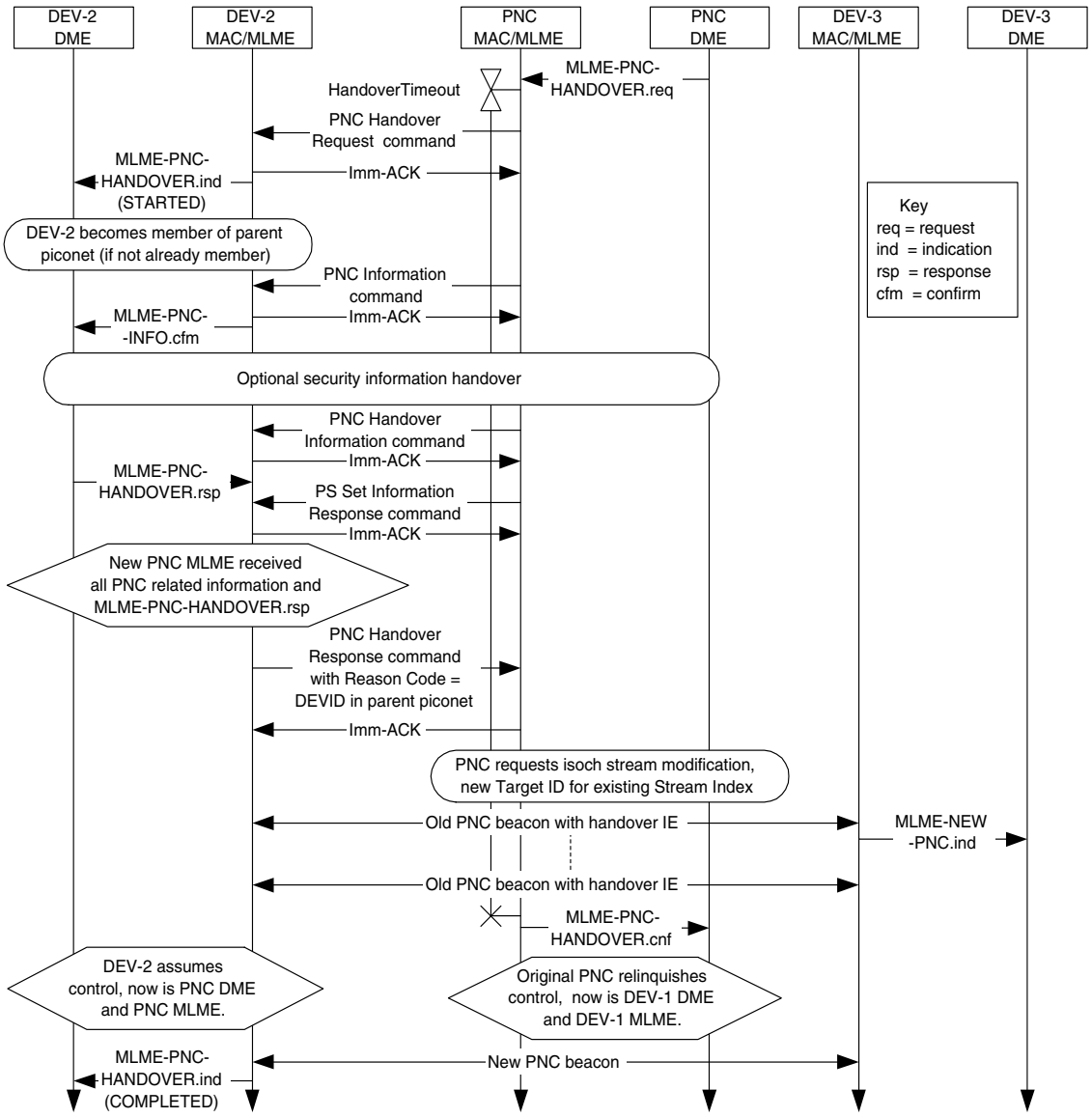

Figure 95—Successful PNC handover in a dependent piconet

Figure 95 references three processes not included in the figure. The MSC for the optional security information handover is shown in Figure 149. The association process that is used to become member of the parent piconet is shown in Figure 102 while the stream modification request to handover control of a CTA is shown in Figure 119.

Once the transfer of the information is complete and the target DEV has joined the parent piconet, the target DEV shall send a PNC Handover Response command to the dependent PNC with a Result Code set to the DEVID that was assigned to it by the parent PNC. This informs the dependent PNC that the target DEV is ready to take over control of the piconet. At this point, the dependent PNC shall send a Channel Time Request command to the parent PNC to handover the control of the dependent piconet CTA to the new

dependent PNC, as described in 8.5.1.2. Once the parent PNC changes the SrcID and DestID of the dependent piconet CTA, the current dependent PNC shall either complete the handover process to the new PNC or it shall shutdown the dependent piconet because it will not be able to regain control of the CTA.

After the dependent PNC receives a beacon from the parent PNC with the change in the SrcID of the dependent piconet CTA, the current dependent PNC shall begin placing the PNC Handover IE in its beacon, using the procedure indicated in as described in 8.2.3, with the Handover Beacon Number field set to indicate the first beacon that will be sent by the new PNC. The last superframe controlled by the current dependent PNC will be the one in which the beacon number is one less than the Handover Beacon Number field. The following superframe will begin when the target DEV, now the new dependent PNC, sends its first beacon.

There are multiple points in the handover process where it is possible for the handover to fail. The current dependent PNC may cancel the handover process up until the time when it requests that the parent PNC handover control of the dependent piconet CTA to the new dependent PNC. The dependent PNC cancels the process by sending a PNC Handover Request command to the target DEV with the Handover Status field set to one to indicate that the process has been cancelled.

The handover process will also fail if the target DEV fails to join the parent piconet. If the target DEV attempts to join as a neighbor PNC but the parent PNC does not support neighbor PNCs or does not wish to allow any more neighbor PNCs, then the association request by the new dependent PNC will be rejected. In that case, the DEV may also try to join as a regular DEV, in which case the dependent piconet would become a child piconet after the handover process.

If the target DEV fails to join the parent piconet as either a regular DEV or a neighbor PNC, it shall send a PNC Handover Response command to the dependent PNC with the Reason Code set to ‘Handover refused, unable to join parent piconet’ as illustrated in Figure 96. The target DEV may refuse the handover at any time while the dependent PNC is sending the information about the piconet.

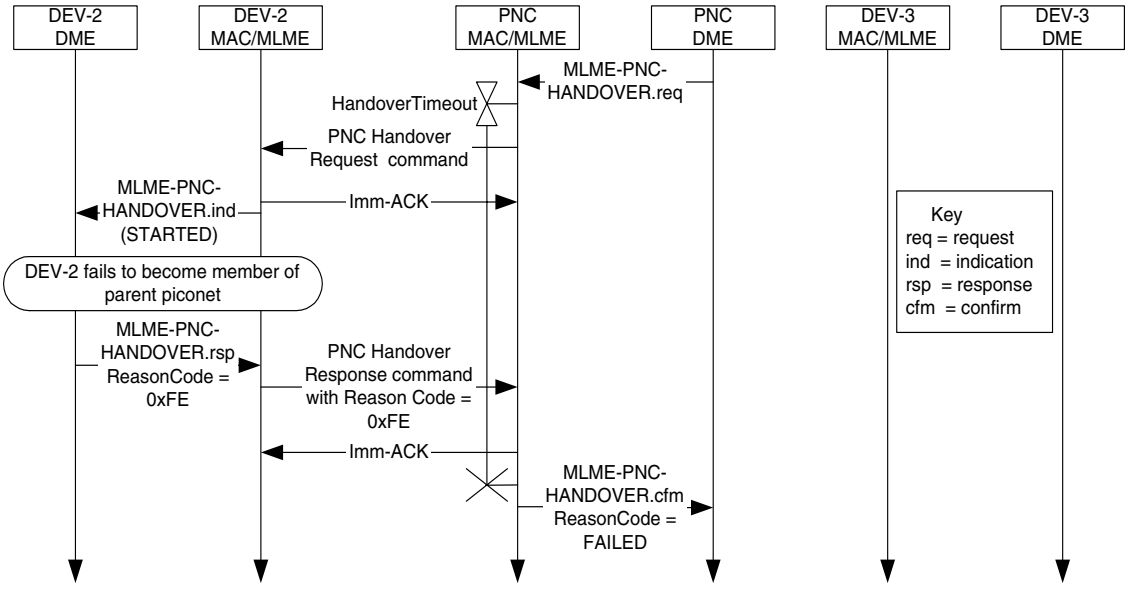

Figure 96—Failed dependent PNC handover when target DEV fails to join parent piconet

If the parent PNC rejects the request to handover control of the CTA to the new dependent PNC, the dependent PNC shall send a PNC Handover Request command to the target DEV with the Handover Status field set to one to indicate that the handover process is being cancelled, as illustrated in Figure 97.

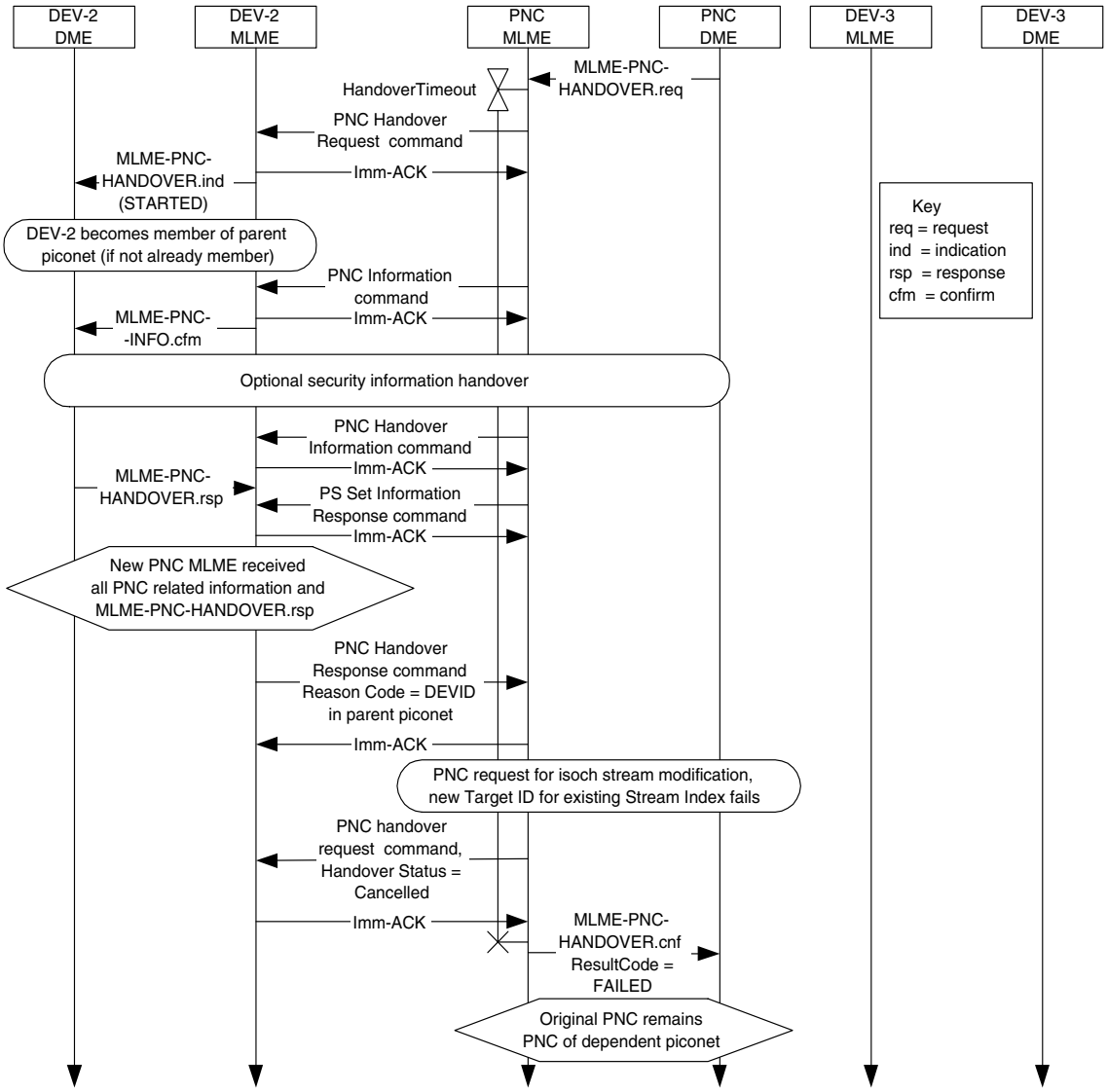

Figure 97—Failed dependent PNC handover when control for the dependent piconet CTA is handed over in the parent piconet

If the dependent PNC cancels the handover process, the target DEV may disassociate from the parent piconet. If DEV-2 joined the parent piconet as a neighbor PNC, it should disassociate from the parent piconet if the handover process is cancelled to free up that resource for other DEVs that need to form a neighbor piconet.

8.2.5 Child piconet

When a PNC capable DEV that is a member of an existing piconet wants to form a child piconet, the DEV shall use the Channel Time Request command, defined in 7.5.6.1 to request a pseudo-static private CTA. A

private CTA is a CTA for which the SrcID and DestID are identical. The DEV shall set the SrcID and TrgtID fields in the Channel Time Request command to the DEVID of the originating DEV, the Stream Index field to zero and the PM CTRq Type field to ACTIVE. The PNC will recognize this as a request for a child piconet. The PNC may allocate a private CTA for the child piconet depending on the availability of network resources, its capabilities and security policy.

If the PNC rejects the formation of a child PNC for any other reason than insufficient channel time or unable to allocate as pseudo-static, it shall send a Channel Time Response command with the Reason Code field set to ‘request denied.’

If the DEV receives a private CTA from the PNC, the DEV DME configures the child PNC parameters using the MLME-START-DEPENDENT.request and confirm primitives, as described in 6.3.3.3 and 6.3.3.4.

The DEV, now the child PNC, shall start sending its beacon in its allocated private CTA. The child PNC shall use a PNID that is distinct from the parent PNID. The child piconet beacon contains a Parent Piconet IE, as described in 7.4.3. Also included in the child piconet beacon is a private CTA for the parent piconet, using the PNCID for both the SrcID and DestID. This is provided to reserve the time, not to convey any information to the parent PNC.

It is possible for more than a single child piconet to be created from a common parent piconet. It is also possible for another dependent piconet to be formed in a child or neighbor piconet. There is no restriction in this standard on the number of levels that may be created. However, there is a practical limitation to the number of dependent piconets and the levels that are able to be supported.

The standard does not provide for the direct frame transfer between a member of a child piconet and a member of a parent piconet. Furthermore, this includes any other child piconets that are dependents of the parent. However, the child PNC DEV is a member of the parent piconet and thus may exchange data with any DEV in the parent piconet. The child PNC DEV is also a member of the child piconet and thus may exchange data with any DEV in the child piconet.

If the child PNC misses mMaxLostBeacons parent PNC beacons, the child PNC shall stop transmitting beacons to its piconet. When the child PNC hears the parent’s beacons again, it shall resume sending its beacon as long as its ATP has not expired.

Figure 98 illustrates the relationship between the parent piconet superframe and the child piconet superframe. In the figure the superframe periodicity is the same for both the child and the parent piconets. Note that the CTA positions CTA1, CTA2, et al., are not to scale in Figure 98 and so are illustrative only.

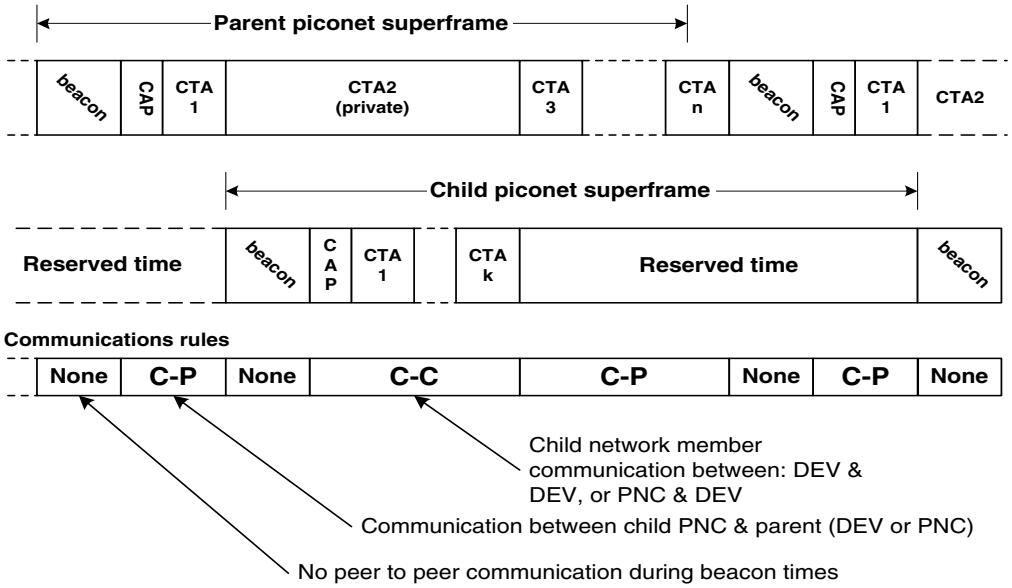

Figure 98—Parent piconet and child piconet superframe relationship

The MSC for creating a child piconet is illustrated in Figure 99. The association and channel time request processes are defined in 8.3.1 and 8.5.1.1, respectively. The MSC for the association process is shown in Figure 102 while the MSC for a channel time request is shown in Figure 114.

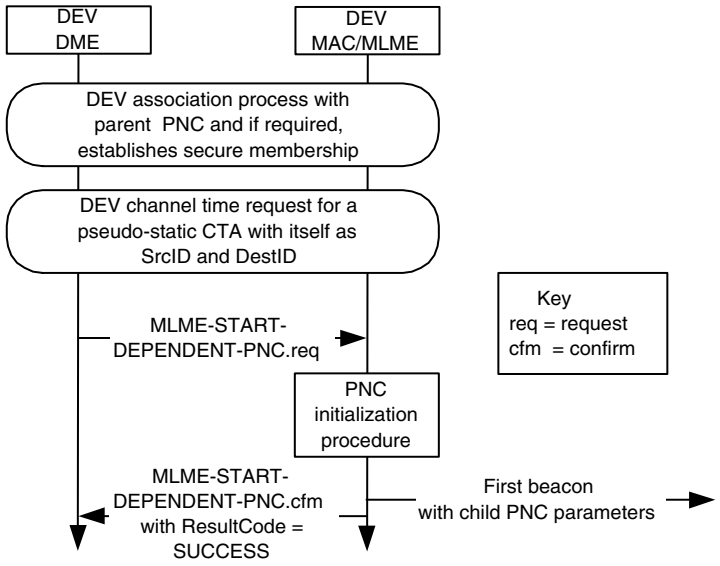

Figure 99—MSC for creating a child piconet

The child piconet is an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association, security, etc. shall be handled within the child piconet and do not involve the parent PNC.

8.2.6 Neighbor piconet

If after following the scan procedure in 8.2.1, no free channels are available, then a neighbor PNC capable DEV (i.e. a PNC capable DEV from a different system), may attempt to start a neighbor piconet on the same channel as the existing piconet. To start a neighbor piconet, the neighbor PNC capable DEV shall send an Association Request command, as described in 7.5.1.1, to the PNC. The Neighbor PNC bit in the DEV Utility field shall be set as indicated in 7.5.1.1 when the Association Request command is sent. A neighbor PNC is not required to establish a secure relationship with the parent PNC, and so a PNC operating in mode 1 may reject the request for the neighbor piconet.

If the neighbor association request is accepted, then the PNC shall set the DEVID in the Association Response command to be one of the unused NbrIDs, as described in 7.2.3. If the request was rejected, as described in 7.5.1.2, depending on the reason code, the neighbor PNC capable DEV may retry the request at a later time. If the reason code in the rejection indicates that neighbor piconets are not supported, then the neighbor PNC capable DEV should not retry the request while that DEV is the PNC of the parent piconet.

After the association request is accepted, the neighbor PNC capable DEV then sends a Channel Time Request command, as described in 7.5.6.1, to obtain a private pseudo-static CTA for the neighbor piconet. The Channel Time Request command shall have both the SrcID and TrgtID set to the NbrID that was assigned to the neighbor PNC capable DEV by the PNC in the Association Response command.

If the PNC permits the formation of a neighbor piconet and there is sufficient channel time available, the PNC shall allocate a private CTA using the NbrID as both the source and destination DEVID. After receiving this channel time allocation in the beacon, the DEV DME configures the neighbor PNC parameters using the MLME-START-DEPENDENT.request and confirm primitives, as described in 6.3.3.3 and 6.3.3.4.

The neighbor PNC capable DEV, now the neighbor PNC, shall start sending its beacon in its private CTA. The neighbor PNC shall use a PNID that is distinct from the parent PNID. If the neighbor piconet is operating an 802.15.3 piconet, its beacon shall contain a Parent Piconet IE, as described in 7.4.3.

If the neighbor PNC is operating an 802.15.3 piconet, a private CTA for the parent piconet is included in its beacon, using the PNCID for both the SrcID and DestID. This is provided to reserve the time for the parent piconet, not to convey any information to the parent PNC.

If the network operated by the neighbor PNC is not an 802.15.3 piconet, the neighbor PNC shall allow communications in its network only during the time allocated by the parent piconet using methods appropriate to its protocol. It shall ensure that its network does not have transmission outside of its allocated CTA.

If the neighbor PNC misses mMaxLostBeacons parent PNC beacons, the neighbor PNC shall stop its own transmissions and the transmissions of the DEVs in its piconet. When the neighbor PNC receives the parent PNC’s beacon again, it may return to normal operation as long as its ATP has not expired.

Figure 100 illustrates the relationship between the parent piconet superframe and the neighbor piconet superframe.

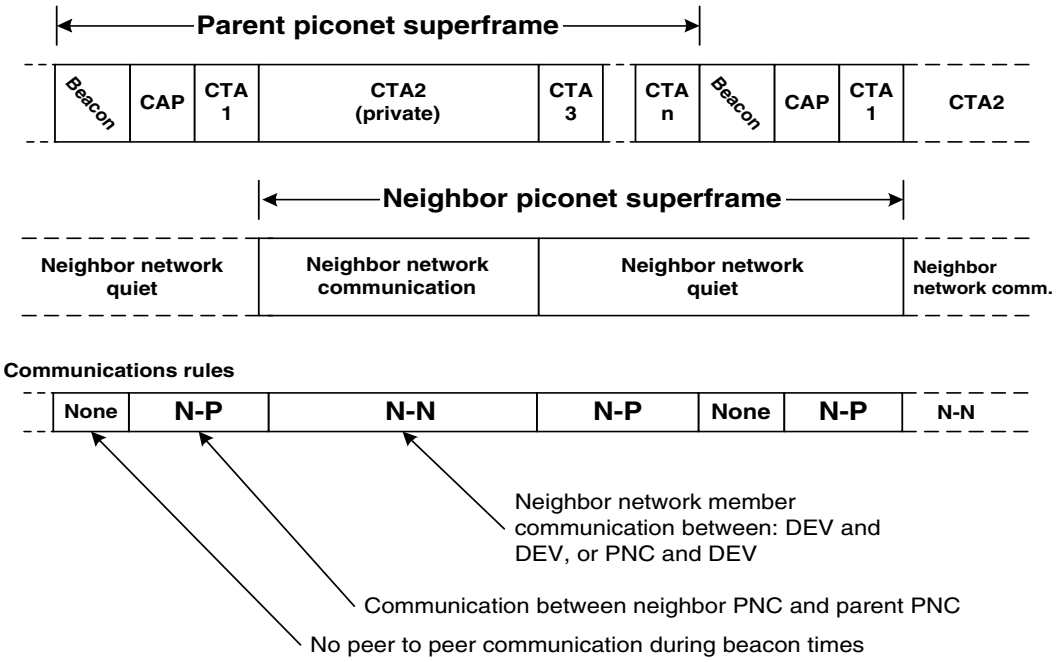

Figure 100—Parent piconet and neighbor piconet superframe relationship

The MSC for the initiation of the neighbor piconet is illustrated in Figure 101. The association and channel time request processes are defined in 8.3.1 and 8.5.1.1, respectively. The MSC for the association process is shown in Figure 102 while the MSC for a channel time request is shown in Figure 114.

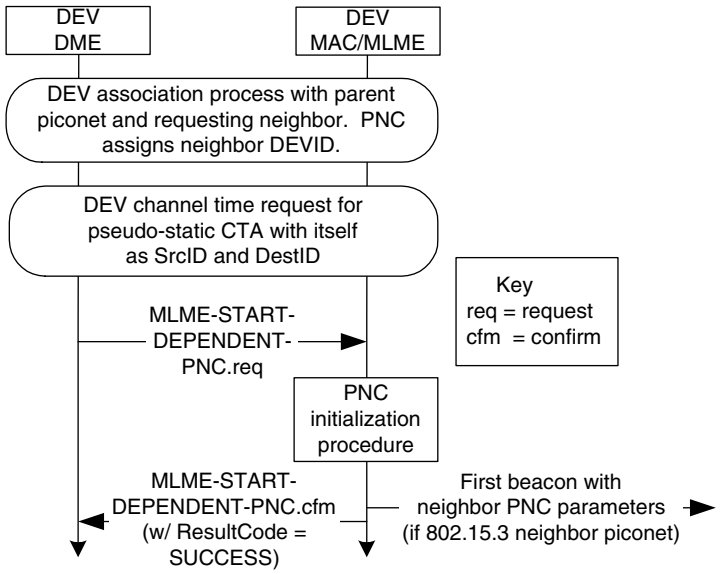

Figure 101—MSC for initiating a neighbor piconet

The neighbor piconet is an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association, security, etc. shall be handled within the neighbor piconet and do not involve the parent PNC.

The neighbor PNC is not a member of the parent piconet and shall only send the following commands to the parent PNC:

— Association Request command
— Disassociation Request command
— Channel Time Request command
— Vendor Specific commands
— Security Message command
— Any Probe Request, Probe Response or Announce commands
— Any required Imm-ACK frames

The parent PNC is not a member of the neighbor piconet.

8.2.7 Stopping piconet operations

If the PNC is going to leave the piconet, perhaps due to a shutdown request from a higher layer and there is no DEV capable of taking over as PNC or there is not sufficient time for a handover, the PNC will shutdown the piconet operations.

8.2.7.1 Stopping an independent or parent piconet

If the PNC is going to remove itself from the piconet and no other DEVs are capable of taking over as the PNC, the PNC shall places the PNC shutdown IE, as described in 7.4.5, in the beacon. The PNC shall ensure that the shutdown announcement complies with the rules for beacon announcements in 8.6.4. The only exception to this requirement is if the PNC will be shutting down and does not have enough time to wait for the next system wake beacon to complete the handover process.

If the parent PNC is not itself a dependent PNC and it is currently supporting one or more dependent piconets, the parent PNC shall select the dependent piconet PNC with the lowest DEVID to continue operation without interruption. The parent PNC shall notify the dependent PNC that it may continue operation by including the dependent DEVID in the PNC Shutdown IE, as described in 7.4.5, in the beacon. The other dependent PNCs, not seeing their DEVID in the PNC Shutdown IE, shall either cease operations, change channels or join another piconet as a dependent piconet by the time of the last beacon sent by the parent PNC. If there is time, the dependent PNC should perform the shutdown procedure for its own piconet.

If the dependent PNC whose DEVID was listed in the parent PNC's PNC Shutdown IE is coordinating an 802.15.3 piconet, it shall remove the Parent Piconet IE from its beacon frame, signifying that it is no longer a dependent piconet.

8.2.7.2 Parent PNC stopping a dependent piconet

If the parent PNC wishes to stop a child piconet, it shall terminate the stream allocated to the child piconet using the isochronous stream termination procedure, as described in 8.5.1.3. If the parent PNC wishes to stop a neighbor piconet, it shall send a Disassociation Request command, as described in 7.5.1.3, to the neighbor PNC. In either case, the dependent PNC shall either change channels, join another piconet as a dependent piconet or immediately initiate its shutdown procedure, as described in 8.2.7.1. The parent PNC shall listen for the dependent PNC shutdown beacon sequence to determine when the dependent piconet CTA should be removed. The parent PNC may set a maximum time for the completion of the dependent shutdown sequence, after which the CTA will be removed regardless of the completion of the dependent shutdown procedure. In the case of a child piconet, this timeout is set by the MLME while for a neighbor

piconet, this time is set via the MLME-DISASSOCIATE. request primitive, as described in 6.3.6.1. If the dependent PNC is a neighbor that is that is operating a piconet that is not an 802.15.3 piconet, the parent PNC shall provide the same time as it would allow for its own shutdown sequence for the neighbor PNC to cease operations as a dependent piconet of the parent piconet before removing its private CTA.

8.2.7.3 Dependent PNC termination of a dependent piconet

After stopping piconet operations for its own piconet, as described in 8.2.7.1, a child PNC shall inform its parent PNC that it no longer requires channel time for child piconet operations by sending the parent PNC a Channel Time Request command terminating the CTA used for the child piconet.

After stopping piconet operations for its own piconet, as described in 8.2.7.1, a neighbor PNC shall inform its parent PNC that it no longer requires channel time for neighbor piconet operations by sending a Disassociation Request command to the parent PNC. Upon receiving a Disassociation Request command from a neighbor PNC, a parent PNC shall remove the CTA used by the neighbor piconet.

8.2.8 Non-PNC capable DEVs

Simple DEVs may be implemented without providing support for the PNC role in a piconet. The implication of this is that these DEVs would be unable to form a piconet by themselves. Therefore, these DEVs should be of a type that are normally used only in conjunction with a DEV that provides PNC capability.

8.3 Association and disassociation with a piconet

Membership in a piconet depends on the security mode of the piconet. For a piconet that does not implement security, as described in 9.2.1, membership occurs immediately upon completion of the association process, as described in 8.3.1. For a piconet that implements security, as described in 9.2.2, membership occurs immediately upon receipt of the MLME-MEMBERSHIP-UPDATE.request primitive with the MembershipStatus parameter set to MEMBER. A DEV is removed from membership in a piconet via the disassociation process.

8.3.1 Association

Prior to the association process, the DME issues an MLME-SYNC.request and receives an MLME-SYNC.confirm.

Before a DEV has completed the association process, all frames sent to the PNC by the DEV shall be exchanged either in the CAP of the superframe or in an association MCTA.

An unassociated DEV initiates the association process by sending an Association Request command, as described in 7.5.1.1, to the PNC. When the PNC receives an Association Request command, it shall send an Association Response command, indicating that the DEV has been associated and the DEVID it has been assigned or that the request has been rejected with the reason for the rejection, as defined in 7.5.1.2. For association using MCTAs, the Association Response command, as described in 7.5.1.2, is sent in an MCTA with PNCID as source and UnassocID as destination.

The PNC shall acknowledge all correctly received Association Request commands by sending an Imm-ACK frame with the DestID set to the UnassocID. The ACK to an Association Request command does not mean that the DEV is associated. The PNC needs some time to ensure that the DEV should be allowed in the piconet, to ensure that there are enough resources available to support another DEV in the piconet and to allocate a DEVID. The PNC may maintain a list of DEV addresses that are allowed to join the piconet. If the list is in use, when the PNC receives an Association Request command, the PNC shall consult the list to determine if the DEV address in the request is included in the list. If the DEV address is not in the list, the PNC shall send

an Association Response command with the reason code set to “association denied,” as described in 7.5.1.2, indicating that the association failed. If the PNC determines that there are not enough resources available to support the new DEV, the PNC shall send an Association Response command with the reason code set to the appropriate value in 7.5.1.2. If the PNC determines that the DEV will be associated, the PNC shall send an Association Response command with the reason code set to “success.” The time difference between when the PNC sends the Imm-ACK to the Association Request command from a DEV and when it sends an Association Response command meant for the same DEV shall not exceed mAssocRespConfirmTime.

The Association Response command is not a directed frame. If an Imm-ACK was required for this command, when there were multiple DEVs trying to associate during the same time interval, all of them would try to ACK and collide. Therefore, the ACK Policy field for the Association Response command is set to no-ACK. Instead each DEV trying to associate shall compare its DEV address with the DEV Address field in the Association Response command and if there is a match, accept the DEVID for all future communications.

The unassociated DEV receiving the Association Response command with the DEV address matching its own shall send during the CAP or an association MCTA a second Association Request command with the SrcID field, as described in 7.2.3, set to its newly assigned DEVID. The PNC upon receiving this Association Request command shall respond with an Imm-ACK with the DestID set to the SrcID of the Association Request command. The PNC after acknowledging this second request shall then initialize the DEV Association IE with the requesting DEV's DEVID, DEV address, DEV Capabilities field, and Association Status field set to “associated.” The requesting DEV upon receiving the Imm-ACK to its second Association Request command shall consider itself associated. All other DEVs that are members of the piconet receiving the beacon containing the DEV Association IE may use the DEV Association IE to update their internal list of associated DEVs in the piconet. The PNC shall ensure that the DEV Association IE announcment for a newly associated DEV complies with the rules for beacon announcements in 8.6.4.

The PNC starts the ATP timer once it has sent the Association Response command for the new DEV. The associating DEV needs to send the second Association Request command before the ATP timer expires. If the PNC receives the second association request command after the ATP timer expires, the PNC shall send the Disassociation Request command, as described in 7.5.1.3, to the DEV requesting association to indicate that it has failed the association process.

Figure 102 illustrates the message flow for a successful association process.

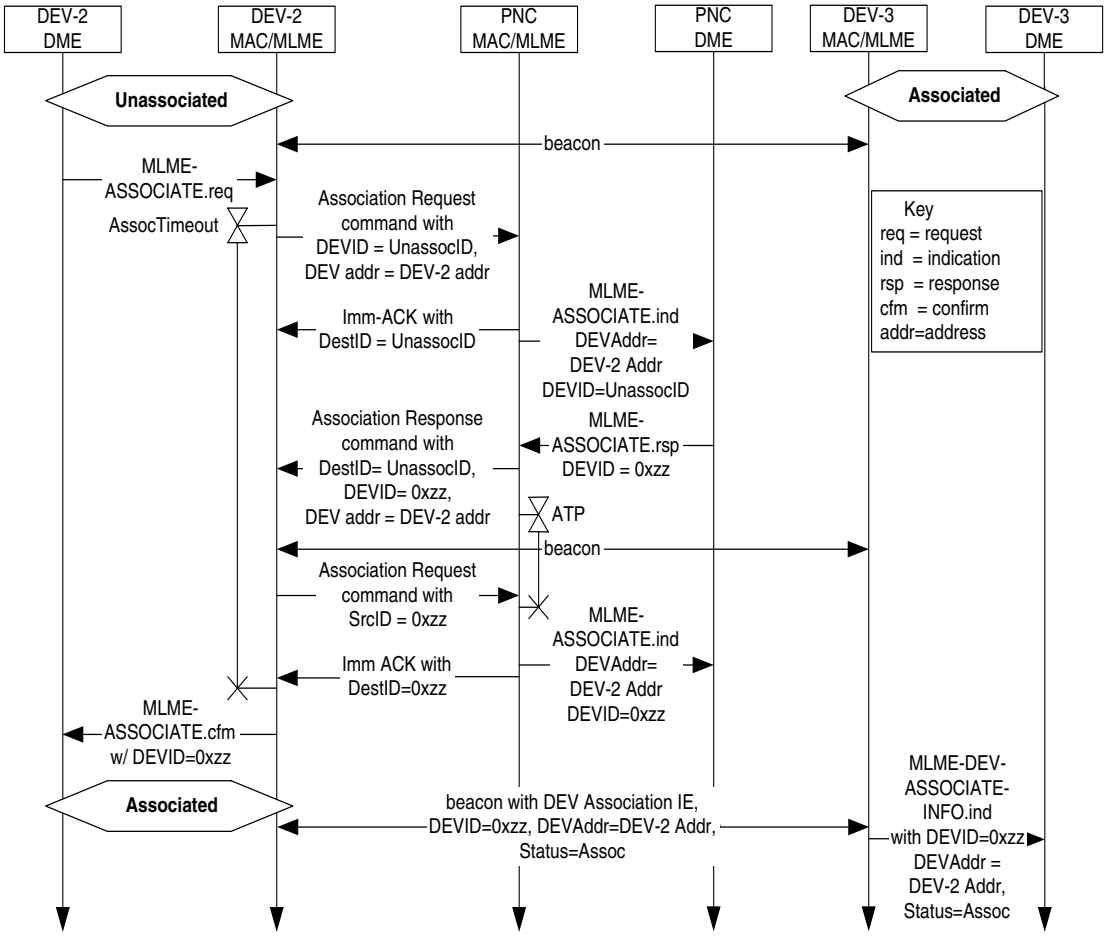

Figure 102—MSC of DEV-2 associating

The device IDs (DEVIDs) shall be assigned in sequence (increasing order) by the PNC except when PNC wishes to use a DEVID that was freed up after a DEV has left the piconet. After the PNC sends a Disassociation Request command, as described in 7.5.1.3, to a DEV, the PNC shall not reuse the same DEVID of that DEV until at least two times the ATP duration for that DEV has passed. The PNC shall ensure that there is only one associated DEV that has been allocated a given DEVID at any given time within the piconet. Similarly any associated DEV shall be allocated only one DEVID. The only exception to this is the PNC itself. The DEV serving as the PNC shall have two values of DEVID associated with it. The PNCID shall be assigned to the PNC function within the DEV and the other value of the DEVID shall be for use for all of the non-PNC traffic. When there is a coordination handover, as described in 8.2.3, the new PNC assumes the PNCID. The former PNC shall continue to use its non-PNCID DEVID for its non-PNC traffic. Hence the PNC is seen as two logical operational entities within the same DEV.

After the association process is complete, the PNC broadcasts the PNC Information command as described in 8.3.3.

8.3.2 Piconet services

Piconet services information is provided by the PNC, if supported, to an associating DEV upon request. In order to request the piconet services information, the associating DEV sets the Piconet Services Inquiry bit

in the Association Request command with the SrcID set to the newly assigned DEVID as described in 7.5.1.1. If the DEV sets the Piconet Services Inquiry bit, the PNC shall send a Piconet Services command, as described in 7.5.5.1, with DestID set to the newly assigned DEVID after that PNC has received the Association Request command with the SrcID set to the newly assigned DEVID from the DEV, as described in 8.3.1. This process is illustrated in Figure 103

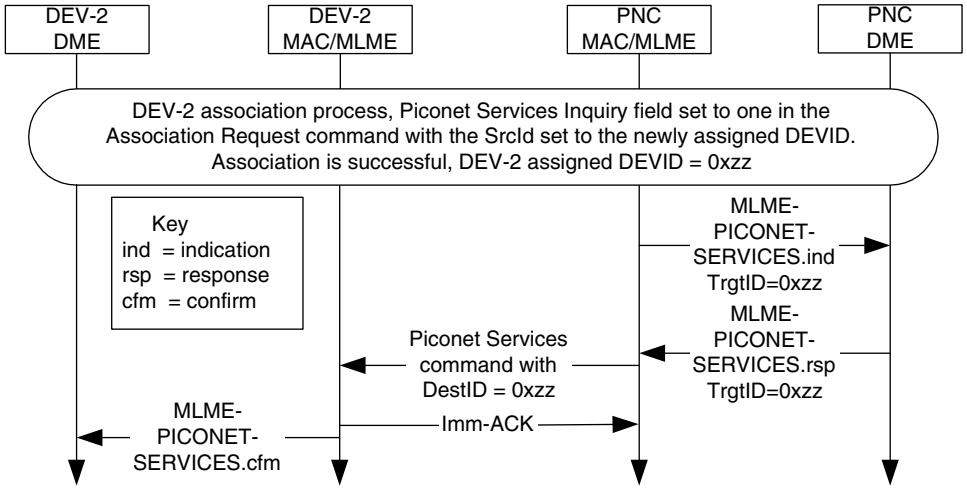

Figure 103—PNC sending the Piconet Services command to a newly associated DEV in response to a request in the association process

An associating DEV may inspect the Piconet Services IEs in the Piconet Services command, as described in 7.5.5.1, returned by the PNC to determine information about other DEVs.

DEVs that are members of the piconet may place their own Piconet Services IE in the PNC's record of piconet services by sending the Piconet Services IE to the PNC using the Announce command. The PNC then sends an Announce command with DestID set to the BcstID containing the Piconet Services IE that it has added to its internal record of piconet services. If the PNC supports this capability, it retains the Piconet Services IEs of DEVs that have been sent to the PNC via the Announce command. The PNC will only save Piconet Services IEs for which it has space. Thus it is possible that the PNC would not retain a DEV's Piconet Services IE. After a DEV disassoicates from the piconet, the PNC shall delete the DEV's Piconet Services IE from its own record.

If a DEV sends a Probe Request command to the PNC requesting the Piconet Services IE, the PNC responds with Probe Response commands that contain all of the Piconet Services IEs that it has in its internal record. If a DEV has not provided a Piconet Services IE to the PNC, the PNC sends the Piconet Services IE in the Probe Response command with the DEVID, a zero Vendor OUI and zero length Piconet Services field. If the PNC did not have enough space to save the Piconet Services IE that a DEV provided, it shall send in the Probe Response command a Piconet Services IE with length 1, i.e. it only contains the DEVID.

If a DEV has a need for privacy, it is not required to provide information that would be available outside of the security operations of the piconet. The MAC PIB element MACPIB_DEVServicesBroadcast, as described in 6.5.2, indicates if the DEV will send the Piconet Services field. Likewise the PNC is not required to furnish this information if it violates the security policy as set in the MAC PIB element MACPIB_PNCServicesBroadcast, as described in 6.5.1.

A DEV may use the Probe Request command, as described in 7.5.4.5, to request the Piconet Services IE from another DEV, the target DEV, in the piconet. If the target DEV supports sending the Piconet Services IE and its internal policy allows sending the information, the target DEV shall respond with a Probe Response command with the requested IE. If the target DEV does not support the piconet services IE or if its policy does not allow sending the piconet services IE, the target DEV shall respond with a Probe Response command with a zero length Piconet Services IE, that is an IE with only the Element ID and Length fields.

It is outside of the scope of this standard to define the content or use of the Piconet Services field.

8.3.3 Broadcasting piconet information

The PNC shall broadcast the piconet information using the PNC Information command, as described in 7.5.4.2, after a DEV becomes a member of the piconet. This means that if security is required for the piconet, the PNC will wait until after secure membership has been established to broadcast the piconet information with the new DEV. In addition, the PNC shall send the piconet information for each of the DEVs that are a member of the piconet and any neighbor PNCs that are associated in the piconet at least once every mBroadcastDEVInfoDuration via a PNC Information command. When the PNC broadcasts this command, the PNC shall include an entry for the DEV personality of the PNC, as well as an entry for the PNCID.

8.3.4 Disassociation

When a PNC wants to remove a DEV from the piconet, the PNC shall send a Disassociation Request command, as described in 7.5.1.3, to that DEV with an appropriate reason code. Similarly when a DEV wants to leave the piconet, the DEV shall send a Disassociation Request command to the PNC with an appropriate reason code.

All Disassociation Request commands, when received correctly, shall be acknowledged by the intended recipient.

All DEVs in the piconet shall send frames to the PNC often enough to assure that the association timeout period (ATP) is not reached. If the PNC does not receive any frame originating from an associated DEV within this timeout duration, the PNC shall disassociate the DEV. The DEV may send a Probe Request command without requesting any information to cause the PNC to reset the ATP if the DEV does not have any other traffic that it needs to send to the PNC.

If the beacons from the PNC are not received by the DEV for longer than the ATP, the DEV shall consider itself disassociated from the piconet and may try to associate again. The DEV notifies the DME that the ATP expired using the MLME-ATP-EXPIRED.ind primitive.

The PNC shall send a Disassociation Request command to a DEV that sends a frame after its ATP has expired.

The PNC upon receiving a Disassociate Request command or an ATP expiration shall include in the beacon a DEV Association IE. The PNC shall ensure that the DEV Association IE announcment for a disassociated DEV complies with the rules for beacon announcements in 8.6.4. The PNC shall perform the stream termination procedures for each of the assigned CTAs with the disassociated DEV as the SrcID or DestID. The other DEVs that are members of the piconet shall use this information to update their internal DEV association table and to determine whether they will discontinue listening or transmitting in a CTA. Note that when a DEV is disassociated, it loses its DEVID and so the PNC will reset the bits that refer to this DEVID in all of the relevant bitmaps, e.g. PS Status IE, PCTM IE, CWB IE. The PNC will also remove the disassociated DEV from any PS sets that it has joined and shall delete the Piconet Services IE, if any, for that DEV from its internal storage.

Figure 104 illustrates the message flow for a disassociation initiated by a DEV.

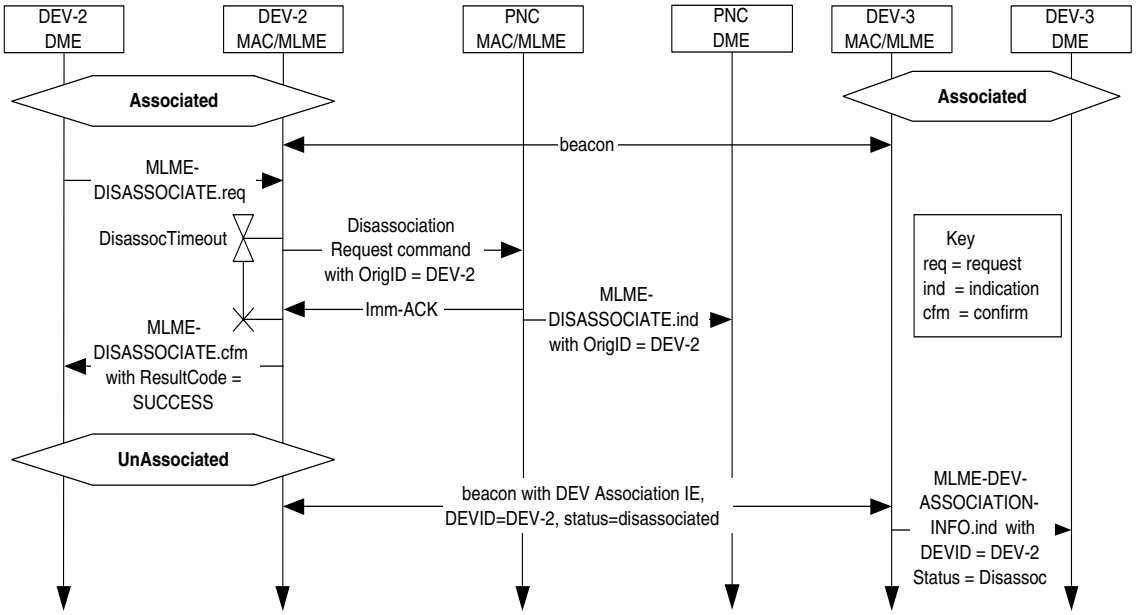

Figure 104—DEV initiated disassociation MSC

Figure 105 illustrates the message flow for a disassociation initiated by the PNC.

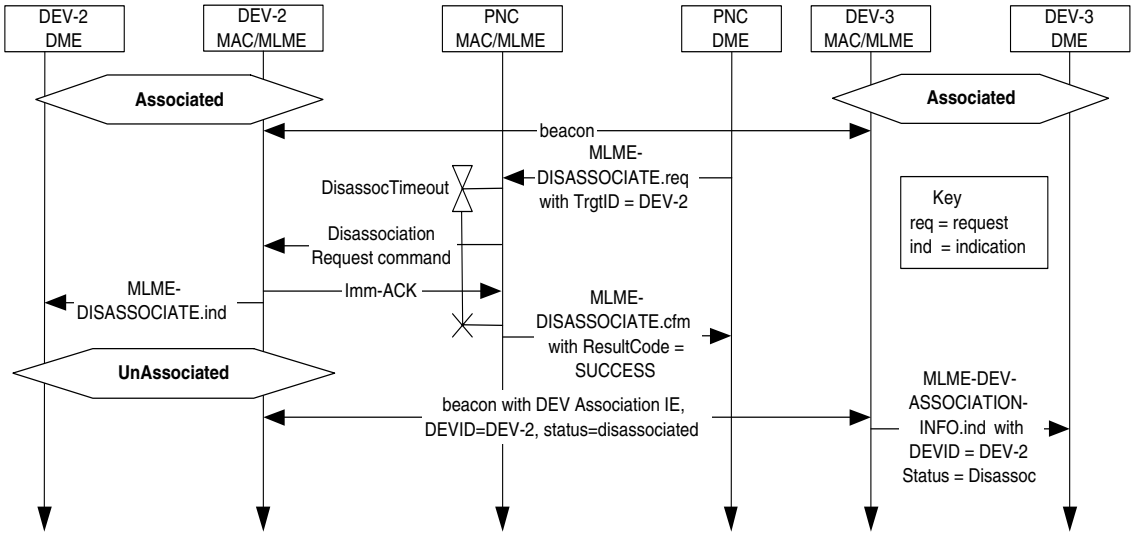

Figure 105—PNC initiated disassociation MSC

8.4 Channel access

The channel time is divided into superframes, with each superframe beginning with a beacon. The superframe is composed of three major parts: the beacon, the optional CAP and the CTAP, as shown in Figure 106. The CTAP is used for asynchronous and isochronous data streams as well as commands while

the CAP may be used for commands and non-stream data, as regulated by the PNC. During the CAP, the DEVs access the channel in a distributed style using CSMA/CA and a backoff procedure. During the CTAP, the PNC controls the channel access by assigning CTAs to an individual DEV or to a group DEVs with each CTA having a fixed start time and duration.

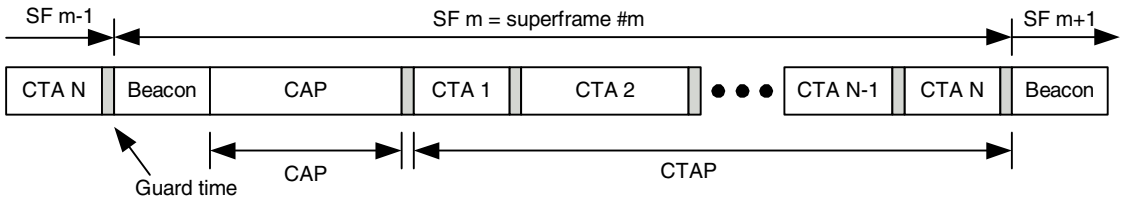

Figure 106—Superframe structure

8.4.1 Interframe space (IFS)

There are four IFSs that are defined; the minimum interframe space (MIFS), the short interframe space (SIFS), the backoff interframe space (BIFS) and the retransmission interframe space (RIFS). The actual values of the MIFS, SIFS, BIFS and RIFS are PHY dependent. For the 2.4 GHz PHY they are listed in 11.2.7.1.

All Imm-ACK frames and Dly-ACK frames shall start transmission over the medium a SIFS duration after the end of the transmission of the previous frame which requested the ACK. A MIFS duration shall be allowed in the CTA between a frame and the next successive frame transmitted over the medium if the first frame either had the ACK Policy field set to either no-ACK or Dly-ACK.

During the CTAP, all DEVs shall use a RIFS for retransmissions. During the CAP, however, the retransmissions shall follow the CAP rules described in 8.4.2. The rules for acknowledgement and retransmissions are described in 8.8. The interframe space requirement for the beacon is ensured by the location of the CTAs which is determined by the PNC, as described in 8.4.3.6.

8.4.2 Contention based channel access

The basic medium access mechanism during the CAP is carrier sense multiple access with collision avoidance (CSMA/CA). The PNC controls the type of data or commands that may be sent in the CAP via the CAP Control bits in the Piconet Mode field of the Piconet Synchronization Parameters field, as described in 7.3.1, in the beacon. A DEV shall only send frames of the type indicated by the Piconet Mode field in the beacon for the current superframe. The CAP Control bits in the Piconet Mode field may be changed by the PNC from superframe to superframe.

To minimize collisions, a transmitting DEV is required to first sense that the medium is idle for a random length of time. The MAC shall use the CCA capabilities of the PHY to detect whether the channel is busy or idle. Only if the medium is idle after that time shall the DEV start its transmission. This process of waiting before transmission is termed “backoff.” The backoff procedure shall not be applied to the transmission of the beacon that is transmitted by the PNC at the beginning of the superframe.

During the CAP a DEV is allowed to transmit one frame at a time with backoff being applied to every frame attempted during the CAP, except for the Imm-ACK frame. The PNC may send a command a SIFS following the Imm-ACK of a frame in the CAP or following a frame with ACK Policy field set to no-ACK in the CAP. In this case, the PNC is not required to perform the backoff procedure before sending its frame.

In no case shall a DEV or the PNC extend its transmissions that started during the CAP into the CTAP. If an Imm-ACK is expected for that frame, the remaining time in the CAP needs to be long enough to accommodate the current frame, 2 SIFS times and the Imm-ACK frame at the same PHY rate as the transmitted frame.

If there is insufficient time remaining in the CAP for the entire frame exchange sequence, then the DEV or the PNC shall not commence transmission of the frame.

The following backoff procedure shall be performed when sending frames (other than Imm-ACK) during the CAP.

The backoff algorithm uses the following information:

— retry_count: An integer that takes on values in the range 0 to 3, inclusive.
— backoff_window(retry_count): A table which has values [7, 15, 31, 63].
— pBackoffSlot: A PHY dependent parameter that is based on the amount of time it takes to sense the channel. For the 2.4 GHz PHY, this is defined in 11.2.7.1.
— bw_random(retry_count): A random integer drawn from a uniform distribution over the interval [0,backoff_window(retry_count)]. The random number generated for a DEV should be statistically uncorrelated with random numbers generated by other DEVs. If the DEV does not possess a random number source, the random integer should be generated using its unique DEV address (and any other information that the implementer wishes to use) and a pseudo-random number generator (PRNG) such as MGF1 as defined in IEEE Std 1363-2000. Note that the current state of the PRNG should be maintained and subsequent backoffs should use subsequent integers in the pseudo-random sequence. It is important that designers recognize the need for statistical independence among the random number streams among DEVs.

The backoff time in the CAP is measured at the air interface and indicates when a DEV may begin transmitting data. The DEV first waits a BIFS duration, as described in 8.4.1, from when the medium is determined to be idle before beginning the backoff algorithm. At the beginning of the CAP, the DEV may begin the backoff algorithm a SIFS after the end of the beacon transmission. If the PNC indicates that it is using an extended beacon, as described in 8.6.2, then the DEV shall wait until a SIFS after the last Announce command sent by the PNC as a part of the extended beacon before beginning the backoff procedure.

The DEV shall then choose backoff_count = bw_random(retry_count) and shall maintain a counter for backoff_count which is decremented only when the medium is idle for the entire duration of pBackoffSlot. The retry_count shall be set to zero for the first transmission attempt of a frame. Whenever the channel is busy, the backoff counter shall be suspended. The channel shall be determined to be idle for the duration of a BIFS period before the backoff slot countdown is resumed. When the backoff counter reaches zero, the DEV may transmit a frame. When a backoff count of zero is drawn the DEV can transmit immediately following the channel having been idle for a BIFS.

The backoff counter shall also be suspended outside of the CAP duration. The backoff counter shall also be suspended if there is not enough time remaining in the CAP for the DEV to send the frame. Note that the backoff counter is maintained across superframes and is not reset with each beacon. If the total time elapsed since the frame was queued for transmission has exceeded the transmission timeout specified for the frame, the backoff counter shall be reset and the attempted transmission shall be canceled.

When a directed frame is transmitted and the expected ACK is not correctly received by the DEV, the retry_count shall be incremented but shall not be set to more than 3. The backoff_count shall then be set to bw_random(retry_count). If the maximum number of retries for that frame has not been exceeded, the backoff procedure is again resumed.

8.4.3 Channel time allocation period channel access

Channel access in the CTAP is based on a TDMA method in which all CTAs have guaranteed start time and duration. The guaranteed start times enable both power saving and good QoS characteristics. All the CTAs for the current superframe are broadcast in the beacon.

8.4.3.1 Channel time allocations (CTA)

The PNC divides the CTAP into channel time allocations (CTAs). A DEV that is give a directed CTA is guaranteed that no other DEVs will compete for the channel during the indicated time duration of the CTA. A DEV with a CTA may or may not make use of all the allocated time duration within the CTA. The selection of a stream, command or asynchronous data for transmission during a CTA is determined locally by the DEV depending on the number of pending frames and their priorities. See A.1.2.1 for more information on priority management.

There are two types of CTAs: dynamic CTA and pseudo-static CTA. The type of a CTA requested is indicated in the Channel Time Request command as specified in 7.5.6.1.

The PNC may move dynamic CTAs within the superframe on a superframe by superframe basis. This allows the PNC the flexibility to rearrange CTA assignments to optimize the utilization of the assignments. The PNC moves a dynamic CTA by simply changing the CTA parameters in the beacon. Dynamic CTAs may be used for both asynchronous and isochronous streams.

If multiple CTAs per superframe were requested by the DEV in the Channel Time Request command, as described in 7.5.6.1, the PNC shall attempt to spread the CTAs out evenly within the superframe.

The PNC should attempt to allocate the CTAs of all SPS DEVs first in the superframe. Exceptions to placing these allocations first are as follows and in order of priority.

- — QoS streams that need multiple CTAs within a superframe and require a location immediately following the beacon, if the CAP is used.
- — If CAP is not used, a single CTA that follows the beacon without the mFirstCTAGap restriction and is:
  - — An MCTA with the PNCID as the SrcID—directed, broadcast or multicast
  - — A single pseudo-static CTA

Pseudo-static CTAs shall be allocated only for isochronous streams and shall not be sub-rate allocations, as described in 7.5.6.1. If the PNC needs to change the duration or location of a pseudo-static CTA within the superframe, it shall change the corresponding CTA blocks in the beacon. The PNC shall not create any new CTAs for other stream indicies that overlap with the old time interval of the pseudo-static CTAs for mMaxLostBeacons number of superframes. However, the PNC may overlap the old and new time intervals of the same psuedo-static CTA within a superframe as it does not create the possibility of frame collisions. If the PNC sees the transmission of a PDU during the new allocation by the source of the old allocation before the expiration of mMaxLostBeacons number of superframes, the PNC may reuse the old allocation for another pair of DEVs. When the source DEV of a pseudo-static CTA receives a beacon with the new CTA, it shall cease using the old CTA and begin using the new CTA. When the destination DEV of a pseudo-static CTA receives a beacon with the new CTA, it shall begin receiving during the new CTA and may also receive during the old CTA.

While the PNC is changing the time interval of the pseudo-static CTA, it is possible for the destination DEV to miss traffic for up to mMaxLostBeacons superframes. If the destination wants to avoid this, it would need to listen for the entire superfame duration whenever it misses a beacon.

A private CTA is a CTA where the same DEV is both the source and the destination. A private CTA is not used for communication in the piconet. Instead, it is used to reserve channel time for some other use. For example, a private CTA would be used for a dependent piconet, as described in 8.2.5 and 8.2.6. Private CTAs shall be pseudo-static CTAs, so that its position and duration remain relatively constant for the other use. A DEV requests a private CTA by using its own DEVID as both the SrcID and TrgtID for Channel Time Request command, as described in 7.5.6.1.

The More Data bit, as described in 7.2.1.6, in the Frame Control field is set to one to indicate that the source DEV could be sending more frames in the CTA. In order to save power at the destination DEV, a source DEV may indicate that it will not use the remaining time in the current CTA by setting the More Data bit to zero. The source DEV may retransmit a frame with More Data set to zero for which an ACK was expected but was not received. If the destination DEV receives a frame with the More Data bit set to zero with ACK Policy field set to Imm-ACK, Dly-ACK or Dly-ACK Request, it should continue to listen for an implementation specific time after sending the ACK frame to make sure that the source DEV is not going to retransmit the frame because it did not receive the ACK. The source DEV may choose to send a zero length frame with the More Data bit set to zero when it has no more frames to send in a CTA.

The More Data bit shall be ignored by the destination for all frames sent in the CAP, with the exception of any Announce commands used for the extended beacon.

8.4.3.2 Channel time allocation (CTA) and channel time usage

The DEVs that are members of the piconet shall use the Channel Time Request command, as described in 7.5.6.1, whenever they wish to make a change in their CTAs. Once a Channel Time Request command is received from a DEV, the PNC shall remember that as the outstanding request for that stream for every superframe until another Channel Time Request command for that stream is received from the DEV. The CTAs within the CTAP are based on the current pending requests from all the DEVs and the currently available channel time within the CTAP. The start time of each CTA is referenced to the start of the beacon frame, as described in 8.6.5. The algorithm used to allocate the channel time and assign CTAs is outside of the scope of this standard.

The PNC shall not allocate any MCTAs or dynamic CTAs within mFirstCTAGap following the end of the beacon except with the PNC as the source.

When a source DEV has a frame of any type for a destination DEV, the source DEV may send it during any CTA for that source DEV and destination DEV pair or to use the CAP to communicate that frame. The source DEV may also send a frame to a destination DEV in any CTA assigned to that source even if the destination DEV is different that that indicated in the CTA block, provided the source DEV has determined that the destination DEV will be receiving in that CTA, as described in 7.4.11.

If the DestID of the CTA is the McstID or the BcstID, the source DEV may still send directed frames to any associated DEV. However, it is possible that the target DEV will not be receiving during the CTA if it is in a power save mode, as described in 8.13, or if it is not receiving multicast traffic, as described in 6.3.17.8.

In any superframe there may be one or more DEVs in the piconet that receives the beacon in error. This may not happen to the same DEV all the time but may happen to different DEVs at different times depending upon their location and type of interference to which they are subjected. If a DEV did not receive the beacon, it shall not transmit during the CAP or during any MCTA or dynamic CTA, except to ACK a directed frame sent to the DEV with the ACK Policy field set to either Imm-ACK or Dly-ACK Request. DEVs with pseudo-static CTAs are allowed to transmit during these CTAs as long as the number of consecutive lost beacons is less than or equal to mMaxLostBeacons. A DEV shall stop transmitting in its pseudo-static CTA when the number of consecutive lost beacons exceeds mMaxLostBeacons. If a DEV that is the destination of a pseudo-static CTA misses a beacon, it should listen for the entire duration of the superframe in case the pseudo-static CTA is in the process of being moved. Any DEV that misses a beacon may also listen for the entire duration of the superframe to receive frames for which it is the destination.

In no case shall a DEV extend its transmissions that started during a CTA beyond the end of that CTA. Hence, the source DEV shall check whether there is enough time remaining in the CTA for the transmission of current frame and SIFS. If an Imm-ACK or Dly-ACK is expected for that frame, the DEV shall check whether there is enough time remaining in the time slot to accomodate the current frame, 2 SIFS periods and the Imm-ACK or Dly-ACK frame at the same PHY rate as the transmitted frame. If there is not enough time remaining for this entire frame exchange sequence, then the DEV shall abort the transmission and not use the remainder of the CTA.

The PNC may compute more than one superframe time allocation at a time and keep them repeating over time until the situation changes. If the PNC wants to increase the allocated channel time allocation on a regular basis, it shall allocate more time up to the maximum requested by sending a Channel Time Response command, as described in 7.5.6.2, to the source DEV and change the allocation(s) in the beacon. If the source DEV requires additional channel time it will need to use the stream modification procedure, as described in 8.5.1.2. The PNC may also reduce the channel time allocation for a stream by sending the Channel Time Response command to the source DEV and changing the allocations in the beacon.

In any individual superframe, the PNC may allocate more time for a dynamic CTA than the amount indicated in the Channel Time Response command.

8.4.3.3 Management CTAs

Management CTAs (MCTAs) are identical to CTAs except that the PNCID is either the SrcID or the DestID in the CTA and the stream index is set to the MCTA stream index, as described in 7.2.5. A PNC may choose to use MCTAs instead of the CAP for sending command frames, unless otherwise restricted by the PHY, as described in 11.2.10. When MCTAs are used, the PNC shall ensure that sufficient MCTAs are allocated to allow for the transmission of commands to and from the PNC. There may not be any MCTAs in a superframe or there may be as few as a single MCTA in a superframe where the ownership of the MCTA changes from superframe to superframe. At the other extreme, there may be one or more uplink and downlink MCTAs per member DEV per superframe plus MCTAs for association. The PNC is responsible for determining the appropriate number of MCTAs in a superframe in the same way that the PNC is responsible for choosing the CAP size if a CAP is used. The PNC determines which DEVs will be allocated MCTAs and the frequency of the allocations. The PNC shall allocate at least one association MCTA every mMCTAAssocPeriod if the CAP is not used.

An open MCTA is one where the SrcID is the BcstID, as described in 7.2.3. Any DEV that is associated in the piconet may attempt to send a command frame to the PNC in an open MCTA. An MCTA with the UnassocID as the SrcID is an association MCTA. Any DEV not currently associated in the piconet may attempt to send an Association Request command to the PNC in an association MCTA. Association Request commands shall not be sent in open MCTAs. Likewise, only Association Request commands shall be sent in association MCTAs. Open MCTAs enable the PNC to service a large number of DEVs with low MCTA requirements by using a minimum number of MCTAs. When there are few DEVs in a piconet it might be more efficient to use MCTAs assigned to a DEV instead of using an open MCTA.

It is the PNC's responsibility to determine the number and type of MCTAs to use for each superframe. A DEV may request the frequency of MCTA allocations by sending a Channel Time Request command, as described in 7.5.6.1, to the PNC with the stream index set to the MCTA stream index, as described in 7.2.5, and the CTA Rate Factor, as described in 7.5.6.1, set to the DEV's desired interval for uplink MCTAs, the Num Targets field set to one and the Target ID field set to the PNCID. All other parameters of the CTRqB shall be set to zero and may be ignored by the PNC upon reception.

If commands are not allowed in the CAP, the PNC shall assign an MCTA with the new DEV's DEVID as the SrcID as soon as possible after a successful association, 8.3.1, preferably in the next superframe, in order to support fast connections.

The access mechanism for regular MCTAs, i.e. neither open nor association MCTAs, is TDMA, as described in 8.4.3.1.

8.4.3.4 Slotted aloha access for open and association MCTAs

Slotted aloha is the access mechanism in an open MCTA or an association MCTA. The access to an open or association MCTA shall be controlled by a contention window CW_a maintained by each DEV. The contention window shall be derived from the number *a*, where *a* is the number of retransmission attempts made by the DEV. For the first access attempt, *a* shall be set to zero. The size of the contention window, CW_a, is defined as follows:

$$CW_a = \begin{cases} 256 & \text{for } 2^{a+1} \geq 256 \\ 2^{a+1} & \text{for } 2^{a+1} < 256 \end{cases} \quad (1)$$

The open or association MCTA used for the a^{th} retransmission attempt shall be chosen by a uniformly distributed random integer value, r_a, within the interval [1, CW_a]. The random number generated for a DEV should be statistically uncorrelated with random numbers generated by other DEVs. If the DEV does not possess a random number source, the random integer should be generated using its unique DEV address (and any other information that the implementer wishes to use) and a pseudo-random number generator (PRNG) such as MGF1 as defined in IEEE Std 1363-2000. Note that the subsequent retransmission attempts should use back-off counters drawn from subsequent entries in the pseudo-random list of integers. It is important that designers recognize the need for statistical independence among the random number streams among DEVs.

The DEV shall start counting r_a beginning with the open or association MCTA(s) in the current superframe and continue across superframes. The lack of an Imm-ACK indicates the failure of the previous access attempt.

The first open or association MCTA after the DEV begins the access process is specified by number '*r*=1'. The open or association MCTA with number equal to r_a is the MCTA that the DEV shall access. The DEV shall not access the MCTA before its counter has reached the open or association MCTA with the number $r=r_a$. After receiving an ACK, *a* shall be reset to zero.

8.4.3.5 Allocation of MCTAs

The PNC shall indicate in every beacon, as described in 7.3.1, the rate at which it will be allocating either open MCTAs or directed uplink MCTAs in the MCTA Allocation Rate field in the Piconet Synchronization Parameters field. If the PNC is not using either open MCTAs or directed uplink MCTAs it shall indicate this with the appropriate value of the MCTA Allocation Rate field, as described in 7.3.1.1. Likewise, if the PNC will not be guaranteeing the rate at which MCTAs will be allocated, it shall also indicate this in the MCTA Allocation Rate field.

The intent of the MCTA Allocation Rate field is to enable the DEVs in the piconet to approximately determine the length of time required to send a command to the PNC. This information might be used to set the timeout parameters for the MLME primitives, as described in 6.3.

8.4.3.6 Guard time

In a TDMA system, guard times are required to keep transmissions in adjacent CTAs from colliding. In addition, a SIFS time is required to ensure sufficient turnaround time between transmissions. A CTA is defined by the start time and the duration as specified in the CTA IE. Guard time is the time between the end of one CTA and the start of the next CTA. Including SIFS as part of CTAs and allocating guard time between CTAs ensures that transmissions are spaced by at least a SIFS. Figure 107 is an illustration of the allocation of the guard time such that the transmissions are separated by at least a SIFS if the owners of adjacent CTAs drift towards the other CTA. The PNC shall allocate sufficient guard time between CTAs to ensure that transmissions in adjacent CTAs do not overlap.

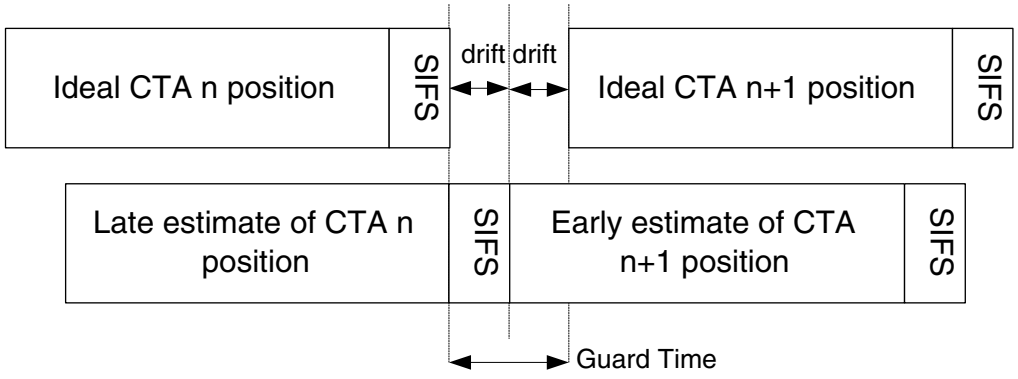

Figure 107—Guard time

The required guard time depends on the maximum drift between a DEV's local time and the ideal time. This drift is a function of the time elapsed since a synchronizing reference event. In an 802.15.3 piconet, the synchronizing event is the start of the preamble of a beacon. The maximum drift, MaxDrift, is calculated as follows:

$$MaxDrift = [\text{Clock accuracy (ppm)} / 1e6] * \text{interval} \quad (2)$$

Propagation delay will also affect timing uncertainty, but in a piconet, the 10 m range limits propagation delay to around 33 ns, or even 66 ns for DEVs 20 m apart at opposite ends of a piconet. This is much lower than the resolution of the CTA timing and it is ignored when calculating the guard time.

The PNC may calculate a single worst case guard time for all CTAs in the superframe, or it may calculate and assign guard time based on the type of CTA (dynamic or pseudo-static) and the position of the CTA in the superframe.

Pseudo-static CTAs require longer guard times than dynamic CTAs because pseudo-static CTAs allow transmission even when up to mMaxLostBeacons are missed by the transmitting DEV. Guard times are calculated based on the worst-case drift in a superframe and the maximum allowed number of lost beacons for each of the adjacent CTAs. Guard time may be calculated by the PNC as follows:

$$GuardTime = (MaxLostBeacons_{CTA\_n} + MaxLostBeacons_{CTA\_n+1} + 2) * MaxDrift \quad (3)$$

MaxLostBeacons for each CTA depends on whether the CTA is pseudo-static or dynamic. MaxLostBeacons is zero for a dynamic CTA and is equal to mMaxLostBeacons if the CTA is pseudo-static. The PNC calculates the MaxDrift using the superframe duration and the clock accuracy, pPHYClockAccuracy. The PNC then calculates the start time and duration of each CTA such that there is sufficient guard time between the end of one CTA and the start of the next CTA.

A DEV transmitting in a CTA starts transmission of the preamble for the first frame at the point which it calculates is the start of the CTA based on its local clock. In the case of no-ACK or delayed-ACK, the transmitting DEV shall ensure that there is enough time remaining in the CTA to transmit the frame and allow for a SIFS before the end of the CTA as calculated by that DEV, as shown in Figure 108.

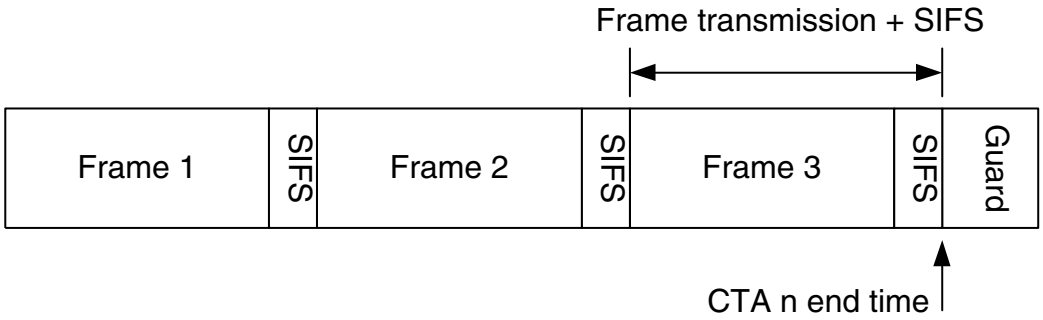

Figure 108—SIFS and Guard time at the end of a CTA—no-ACK

If Imm-ACK is used, the transmitting DEV shall also ensure there is enough time for the ACK and another SIFS as shown in Figure 109.

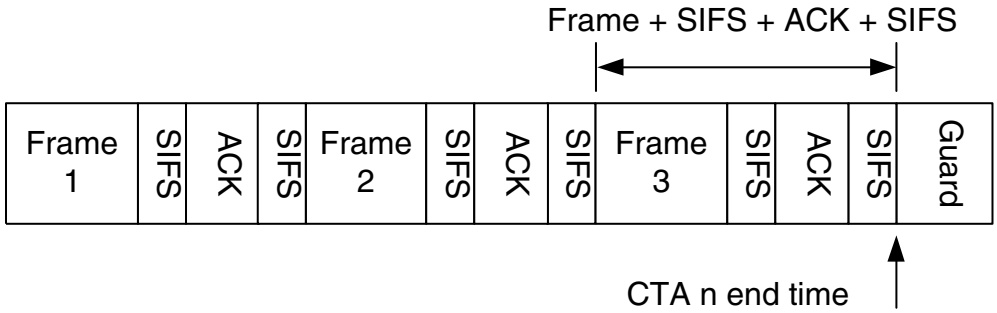

Figure 109—SIFS, ACK, SIFS and Guard time at the end of a CTA - Imm-ACK

As with any CTA, the PNC shall include sufficient guard time between the last CTA in the superframe and the beacon as shown in Figure 110.

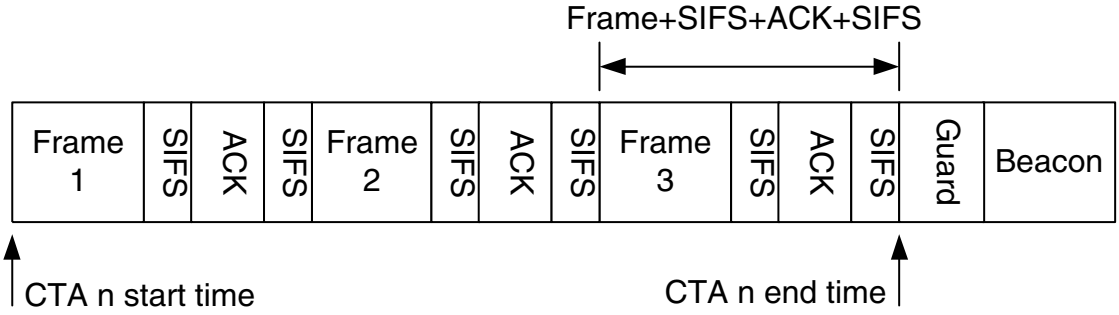

Figure 110—Guard time at the end of the superframe

The PNC shall begin transmission of the beacon preamble at the point in time that it calculates is the start of the superframe based on its local clock. All DEVs will resynchronize their clock based on the beacon arrival.

Because the clock in one DEV may be fast and another may be slow relative to the ideal time, a DEV that is expecting to receive either the beacon or a frame during the CAP or in a CTA shall begin receiving before the time that it calculates to be the start of the beacon, CAP or CTA and shall continue receiving after the time that it calculates to be within one SIFS of the end of the CTA. The amount of time that the DEV listens before the start of the CTA and after the end of the CTA is up to the implementer. It may be calculated based on the type of CTA, the superframe duration and pPHYClockAccuracy. The DEV shall be able to receive a frame that is transmitted within the bounds of allowable transmission for the CTA, accounting for the worst-case drift.

8.4.3.7 Calculating channel time requests

Each DEV sends channel time requests to the PNC to indicate the amount of channel time required for transmission.

The requesting DEV shall include the frame transmission time, if known *a priori*, and the ACK transmission time, if used, and one MIFS or SIFS as appropriate per frame or ACK when calculating channel time requests. Figure 111 shows an example of channel time being requested for a CTA where Imm-ACKs are used.

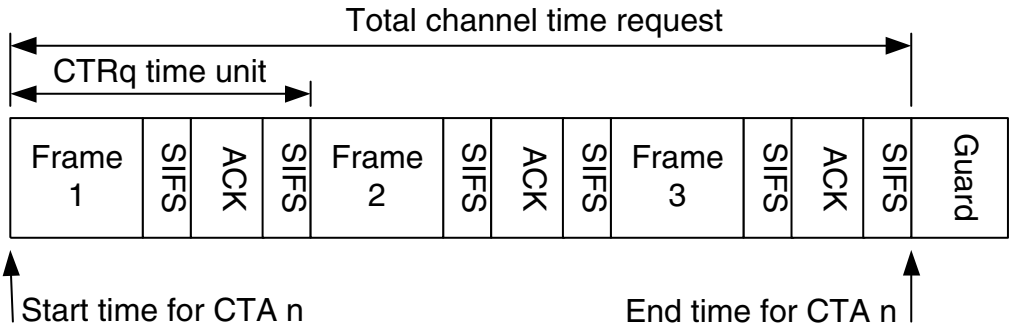

Figure 111—Channel time request for frames with immediate ACKs

When No-ACK is used, the channel time request is calculated differently because there is a MIFS in between each frame in the CTA instead of a SIFS. However, there is a SIFS at the end of the CTA to allow time for the DEVs to switch from transmit to receive and from receive to transmit. Figure 112 shows an example of a channel time request when no-ACK is used.

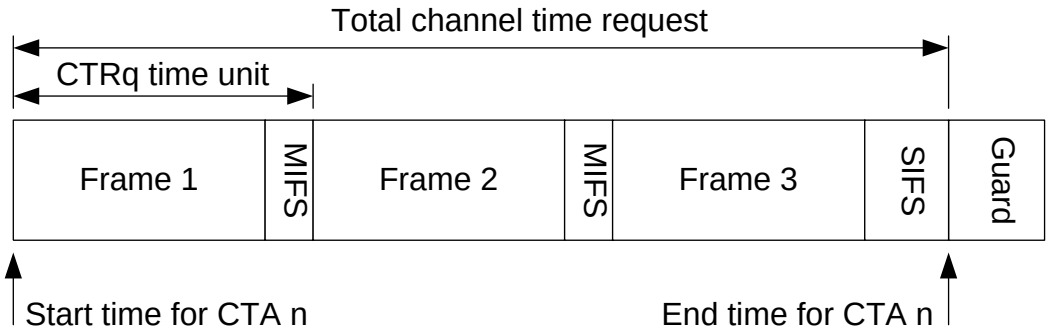

Figure 112—Channel time request with no ACKs

A CTRq time unit in the CTA may cover more than one frame as shown in Figure 113.

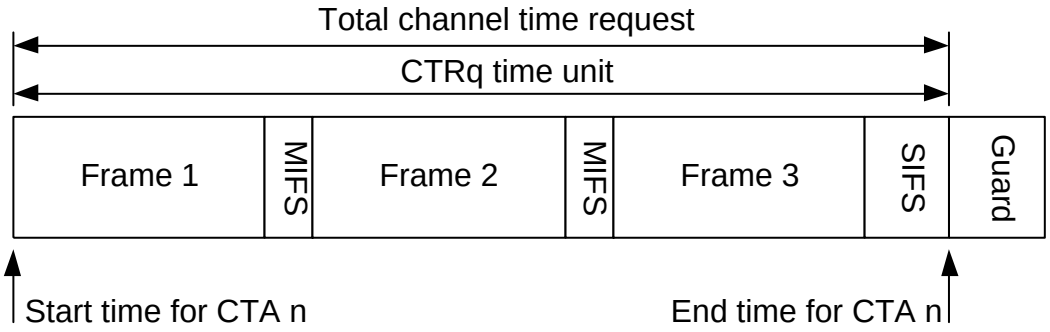

Figure 113—CTRq TU covering multiple frames

If the frame size is not known a priori, it is up to the requesting DEV to decide the amount of channel time to request for the CTA.

8.5 Channel time management

Channel time management in an 802.15.3 piconet involves:

— The creation, modification, and termination of isochronous data streams between two or more DEVs in the piconet.
— The reservation and termination of asynchronous channel time for the exchange of asynchronous data between two or more DEVs in the piconet.

A DEV may support one or more streams depending upon the application it is designed to support. A PNC needs to support as many isochronous streams as it desires to source and sink.

8.5.1 Isochronous stream management

Creating, modifying, and terminating isochronous streams between two or more DEVs in a piconet is accomplished via negotiation between the originating DEV and the PNC using the Channel Time Request and Channel Time Response commands, as described in 7.5.6.1 and 7.5.6.2. Once a stream index and its CTA are established, the CTA may be modified or terminated.

Only a DEV that is either a member of the piconet or associated as a neighbor PNC shall send a Channel Time Request command to the PNC.

There is no absolute guarantee of the length of delay between the time of the request and the reception of a beacon containing the requested CTA.

8.5.1.1 Isochronous stream creation

In the case where the originating DEV is going to request a new isochronous stream with a target DEV, the originating DEV shall send a Channel Time Request command, as described in 7.5.6.1, to the PNC with these parameter values:

— Target ID List field is set to the DEVID with which the originating DEV is requesting a new stream.
— Stream Index field is set to the unassigned stream value, as described in 7.2.5.
— Stream Request ID field is set to a unique value between 1 and 255 for the duration of the negotiation.
— Priority field is set to a value between 0b011 and 0b110 as defined in Figure 74.
— All the other Channel Time Request command parameters are set to appropriate values as defined in 7.5.6.1.

The PNC upon receiving the Channel Time Request command from the originating DEV shall respond with a Channel Time Response command, as described in 7.5.6.2, to the originating DEV with the following Channel Time Response command field values if the requested channel time is available:

— The Stream Index field is set to an unused value other than the asynchronous stream index to indicate that the isochronous stream has been allocated channel time.
— The Available Number Of TUs field is set to a value greater than or equal to the Minimum Number Of TUs and less than or equal to the Desired Number Of TUs requested.
— The Reason Code field is set to ‘success.’

The PNC may update the beacon with the newly assigned isochronous stream CTAs before it receives an ACK to the Channel Time Response command from the originating DEV.

The PNC shall announce the creation of all pseudo-static streams and of all sub-rate streams. It shall also announce creation of a streams for which the destination DEV, or any intended destination DEV in the case of broadcast and multicast streams, is in power save mode. The PNC shall make the announcement with the CTA Status IE, as described in 7.4.10, using the beacon information announcement mechanism, as described in 8.6.4. The PNC shall issue the initial CTA for the stream in the superframe indicated in the CTA Status IE.

The CTA Status IE shall have the stream index of the new allocation and the CTRq Control field and CTA Rate Factor field from the Channel Time Request command. In addition, the PNC shall allocate the first CTA of the stream in the superframe with the beacon number, as described in 7.3.1.1, indicated by the Start Beacon Number field of the CTA Status IE.

If, however, either the requested channel time is not available or the PNC is not able to support the requested priority, as described in A.1.2.1, the PNC shall respond to the requesting DEV with these parameter values:

— The Stream Index field shall be set to the unassigned stream value, as described in 7.2.5.
— The Available Number Of TUs field shall be set to the number of TUs that the PNC had available for allocation to this request.
— The Reason Code field shall be set to ‘priority unsupported,’ ‘channel time unavailable,’ or ‘unable to allocate as pseudo-static CTA’ value.

If the request is for a private pseudo-static CTA, and the PNC will not support the creation of a child piconet, it shall respond with the reason code set to ‘request denied.’

The requesting DEV upon receiving this Channel Time Response command and indicated parameters may accept its denied request as final, it may resend its original request, or it may modify its original request with new parameters.

If the target DEVID is not a member of the piconet, the PNC shall respond to the requesting DEV with these parameter values:

— The Stream Index field shall be set to the unassigned stream value.
— The Available Number Of TUs field set to zero.
— The Reason Code field shall be set to either ‘target DEV not a member’ or ‘target DEV unassociated’ depending on the status of the target DEVID.

DEVs perform multicast negotiations at a higher layer. A DEV sets up a multicast stream at the request of the upper layer by sending a request to the PNC for a stream with the multicast ID as the destination. A DEV enables reception of a multicast stream by using the MLME-MULTICAST-RX-SETUP.request. This tells the MAC to receive frames from a particular source DEV with the DestD set to the McstID and with the stream index specified in the MLME.

If the target DEV is in either DSPS or APS mode and the PNC grants the channel time request, the PNC shall set the Reason Code in the Channel Time Response command to “Success, DEV in PS mode.” The PNC shall place the PCTM IE in the beacon with a bit set for the target DEV, as described in 7.4.8.

When the Target DEV in DSPS or APS mode receives a beacon with its bit set in the PCTM IE, it shall send a PM Mode Change command to the PNC. If the DEV is going to remain in a power save mode it shall set the PM Mode field in the PM Mode Change command to the appropriate value, either ‘SPS’ or ‘APS.’ The PNC shall then terminate the stream, as described in 8.5.1.3.

If the power save DEV is going to listen to the new allocation, it shall set the PM Mode field in the PM Mode Change command to ‘ACTIVE.’ The PNC shall then begin allocating the channel time in the beacon for the stream. The PNC shall no longer set the bits for the DEV in the PS Status IEs.

If the PNC does not receive the PM Mode Change command from the power save DEV within a timeout determined by the PNC, the PNC shall terminate the channel time request, as described in 8.5.1.3, and unset the PS DEV’s bit in the PCTM IE.

If the Target DEV is DSPS mode, after the PNC sets the DSPS DEV’s bit in the PCTM IE the PNC shall provide in the DSPS DEV’s next wake superframe an MCTA with the DSPS DEV as the source and the PNC as the destination that is long enough to handle a PM Mode Change command, a Channel Time Request command with 4 isochronous CTRqBs, and the associated Imm-ACKs and SIFSs. This allows the DSPS DEV to request a change to one of the current channel time allocations, to request new channel time or to request that a channel time allocation be terminated.

Figure 114 illustrates the sequence of messages involved in successfully establishing a DEV-2 to DEV-3 stream in a piconet.

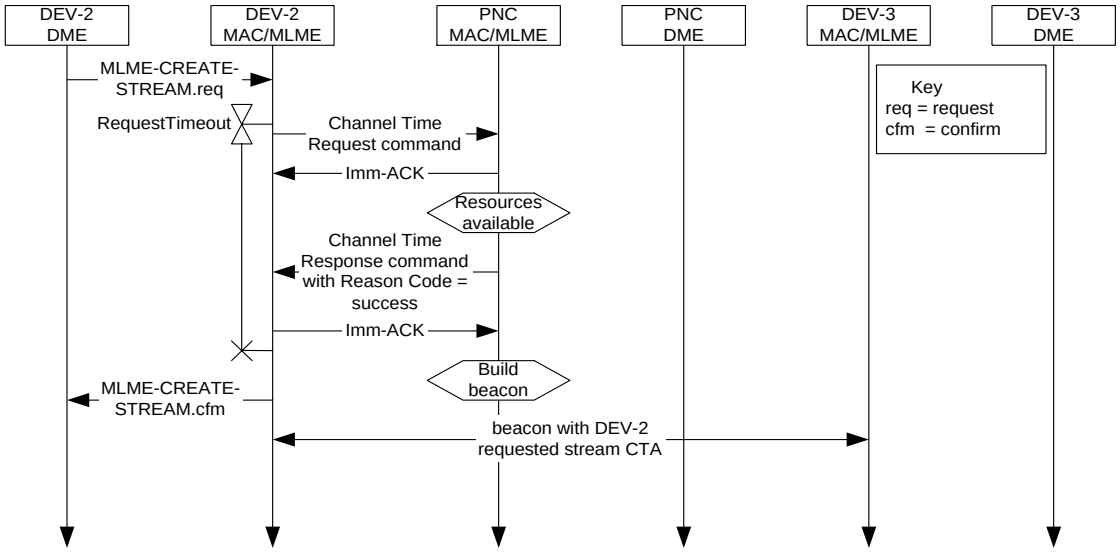

Figure 114—MSC for creating a DEV-2 to DEV-3 stream

Figure 115 illustrates the sequence of messages involved in an unsuccessfull attempt to establish a DEV-2 to DEV-3 stream in a piconet.

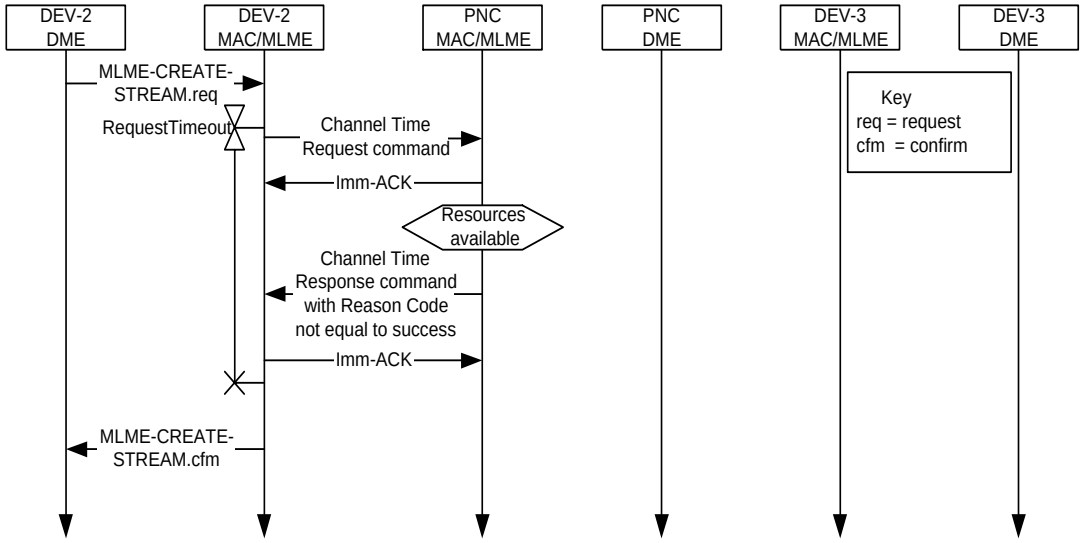

Figure 115—MSC for a denied DEV-2 to DEV-3 stream

Figure 116 illustrates the sequence of messages involved in successfully establishing a DEV-1/PNC to DEV-2 stream in a piconet. In this standard, the MSC convention is to refer to the DEV role of the PNC as DEV-1/PNC.

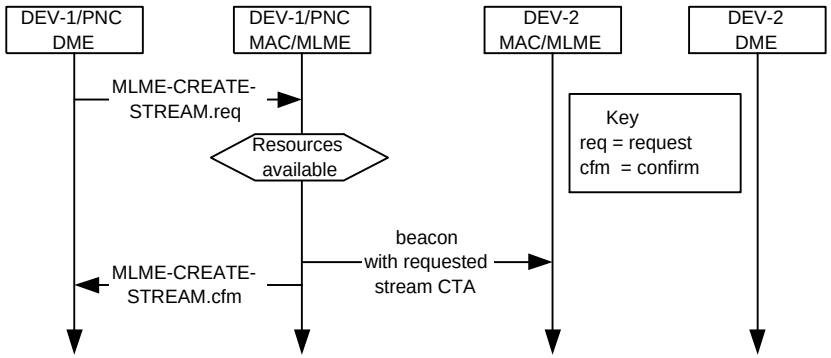

Figure 116—MSC for creating a DEV-1/PNC to DEV-2 stream

8.5.1.2 Isochronous stream modification

Only the originating DEV or the PNC may modify an established isochronous stream. The originating DEV that is requesting a modification of the channel time allocated to one of its streams shall send the PNC a Channel Time Request command with these parameter values:

— Target ID List field is set to the DEVID with which the originating DEV has an established stream.
— Stream Request ID field set to zero.
— Stream Index field set to the index of the stream to be modifed.
— The CTA Type field shall be set to the same value as in the original request for that stream index.
— All the other Channel Time Request command parameters are set to appropriate values as defined in 7.5.6.1.

The PNC upon receiving the Channel Time Request command shall check to see if the requested resources are available. If the requested channel time is not available the PNC shall:

— Send a Channel Time Response command with the Available Number Of TUs equal to the previously assigned Available Number Of TUs.
— The Stream Request ID field set to zero.
— The Reason Code field set to ‘channel time unavailable’.
— Await an Imm-ACK from the requesting DEV.
— Make no modification to the existing beacon CTA blocks for the DEV requesting the modification.

If the requested channel time is available, the PNC shall:

— Reserve the requested channel time.
— Send a Channel Time Response Command where:
  the “new" Available Number Of TUs field is greater than the "previous" Available Number Of TUs field and is less than or equal to the Desired Number Of TUs (for a requested an increase in channel time),
  or
  the "new" Available Number Of TUs field less than or equal to the Desired Number Of TUs field (for a requested decrease in channel time).
— The Stream Request ID field set to zero.
— The Reason Code field set to ‘success’.

— Await an ACK from the requesting DEV.

— Build a new beacon with the modified CTA.

The PNC shall announce the modification of those streams for which the destination DEV, or any intended destination DEV in the case of broadcast and multicast streams, is in power save mode. The PNC shall announce the modification of those streams for which one or more of the PM CTRq Type, CTA Rate Type, and the CTA Rate Factor fields are modified. The PNC shall make the announcement with the CTA Status IE, as described in 7.4.10, using the beacon information announcement mechanism, as described in 8.6.4. The PNC shall issue the first modified CTA for the stream in the superframe indicated in that IE. If the target DEV is in DSPS mode, the PNC shall also allocate an uplink MCTA in the same superframe as when the CTA is first allocated with the DSPS DEV as the source and the PNC as the destination that is long enough to handle a PM Mode Change command, a Channel Time Request command with 4 isochronous CTRqBs, and the associated Imm-ACKs and SIFSs.

A dependent PNC, the originator DEV, may handover control of the dependent piconet's CTA to another DEV, the target DEV, in the parent piconet. The target DEV shall be either a member of the piconet or a DEV that has associated as a neighbor PNC, as described in 8.2.6. To handover control of the dependent piconet's CTA, the originator DEV shall send a Channel Time Request command to the parent PNC with the following parameters:

— The Num Targets field set to one.

— The Target ID List field containing the DEVID of the target DEV that is to receive control of the CTA

— The Stream Request ID field set to zero.

— The Stream Index field set to the stream index of a CTA that has already been allocated to the dependent PNC as a private, pseudo-static CTA.

— All other fields set to the same values as in the last successful Channel Time Request for this Stream Index.

If the target DEV indicated in the Target ID List is either a member of the parent piconet or is an associated neighbor PNC and the Channel Time Request command has the correct entries as indicated above, the parent PNC shall grant the request to change the source and destination for the stream and shall send a Channel Time Response command to the originator with the Reason Code set to 'Success.' The PNC shall continue to place the CTA block for the allocation in the beacon but shall change the SrcID and DestID to be equal to the target DEV's DEVID. Once the PNC has changed the SrcID and DestID in the CTA block, the target DEV will have gained control of the CTA and will be allowed to request modification or termination of the allocation.

If the target DEV is not a member of the piconet and it is not an associated neighbor PNC, the parent PNC shall reject the request and shall send a Channel Time Response command to the originator with the Reason Code set to either 'target DEV unassociated' or 'target DEV not a member' depending on the status of the target DEV.

If the Channel Time Request command has improper entries, e.g. the Stream Index does not exist or the Stream Index is not associated with a private, pseudo-static CTA, then PNC shall reject the request and shall send a Channel Time Response command to the originator DEV with the Reason Code set to 'request denied.'

Figure 117 illustrates the message sequence involved in requesting a modification to an existing stream.

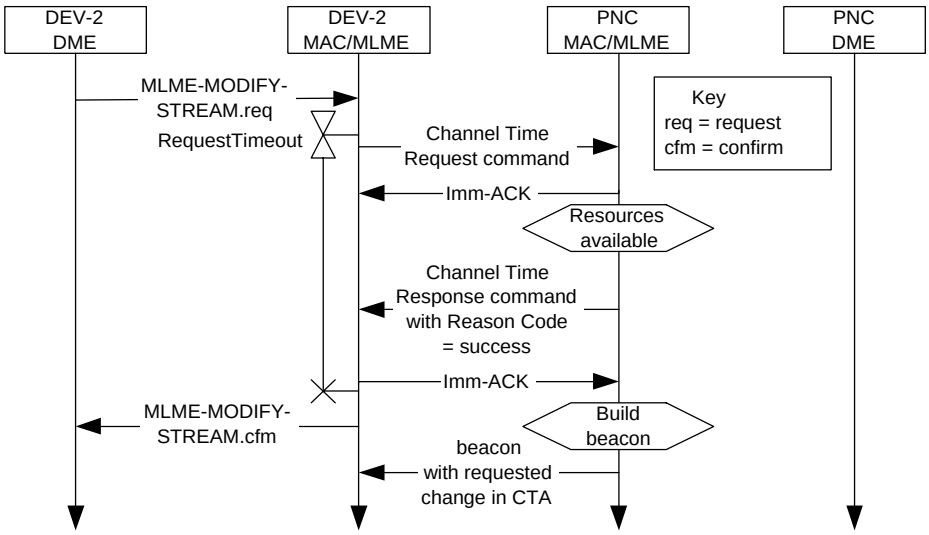

Figure 117—MSC for modifying a stream

Figure 118 illustrates the message sequence involved when a requested stream modification for an existing stream is denied.

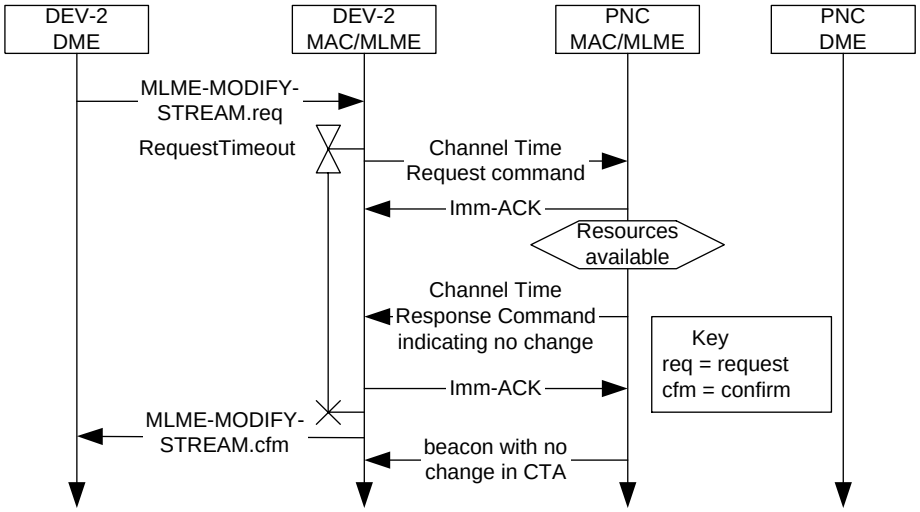

Figure 118—MSC for a denied stream modification

The MSC for the handover of the control of a private, pseudo-static CTA is illustrated in Figure 119.

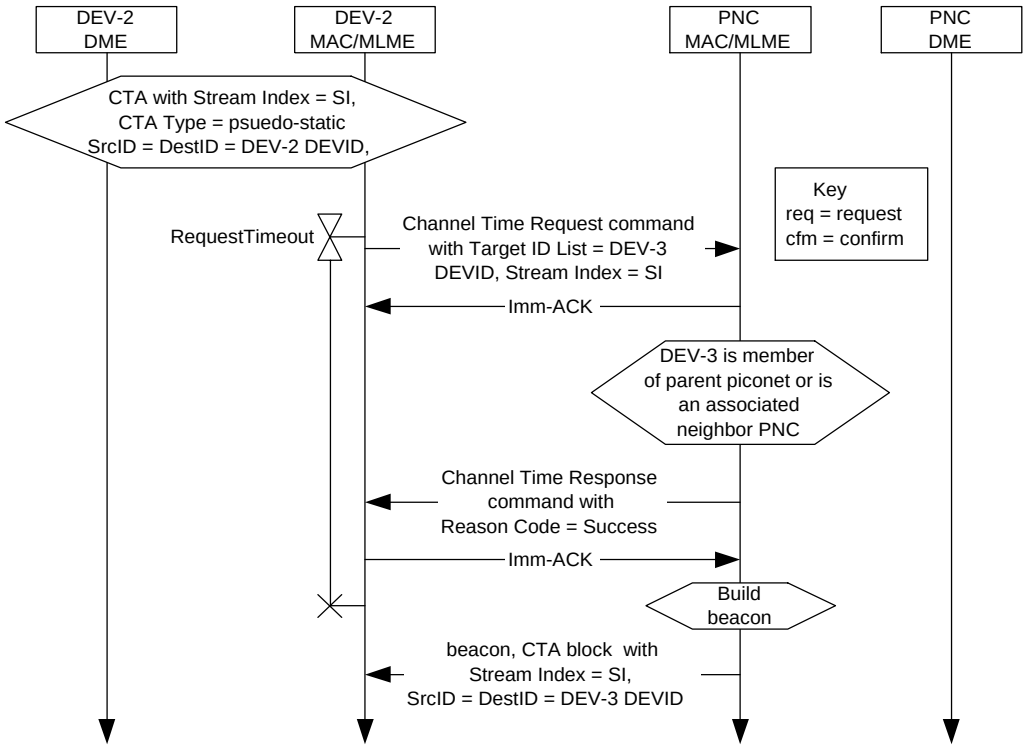

Figure 119—MSC for the handing over control of a private, pseudo-static CTA

8.5.1.3 Isochronous stream termination

Only the PNC, the originating DEV, or the target DEV may terminate an established stream. In the case of multicast or broadcast streams, only the originating DEV or the PNC may terminate the stream. In the case where either the originating DEV or the target DEV desires to terminate a specific stream, it shall send the PNC the Channel Time Request command with these parameter values:

— Target DEVID List field is set to the DEVID of the DEV to which the originating DEV has an established stream.

— Stream Index field shall be set to the value of the stream to be terminated.

— All other fields shall be set to zero.

The PNC, upon receiving a Channel Time Request command from a DEV requesting stream termination, shall respond with an Imm-ACK. In the case where the originating DEV is requesting a stream termination, the PNC shall then notify the target DEV of the termination via a null CTA block in the beacon. The null CTA block shall appear in at least mMinBeaconInfoRepeat consecutive beacons. For CTAs that were not allocated every beacon, i.e. sub-rate CTAs, the first null CTA block shall be placed starting in the beacon when the next CTA would have been allocated. A null CTA block has the stream index, SrcID and DestID set to the appropriate values with zero values for the CTA location and CTA duration, 7.4.1. Figure 120 illustrates the MSC for termination of a stream by a source DEV.

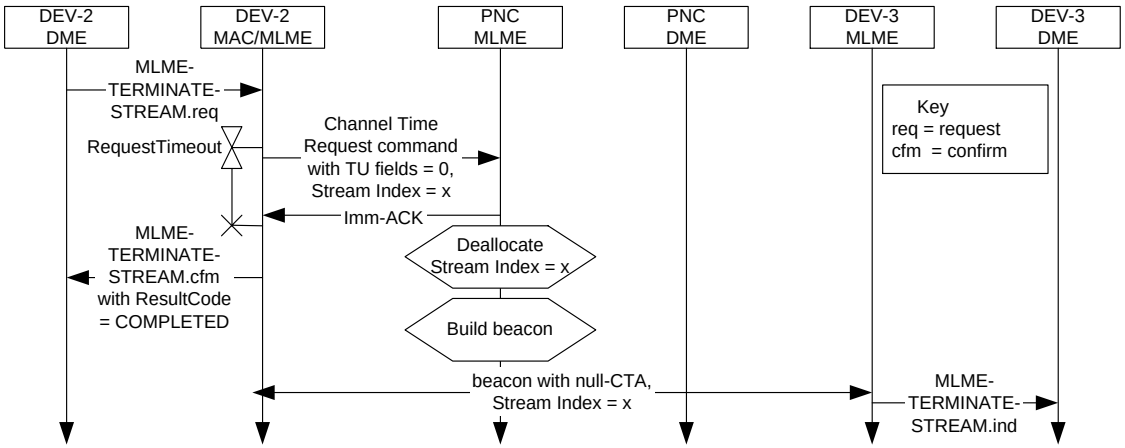

Figure 120—MSC of source DEV-2 requesting termination of its stream

In the case where the target DEV is requesting a stream termination, the PNC shall then notify the originating DEV of the termination via a Channel Time Response command. Figure 121 illustrates the MSC for termination of a stream by a target DEV.

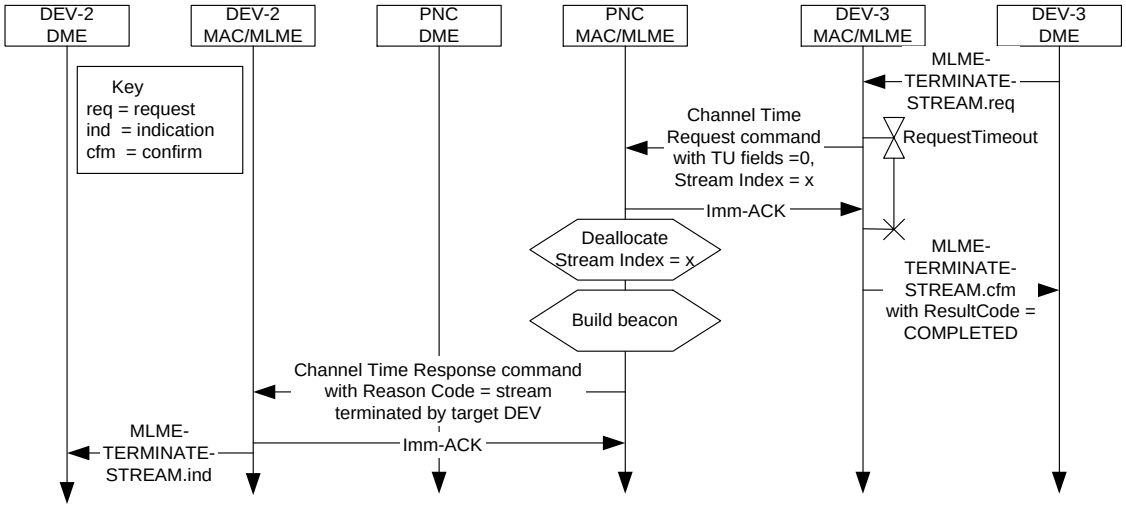

Figure 121—MSC of target DEV-3 requesting termination of source DEV-2's stream

In the case where the PNC decides to terminate an originating DEV's stream, the PNC shall notify the source DEV via a Channel Time Response command and the target DEV via a null CTA in at least mMinBeaconInfoRepeat consecutive beacons. For CTAs that were not allocated every beacon, e.g. sub-rate CTAs, the first null CTA block shall be placed starting in the beacon where the next CTA would have occured. Figure 122 illustrates the termination of a source DEV's stream by the PNC.

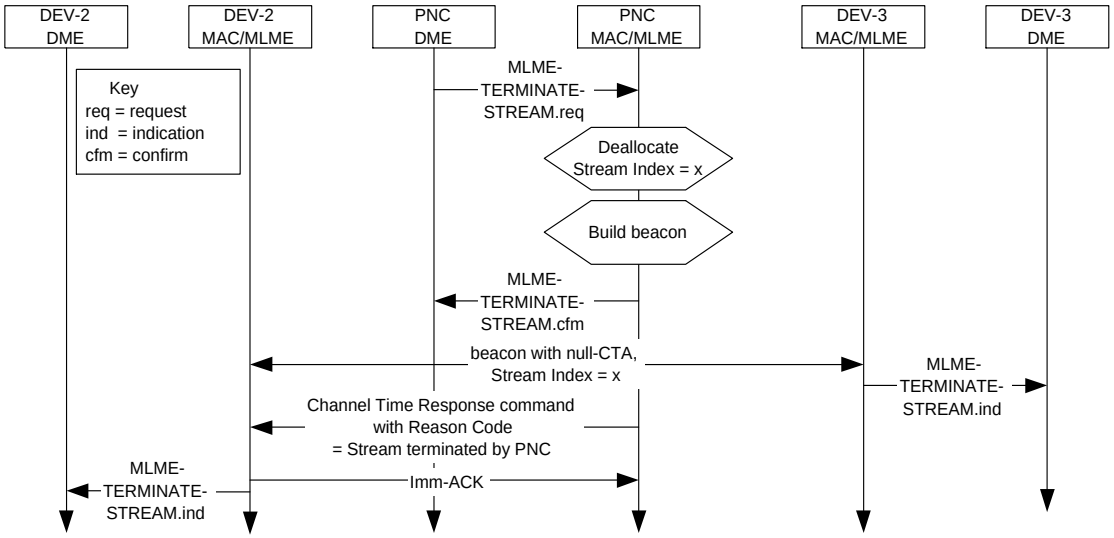

Figure 122—MSC of PNC terminating a stream

8.5.2 Asynchronous channel time reservation and termination

This subclause describes the process used to allocate asynchronous channel time with other DEVs in the piconet.

8.5.2.1 Asynchronous channel time reservation

There are two methods for requesting asynchronous channel time:

— Request a single CTA for multiple target DEVs, i.e. group asynchronous CTRq.
— Request individual CTAs for each of the target DEVs, i.e. a individual asynchronous CTRq.

The DEV requesting asynchronous channel time shall only use one of the two methods at a time. The DEV switches between the two methods by sending a Channel Time Request command that utilizes the new method. If the DEV changes methods, the PNC shall drop previously received asynchronous CTRqBs from that DEV. A DEV shall not send a Channel Time Request that requests both types of asynchronous allocations and the PNC shall reject any request received from a DEV that requests both types of asynchronous allocations.

When a DEV is requesting the creation or modification of channel time for asynchronous data transmissions with a target DEV or DEVs, the originating DEV shall send a Channel Time Request command, 7.5.6.1, to the PNC with these parameter values:

— The target ID list shall contain either:
  — A list of all of the target DEVs. Only one CTRqB is used for all destinations with the same TU for all of the target DEVs.

  or

  — Only one DEV in the destination list. In this case the originating DEV may send multiple CTRqBs in the command and the TU may be different in each of the CTRqBs.
— Stream Index field shall be set to zero.
— Priority field shall be set to a value of either 0b000 or 0b001 as defined in A.1.2.1.

— The DSPS Set Index, PM CTRq Type, CTA Type and CTA Rate Type fields shall be set to zero and may be ignored upon reception.
— All the other Channel Time Request command parameters are set to appropriate values as defined in 7.5.6.1.

The PNC upon receiving the Channel Time Request command from the originating DEV shall respond with an Imm-ACK to the requesting DEV. If the requested channel time is available the PNC places the CTA block(s) in a beacon with the source and target DEVID fields appropriately set.

In the case of a group asynchronous allocation, the PNC shall place multiple CTA blocks in the beacon, one for each of the destinations. Each CTA block shall have the asynchronous stream index and the same SrcID, start time and duration but different DestIDs. The PNC may also split a group asynchronous allocation into several CTAs in a single superframe, with any such CTA again announced by multiple CTA blocks that overlap in time but have different DestIDs. Such splits shall only be done on the TU boundaries.

For an individual asynchronous allocation, if any of the DestIDs is currently in a power save mode, the PNC shall allocate the CTAs for the power save DEV(s) in the DEV's next wake beacon, either system or DSPS. It is the responsibility of the source DEV to be able to handle different CTAs for destination DEVs in power save mode as opposed to DEVs in ACTIVE mode.

For group asynchronous allocations, if some of the DEVs are in a power save mode, then the PNC shall allocate as much as possible of the requested channel time in their wake beacons. The rest of the channel time may be allocated in non wake beacons.

There is no guarantee of the delay between the time of the request and the reception of a beacon containing the requested CTA. If a frame's timeout interval expires while waiting for its requested CTA in the beacon, an MAC-ASYNC-DATA.confirm shall be sent with the ReasonCode set to TX_TIMEOUT.

When the PNC allocates an asynchronous CTA it decrements its count of unallocated TUs by the number of TUs allocated in the CTA. When the count of unallocated TUs reaches zero, the PNC no longer allocates the CTA and drops the CTRqB.

If the request is rejected, the PNC shall send a Channel Time Response command indicating the rejection. Note that if an asynchronous request is queued by the PNC but that channel time is not immediately available due to resource constraints, that does not constitute a rejection of the request.

A new asynchronous CTRq to a target DEV replaces the previous request for that target DEV and unallocated TUs from the previous request shall be replaced by the current request. A new group asynchronous CTRq replaces all previous asynchronous requests and unallocated TUs from all previous asynchronous requests shall be replaced by the current request. The originating DEV may change the request method, TU size, destinations and desired TUs between subsequent asynchronous CTRq's.

The PNC may time out the request for an asynchronous CTA and purge them after mAsyncRequestLifetime. The requesting DEV should resend a new request after mAsyncRequestLifetime if it desires more channel time.

Figure 123 illustrates the sequence of messages involved in reserving channel time for the exchange of asynchronous data between DEV-2 and DEV-3 in a piconet.

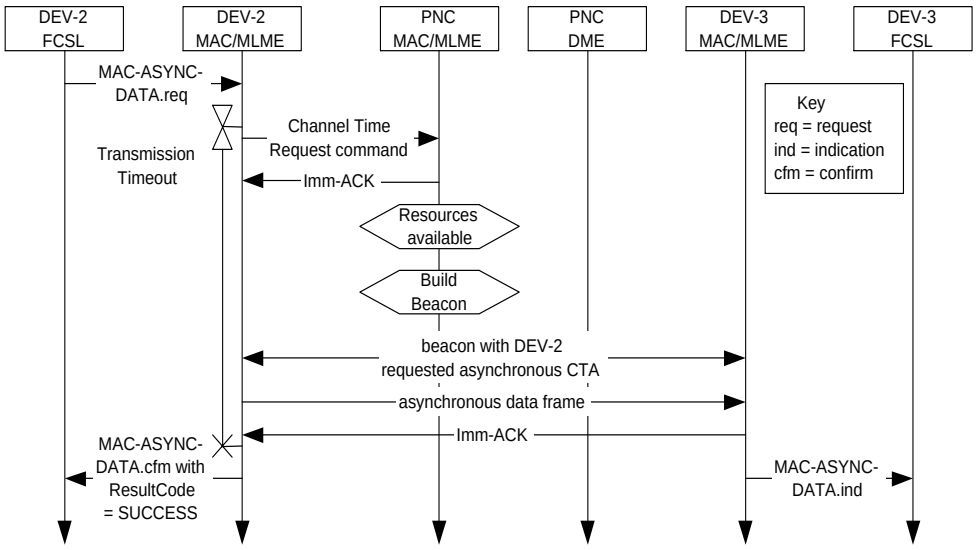

Figure 123— MSC for reserving asynchronous data channel time

If the target DEV is in APS mode and the PNC grants the channel time request, the PNC shall set the Reason Code in the Channel Time Response command to ‘Success, DEV in PS mode.’ The PNC shall place the PCTM IE in the beacon with a bit set for the target DEV, as described in 7.4.8.

When the Target DEV in APS mode receives a beacon with its bit set in the PCTM IE, it shall send a PM Mode Change command to the PNC. If the DEV is going to remain in APS mode it shall set the PM Mode field in the PM Mode Change command to ‘APS.’ The PNC shall then terminate the asynchronous channel time, as described in 8.5.2.2.

If the power save DEV is going to listen to the new allocation, it shall set the PM Mode field in the PM Mode Change command to ‘ACTIVE.’ The PNC shall then begin allocating the channel time in the beacon for the asynchronous allocation. The PNC shall no longer set the bits for the DEV in the PS Status IEs.

If the PNC does not receive the PM Mode Change command from the APS DEV within a timeout determined by the PNC, the PNC shall terminate the channel time request, 8.5.2.2, and unset the DEV’s bit in the PCTM IE.

8.5.2.2 Asynchronous channel time termination

Only the PNC or the originating DEV shall be allowed to terminate an asynchronous CTA. In the case where the originating DEV is going to terminate a specific asynchronous CTA, it shall send to the PNC the Channel Time Request command with these parameter values:

— TargetID List field shall be set to the DEVIDs of the DEVs with whom the originating DEV is going to terminate the asynchronous connection.
— Stream Index field shall be set to the asynchronous stream index, as described in 7.2.5.
— All other fields shall be set to zero.

The PNC, upon receiving the Channel Time Request command from a DEV requesting termination of the asynchronous channel time, shall respond with an Imm-ACK and shall cease allocating the channel time.

In the case where the PNC terminates the asynchronous channel time, the PNC shall notify the source DEV via a directed Channel Time Response command, as described in 7.5.5.2, with the reason code set to the appropriate value.

8.6 Synchronization

All DEVs within a single piconet shall be synchronized to the PNC's clock. In addition, child or neighbor PNCs shall synchronize their piconet's time usage to the parent PNC's beacon and their CTA. The beacon sent at the beginning of every superframe contains the information necessary to time-synchronize the DEVs in the piconet. See 7.3.1 for the definition of the timing parameters sent in the beacon.

Each DEV in the piconet, including the PNC, shall reset its superframe clock to zero at the beginning of the beacon preamble, as shown in Figure 124. All times in the superframe shall be measured relative to the beginning of the beacon preamble. If a DEV does not hear a beacon, it should reset its superframe clock to zero at the instant where it expected to hear the beginning of the beacon preamble.

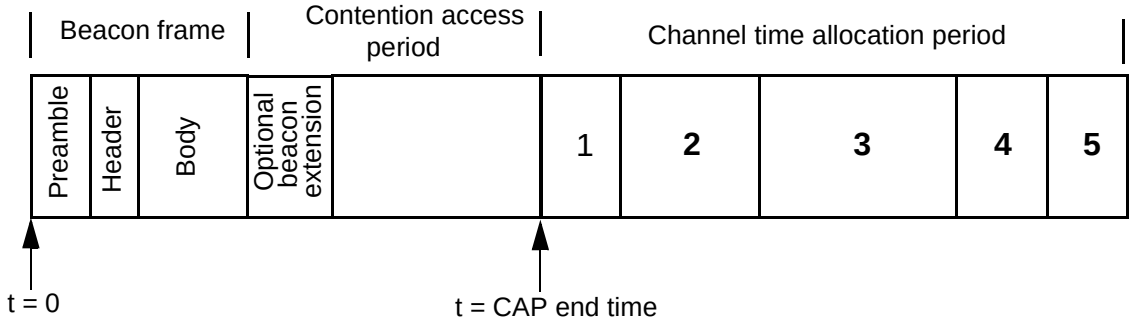

Figure 124—Piconet timing relative to the beacon

8.6.1 Time accuracy

A compliant implementation shall maintain the accuracy of the timer to be at least as accurate as pPHY-ClockAccuracy.

8.6.2 Beacon generation

The PNC shall send a beacon at the beginning of each superframe using the beacon frame described in 7.3.1.

If the PNC determines that the beacon frame is too large or if it is going to split the information in the beacon frame, it may send one or more Announce commands with the SrcID set to the PNCID and the DestID set to the BcstId following the beacon. This is called an extended beacon. Unless it is specified otherwise, the term beacon applies to both the beacon frame and the Announce commands that make up the extended beacon. The first Announce command shall be sent one MIFS following the beacon with any additional Announce commands following one MIFS after the the prior Announce command. If the PNC sends some of the beacon information in the broadcast Announce commands, it shall set the More Data bit to indicate 'more data' in the Frame Control field of the beacon frame and in all but the last Announce command frame used to communicate the IEs. The CAP or the CTAP, if the CAP isn't present, begins after the last Announce command that is part of the extended beacon. The PNC shall send CTA IEs, BSID IE, and the Parent Piconet IE, if present, only in the beacon frame and not in any of the broadcast Announce commands. The Announce commands are sent to the BcstID and so the ACK Policy field shall be set to no-ACK in these frames.

The PNC shall transmit the beacon such that the time between beacons is the superframe duration with an error of no more than pPHYClockAccuracy times the superframe duration. The PNC changes the superframe position or duration using the procedures indicated in 8.10.1 and 8.10.2, respectively.

8.6.3 Beacon reception

All of the DEVs that are associated shall use the beacon start time, CAP end time and the CTA IEs contained in the beacon to start their transmissions. The superframe duration and the CAP end time in the beacon, as described in 7.3.1, are used to accurately mark the beginning and the end of the CTAP. A lost beacon is defined as one for which the FCS is not valid or when a DEV has not received a beacon at the expected time.

8.6.4 Beacon information announcement

The PNC sends several IEs in its beacons to inform the DEVs in the piconet about constant or temporary conditions. Some IEs are sent in every beacon, while others are only sent if certain operations are in use, such as power save or a dependent piconet. Some of these IEs are listed in Table 56.

Table 56—IEs include in beacons as needed

Information element	Format	Usage
Application specific	7.4.7	8.14
Pending channel time map (PCTM)	7.4.8	8.13.2, 8.13.3
PS status	7.4.13	8.13
Continued wake beacon (CWB)	7.4.14	8.13.2

Other IEs are only sent as an announcement of a changed condition in the piconet. These IEs could be for the benefit of all DEVs or for a particular DEV. IEs that are not sent in every beacon are called announcements and are listed in Table 57.

Table 57—Repeated beacon announcements

Information element	Clause	Announced in	Intended for	Clause
DEV association	7.4.4	mMinBeaconInfoRepeat	All DEVs	8.3.1, 8.3.4
PNC shutdown	7.4.5	mMinBeaconInfoRepeat	All DEVs	8.2.7.1
Piconet parameter change	7.4.6	mMinBeaconInfoRepeat	All DEVs	8.10, 8.11.1, 8.11.2
PNC handover	7.4.9	mMinBeaconInfoRepeat	All DEVs	8.2.3
CTA status	7.4.10	mMinBeaconInfoRepeat	DestID	8.5.1.1, 8.5.1.2

If the intended recipient of an IE is all DEVs, the following rules apply:

— The IE shall be sent in at least mMinBeaconInfoRepeat consecutive beacons.
— If any DEV is in PSPS, the IE announcement shall be made in a system wake beacon and in at least mMinBeaconInfoRepeat-1 consecutive beacons following the system wake beacon.

If the intended recipient of an IE is one individual DEV, the following rules apply:

— The IE shall be sent in at least mMinBeaconInfoRepeat consecutive beacons.
— If the DEV is in PSPS mode, the IE announcement shall be made in a system wake beacon and in at least mMinBeaconInfoRepeat-1 consecutive beacons following the system wake beacon.
— If the DEV is in DSPS mode, the IE announcement shall be made in one of the DEV's DSPS set wake beacons and in at least mMinBeaconInfoRepeat-1 consecutive beacons following the DEV's DSPS set wake beacon.

A CTA Status IE is considered to be intended for all DEVs if the DestID contained in that IE is the BcstID or McstID. Otherwise the CTA Status IE is intended for the DEV defined by DestID.

8.6.5 Acquiring synchronization

All DEVs acquire synchronization through beacons from the PNC. Unassociated DEVs that wish to associate with the piconet use the information within the beacons for synchronization.

Figure 125 illustrates the message flow for a successful synchronization.

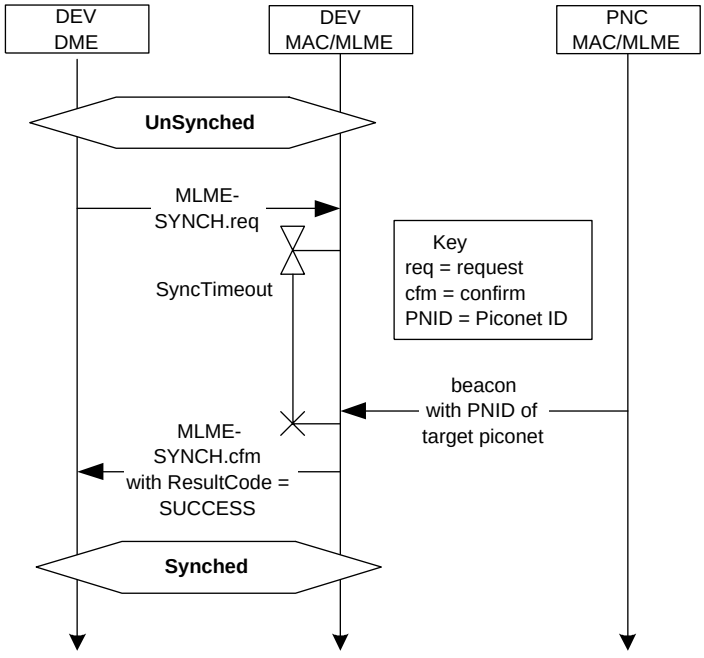

Figure 125—MSC of a DEV synchronizing with a PNC

8.7 Fragmentation and defragmentation

Fragmentation may be performed at the transmitting DEV on each MSDU. In addition, certain commands, i.e. MCDUs, may be fragmented, as indicated in 7.5. All fragments shall be of equal size, except the last fragment which may be shorter. Once the MSDU/MCDU is fragmented and a transmission attempted, it shall not be refragmented. The smallest size of a fragment, excluding the last fragment, shall be at least pMinFragmentSize. A DEV indicates its preferred fragment size for reception in the Preferred Fragment Size field in the DEV Capabilities field, as described in 7.4.11, that it sends to the PNC when the DEV associates with the piconet.

Each fragment shall be sent with the Last Fragment Number field set to the highest fragment number of the current MSDU/MCDU, which is one less than the total number of fragments of the current MSDU/MCDU.

The first fragment shall be sent with the Fragment Number field set to zero. Each subsequent fragment shall be sent with the Fragment Number field incremented by one. However, the Fragment Number field shall not be increased when a fragment is retransmitted.

All fragments of the same MSDU/MCDU shall have the same MSDU/MCDU number.

Defragmentation of an MSDU/MCDU is the reassembly of the received fragments into the complete MSDU/MCDU. The MSDU/MCDU shall be completely reassembled in the correct order before delivering it to the frame convergence sublayer (FCSL). Any MSDU/MCDU with missing fragments shall be discarded.

The receiving DEV shall not deliver an MSDU/MCDU to the FCSL until all of the fragments have been obtained. The receiving DEV may discard the fragments of an MSDU/MCDU if it is not completely received within a timeout determined by the receiving DEV. The destination DEV may also discard the oldest incomplete MSDU/MCDU if otherwise a buffer overflow would occur. If the no-ACK policy is used, the destination DEV shall discard an MSDU/MCDU immediately if a fragment is missing.

A DEV shall support concurrent reception of fragments of at least three MSDU/MCDUs including isochronous streams, asynchronous data and commands.

For frames with Imm-ACK mechanism, a DEV shall not send another fragment or frame with the same stream index to the same DEV until the sending DEV has received an Imm-ACK frame response to that frame or it has timed out on sending the frame.

If Dly-ACK is used, unacknowledged fragments from multiple MSDUs belonging to the same stream may be retransmitted in the same burst. In this case it is the responsibility of the destination DEV to deliver the MSDUs in the correct order to the FCSL.

8.8 Acknowledgement and retransmission

There are three acknowledgement types defined for this standard; no acknowledgement (no-ACK), immediate acknowledgement (Imm-ACK) and delayed acknowledgement (Dly-ACK).

8.8.1 No-ACK

A transmitted frame with the ACK Policy field set to indicate no-ACK shall not be acknowledged by the intended recipient(s). The transmitting DEV assumes that the frame is successful for all its local management entities and proceeds to the next frame scheduled for transmission. The ACK Policy field in broadcast and multicast addressed frames shall be set to no-ACK upon transmission.

8.8.2 Immediate ACK

A directed frame that expects an Imm-ACK shall have the ACK Policy field in that directed frame set to indicate the same, as defined in 7.2.1.4. If the intended recipient of a directed frame correctly receives the frame, it shall send the Imm-ACK frame as described in 8.4.1.

8.8.3 Delayed acknowledgement

Delayed acknowledgement (Dly-ACK) shall be used only for directed stream data frames, i.e. isochronous connections, where the Dly-ACK mechanism has been set up with negotiation between the source and desti-

nation DEVs. The Dly-ACK mechanism is initiated by the source DEV sending a single data frame with the ACK Policy field set to Dly-ACK Request.

If the destination DEV accepts the use of Dly-ACK, it shall respond with a Dly-ACK frame, acknowledging the received data frame and setting the Max Burst field to a value representing the maximum number of pMaxFrameBodySize MPDUs the source DEV may send in one burst. Because the receiver buffer requirement is equal to Max Burst field times pMaxFrameBodySize, the source may send as many smaller frames as will fit in the receive buffer window, up to a maximum of Max Frames, as provided in the Dly-ACK frame, as described in 7.3.2.2. The destination DEV may change the Max Burst value in each Dly-ACK frame. The MPDUs ACKed field shall be set to one and the MPDU ID field shall contain the information for the frame that was sent to negotiate the Dly-ACK.

If the destination DEV wants to decline the use of the Dly-ACK mechanism, it shall reply with an Imm-ACK frame. The source upon reception of the Imm-ACK shall send a MAC_ISOCH_DATA.confirm with the ResultCode set to DLY_ACK_FAILED to the FCSL. This implies acknowledgment of the data frame and additionally indicates that the use of the Dly-ACK policy has been refused by the destination. The FCSL would then notify the DME that the Dly-ACK negotiation failed, which might require a modification of the channel time allocation.

If the max burst value is set to zero, the source DEV shall stop transmitting in the current CTA and reopen the Dly-ACK mechanism by sending a single frame with the ACK Policy field set to Dly-ACK Request in the next CTA for this stream. If the value is not zero the source DEV may continue transmission in the current CTA, if the time is available.

The source DEV may change the ACK policy in a stream from Dly-ACK to Imm-ACK or no ACK by sending a frame with the ACK Policy field set to one of those values. This has the effect of canceling the Dly-ACK policy and the source shall use the Dly-ACK negotiation procedure before restarting the Dly-ACK mechanism. The receiver shall no longer maintain the ACK status of any previous frames sent with the ACK Policy field set to Dly-ACK.

The source DEV may send any MPDUs including retransmissions and new MPDUs, as long as the total size of transmitted data is less than size indicated by the Max Burst value set by the destination DEV.

If the Dly-ACK frame is not received when requested, the last data frame of the burst is repeated until the Dly-ACK frame is received. The source DEV may send an empty data frame which was not in the original burst, as an alternative to resending the last data frame, as long as the total number of frames, including the empty one, does not exceed Max Frames. The source DEV shall not start or resume burst transmissions until a Dly-ACK frame is received.

The destination MAC shall deliver MSDUs for each isochronous stream in ascending MSDU number order to its FCSL. If necessary to accomplish this, a destination MAC may discard correctly received (and potentially acknowledged) frames.

Figure 126 illustrates the Dly-ACK negotiation. In the figure, the term “Alt” indicates that two alternate outcomes are illustrated in the MSC.

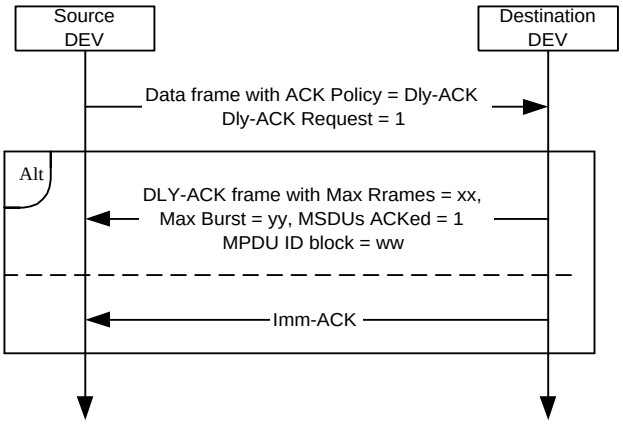

Figure 126—MSC for Dly-ACK negotiation

8.8.4 Retransmissions

During the CAP, retransmissions shall follow the backoff rules as specified in 8.4.2.

During CTAs within the CTAP when an Imm-ACK or Dly-ACK is expected, but is not received during a RIFS, the source DEV shall start the retransmission of the frame (or new frame if the failed frame’s retransmission limit has been met) after the end of RIFS as long as there is enough channel time remaining in the CTA for the entire frame exchange.

A DEV determines the number of times a frame is retried before the DEV discards that frame. If the DEV gives up on a fragment of an MSDU/MCDU, the DEV shall discard all MPDUs of that MSDU/MCDU.

8.8.5 Duplicate detection

Because the DEV sending the data frame may not correctly receive an ACK, duplicate frames may be sent even though the intended recipient has already received and acknowledged the frame. Hence all DEVs shall detect such multiple receptions and indicate the data frames to the higher layers only once. The SrcID, Stream Index, Fragmentation Control field, Retry bit, and PNID are used to detect multiple receptions of the same frame.

8.9 Peer discovery

Each DEV that is a member of the piconet may use the PNC Information Request command, as described in 7.5.4.1, to obtain information about other DEVs in the piconet. In addition the DEV may use the Probe Request command, as described in 7.5.4.5, to obtain other information required for peer-to-peer communication. The remote scan procedure is used by the PNC to determine channel conditions. All DEVs in the piconet are able to use the Channel Status Request, as described in 7.5.7.1, and Channel Status Response, as described in 7.5.7.2, commands to gather information about the quality of their link with another DEV.

8.9.1 PNC information request

A DEV may request information about either a single DEV in the piconet or about all of the DEVs in the piconet by sending a PNC Information Request command, as described in 7.5.4.2, to the PNC. If the DEV is requesting information about only a single DEV in the piconet, it shall set the DEVID in the command to the ID of that DEV. If the DEV is requesting information about all of the DEVs in the piconet, it shall set the DEVID in the command to the BcstID, as described in 7.2.3. If the originating DEV requests information about a single DEV that is not a member of the piconet, then the PNC shall send a PNC Information command with length zero. Otherwise, the PNC shall send the PNC Information command with the information requested by the originating DEV.

Figure 127 illustrates the sequence of messages involved in acquiring PNC information regarding a specific DEV or all of the DEVs that are members of the piconet.

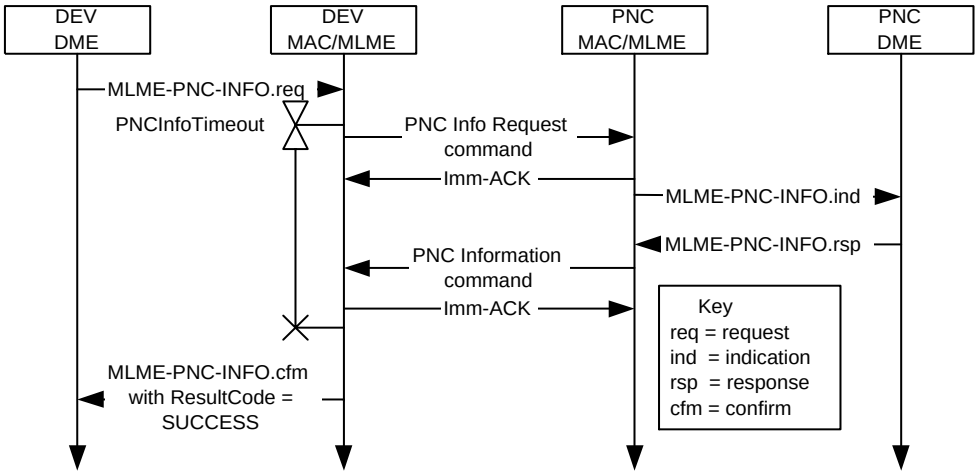

Figure 127—MSC for acquiring information regarding a specific DEV or all of the DEVs from the PNC using the PNC Information Request and PNC Information commands

8.9.2 Probe request and response

The Probe Request command provides the ability to request IEs from a target DEV. If the target DEV receives the Probe Request command, it shall respond to the originator with Probe Response command(s) that shall have the IEs requested by the originator.

A DEV may request information about an isochronous stream by sending a Probe Request command requesting the CTA Status IE, as described in 7.4.10, with the Request Index field set to the stream index of the stream for which CTA information is requested. If the Request Index field is set to zero, the DEV is requesting information about all isochronous streams directed to the requesting DEV and to the BcstId and McstId. The PNC shall respond to a Probe Request command containing a request for the CTA Status IE by sending Probe Response command(s) containing the appropriate CTA Status IE(s).

Any DEV may send the Probe Request command with the Information Requested field set to zero and ACK Policy field set to Imm-ACK to any other DEV in the piconet to determine if the destination DEV is still present in the piconet and is within range of the sending DEV.

A DEV that is going to send a Probe Request command to a DEV operating in a power save mode should consider the operation of those modes as described in 8.13 to determine the appropriate time to send the Probe Request command and the time to expect a response.

Figure 128 illustrates the sequence of messages involved in acquiring IEs from a target DEV using the Probe Request and Probe Response commands.

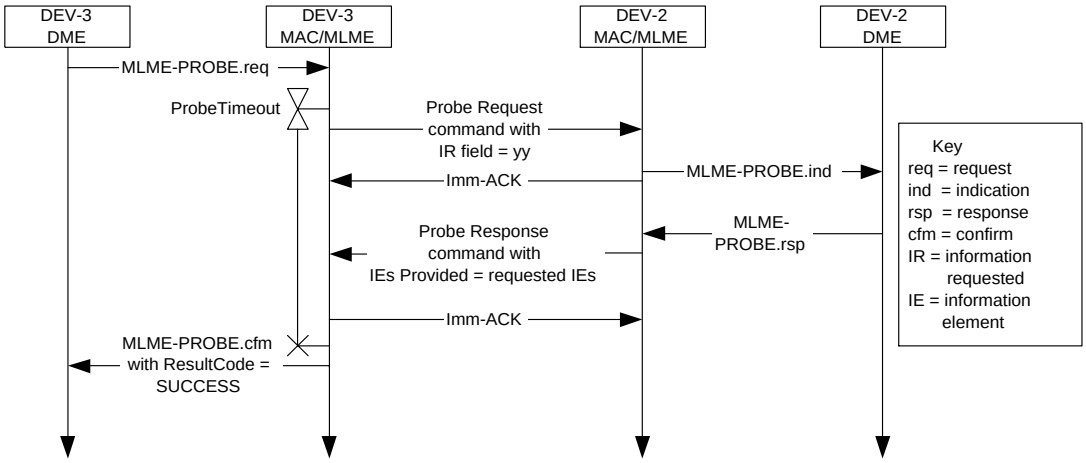

Figure 128—MSC for acquiring DEV IEs using the Probe Request and Probe Response commands

The types of IEs that are allowed to be requested or to be sent in the response depend on the status of the originator as either a DEV or PNC. The rules for requesting a specific IE are listed in Table 51 while the rules for responding to the request for a specific IE are listed in Table 52.

8.9.3 Announce

The Announce command provides the ability to send unrequested IEs to a target DEV. This command shall have one or more IEs that the originator is sending to the target.

A DEV that is going to send an Announce command to a DEV operating in a power save mode should consider the operation of those modes as described in 8.13 to determine the appropriate time to send the Announce command.

Figure 129 illustrates the sequence of messages involved in using the Announce command for sending information.

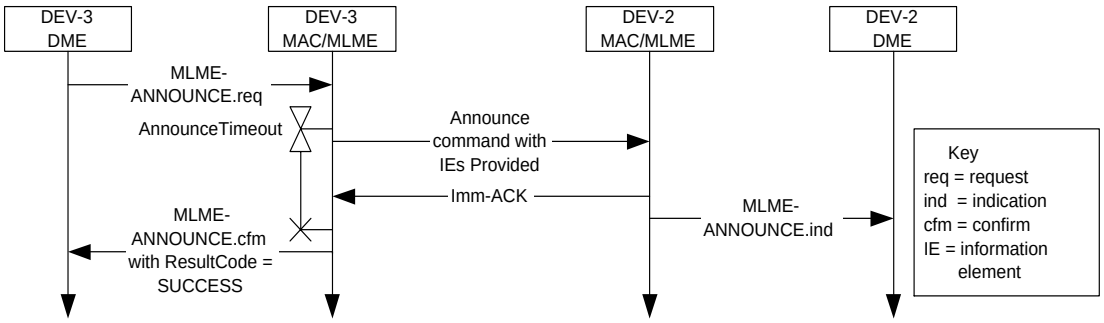

Figure 129—MSC showing sending of information using the Announce command

The types of IEs that are allowed to be sent depend on the status of the originator as either a DEV or PNC. The rules for sending a specific IE are listed in Table 53.

8.9.4 Channel status request

The Channel Status Request command, as described in 7.5.7.1, may be used by any DEV in the piconet to get information from a target DEV about the link quality between the two DEVs. The originating DEV sends the Channel Status Request command to the target DEV to start the process. When the target DEV receives the request, it shall send a Channel Status Response command to the originating DEV with all of the information specified in 7.5.7.2.

The information conveyed in the channel status process is based on the results of an attempted data transfer between the two DEVs. Thus, the Channel Status Request command should only be used for DEVs that are actively participating in a data transfer as the information would not have much meaning otherwise. The PNC also uses this command to get the channel status information from the DEVs in the piconet, as described in 8.11.1, 7.5.7.1, and 7.5.7.2

Figure 130 illustrates the sequence of messages involved in determining the channel status between any two DEVs.

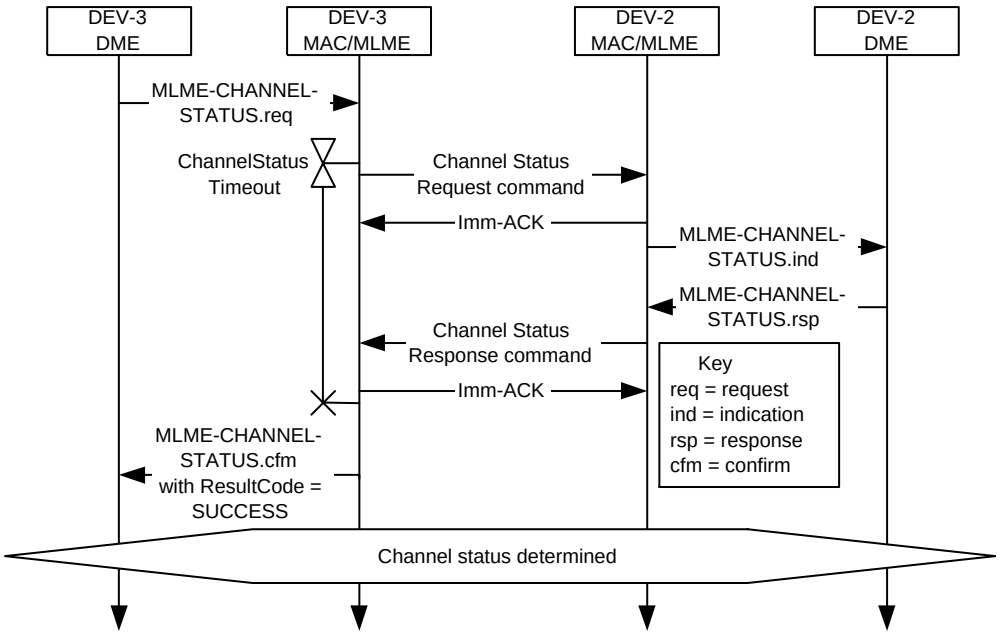

Figure 130— MSC for determining the channel status between two DEVs using the Channel Status Request and Channel Status Response commands

8.9.5 Remote scan

Remote scan is a procedure by which a PNC may request that a target DEV in the piconet initiate a channel scan on the PNC's behalf of the specified channels and to report the results of the scanned channels back to the PNC. The PNC may then use the results of the remote scan to initiate a change in maximum transmit power for the piconet, initiate a channel change to a channel with better channel characteristics, or other action. The algorithm for determining which procedure to execute is outside of the scope of this standard.

The PNC may optionally allocate channel time in the CTAP so that there is quiet time for the remote DEV to scan the channel for other piconets.

One of the reasons that the PNC requests a remote scan is because it determines that the current channel is unsatisfactory for continued operation of the piconet. This allows the PNC to get information about other wireless networks that may be out of range of the PNC but in range of some of the DEVs in the piconet. The PNC is able to get this information while it continues allocating channel time and generating beacons for its piconet.

The PNC initiates a remote scan by sending the Remote Scan Request command with a list of channels to a DEV in the piconet. The target DEV should accept the request to perform a channel scan on behalf of the PNC. If the DEV does not accept the request, it shall respond to the PNC by sending a Remote Scan Response command with a ReasonCode set to ‘request denied’. The PNC upon receiving this response, may send a Remote Scan Request command to another piconet DEV, initiate its own channel scan or take other action.

In the case where the target DEV does accept the request, the target DEV shall initiate a series of OpenScan channel scans, as described in 8.2.1, based upon the channel list passed to it in the Remote Scan Request command. The target DEV shall scan each of the channels requested by the PNC. When the target DEV has finished scanning the channels, it shall respond via the Remote Scan Response command to the PNC with the Remote Piconet Description Set parameters and the Channel Rating List field. The PNC upon receiving this information may then determine that there is a new channel with better characteristics than its current channel. The PNC may use this information to initiate the change channel procedure. The PNC may decide instead that a more appropriate solution to the current channel impairment is to increase/decrease transmission power level of the piconet. The PNC may also decide to do nothing or to take other unspecified action.

A DEV that is receiving beacons from more than one PNC may send an unsolicited Remote Scan Response command to its PNC with a Piconet Description Set representing the interfering PNCs. A DEV shall not report overlapping piconets if it determines that the beacons were received from a child or 802.15.3 neighbor piconet that is associated with the DEV's current piconet.

Figure 131 illustrates the sequence of messages involved in a remote scan initiated by the PNC.

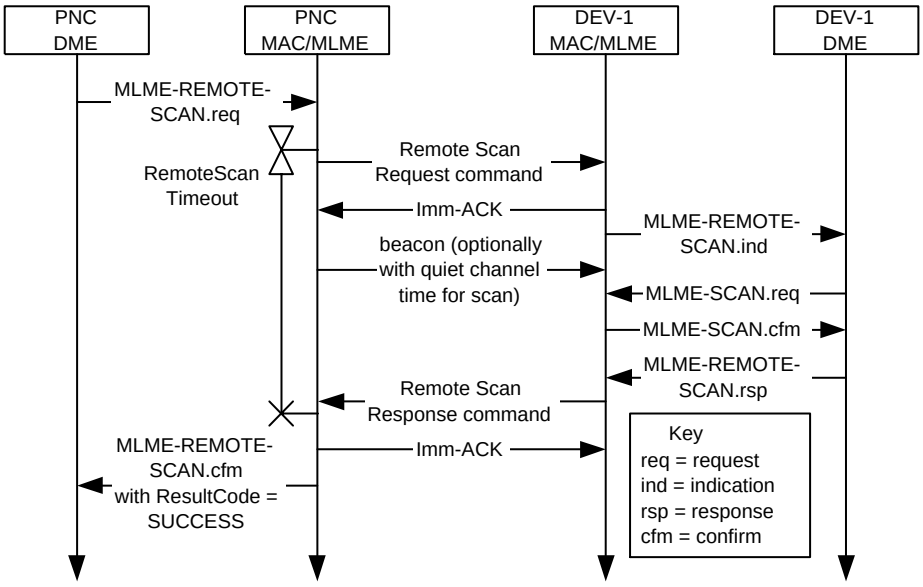

Figure 131— Remote scan MSC

8.9.6 PNC channel scanning

PNC channel scanning is a procedure by which the PNC is able to determine the channel characteristics of not only its current operating channel but also the channel characteristics of other channels. The PNC may use the results of its channel scans to determine whether the current channel in which it is operating has acceptable characteristics or that there is another channel(s) with better channel characteristics than its current channel.

The PNC may allocate CTAs such that there is unallocated channel time in the CTAP. This provides quiet time for the PNC to scan channels for other 802.15.3 piconets, non 802.15.3 wireless networks, or interference.

If the PNC initiates a scan of one or more alternate channels, the PNC shall not transmit a beacon for one or more beacon intervals. The PNC shall not suspend beacon transmissions for more than twice aMinChannelScan. The PNC, upon returning to its current channel and resuming the transmission of its beacons, shall increment the Time Token field by the number of beacons not sent during the time the PNC was scanning one or more alternate channels.

After scanning both its current channel and other channels, the PNC may initiate one of these options:

1) Do nothing since the PNC has determined that none of the alternate channels were better than its current channel.
2) Initiate the dynamic channel change procedure described in 8.11.1.
3) Increase or decrease the max tx power level of the piconet, as described in 8.11.2.
4) Initiate some other unspecified action.

The algorithm for determining when to initiate any of these actions is outside the scope of this standard.

8.10 Changing piconet parameters

This subclause describes the methods used to change certain key characteristics of the piconet.

A PNC shall not change either the pseudo-static CTAs or the pseudo-static CTA blocks during a piconet parameter change. If the parent needs to move a pseudo-static CTA because the superframe duration is being reduced, it shall do so prior to using the superframe duration change process, as described in 8.10.2. If a child or 802.15.3 neighbor piconet has the same superframe duration as the parent, then it shall use the value of the Change Beacon Number field in the Piconet Parameter Change IE from the parent's beacon in the Piconet Parameter Change IE in its own beacon. The exceptions to this are:

— When the parent is changing its PNID or BSID.
— A child or neighbor PNC decides not to change channels with the parent PNC and is shutting down, as described in 8.11.1.

When the Change Beacon Number field is used with changing parameters, the PNC shall ensure that the Piconet Parameter Change IE announcement complies with the rules for beacon announcements in 8.6.4. The PNC shall not place more than one Piconet Parameter Change IE, as described in 7.4.6, in any beacon.

The PNC shall not be required to wait until all of the DEVs in power save modes are in ACTIVE mode before changing a piconet parameter. A power save DEV is not required to switch to ACTIVE mode when there is a piconet parameter change in progress. However the power save DEV shall update its internal values for the new piconet parameter at the time indicated in the Piconet Parameter Change IE.

8.10.1 Moving beacon

The PNC may move the relative position of its beacon. Moving a beacon means that the superframe duration is unchanged while the position of the beacon is moved.

If the PNC wishes to move its beacon position, it shall insert the Piconet Parameter Change IE, as described in 7.4.6, into NbrOfChangeBeacons beacons with the change type set to MOVE and the superframe timing set to the delay of the first beacon after this sequence compared to previous beacon transmission time.

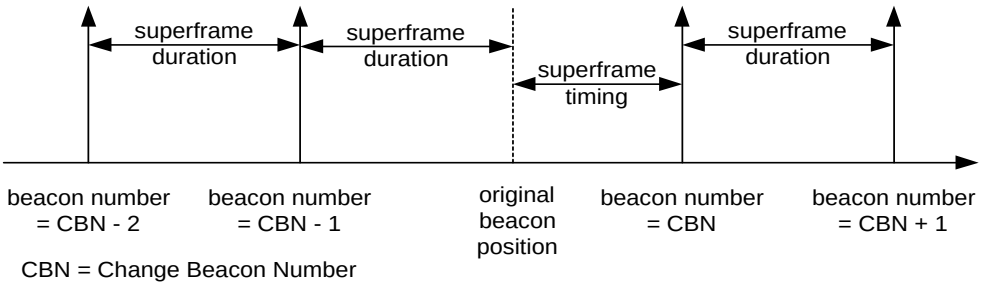

Figure 132—Moving the beacon position

8.10.2 Changing superframe duration

The PNC may change the duration of its superframe. If the PNC wishes to change its superframe duration, it shall insert the Piconet Parameter Change IE, as described in 7.4.6, into NbrOfChangeBeacons beacons with the Change Type set to SIZE and the Superframe Timing set to the size of the superframe which follows the first beacon after this sequence.

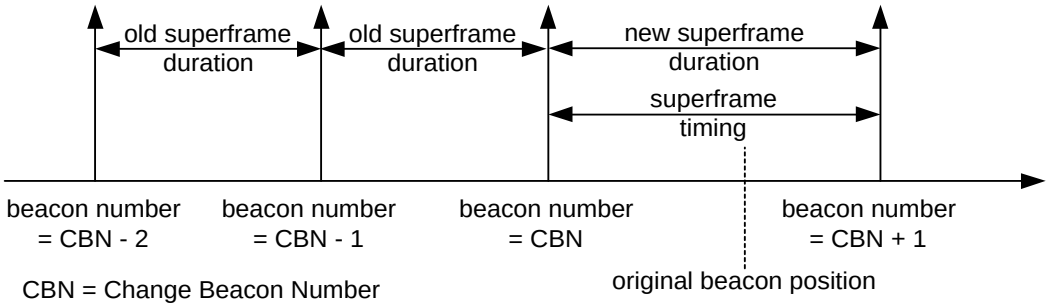

Figure 133—Changing superframe duration

8.10.3 Setting the PNID or BSID

The BSID is used to provide a way to identify the piconet. The PNC is able to change the BSID via the Piconet Parameter Change IE in the beacon, as described in 7.4.6, using the process described in this subclause. The BSID is preserved in the PNC handover process and it may be persistent when the PNC restarts a piconet that ended without handing over control to a PNC capable DEV.

The PNID is chosen by the PNC when it starts the piconet and shall only be changed if the PNC detects another piconet with the same PNID on any channel. The same PNID may be persistent when the PNC restarts a piconet that ended without handing over control to a PNC capable DEV.

If the PNC detects that another piconet is using the same BSID in its operational area, it may change the BSID, but it is not required to change it. If the PNC detects that another piconet is using the same PNID within range of the PNC on any channel or if it is informed of this via the Overlapping PNID IE, as described in 7.4.15, the PNC shall choose another PNID and change it via the Piconet Parameter Change IE in the beacon. The PNC shall not simultaneously change both the PNID and the BSID.

If a DEV detects a piconet within its range on any channel with the same PNID, it shall send an Announce command, as described in 7.5.5.2, to the PNC including an Overlapping PNID IE, as described in 7.4.15, that contains the current PNID and channel index. Once this command has been sent successfully, the DEV shall not send this information again until after the current PNID has been changed by the PNC.

Before changing its PNID, a parent PNC shall scan for the PNIDs of other piconets, including all of its dependent piconets. The PNC shall not change its PNID to the same value as that of any other piconet that it detects.

If the PNC decides to change the PNID or BSID, the PNC shall send a beacon with the Piconet Parameter Change IE indicating the new PNID or BSID. The DEVs that received the beacon with Piconet Parameter Change IE shall change the PNID or BSID to the new value at the time of the beacon with a beacon number equal to the Change Beacon Number field in the previous Piconet Parameter change IEs.

The MSC in Figure 134 describes the PNID and BSID change processes.

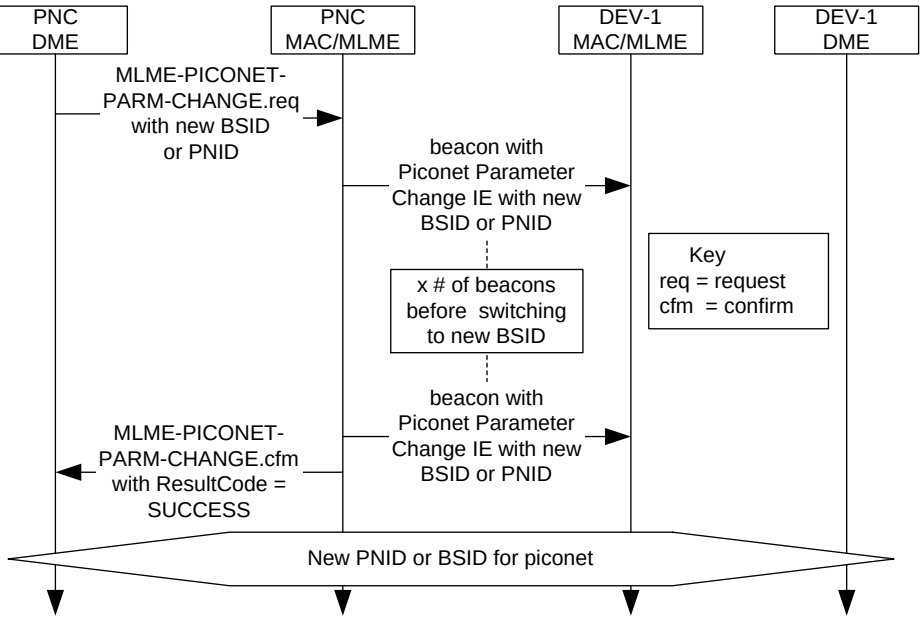

Figure 134—MSC for changing either the PNID or BSID for a piconet

8.10.4 Maintaining synchronization in dependent piconets

A dependent PNC receiving a parent beacon with a Piconet Parameter Change IE, that has the Change Type field set to MOVE or SIZE, shall immediately insert the appropriate Piconet Parameter Change IE into its beacons. If the Change Type field is set to SIZE, a dependent PNC shall change the length of the private CTA used to reserve time for the operations of the parent piconet in the first beacon it broadcasts in the parent superframe of the new superframe size.

8.11 Interference mitigation

The PNC should periodically listen in the current channel to detect interference, the presence of other 802.15.3 piconets or the presence of other wireless networks. The PNC should also periodically listen to other overlapping channels for the presence of the same types of DEVs.

If the PNC detects the presence of another 802.15.3 piconet, four of the methods that are available to mitigate the interference between the two piconets are:

a) The PNC may join the other piconet to form a child piconet, as described in 8.2.5.
b) The PNC may join the other piconet to form a neighbor network, as described in 8.2.6.
c) The PNC may change channels to one that is unoccupied, as described in 8.11.1.
d) The PNC may reduce the maximum transmit power in the piconet, as described in 8.11.2.

If the PNC determines that there is either an interferer or a non 802.15.3 wireless network operating in the PNC's current channel or overlapping with the current channel, two of the methods that the PNC are available to mitigate the interference are:

a) The PNC may change channels to one that is unoccupied, as described in 8.11.1.
b) The PNC may reduce the maximum transmit power in the piconet, as described in 8.11.2.

8.11.1 Dynamic channel selection

The PNC initiates dynamic channel selection if it determines that the current conditions of its channel are poor. The PNC may use one or more of these methods to make this determination.

1) The PNC may perform a PNC channel scanning procedure, defined in 8.9.6.
2) The PNC may use the remote scan procedure, defined in 8.9.5.
3) The PNC may collect the channel status from its member DEVs by sending a Channel Status Request command, defined in 7.5.7.1, to request that the DEVs provide their channel status via a Channel Status Response command, as described in 7.5.7.2.

The algorithm required to use the channel status information when deciding whether to change channels is outside the scope of this standard.

The PNC shall initiate a dynamic channel change procedure only after it has performed a PNC channel scan, as described in 8.9.6, and has determined that there is one or more channels with better characteristics than exist in its current operating channel. Note that in addition to the PNC channel scan, the PNC is able to use other methods, described above, to determine which channel to use as the new channel. If the PNC decides to initiate a dynamic channel change, the PNC shall broadcast the Piconet Parameter Change IE, as described in 7.4.6, with the change type set to CHANNEL in its current channel via its beacon for NbrOfChangeBeacons consecutive beacons. The Piconet Parameter Change IE shall contain the channel index of the new channel to which the PNC will be moving the piconet, and the Change Beacon Number field that contains the beacon number of the first beacon that will be sent on the new channel. The channel change shall take effect starting with the first beacon with a beacon number equal to the Change Beacon Number field in the previous Piconet Parameter Change IEs. The DEVs that received a beacon containing the Piconet Parameter Change IE shall change from their current channel to the new channel before the first expected beacon on the new channel.

All DEVs shall not transmit on the new channel until a beacon has been correctly received on the new channel.

Dependent piconets shall either change to the new channel with the parent PNC or cease operations. If a dependent PNC ceases operation due to a channel change, it may attempt to restart its piconet in the original channel, as described in 8.2.2.

The PNC is not requried to wait until all of the DEVs in power save modes are in ACTIVE mode before changing channels. A power save DEV is not required to switch to ACTIVE mode when there is a channel change in progress. However, the power save DEV should change channels at the time indicated in the Piconet Parameter Change IE.

The PNC shall ensure that the Piconet Parameter Change IE announcement complies with the rules for beacon announcements in 8.6.4.

Figure 135 illustrates the message sequence involved in transitioning the current piconet to a new channel.

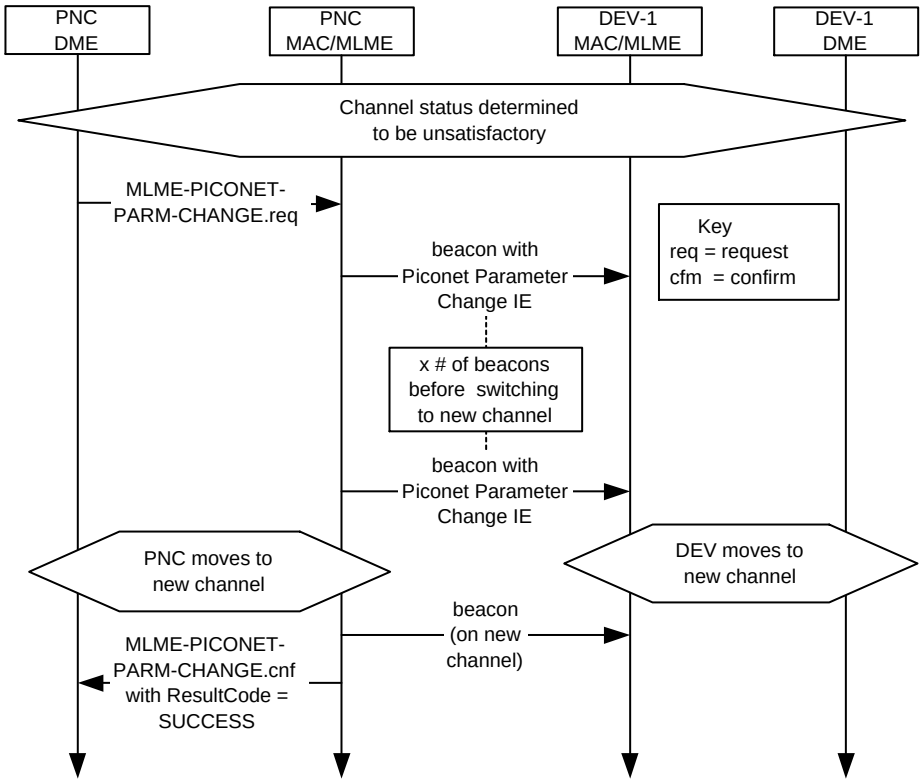

Figure 135—MSC for changing piconet parameters

When a parent PNC changes channels, the dependent PNC may change channels as well. If the dependent PNC is going to change channels with the parent PNC, it puts the appropriate Piconet Parameter Change IE into its beacons when it receives a beacon from the parent indicating a pending channel change. The dependent PNC shall then switch channels to the channel indicated in the Piconet Parameter Change IE at the appropriate time.

A dependent PNC may change to a different channel even if the parent PNC does not change channels. In this case, the child or 802.15.3 PNC shall remove the Parent Piconet IE from its beacon after the channel change since it is no longer a child or neighbor of that piconet. The dependent PNC should also disassociate from the parent piconet when it changes channels without the parent PNC.

8.11.2 Transmit power control

Two forms of transmit power control (TPC) are available for 802.15.3 systems, a maximum power for the CAP, the beacon and directed MCTAs, and adjustable power in a CTA. The goal of TPC in the CAP is to prevent one DEV from having better access to the medium in the CAP due to a higher transmit power level. Adjustable transmitter power in the CTA is intended to support reduced power usage as well as reducing the overall interference levels generated by the piconet.

8.11.2.1 Maximum transmitter power for the CAP, beacon and directed MCTAs

The PNC may choose a maximum transmit power level for the CAP, beacon and directed MCTAs, excluding association MCTAs. The PNC shall convey this information to the DEVs via the beacon frames using the Max TX Power Level field in the beacon. The PNC shall not set the Max TX Power Level below the pMinTPCLevel for the PHY, which is defined in 11.5.9 for the 2.4 GHz PHY. All DEVs within the piconet shall set their nominal transmit power level for frames in the CAP or directed MCTAs, excluding association MCTAs, to be no more than the value indicated in the Max TX Power Level field in the beacon, as described in 7.3.1. DEVs shall comply with the maximum transmit power within 10*mMaxLostBeacons superframes following the beacon in which the DEV detects the change. Likewise, the PNC shall set its nominal transmit power for the beacon to be no more than the value that is indicated in the beacon.

8.11.2.2 Adjustable transmitter power in a CTA

Each DEV participating in a CTA may request that the other DEV with which it is communicating in the CTA either increase or decrease its transmitter power level. The originating DEV shall use the Transmit Power Change command, as described in 7.5.7.5, to request a change in the power level setting of the target DEV for all CTAs assigned between the two DEVs. The target DEV shall increase or decrease its transmit power level as indicated in the Transmit Power Change command if the power level setting is supported by the target DEV. If the power level change is not supported by the target DEV, it shall use the closest implemented TX power level that is greater than the requested level provided that is within the allowable range. The target DEV shall apply the change in the power level for all CTAs assigned between the two DEVs.

A DEV may also change its transmit power based on its own estimation of the channel.

Figure 136 illustrates the message sequence involved in DEV-3 requesting a change in the TX power of DEV-2.

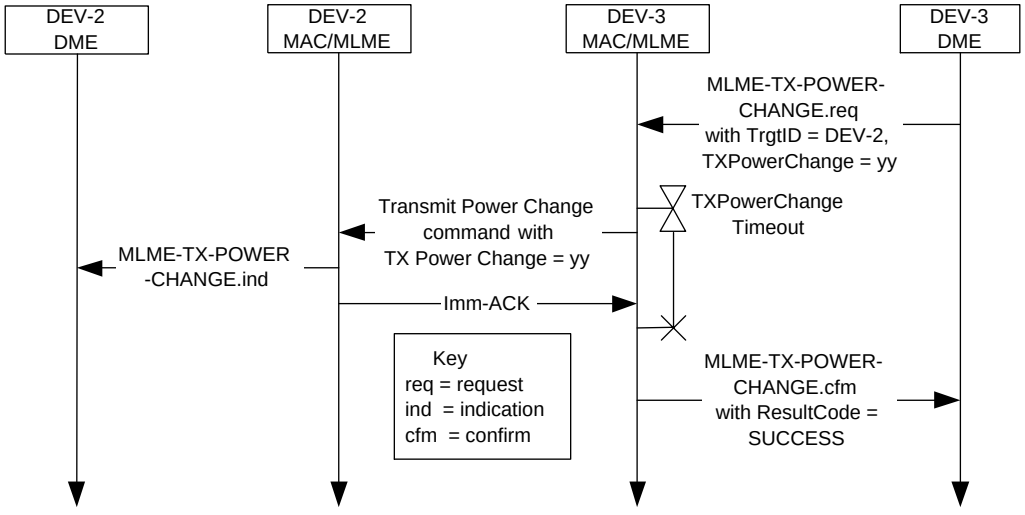

Figure 136—Transmit power change MSC

8.12 Multi-rate support

A compliant PHY may support more than one data rate. In each PHY there will be one mandatory base rate specified for the purposes described in this subclause. In addition to the base rate, the PHY may support rates that are both faster and slower than the base rate. A DEV shall send a frame with a particular data rate

to a destination DEV only when the destination DEV is known to support that rate. The allowed data rates and the mandatory base rate are PHY dependent. For the 2.4 GHz PHY, the supported rates are defined in 11.3. In order to determine the rates that are supported by a target DEV in the piconet, the DEV shall use one of three methods:

a) Check the capabilities of the target DEV broadcast by the PNC when the DEV becomes a member of the piconet;
b) Send a Probe Request command, as described in 7.5.4.5, to the target DEV to request its Capability IE, as described in 7.4.11;
c) Request the information from the PNC using the PNC Information Request command, as described in 7.5.4.1.

Similarly, each DEV may periodically use the Channel Status Request command, as described in 7.5.7.1, to obtain the channel status information from other DEVs and decide the PHY rate to be used in transmissions to those other DEVs. Additionally, the channel quality may be evaluated by the presence or absence of ACKs to transmitted frames. This information may be used to determine if the rate of transmssion or the power level needs to change.

If the Dly-ACK mechanism is used, all frames in a burst shall be sent with the same rate. The Dly-ACK frame shall be sent with the same rate as the rate of the last frame in the burst that requested the Dly-ACK.

The allowed PHY rates for each of the different types of frames are listed in Table 58.

Table 58—Allowed PHY data rates for frames

Frame type	Allowed PHY data rates
All broadcast and multicast addressed frames (including the beacon and commands)	Base rate
Imm-ACK	Same rate as the frame that is being ACKed
Dly-ACK	Same rate as the last frame of the burst being ACKed
Association request	Base rate
Association response	Base rate
Disassociation request	Base rate
Directed command frame	Any rate supported by both the source and destination.
Directed data frame	Any rate supported by both the source and destination.

8.13 Power management

There are four power management (PM) modes defined in this standard, ACTIVE, APS, PSPS, and DSPS modes. The latter three modes are collectively referred to as power save (PS) modes. A DEV that is in ACTIVE, APS, PSPS, or DSPS mode is said to be an ACTIVE DEV, an APS DEV, a PSPS DEV, or a DSPS DEV, respectively. In any given PM mode, a DEV may be in one of two power states, either AWAKE or SLEEP states. AWAKE state is defined as the state of the DEV where it is either transmitting or receiving. SLEEP state is defined as the state in which the DEV is neither transmitting nor receiving. A DEV, regardless of its PM mode, is allowed to enter the SLEEP state during a CTA for which it is neither the source nor the destination. A DEV is also allowed to enter the AWAKE state during any time when it is in a power save mode.

The wake beacon for a DEV is defined as the PNC-defined system wake beacon for DEVs in PSPS mode, as described in 8.13.1, and the wake beacon of the DSPS set for a DEV in DSPS mode, as described in 8.13.2. A DEV in DSPS mode may be in multiple DSPS sets and therefore may have multiple wake beacons because each of those DSPS sets may have its own wake beacon. The wake beacon for a DEV in APS mode occurs at times determined by the DEV and is unknown to the PNC and other DEVs in the piconet. Unlike the DSPS and PSPS wake beacons, the wake beacon of the DEV in APS mode is not periodic and is only guaranteed to happen once per ATP period for that DEV.

A DEV shall always establish membership with the piconet in ACTIVE mode. If the DEV MLME changes its PM mode to ACTIVE without the prompting of the DME, it notifies the DME with the MLME-PM-MODE-ACTIVE.indication primitive as described in 6.3.22.7.

Table 59 lists the rules for the four modes of operation defined in this standard. Each entry indicates the state required, either AWAKE or SLEEP, for the DEV.

Table 59—Power management rules for superframe elements

Superframe portion	ACTIVE	APS in wake superframe	PSPS or DSPS DEV in wake superframe
Beacon	AWAKE	AWAKE	AWAKE
CAP	AWAKE	May SLEEP	AWAKE
CTA with BcstID as DestID (including MCTAs)	AWAKE	May SLEEP	AWAKE
CTA with McstID as DestID (including MCTAs)	May SLEEP	May SLEEP	May SLEEP
CTA with DEV as SrcID or DestID (including MCTAs)	AWAKE	May SLEEP	AWAKE
All other CTAs and unallocated time (between CTAs)	May SLEEP	May SLEEP	May SLEEP

The PNC shall support one PS set for APS and one PS set for PSPS. In addition the PNC shall support at least one DSPS set, i.e. a PS set with PS Set Index between 2 and 253.

8.13.1 Piconet synchronized power save (PSPS) mode

A DEV in PSPS mode shall listen to all system wake beacons, as announced by PNC and is required to be in the AWAKE state during system wake superframes as indicated in Table 59. The wake beacon for PSPS DEVs is determined by the PNC. PSPS mode is identified by a PS Set Index equal to one. If a DEV in PSPS mode does not correctly receive the system wake beacon, it shall be in the AWAKE state during the expected beacon tranmsision times to receive the following beacons until a beacon is correctly received.

The PNC shall announce the system wake beacon in the Next Wake Beacon field in the PS Status IE with PS Set Index field equal to one in the beacon. If none of the DEVs are in PSPS mode, the PNC may omit the PS Status IE for PS Set Index one from the beacon. In that case every beacon is a system wake beacon for the purpose of beacon information announcements, as described in 8.6.4.

It is the responsibility of the DEV using PSPS mode to synchronize with the system wake beacon before entering the sleep state. Because the PSPS DEV at some point will need to send commands to the PNC, e.g. the PM Mode Change command, the PNC needs to take this into consideration when allocating MCTAs if the CAP is not available for sending commands.

Any DEV that is going to use the PSPS mode shall send the SPS Configuration Request command, as described in 7.5.8.3, to the PNC with the Operation Type field set to ‘join’, SPS Set Index set to one and the Wake Beacon Interval set to its desired system wake beacon interval. The valid range for a requested Wake Beacon Interval is defined in 7.5.8.3. Upon reception of this command, the PNC shall ACK the command and respond with an SPS Configuration Response command with the next wake beacon. The PNC uses the information in the Wake Beacon Interval field from all particpating PSPS DEVs to determine the system wake beacon interval. The actual system wake beacon interval may not correspond to any of the PSPS DEVs requested wake beacon interval.

A DEV may send the SPS Configuration Request command more than once if its system wake beacon interval requirement changes. If the DEV no longer wishes to use PSPS mode, it shall send the SPS Configuration Request command with the Operation Type field set to ‘leave’ and the PS Set Index set to one to leave the PSPS set.

A DEV shall send a PM Mode Change command to the PNC with the PM Mode field set to SPS and receive the ACK from the PNC before entering the PSPS mode. When the PNC receives this command, it shall terminate all super-rate streams for which the DEV is the destination, as described in 8.5.1.3, and set the DEVID Bitmap field in the PS Status IE appropriately, as described in 7.4.13.

The PS Status IE in the beacon for PS Set Index value of one with the bit for the DEV’s DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched into PSPS mode.

When the DEV is going to switch to ACTIVE mode, it shall send a PM Mode Change command to the PNC with the PM Mode field set to ACTIVE. Once this command is sent, the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC. The PNC is not required to align sub-rate allocations for a PSPS DEV with the system wake beacon.

8.13.2 Device synchronized power save (DSPS) mode

DSPS mode allows a DEV that is sensitive to power utilization to synchronize its AWAKE state with other DEVs. The DSPS mode is based on grouping DEVs that have similar power save requirements into DSPS sets. These DSPS sets are managed by the PNC, but the parameters of the sets are determined by the DEVs.

8.13.2.1 Creation, use and management of DSPS sets

In order to use DSPS mode, a DEV is required to first join a DSPS set. Each DSPS set has two associated parameters: the Wake Beacon Interval and the Next Wake Beacon. The DSPS set is identified by an index value called the DSPS Set Index that is between 2 and 253, inclusive. The Wake Beacon Interval is the number of superframes between two successive wake beacons of that DSPS set. This value is set by the DEV when it creates the DSPS set. The Next Wake Beacon parameter is the beacon number, as described in 7.3.1.1, corresponding to the immediate next wake beacon of that DSPS set. This parameter is set by PNC when it creates the DSPS set. Both of these parameters shall be maintained by the PNC.

Any DEV that is a member of the piconet may request the information about the existing DSPS sets by sending a PS Set Information Request command, as described in 7.5.8.1, to the PNC. The PNC shall respond by sending a PS Set Information Response command, as described in 7.5.8.2, that provides the parameters of all of the PS sets currently in use within the piconet.

The DEV may select a DSPS set to join. If there are not any DSPS sets currently in existence that match the DEV's requirements, the DEV may request the formation of a new DSPS set by setting the SPS Set Index field in the SPS Configuration Request command, as described in 7.5.8.3 to the 'Unallocated DSPS set' value and the Operation Type field set to 'join'. The DEV shall set the Wake Beacon Interval field to its requested value. The valid range for the Wake Beacons Interval field is defined in 7.5.8.3. This value shall not be changed while the DSPS set has any members. The PNC shall respond to the request by sending the SPS Configuration Response command, as described in 7.5.8.4, to the DEV indicating success or the reason that the request failed. If the DSPS set is created, the PNC assigns a DSPS Set Index to it. The PNC shall assign a unique number between 2 and 253 for the DSPS set index. The PNC includes in the SPS Configuration Response command a value for the Next Wake Beacon field set to the beacon number, as described in 7.3.1.1, of the first wake beacon for the new DSPS set. Once a DSPS set is created, the PNC shall keep the Next Wake Beacon for that set updated at all times. The maximum number of DSPS sets supported by the PNC is implementation dependent up to a maximum of 252.

The PNC may require that all PS sets have a unique Wake Beacon Interval. For example, the PNC may reject a request to create a PS set with a Wake Beacon Interval of 4 if there is a PS set that already has this value. If the DEV requires this Wake Beacon Interval, it may join the existing PS set.

A DEV may join an existing DSPS set by sending the SPS Configuration Request command to the PNC with the SPS Set Index field set to the index of an existing DSPS set and the Operation Type field set to 'join'. All other parameters shall be ignored. The PNC shall confirm or reject the request by sending the SPS Configuration Response command to the DEV. Since a DEV may support multiple applications with different requirements, a DEV may register in more than one DSPS set at a given time.

A DEV that no longer needs to be in a DSPS set shall send the SPS Configuration Request command to the PNC with Operation Type field set to 'leave.' The PNC shall not send the SPS Configuration Response command to the requesting DEV if the Operation Type field was set to 'leave'.

When the last member of a DSPS set has left, the DSPS set shall be terminated by the PNC.

8.13.2.2 Changing DSPS mode and operation

DSPS DEVs alternate between DSPS mode and ACTIVE mode depending on the amount and type of data traffic without leaving any of the DSPS sets that the DEV has joined. The PM Mode Change command, as described in 7.5.8.5, is used by a DEV to inform the PNC that it is changing its power management mode. A DEV shall have joined one or more DSPS sets before it will be allowed to switch into DSPS mode.

If the DEV is going to change its power management mode from ACTIVE to DSPS, the DEV shall send the PM Mode Change command, as described in 7.5.8.5, to the PNC with the PM Mode field set to SPS. The PNC shall then set the bit in the DEVID Bitmap field for the DEV in each PS Status IE that corresponds to an SPS set of which the DEV is a member. If the DEV is the source or destination of any streams, not including broadcast or multicast, which are not using a DSPS wake beacon interval, the PNC shall terminate those streams, as described in 8.5.1.3, when the DEV changes to DSPS mode. The PNC does not automatically terminate any streams when the DEV changes from DSPS to ACTIVE mode.

If the DEV has joined PS set one (PSPS) in addition to other DSPS sets before issuing the PM Mode Change command, the DEV shall transition into a combined PSPS and DSPS mode. In either case, the DEV shall not consider itself in either DSPS or PSPS mode until it has received an Imm-ACK to the PM Mode Change command.

The presence of a PS Status IE in the beacon with the bit for the DEV's DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched into PSPS mode.

If the DEV is going to change its power save mode from DSPS to ACTIVE, the DEV shall send the PM Mode Change command, as described in 7.5.8.5, to the PNC with the PM Mode field set to ACTIVE. Once this command is sent the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap field in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC. When the PNC correctly receives the PM Mode Change command with the PM Mode field set to ACTIVE, it shall no longer set the bit(s) for the DEV in the PS Status IE(s). If the PM Mode field indicates the current mode of the DEV, then no change is requested.

The PNC shall create one PS Status IE in the beacon for each DSPS set that has at least one member currently in DSPS mode. When a DSPS set is not in use, the PNC shall discontinue inserting its PS Status IE in the beacon. The PNC needs to ensure that the number and size of the PS Status IEs do not cause the beacon or extended beacon to exceed its maximum allowed size, as described in 7.2.

Other DEVs may use the information in the PS Status IE to learn when to transmit to a DSPS DEV. In addition, the PS Status IE informs DSPS capable DEVs which DSPS set is required in order to synchronize data transfers to the DEVs in DSPS mode.

It is possible that DEVs in DSPS mode will not receive broadcast messages. If a DEV requires the opportunity to receive all of the broadcast streams, it should not use DSPS mode. If a source DEV needs that the DEVs in DSPS mode have an opportunity to receive the broadcast frame, then the source DEV should send the frame in all of the superframes required to reach the DSPS DEVs in their wake beacons.

The PNC may grant an ACTIVE DEV’s channel time request to create or modify a stream with PM CTRq Type field set to ACTIVE and a DSPS mode DEV as the TargetID using the processes defined in 8.5.1.1 and 8.5.1.2.

If the CTA Rate Type field of a new allocation is set to sub-rate with the DSPS DEV as the destination but not aligned with one of the DEV’s DSPS sets, the DSPS DEV may either:

— Stay in DSPS mode while listening to the additional beacons required for the new allocation
— Switch to ACTIVE mode using the PM Mode Change command.
— Terminate the stream as described in 8.5.1.3.

8.13.2.3 CTA timing in DSPS mode

A DSPS CTRq is a Channel Time Request command, as described in 7.5.6.1, with the PM CTRq Type field set to DSPS. This requests that the PNC allocate channel time during the wake superframes of the specified DSPS set. An additional condition placed on the timing is that the value of the CTA Rate Factor field shall not be less than the number of superframes between wake beacons, i.e. the Wake Beacon Interval. Since the CTA Rate Factor field, like the Wake Beacon Interval, is a power of 2, the rate of DSPS CTAs also is a power of 2 sub-rate of the wake beacon rate, as illustrated in Figure 137. For example, in case 3 of Figure 137 the Wake Beacon Interval is $2=2^1$ while the CTA Rate Factor is $8=2^3$. Thus, the CTAs occur every fourth wake beacon, 4 = 8/2. An example of values that are not allowed would be a Wake Beacon Interval of 4 and a CTA Rate Factor of 2. The reason that this is not allowed is that the CTAs would occur more often than the DEV was waking to listen to the beacon.

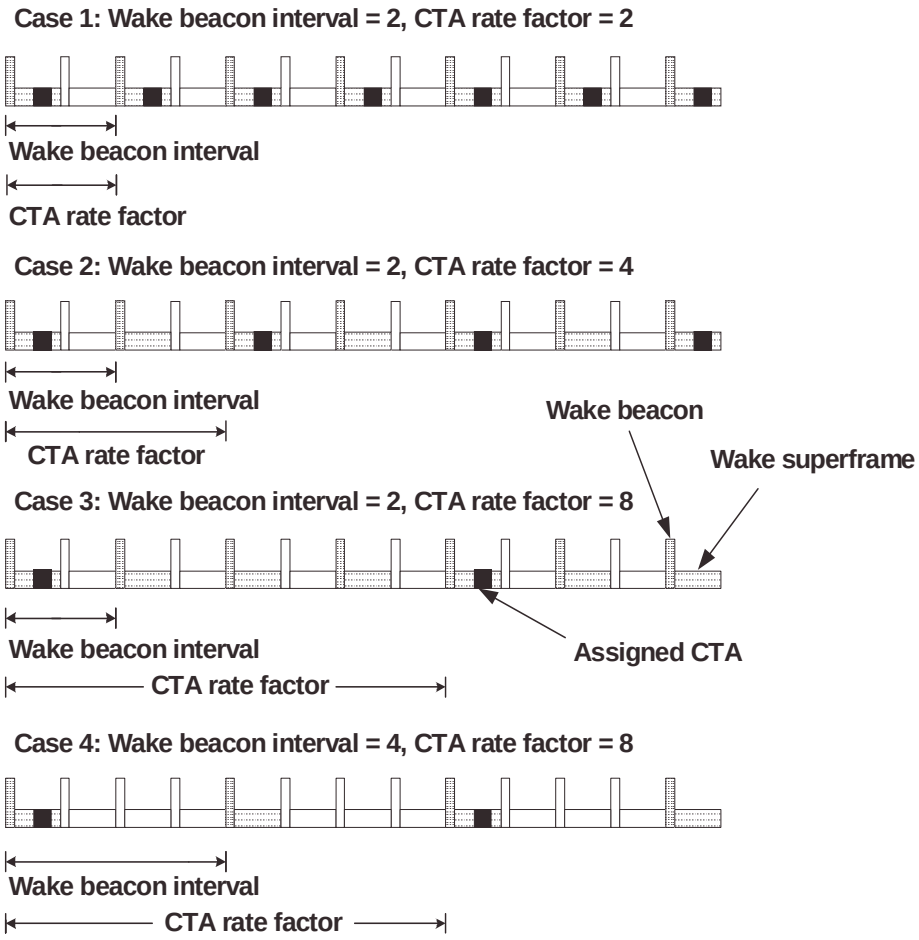

Figure 137—Illustration of DSPS set and sub-rate CTAs

If the PNC grants a DSPS CTRq, the PNC shall allocate CTAs only in the wake beacons of that DSPS set unless there is insufficient channel time for the allocation. If the PNC determines that it is unable to provide the requested CTA in a wake beacon, the PNC shall not allocate the CTA and shall take one of three actions:

1) Continue attempting to allocate the CTA at the appropriate times.
2) Terminate the stream, as described in 8.5.1.3.
3) Set the appropriate bits in the CWB IE, as described in 7.4.8, for the SrcID and DestID to indicate that those DEVs should wake up for the next beacon to see if there will be a CTA in the next beacon. The PNC may set the CWB bits in up to three consecutive beacons until it is able to allocate the CTA. Although this behavior is permitted, an undesirable reduction in battery life will result from unnecessary wake periods for DSPS DEVs.

A DSPS DEV shall listen to every wake beacon regardless of the frequency with which CTAs are allocated. If the DSPS DEV is the DestID of any CTA in the wake beacon, then the DSPS DEV shall listen during the assigned CTA in that wake superframe.

Figure 138 shows four equivalent PNC CTA arrangements for a Wake Beacon Interval of 2 and the CTA Rate Factor of 8. The PNC chooses the wake superframes to position the assigned CTAs at an overall rate of 1 CTA per 8 superframes.

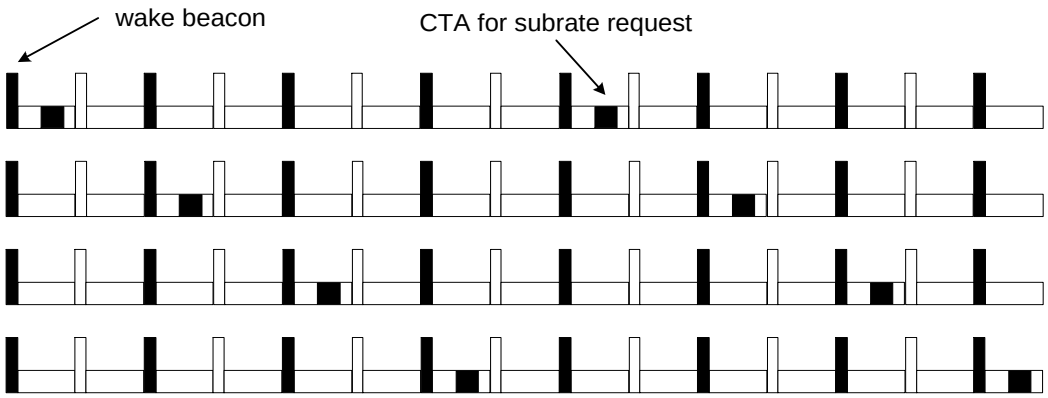

Figure 138—Equivalent CTAs for wake beacon interval = 2 and CTA rate factor = 8

The example in Figure 139 shows that Wake Beacon Interval and CTA Rate Factor together provide the ability of the DEV to tradeoff power savings and superframe loading, i.e. the number of CTAs allocated to any one superframe. In this example, the PNC minimizes the superframe loading for the eight DSPS DEVs, if they are members of different DSPS sets, because there is one wake beacon every other superframe, rather than the minimum, one wake beacon every eight superframes if power saving were maximized.

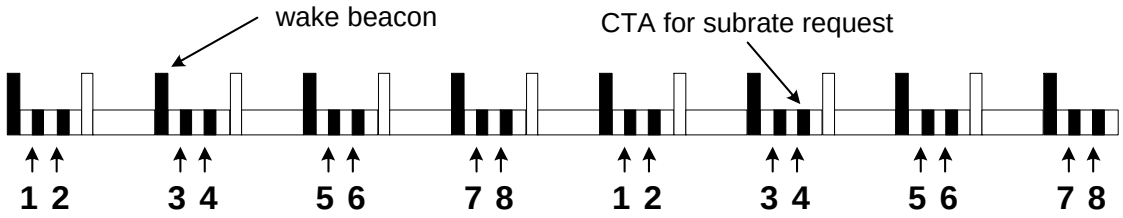

Figure 139—Minimum superframe loading of CTAs for 8 DEVs requesting wake beacon interval = 2 and CTA rate factor = 8

8.13.3 Asynchronous power save (APS) mode

APS mode allows a DEV to conserve power by remaining in SLEEP state for extended periods of time. The only responsibility of a DEV in APS mode is to communicate with the PNC before the end of its ATP in order to preserve its membership in the piconet

In the APS mode the DEV is not required to listen to any beacons or other traffic until it changes to either ACTIVE or a different power save mode using the PM Mode Change command, as described in 7.5.8.5. APS mode shall not be used in combination with any other power save mode. A DEV shall not use the SPS Configuration Request command to set parameters for the APS Set Index.

All DEVs in APS mode need to send at least one acknowledged frame to the PNC during their ATP in order to avoid being disassociated from the piconet, as described in 8.3.4. Because the APS DEV will need to send a frame to the PNC at least once during its ATP, the PNC needs to take this into consideration when allocating MCTAs if the CAP is not available for sending commands.

A DEV shall send a PM Mode Change command to the PNC with the PM Mode field set to APS and receive the ACK before entering APS mode. When the PNC receives this command, it shall set the DEVID Bitmap

field in the PS Status IE appropriately, as described in 7.4.14. The PNC shall terminate all streams and asynchronous data allocations that have the APS DEV as either the source or destination ID.

The PS Status IE in the beacon with the bit in the DEVID Bitmap field for the DEV's DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched to APS mode. The PS Set Index of 0 shall only be used for APS DEVs. Although a PS Set Index is assigned to the DEVs in APS mode, the DEVs in this mode all act independently, unlike the DEVs that are members of other PS sets.

The DEV may leave APS mode by sending a PM Mode Change command to the PNC with the PM Mode field set to ACTIVE. Once this command is sent the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC.

The PNC may grant an ACTIVE mode DEV's channel time request, with PM CTRq Type field set to ACTIVE and with an APS mode DEV as the DestID using the processes defined in 8.5.1.1 for isochronous allocations and in 8.5.2.1 for asynchronous allocations.

8.13.4 Message sequence charts for power save modes

Figure 140 illustrates the message flow for a DEV inquiring about the PS sets that are currently defined within the PNC.

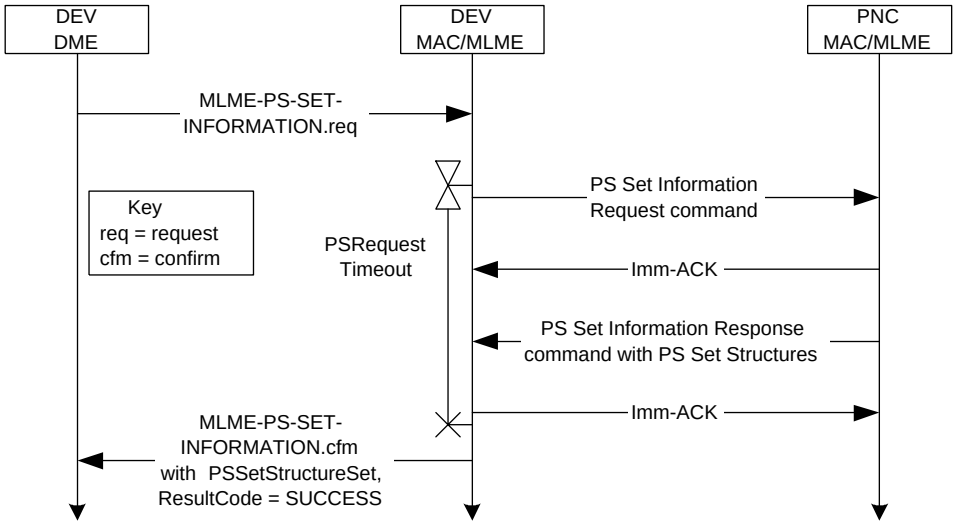

Figure 140—MSC for PS set information exchange

Figure 141 illustrates the message flow for a DEV requesting the creation of a new DSPS set.

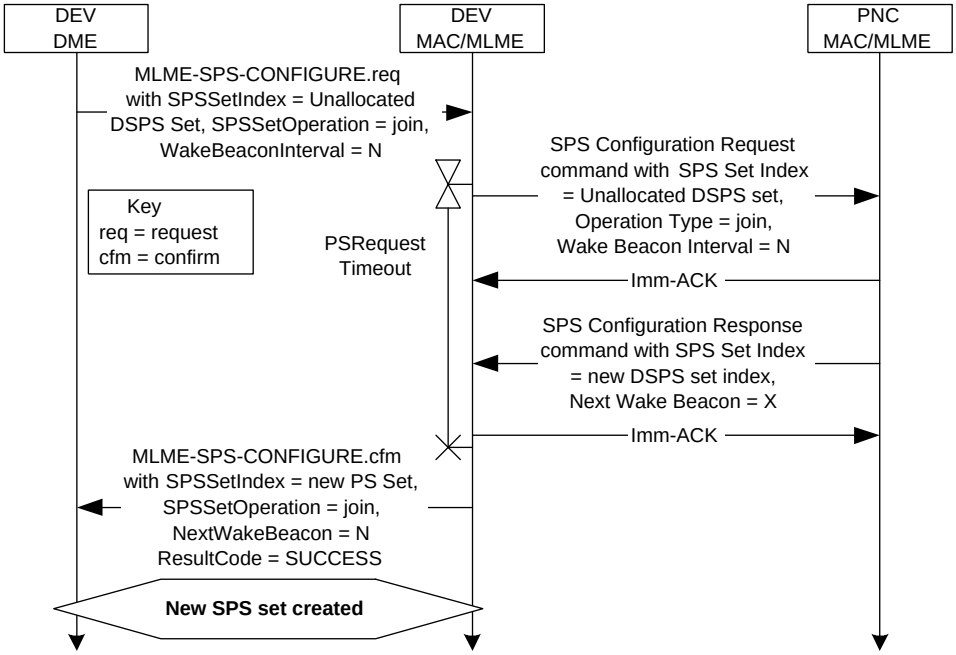

Figure 141—MSC for DSPS set creation

Figure 142 illustrates the message flow for a DEV requesting to add itself to an existing SPS set.

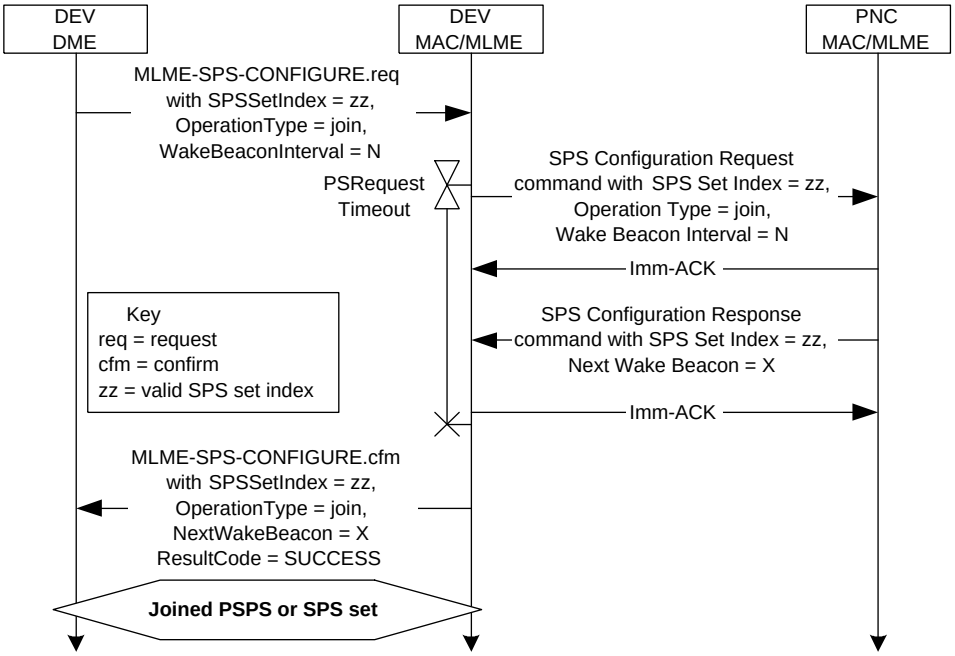

Figure 142—MSC showing a DEV joining an existing SPS set

Figure 143 illustrates the message flow for a DEV requesting to leave an SPS set.

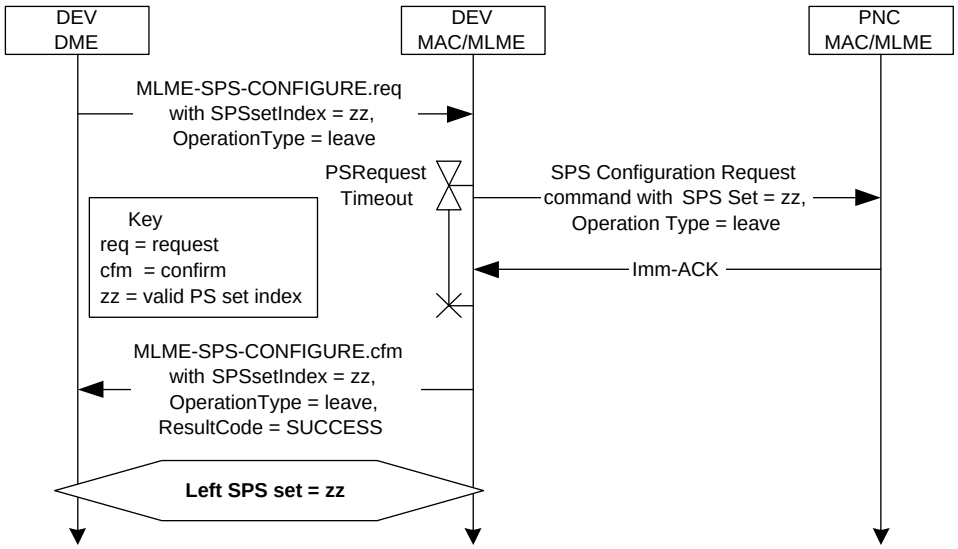

Figure 143—MSC showing a DEV leaving an SPS set

Figure 144 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from ACTIVE to an SPS mode when that DEV is a member of one or more SPS sets.

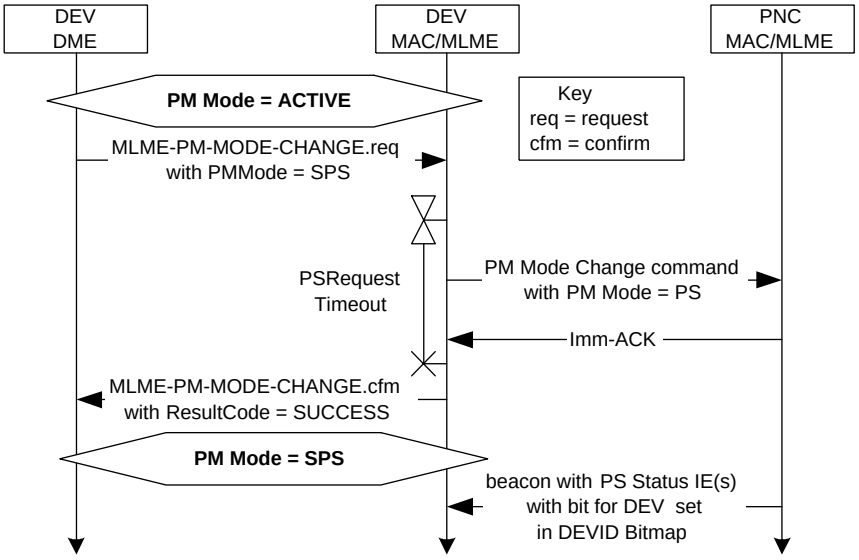

Figure 144—MSC showing DME initiated PM mode change from ACTIVE to an SPS mode

Figure 145 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from ACTIVE to APS mode.

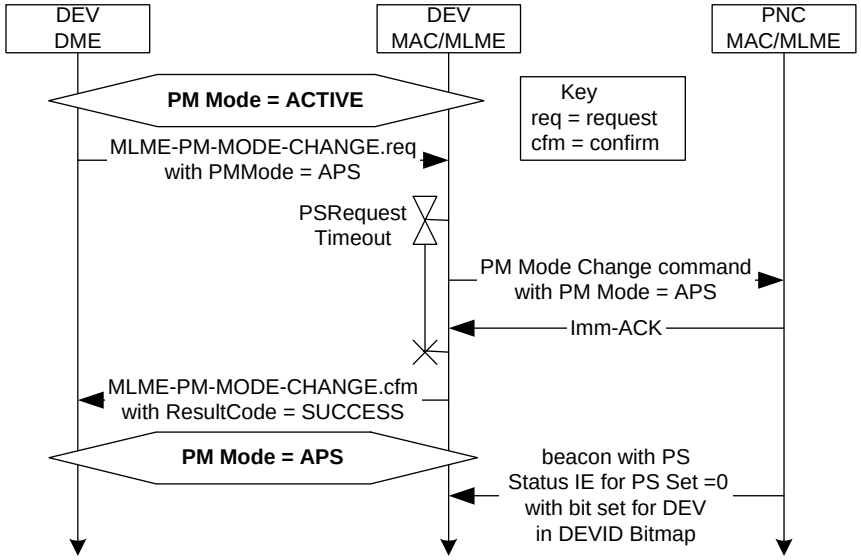

Figure 145—MSC showing DME initiated PM mode change from ACTIVE to APS

Figure 146 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from one of the power save modes to ACTIVE mode.

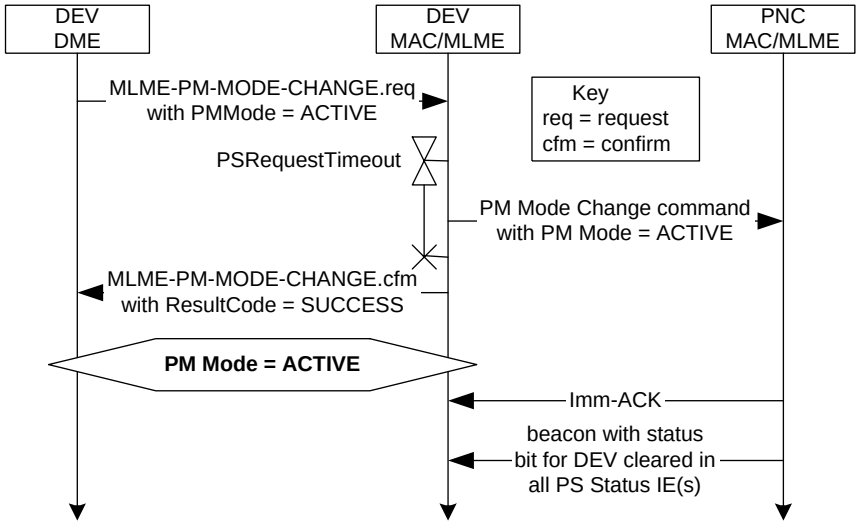

Figure 146—MSC showing DME initiated PM mode change from any PS mode to ACTIVE

Figure 147 illustrates the message flow for a DEV MLME changing the PM mode of operation from DSPS to ACTIVE due to an ACTIVE mode channel time allocation.

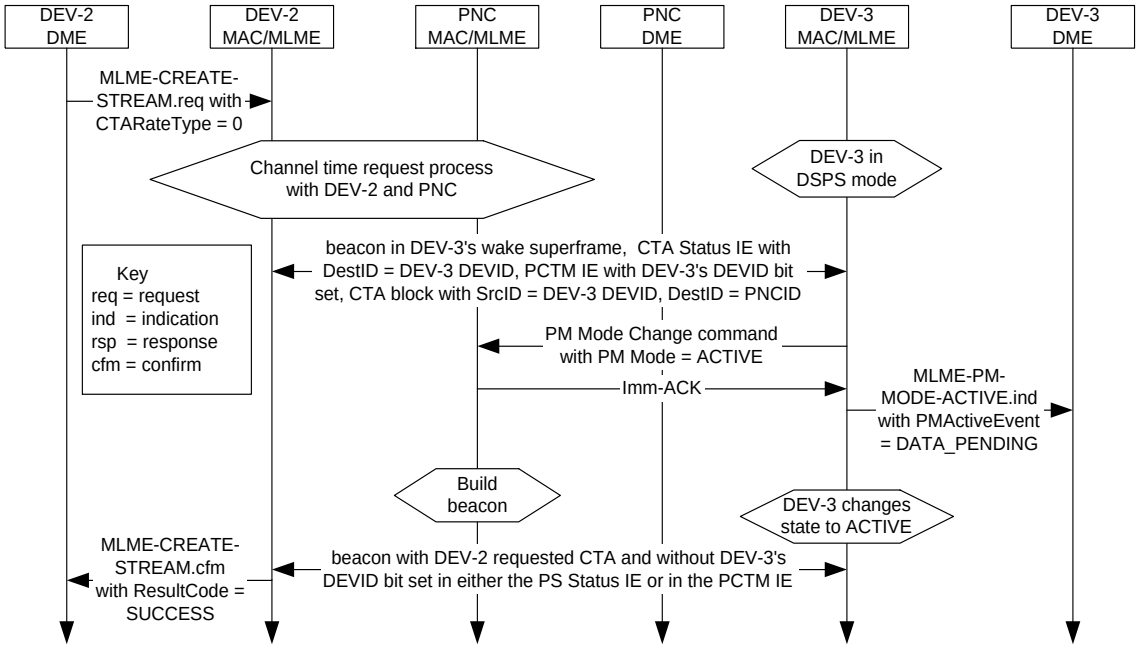

Figure 147—Message sequence showing MLME initiated PM mode change from DSPS to ACTIVE in response to a new chanel time allocation

8.14 ASIE operation

The ASIE is used to implement out of scope features that require additional functionality by both the PNC and one or more of its piconet member DEVs. The “additional functionality” is defined as an enhancement that does not violate the standard and allows DEVs that do not have the functionality to operate normally. The Application Specific Data field in this IE provides the messages that are only interpreted by the targeted DEV.

The ASIE communicates to potential DEVs that the PNC is providing its part of the additional functionality, and it is also used to control that functionality. Multiple ASIEs may be added by the PNC unless limited by standard services that are required by the PNC or by the size of the beacon. The designer should minimize the size of each ASIE used to support the custom application.

The PNC DME uses MLME-PNC-CREATE-ASIE.request to tell the PNC MLME to place the specified ASIE in one or more beacons. If the DME sets the Cmd parameter to ‘NEW,’ the PNC shall either create a new ASIE or modify an existing one. The MAC/MLME uses the ASIEIndex to identify the ASIE to create or modify.

The MLME-PNC-CREATE-ASIE.confirm is used by the PNC MLME to inform the PNC DME that the ASIE requested will be generated.

The MLME-RECEIVE-ASIE.indicate is used by the MLME of the DEV addressed in the ASIE, to pass the ASIE data up to the DME. The MLME shall pass the data to the DME only once for each ASIE that it receives. Since each ASIE may persist in multiple beacons, this persistence shall be detected in the DEV MLME.

8.15 MAC sublayer parameters

The parameters that define some of the MAC characteristics are given in Table 60.

Table 60—MAC sublayer parameters

Parameter	Value
mMinChannelScan	mMaxSuperframeDuration
mBroadcastDEVInfoDuration	64*mMaxSuperframeDuration
mAssocRespConfirmTime	4*mMaxSuperframeDuration
mMinSuperframeDuration	1 ms
mMaxSuperframeDuration	65,535 μs
mMaxLostBeacons	4
mMCTAAssocPeriod	150 ms
mFirstCTAGap	100 μs
mMinBeaconInfoRepeat	4
mAsyncRequestLifetime	1 s
mMaxKeyChangeDuration	65,535 ms
mMaxTimeTokenChange	65,535
mMaxNumValidDEVs	243

Additional characteristics that are PHY dependent are indicated in Table 61 for the 2.4 GHz PHY.

Table 61—MAC sublayer parameters - 2.4 GHz PHY dependent

Parameter	Subclause
SIFS	11.2.7.1
MIFS	11.2.7.1
BIFS	11.2.7.1
RIFS	11.2.7.1
pBackoffSlot	11.2.7.1
pMinTPCLevel	11.5.9
pMaxTransferUnitSize	11.2.8.2
pMaxFrameBodySize	11.2.8.1
pPHYClockAccuracy	11.5.5
pLengthHCS	11.2.9
pMinFragmentSize	11.2.8.3

9. Security

Wireless networks face unique security challenges and piconets are no exception. Recognizing the diversity of piconet applications and entities, this standard supports two different modes of security, no security and the use of strong cryptography. The standard supports the protection of command, beacon and data frames using an 128-bit AES security suite, and the distribution of keys for command and data frame protection.

The background assumptions used in designing this security solution are outlined in Annex B.1.

9.1 Security mechanisms

Security mechanisms provided by this standard allow security services to be implemented to control the admission of DEVs into a security relationship between the PNC and a DEV or between two ordinary DEVs and protect the information and integrity of communications between DEVs in a security relationship. This standard also provides a symmetric cryptography mechanism to assist in providing security services. Additional security services need to be provided by the higher layers to ensure proper management and establishment of the symmetric keys used in this standard.

9.1.1 Security membership and key establishment

The method by which a DEV becomes a member of a security relationship and obtains the appropriate key is outside of the scope of this standard. The Security Message command has been included as a special command to assist in the implementation of vendor specific protocols for establishing security relationships and any related data. It can be achieved with higher layer protocols that are not specified in this standard. The MAC/MLME is informed of changes to the membership of a security relationship and the key for that relationship with the MLME-MEMBERSHIP-UPDATE primitive, as described in 6.3.9.1.

9.1.2 Key transport

All keys that are transmitted from one DEV to another shall be encrypted as specified in the key request, as described in 9.4.3, and distribute key protocols, as described in 9.4.2. For example, key transport is used to provide a copy of the piconet group data key to a DEV.

9.1.3 Data encryption

Data encryption uses a symmetric cipher to protect data from being read by parties without the cryptographic key. Data may be encrypted either by using a key shared by all piconet DEVs or by using a key shared between only two DEVs.

9.1.4 Data integrity

Data integrity uses an integrity code to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. Integrity may be provided using a key shared by all piconet DEVs or using a key shared between only two DEVs. All secure data frames that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indicate and no other action is taken on the frame by the MLME.

9.1.5 Beacon integrity protection

The beacon may be integrity-protected. This integrity protection provides evidence to all the DEVs in the piconet that the PNC of the secure piconet transmitted the beacon. Under normal operations, the integrity check on the beacon provides evidence that the piconet is operating properly and that no security changes

have occurred. If the integrity check on the beacon fails, the DEV is alerted to the fact that the DEV does not have its security state synchronized with the PNC.

9.1.6 Command integrity protection

The integrity of commands may be protected just like any other data. Integrity protected commands sent between the PNC and a DEV shall be protected using the PNC-DEV management key. All secure commands that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indicate and no other action is taken on the frame by the MLME.

9.1.7 Freshness protection

To prevent replay of old messages, a strictly-increasing time token is included in the beacon. A DEV may reject as invalid a received beacon with a time token less than or equal to the current time token. In addition, the time token is included in the CCM nonce, as described in 10.2.4, for each secure frame, as described in 7.2, so the integrity check will fail if a frame is replayed in a different superframe. A DEV in a secure piconet maintains two values for freshness. The CurrentTimeToken is the time token value found in the beacon for the current superframe and is used to protect all messages sent and check all messages received during that superframe. The LastValidTimeToken is used by the DEV to ensure that the security of the beacons have not been compromised.

9.2 Security modes

The security mode indicates whether a DEV is currently implementing frame protection in the piconet. The security mode in use is determined by the MACPIB_SecurityOptionImplemented entry in the MAC PIB.

9.2.1 Security mode 0

A DEV operating in security mode 0 shall not perform any cryptographic operations on MAC frames. While in this mode, if the MAC receives a frame with the SEC field set to one, the MAC shall discard the frame and the MLME shall return an MLME-SECURITY-ERROR.indication to the higher layer with the ReasonCode set to UNAVAILABLE-KEY.

9.2.2 Security mode 1

Security mode 1 provides a mechanism for a device to perform cryptographic security on frames transmitted in the piconet. DEVs operating in security mode 1 use symmetric-key cryptography to protect frames using encryption and integrity.

While in mode 1, the cryptographic operations used for secure frames exchanged with the PNC and with other members of the piconet security group shall be performed as specified by the symmetric key security operations. While in this mode, if the MAC receives a frame with the SEC field in the Frame Control field set to a value different than expected as defined in Table 50, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

9.3 Security support

The security policies determine the actions taken to preserve the security of the piconet. Subclauses 9.3.1 through 9.3.8 specify the methods that are provided in this standard to support specific security policies.

9.3.1 PNC handover

When a PNC chooses to handover the PNC role to another DEV in the piconet, the security relationships with the old PNC no longer apply to the new PNC. When the old PNC hands over the piconet information using a PNC Information command, as described in 7.5.4.2, the list of associated DEVs is passed to the new PNC.

PNC handover does not affect the group membership, so it does not require a rekey of the group key. However, in a piconet with payload protection, the command functions of the PNC that relate to specific DEVs are not supported until the new PNC has established secure membership with each DEV in the piconet. When the PNC role has been handed over, the new PNC should create CTAs for each of the associated DEVs to establish secure membership with the new PNC.

The old PNC may send security information about the new PNC to the other DEVs in the piconet and send security information about all of the DEVs that are secure members in the piconet to the new PNC when it hands over the role of the PNC. This is accomplished by sending a directed Security Information command, as described in 7.5.4.4, to the new PNC with the security information of the piconet in it and by sending a broadcast Security Information command or a directed Security Information command to each member of the piconet with the security information of the new PNC.

9.3.2 Changes in the piconet group data key

When the PNC changes the piconet group data key, the PNC shall transmit the new key to all of the members of the piconet that are in ACTIVE mode using the Distribute Key Request command, as described in 7.5.2.3. Once the Distribute Key Request command has been issued for all of the members of the piconet that are in ACTIVE mode, the PNC may change the SECID in the beacon. When a DEV receives a valid Distribute Key Request command, as described in 7.5.2.3, from the PNC, the DEV shall use the new key for all outgoing secure frames that require the use of the piconet group data key once it sees the corresponding SECID in the beacon. The DEV may continue to accept frames protected by the old piconet group data key for up to mMaxKeyChangeDuration since the DEV last received a valid beacon protected by the old piconet-wide group data key.

If a DEV receives a beacon with a time token greater than the last known time token, but with a SECID that does not match the SECID of the known key, the DEV shall send a Key Request command to the PNC to obtain the new key.

9.3.3 Joining a secure piconet

If a DEV wishes to join a secure piconet, it should associate with the PNC in order to be assigned a local DEVID. Once the DEV is associated, the PNC shall allocate an MCTA if commands are not allowed in the CAP. The DEV or PNC may choose to send Probe Request and/or Announce commands to each other to either request or transmit IEs, including Vendor Specific IEs. The DEV and PNC may also exchange additional data frames or Security Message comamnds. After the DEV has associated and exchanged the desired information with the PNC, the DEV shall establish secure membership. The process by which secure membership is established is outside of the scope of this standard.

9.3.4 Membership update

When the DME determines that there has been a change of membership status with a particular DEV or when a management or data key is changed, the DME shall issue an MLME-MEMBERSHIP-UPDATE.request to its MLME. This membership status change or key change may be the result of a successful establishment of a security relationship, key update process, termination of a security relationship or some other event. The process by which this change occurs is outside the scope of this standard.

When the MLME receives the MLME-MEMBERSHIP-UPDATE.request, it shall first examine the TrgtID to determine the membership relationship to modify. If the TrgtID is the PNCID, the data key corresponds to the piconet group data key, the management key corresponds to the management key for the relationship with the PNC and the MembershipStatus indicates whether the DEV is a secure member of the piconet. Otherwise, the management key and data key correspond to keys for a peer-to-peer relationship with the DEV indicated by the TrgtID and the MembershipStatus indicates whether the DEV shares a secure relationship with that peer DEV.

If the TrgtID is the PNCID and the MembershipStatus is set to MEMBER, the DEV is a secure member of the piconet. If the TrgtID is the PNCID and the MembershipStatus is set to NON-MEMBER, the DEV is no longer a secure member of the piconet.

The MembershipStatus field indicates to the MLME whether the DEV is currently maintaining secure relationship information with the target DEV. If the MembershipStatus is set to NON-MEMBER, the MLME shall securely delete the management key, the data key and the related SECID, key type and key originator values corresponding to that TrgtID. When a DEV is not a member of a security relationship with a peer DEV, the DEV shall select keys for secure frame processing as if the DEV does not have an individual relationship with that peer DEV, as described in 9.3.8. When a DEV is not a member with the PNC, the DEV is not a secure member of the piconet and shall select keys for secure processing as if the DEV does not have a piconet group data key or PNC-DEV management key, as described in 9.3.8.

If the MembershipStatus is set to MEMBER, the MLME shall examine the KeyInfoLength field to determine if a new key is being added or a key is being deleted. If the KeyInfoLength field is set to zero, the MLME shall securely delete the key and SECID corresponding to the management key or data key for that relationship depending on whether the KeyType is set to MANAGEMENT or DATA. If the deleted key is the management key stored for the relationship, the DEV is unable to transmit or successfully receive frames to any DEV that require protection with the management key, as described in 9.3.8, but the DEV may continue to use the data key corresponding to that relationship and the piconet group data key. This may occur, for instance, during PNC handover, in which the management key with the PNC is no longer valid (since the PNC has changed), but the piconet group data key is still valid. If the deleted key is a data key stored for a peer-to-peer relationship, the DEV is unable to transmit or successfully receive frames that require protection with the data key, as described in 9.3.8, but the DEV may continue to use the management key and piconet group data key. If the deleted key is the piconet group data key, the DEV is unable to transmit or successfully receive frames that require protection with that key, as described in 9.3.8, with the exception of secure beacons, as described in 9.3.6.

If the MembershipStatus is set to MEMBER and the KeyInfoLength field is not 0, the MLME shall examine the KeyType field to determine which key is to be updated. If the KeyType is set to MANAGEMENT, the MLME shall set the SECID, key originator field and key for the management key of this relationship to the values in the SECID, KeyOriginator and KeyInfo fields respectively from the MLME-MEMBERSHIP-UPDATE.request. If the KeyType is set to DATA, the MLME shall ignore the key originator field and set the SECID and key to the values in the SECID and KeyInfo fields respectively from the MLME-MEMBERSHIP-UPDATE.request.

9.3.5 Secure frame generation

When a DEV wishes to send a secure frame, it shall use the keying material required for the type of frame and by the relationship between the sending DEV and the receiving DEV. For each security relationship, there are two keys used to protect secure frames: a management key and a data key. Table 62 provides a listing of which of the keys shall be used to protect secure frames and which frames shall be sent without security. A DEV shall not send a secure frame if the only key selection in Table 62 is ‘none’. A DEV shall not send an unprotected frame or a frame with an incorrect SECID when security is required for that frame. If the DEV is unable to find the corresponding key that is to be used, the MLME shall return an MLME-

SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not transmit the requested frame.

A PNC in a piconet using security shall send secure beacons protected with the piconet group data key stored in the MAC/MLME. For each superframe, the PNC shall increment the time token and transmit a secure beacon with the SEC field in the Frame Control field set to one.

Key selection for secure frames is described in 9.3.8.

If the DEV is able to obtain the appropriate keying material, the DEV shall use the CurrentTimeToken and secure frame counter for the corresponding SECID to construct the CCM nonce, Figure 154, used to protect the secure frame. The SECID included in the frame shall be the value corresponding to the keying material being used. The integrity code shall be computed as specified in 10.3.2. The result of the integrity code computation shall be encrypted as specified in 10.2.2 and placed in the Integrity Code field in the secure frame. The encryption operation shall be applied only to the integrity code, the key that is transmitted in a Distribute Key command or Request Key Response command and the payload of data frames. The result of the encryption operation shall be inserted into the frame in the place of the data that was encrypted. The DEV shall then compute the FCS over the modified frame.

9.3.6 Secure frame reception

Before any security operations have been performed on a received frame, the DEV shall check the FCS. Table 62 provides a listing of the keys that shall be used to protect secure frames and the frames that shall be sent without security. A DEV may ignore any secure frame if the only key selection in Table 62 is ‘none’. A DEV may ignore any non-secure frame or a secure frame with an incorrect SECID when security is required.

An associated device that has not yet received the piconet group data key shall accept all secure beacons and ignore the integrity code, SECID and secure frame counter. When the DEV has received the piconet group data key, it shall set the LastValidTimeToken and CurrentTimeToken to be the time token in that beacon.

When a DEV receives a secure beacon frame (a beacon with the SEC field in the Frame Control field set to one, the DEV shall determine if the received time token is greater than the CurrentTimeToken and less than the LastValidTimeToken + aMaxTimeTokenChange. If not, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-TIME-TOKEN and shall not perform any additional operations on the received beacon. The DEV shall also determine if the SECID matches the SECID of the piconet group data key stored in the MAC/MLME, or the SECID of a valid old piconet group data key, as described in 9.2.5. If the SECID does not match, the DEV may set the CurrentTimeToken to the value in the beacon and request a new piconet group data key, as described in 9.3.2. If both of these checks succeed, the DEV shall check the integrity code on the beacon using the piconet group data key. If this succeeds, the DEV shall accept the beacon and set the LastValidTimeToken and CurrentTimeToken to be the time token in the beacon. If the DEV is able to determine that it missed a beacon or that the beacon was corrupted and if CurrentTimeToken is less than LastValidTimeToken + aMaxTimeTokenChange – 1, the DEV should increment the CurrentTimeToken to maintain synchronization with other DEVs in the piconet.

When a DEV receives a secure non-beacon frame, it shall use the appropriate keying material depending on the type of frame, SECID and source address found in the frame. If the SECID in the frame does not correspond to known keying material in the receiving DEV, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not perform any additional operations on the received frame.

If there are no previous security errors in the processing of the frame, the DEV shall apply the operations defined by the symmetric key security operations to the frame, see Table 10.3.2. If any of the security operations fail, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the Reason-

Code set to FAILED-SECURITY-CHECK and shall not perform any additional operations on the received frame. If the security operations have been successfully performed and the frame has been modified appropriately, the DEV may then continue to process the frame.

While operating in Mode 1, if the MAC receives a command frame with the SEC field in the Frame Control field set to a value different than expected as defined in Table 50, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

9.3.7 Selecting the SECID for a new key

For each management and data key used in the piconet, the key originator in the relationship shall select the 2-octet SECID, as described in 7.2.7.2, that identifies the key. A DEV shall reject any SECID that it receives where the first octet does not contain the correct DEVID as described in 7.2.7.2.

When a PNC capable DEV starts a secure piconet, as described in 8.2.2, it shall select a SECID and a symmetric key to be used for beacon protection. Because there are no other DEVs in the piconet when the PNC capable DEV starts a piconet, this key is not distributed to any other DEVs. Once another DEV joins the piconet, the PNC will update the key and SECID as indicated in 9.3.3.

9.3.8 Key selection

The key used to protect a particular frame depends on the purpose of the frame and the membership states of the DEV. If the DEV is a member of a secure piconet (i.e. the DEV is the PNC or the DEV is a secure member with the PNC), the DEV will have entries for the piconet group data key and for the PNC-DEV management key. If the DEV has a secure relationship with a peer-DEV (i.e. the DEV is a secure member with a peer DEV), the DEV will have entries for a peer-to-peer data key and a peer-to-peer management key that it shares with that DEV. For any given frame, the DEV shall either send the frame without security or with the single key that is required for that frame, as indicated in Table 62. All secure commands between the PNC and other DEVs shall be protected with the PNC management key. All secure data frames with the PNC as either the DestID or SrcID, all secure broadcast frames and all secure beacons shall be protected with the piconet group data key. If two DEVs in a secure piconet do not have a peer-to-peer security relationship, they shall use the piconet group data key for commands that are required to be sent securely and they shall use the piconet group data key for secure data frames transmitted between them. Table 62 summarizes the keys that shall be used for each type of frame.

Table 62—Key selection for secure frames

Frame type or command	None	PNC-DEV mgmt. key	Piconet group data key	Peer-to-peer mgmt. key	Peer-to-peer data key	Comment
Beacon frame			X			All secure beacon frames shall be protected by the piconet group data key.
Imm-ACK frame	X					Immediate acknowledgement frames shall not be secured with any key.
Dly-ACK frame	X					Delayed acknowledgement frames shall not be secured with any key.

Table 62—Key selection for secure frames (*Continued*)

Frame type or command	None	PNC-DEV mgmt. key	Piconet group data key	Peer-to-peer mgmt. key	Peer-to-peer data key	Comment
Data frame	X		X		X	Data frames may be sent at any time with or without security. Secure data frames between DEVs that share a peer-to-peer key shall use the peer-to-peer data key, otherwise they shall use the piconet group data key.
Association request	X					Association Request commands shall not be secured with any key.
Association response	X					Association Response commands shall not be secured with any key.
Disassociation request	X	X				Disassociation Request commands shall not be secured with any key before the DEV establishes secure membership in the piconet and shall be protected by the PNC-DEV management key otherwise.
Request key		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Request key response		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Distribute key request		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Distribute key response		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
PNC handover		X				
PNC handover response		X				
PNC handover information		X				
PNC information request		X				
PNC information		X	X			If the PNC Information command is sent as a directed frame from the PNC to a DEV, the PNC-DEV management key shall be used. If the PNC Information command is sent as a broadcast frame, the piconet group data key shall be used.

Table 62—Key selection for secure frames (*Continued*)

Frame type or command	None	PNC-DEV mgmt. key	Piconet group data key	Peer-to-peer mgmt. key	Peer-to-peer data key	Comment
Probe request	X	X	X	X		If the Probe Request command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Probe response	X	X	X	X		If the Probe Response command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Piconet services	X					
Announce	X	X	X	X		If the Announce command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Channel time request	X	X				If the communicating parties are the PNC and a neighbor PNC, the Channel Time Request command shall not be protected with any key. Otherwise, the PNC-DEV management key shall be used.
Channel time response	X	X				If the communicating parties are the PNC and a neighbor PNC, the Channel Time Response command shall not be protected with any key. Otherwise, the PNC-DEV management key shall be used.

Table 62—Key selection for secure frames (*Continued*)

Frame type or command	None	PNC-DEV mgmt. key	Piconet group data key	Peer-to-peer mgmt. key	Peer-to-peer data key	Comment
Channel status request		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key for the relationship (peer-to-peer or PNC-DEV) shall be used.
Channel status response		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key for the relationship (peer-to-peer or PNC-DEV) shall be used.
Remote scan request		X				
Remote scan response		X				
Transmit power change		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
PM mode change		X				
SPS configuration request		X				
SPS configuration response		X				
PS set information request		X				
PS set information response		X				
Security message	X					
Vendor specific		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.

9.4 Protocol details

The following protocol details include all cryptographic components and headers for the frames. The headers should be interpreted as being headers in the MAC frames. In addition, each element should be interpreted as specified in Clause 7. Note that all frames transmitted in this sub-clause are sent with the ACK Policy field set to Imm-ACK unless specified otherwise. The ACK frames do not affect the security of the protocols and are omitted from all diagrams.

9.4.1 Security information request and distribution

A DEV establishing membership in a security relationship, a DEV may request or send security information to another DEV. This most often is done directly before or during the PNC handover process, but may be done at any time.

Figure 148 illustrates the message flows for Security Information Request and Security Information commands between two peer DEVs.

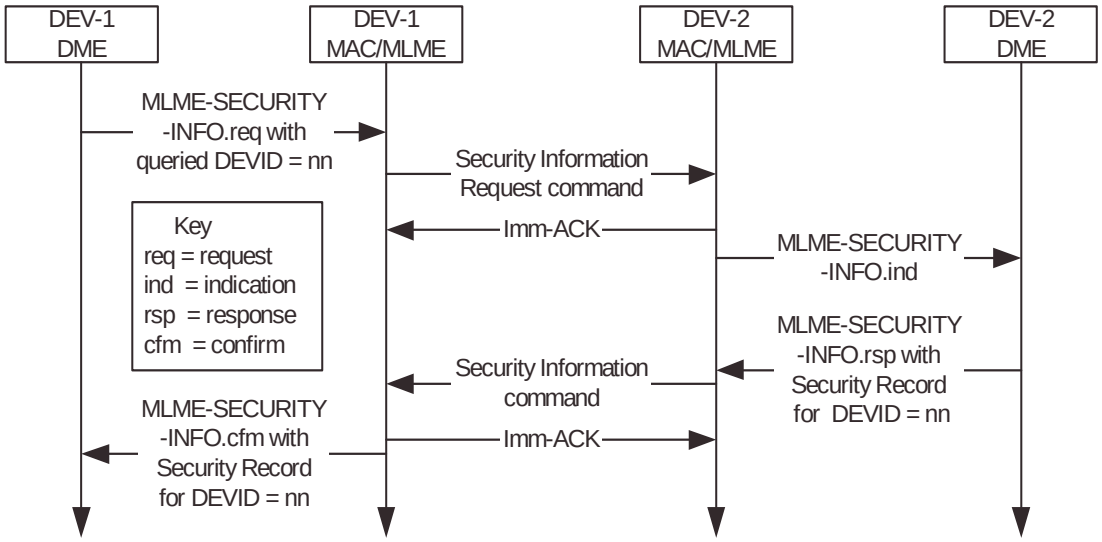

Figure 148—Message sequence chart for DEV-DEV Security information request

Figure 149 illustrates the message flows for an Security Information Request from the new PNC to the old PNC.

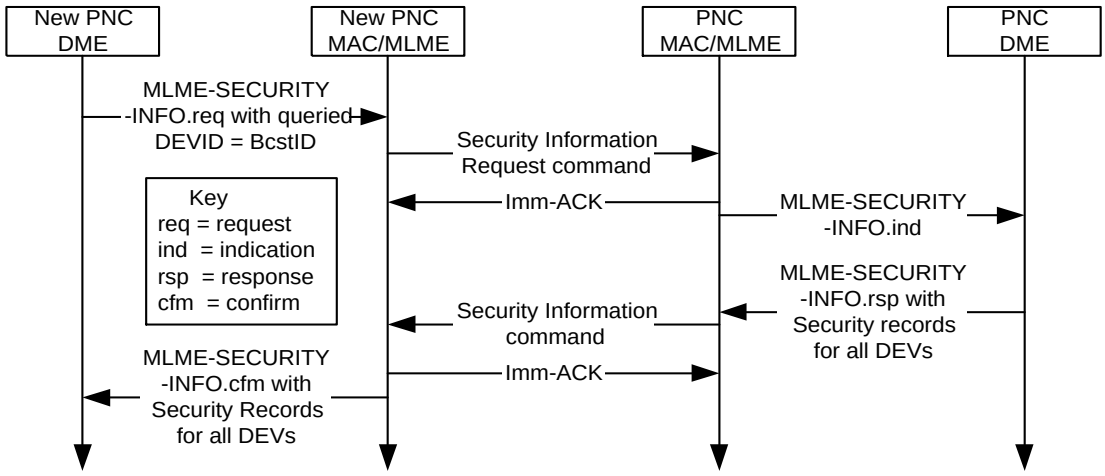

Figure 149—Message sequence chart for New PNC-Old PNC Security Information transfer

Figure 150 illustrates the message flows for an Security Information Request from the new PNC to the old PNC.

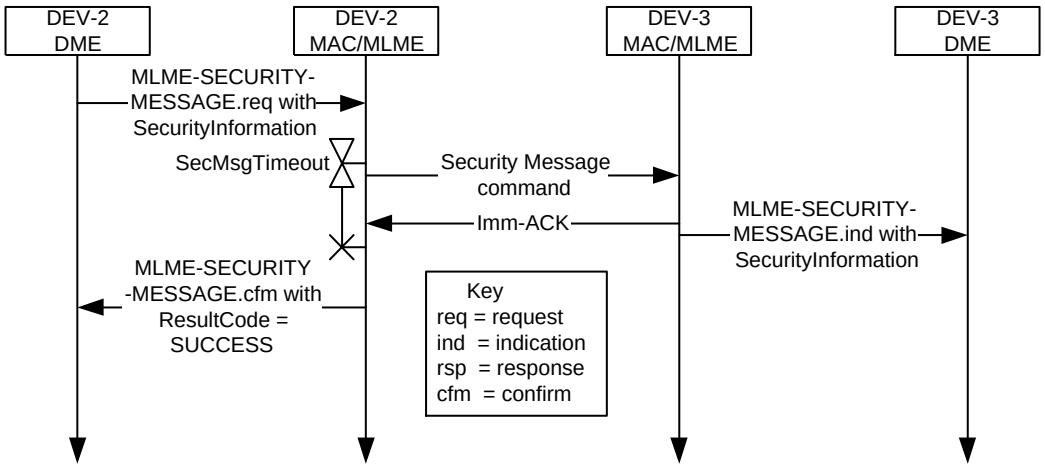

Figure 150—Message sequence chart for sending security information with the Security Message command

9.4.2 Key distribution protocol

In a secure piconet or in a secure peer-to-peer relationship, the key originator may wish to update the current data protection key by initiating the distribute key protocol described here. For a change in the piconet group data key, the PNC sends the new piconet group data key to each member of the piconet. For a change in a peer data key, the key originator in the relationship initiates the distribute key protocol. The key originator should initiate this protocol with each DEV with their respective shared key whenever the key is updated.

Figure 151 illustrates the message flows for the key distribution protocol between the PNC and a DEV. Note that for the PNC-DEV key distribution, the DEV does not send a distribute key response to the PNC.

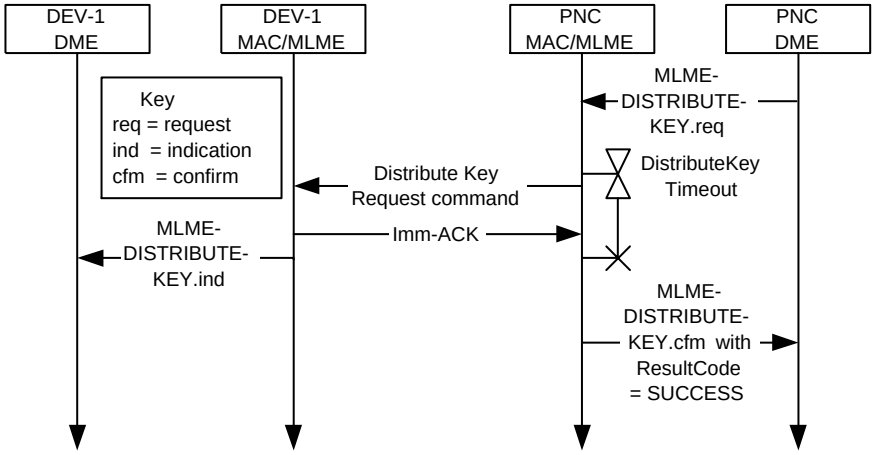

Figure 151—Message sequence chart for PNC-DEV key distribution

Figure 152 illustrates the message flows for the key distribution protocol between a DEV acting as the key originator of the relationship and a DEV.

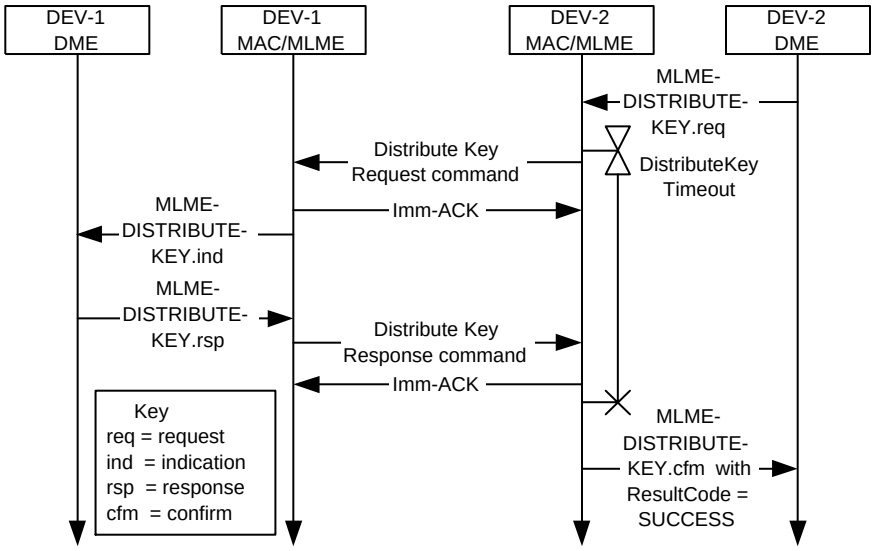

Figure 152—Message sequence chart for peer-to-peer key distribution

9.4.3 Key request protocol

In a secure piconet, if a DEV receives a frame or beacon with an unknown SECID, it may initiate the request key protocol described here in order to obtain the unknown key from the key originator of the relationship. Figure 153 illustrates the message flows for the key request protocol between a DEV and the key originatory.

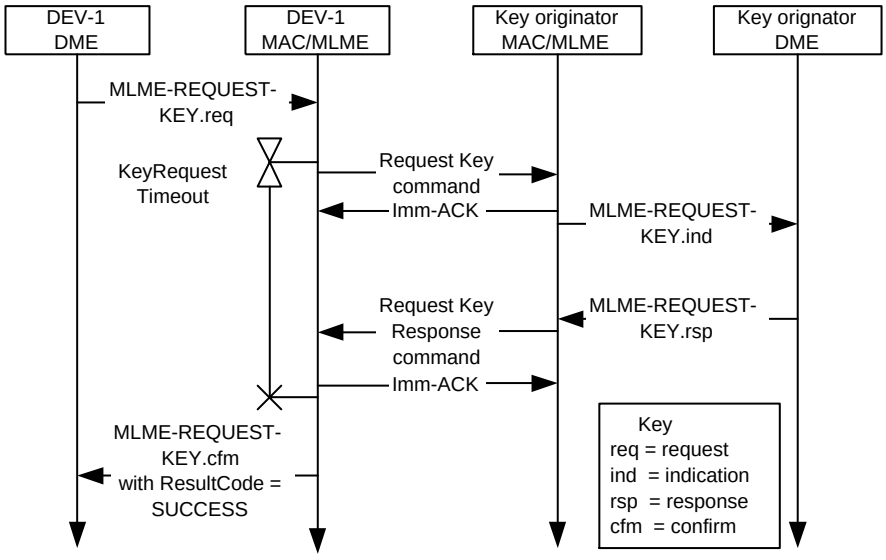

Figure 153—Message sequence chart for DEV key request

10. Security specifications

This clause specifies the security operations that shall be used when security is implemented in the piconet. or in a peer-to-peer security relationship. DEVs that engage in peer-to-peer security relationships may also use different security operations than those being used for piconet protection.

10.1 Modes for security

When symmetric key security operations are selected, DEVs perform secure operations in mode 1. This mode is defined in 9.2. Symmetric key security operations are not defined for mode 0.

10.2 Symmetric cryptography building blocks

The following cryptographic primitives and data elements are defined for use in this standard.

10.2.1 Security interfaces

When transmitting and interpreting security material in this standard, the first byte transmitted shall be the first byte of the security material and represented on the left of the other bytes. The bit ordering within the byte for security operations shall be most significant bit first and least significant bit last. This ordering shall be irrespective of the transmission order of the bits. See Figure 4 for the mapping of bit transmission order to most significant or least significant.

10.2.2 CTR + CBC-MAC (CCM) combined encryption and data authentication

The combined symmetric encryption and data authentication mechanisms used in the symmetric key security operations consists of the generation of an integrity code followed by the encryption of plaintext data and the integrity code. The output consists of the encrypted data and the encrypted integrity code.

The symmetric authentication operation consists of the generation of an integrity code using a block cipher in CBC mode computed on a nonce followed by (optional) padded authentication data followed by (optional) padded plaintext data. The verification operation consists of the computation of this integrity code and comparison to the received integrity code.

The symmetric encryption operation consists of the generation of a key stream using a block cipher in counter mode with a given key and nonce and performing an XOR of the key stream with the plaintext and integrity code. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext and integrity code.

All of the above operations shall be performed as specified in 10.4. The parameters for these operations shall be as specified in 10.2.3.

10.2.3 CCM parameters

The CCM operations shall be parameterized by the following selections: the AES encryption algorithm as specified in 10.2.5, the length in octets of the length field L shall be 2 octets, the length of the authentication field M shall be 8 octets, the nonce shall be formatted as specified in 10.2.4.

10.2.4 Nonce value

The nonce used for CCM encryption and authentication shall be a 13-octet field, dependent on the frame in which it is used, consisting of the 8-bit SrcID followed by the 8-bit DestID followed by the current 6-octet time token followed by the 2-octet secure frame counter followed by the 3-octet Fragmentation Control field from the MAC header. In order to preserve the security of the symmetric algorithms, this nonce shall be unique. As a result, the DEV shall not reuse any 2-octet sequence number within a single superframe that is intented for a particular DEVID (as this would cause a repeated nonce). This uniqueness is guaranteed by the use of the SrcID, which guarantees that different DEVs sharing the same key will use a different nonce, by the time token, which is different for every superframe with a given key and by the DestID and secure frame counter, which guarantee uniqueness within a superframe as long as a DEV does not send more than 65536 frames to a particular DestID within that superframe. If a frame is retransmitted and a single bit in the header or frame body has changed, a new nonce shall be used. To ensure this, each time a frame is retransmitted the secure frame counter shall be incremented.

Figure 154 specifies the format of the nonce that is input to the CCM algorithm. The SrcID, DestID, Secure Frame Counter and Fragmentation Control field shall be included in the frame that is being protected. The time token shall be the time token from the beacon for this superframe.

Octets: 3	**2**	**6**	**1**	**1**
Fragmentation control field	Secure frame counter	Time token	DestID	SrcID

Figure 154—CCM nonce format

10.2.5 AES encryption

The advanced encryption standard (AES) encryption algorithm used for symmetric key security operations shall be performed as specified in NIST FIPS Pub 197. This encryption algorithm is parameterized by the use of 128-bit keys and 128-bit block size.

10.3 Symmetric cryptography implementation

10.3.1 Symmetric cryptography data formats

Table 63 specifies the length and meaning of the symmetric cryptography related specific data elements from Clause 7. The operations performed to obtain the variable data values are specified in 10.3.2.

Table 63—Symmetric cryptography frame object formats

Notation	Length	Value	Description
Encrypted key	16	Variable	The encrypted key consists of the result of the encryption of a 16-octet key (not including the integrity code) using CCM encryption as specified in 10.2.2.
Integrity code	8	Variable	The integrity code consists of the encrypted integrity code that is the result of a CCM computation as specified in 10.2.2 that is computed along with the encrypted seed.
Encrypted data	Variable	Variable	The encrypted data consists of the result of the encryption of the specified data (not including the integrity code) using CCM encryption as specified in 10.2.2.

10.3.2 Symmetric cryptographic operations

Table 64 specifies the symmetric cryptography related operations on all secure frames defined in Clause 7.

Table 64—Symmetric cryptographic operations

Operation	Specification
Secure beacon integrity code generation	The integrity code included in the beacon is generated by computing the encrypted integrity code with the piconet group data key using CCM encryption and data authentication as specified in 10.2.2 with the entire beacon up to the integrity code as the authentication data input *a* and the empty string as the plaintext input *m* for encryption.
Secure command integrity code generation	The integrity code included in command frames is generated by computing the encrypted integrity code with the payload protection key using CCM encryption and data authentication as specified in 10.2.2 with the entire command up to the integrity code as the authentication data input *a* and the empty string as the plaintext input *m* for encryption.
Data integrity code generation	The integrity code included in data frames is generated by computing the encrypted integrity code with the payload protection key using CCM encryption and data authentication as specified in 10.2.2 on the entire data frame up to the encrypted data field as the authentication data input *a* and the data to be encrypted as the plaintext input *m* for encryption (and authentication).
Key encryption operation	The key for key transport is encrypted using CCM encryption and data authentication on the key as specified in 10.2.2 using the management payload protection key with the entire command frame up to the encrypted key field as the authentication data input *a* and the 16-octet pre-encrypted key as the plaintext input *m* for encryption.
Data encryption generation	Data in a data frame is encrypted using CCM encryption and data authentication as specified in 10.2.2 using the data payload protection key with the entire data frame up to the encrypted data field as the authentication data input *a* and the data to be encrypted as the plaintext input *m* for encryption (and authentication).

Figure 155 specifies the length information and data input to the CCM operation for secure beacons. The auth data length *l(a)* shall be set to the length of all of the protected data and the enc data length *l(m)* shall be set to zero. The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and integrity code.

Octets: 2	**2**
Enc Data Length *l(m)* = 0	Auth Data Length *l(a)* = $27+L_1+...+L_{n-1}$

L_{n-1}	...	L_1	13	2	2	10
Information element-(n-1)	...	Information element-1	Piconet synch. parameters	Secure frame counter	SECID	Frame header

Figure 155—CCM input for secure beacons

Figure 156 specifies the length information and data input to the CCM operation for secure commands. For all commands except for the Request Key Response command and Distribute Key Request command, the auth data length *l(a)* shall be set to the length of all of the protected data and the length of encrypted data *l(m)* shall be set to zero. For the Request Key Response command and Distribute Key Request command, the auth data length *l(a)* shall be set to the length of all of the protected data minus 16 (the length of the key) and the enc data length shall be set to 16 (the length of the key). The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and integrity code.

Octets: 2	**2**
Enc Data Length *l(m)* = L_2	Auth Data Length *l(a)* = $18+L_1$

L_2	**L_1**	**2**	**2**	**2**	**2**	**10**
Enc data	Auth data	Length ($=4+L_1+L_2$)	Command type	Secure frame counter	SECID	Frame header

Figure 156—CCM input for secure commands

Figure 157 specifies the length information and data input to the CCM operation for secure data frames. The auth data length *l(a)* shall be set to 14 and the length of encrypted data *l(m)* shall be set to the length of the data payload. The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and integrity code

Octets: 2	**2**
Enc Data Length *l(m)* = L_1	Auth Data Length *l(a)* = 14

L_1	**2**	**2**	**10**
Pre-encrypted data	Secure frame counter	SECID	Frame header

Figure 157—CCM input for secure data frames

10.4 CCM mode

CCM is a generic authenticate-and-encrypt block cipher mode. CCM is defined for use with block ciphers with a 128-bit block size, such as AES.

For CCM mode there are two parameter choices to be made. The first choice is *M*, the size of the authentication field. The choice of the value for *M* involves a trade-off between message expansion and the probability that an attacker will be able to undetectably modify a message. Valid values are 4, 6, 8, 10, 12, 14, and 16 octets. The second choice is *L*, the size of the length field. This value requires a trade-off between the maximum message size and the size of the nonce. Different applications require different trade-offs, so *L* is a parameter. Valid values are 2 to 8 octets (the value $L=1$ is reserved).

Table 65—Parameters of CCM mode

Name	Description	Field Size	Encoding of field
M	Number of octets in authentication field	3 bits	(*M*-2)/2
L	Number of octets in length field	3 bits	*L*-1

10.4.1 Inputs

To send a message the sender shall provide the following information:

— An encryption key *K* suitable for the block cipher.

— A nonce *N* of 15-*L* octets. Within the scope of any encryption key *K*, the nonce value shall be unique. That is, the set of nonce values used with any given key shall not contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties of this mode.

— The message *m*, consisting of a string of *l(m)* octets where $0 \le l(m) < 2^{8L}$. The length restriction ensures that *l(m)* will be able to be encoded in a field of *L* octets.

— Additional authenticated data *a*, consisting of a string of *l(a)* octets where $0 \le l(a) < 2^{64}$. This additional data is authenticated but not encrypted, and is not included in the output of this mode. It may be used to authenticate plaintext headers, or contextual information that affects the interpretation of the message. If there isno additional data to authenticate, the string shall be zero length.

Table 66—Inputs for CCM

Name	Description	Field Size	Encoding of field
K	Block cipher key	Depends on block cipher	String of octets.
N	Nonce	15-*L* octets	Not specified
m	Message to be encrypted and sent	*l(m)* octets	String of octets.
a	Additional authenticated data	*l(a)* octets	String of octets.

10.4.2 Data authentication

The first step is to compute the authentication field *T*. This is done using CBC-MAC. First a sequence of blocks $B_0, B_1, \ldots B_n$ is defined and then CBC-MAC is applied to these blocks.

The blocks shall be ordered as shown in Figure 158

Octets: 16	16* (n-2)	16	16
B_n	B_{n-1} to B_2	B_1	B_0

Figure 158—Authentication block ordering

The first block B_0 shall be formatted as illustrated in Figure 159.

16-*L* ... 15	**1 ... 15-*L***	**Octets: 0**
Flags	Nonce *N*	*l(m)*

Figure 159—First authentication block B_0

The value *l(m)* is encoded in most-significant-octet first order.

The Flags field shall be formatted as illustrated in Figure 160.

Bits: 7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	Adata	M			L		

Figure 160—Authentication flags octet

The Reserved bit is reserved for future expansions and should always be set to zero. The Adata bit is set to zero if *l(a)*=0, and set to one if *l(a)*>0. The *M* field is assigned the value of 4*(bit 5) + 2*(bit 4) + (bit 3) and encodes the value of *M* as (*M*-2)/2. As *M* may take on the even values from 4 to 16, the 3-bit field may take on the values from 1 to 7. The *L* field is assigned the value of 4*(bit 2) + 2*(bit 1) + (bit 0) and encodes the size of the length field used to store *l(m)*. The parameter *L* may take on the values from 2 to 8 (the value L=1 is reserved). This value is encoded in the 3-bit field using the values from 1 to 7 by choosing the field value as *L*-1 (the zero value is reserved).

If *l(a)*>0 (as indicated by the Adata field) then one or more blocks of authentication data are added. These blocks contain *l(a)* and *a* encoded in a reversible manner. The string that encodes *l(a)* shall be constructed as follows:

— If $0 < l(a) < 2^{16}-2^8$ then the length field shall be encoded as two octets which contain the value *l(a)* in most-significant-octet first order.
— If $2^{16}-2^8 \le l(a) < 2^{32}$ then the length field shall be encoded as six octets consisting of the octets 0xff, 0xfe, and four octets encoding *l(a)* in most-significant-octet-first order.
— If $2^{32} \le l(a) < 2^{64}$ then the length field shall be encoded as ten octets consisting of the octets 0xff, 0xff, and eight octets encoding *l(a)* in most-significant-octet-first order.

This is summarized in Table 67. Note that all fields are interpreted in most-significant-octet first order.

Table 67—Length encoding for additional authentication data

First two octets	Followed by	Comment
0x0000		Reserved
0x0001 ... 0xFEFF		For $0 < l(a) < 2^{16} - 2^8$
0xFF00 ... 0xFFFD		Reserved
0xFFFE	four octets $l(a)$	For $2^{16} - 2^8 \leq l(a) < 2^{32}$
0xFFFF	eight octets $l(a)$	For $2^{32} \leq l(a) < 2^{64}$

The blocks encoding a shall be formed by the string that encodes $l(a)$ followed by a itself, and splitting the result into 16-octet blocks, padding the last block with zeroes if necessary. These blocks shall be appended as the octets following the first block B_0. These blocks, if created shall be formatted as shown in Figure 161, where the length of a is 16*(k–1)–L1+L2.

Octets: 16 - L_2	L_2 = 1 to 16	16*(k – 2)	16 – L_1	L_2 = 2, 6 or 10
0	Final octets of a	Next octets of a	First octets of a	$l(a)$
B_k		B_{k-1} to B_2	B_1	

Figure 161—Authentication block ordering for additional authentication blocks

After the (optional) additional authentication blocks have been added, the next step is to form the message blocks. The message blocks are formed by splitting the message m into 16-octet blocks, padding the last block with zeroes if necessary. If the message m consists of the empty string, then no blocks shall be added in this step.

These blocks, if created shall be formatted as shown in Figure 162, where the length of a is 16*(k–1)–L1+L2.

Octets: 16 – L_3	L_3	16*(n – k – 2)	16
0	Final octets of m	Next octets of m	First octets of m
B_n		B_{n-1} to B_{k+2}	B_{k+1}

Figure 162—Authentication block ordering for message blocks

The result is a sequence of blocks B_0, B_1, ..., B_n. The CBC-MAC shall be computed by:

$$X_1 := \mathrm{E}(K, B_0)$$

$$X_{i+1} := \mathrm{E}(K, X_i \oplus B_i) \text{ for } i=1, \ldots , n$$

$$T := \text{first-}M\text{-octets} (X_{n+1})$$

where E() is the block cipher encryption function and T is the integrity code value. Note that the last block B_n is XORed with X_n and encrypted with the block cipher to give T.

10.4.3 Encryption

The message data shall be encrypted with CTR mode. The key stream blocks are defined by

$$S_i := \mathrm{E}(K, A_i)$$

for i=0, 1, 2, ...

The values A_i are formatted as

16–*L* ... 15	**1 ... 15–*L***	**0**
Flags	Nonce *N*	Counter *i*

Figure 163—Encryption blocks A_i

where *i* is encoded in most-significant-octet first order.

The flags field is formatted as

Bits: 7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	Reserved	0			L		

Figure 164—Encryption flags octet

The reserved bits are reserved for future expansions and shall be set to zero. Bit 6 corresponds to the Adata bit in the B_0 block, but this bit is not used here. Bits 3, 4, and 5 shall be set to zero. This ensures that all the *A* blocks are distinct from B_0, which has the non-zero encoding of *M* in this position. Bits 0, 1, and 2 contain *L*, using the same encoding as in B_0.

The message is encrypted by XORing the octets of message *m* with the first *l(m)* octets of S_1, S_2, S_3, ..., S_{n-k} ordered as shown in Figure 165. Note that S_0 is not used to encrypt the message.

Octets: L_3	**16* (n – k – 2)**	**16**	**16**
First L_3 octets of S_{n-k}	S_{n-k-1} to S_3	S_2	S_1

Figure 165—Block ordering for encryption

The authentication value *U* shall be computed by encrypting *T* with the key stream block S_0 and truncating it to the desired length.

$$U := T \oplus \text{first-}M\text{-octets}(S_0)$$

10.4.4 Output

The final result *c* consists of the encrypted message, followed by the encrypted authentication value *U*.

10.4.5 Decryption

To decrypt a message the following information is required:

— The encryption key *K*.
— The nonce *N*.
— The additional authenticated data *a*.
— The encrypted and authenticated message *c*.

Decryption starts by recomputing the key stream to recover the message *m* and the integrity code value *T*. The message and additional authentication data is then used to recompute the CBC-MAC value and check *T*.

If the *T* value is not correct, the receiver shall not reveal any information except for the fact that *T* is incorrect. In particular, the receiver shall not reveal the decrypted message, the value *T*, or any other information.

10.4.6 Restrictions

All implementations shall limit the total amount of data that is encrypted with a single key. The sender shall ensure that the total number of block cipher encryption operations in the CBC-MAC and encryption together shall not exceed 2^{61}. (This allows close to 2^{64} octets to be encrypted and authenticated using CCM, which should be more than enough for most applications.) Receivers that do not expect to decrypt the same message twice may also implement this limit.

The recipient shall verify the CBC-MAC before releasing any information such as the plaintext. If the CBC-MAC verification fails, the receiver shall destroy all information, except for the fact that the CBC-MAC verification failed.

10.4.7 List of symbols

Table 68 provides a list of the symbols used for the above specification of CCM.

Table 68—List of symbols

Name	Description	Size	Comment
a	Additional authenticated data	*l(a)* octets	Use empty string if not desired.
A_i	Counter block to generate key stream	16 octets	Contains block counter, nonce, and flags.
B_i	Input block for CBC-MAC	16 octets	Together encode *N*, *L*, *M*, *m*, and *a* uniquely.
c	Ciphertext	*l(m)* + *M* octets	Includes the encrypted integrity code.
K	Block cipher key	N/A.	At least 128 bits, preferably 256 bits.
L	Number of octets in length field	3 bits	Values 1 ... 8, encoded in 3 bits as *L*-1.
m	Message to be encrypted and sent	*l(m)* octets	Subject to $0 \le l(m) < 2^{8L}$
M	Number of octets in authentication field	3 bits	Values 4, 6, 8, ..., 16. Encoded value is (*M*-2)/2
N	Nonce	15–*L* octets	Nonce should never be repeated for same key.

Table 68—List of symbols

S_i	Block of the encryption key stream	16 octets	Use S_0, S_1, S_2, ... to encrypt *m* and *T*.
T	Unencrypted authentication tag	*M* octets	
U	Encrypted authentication tag	*M* octets	Appended as the higher order octets to the message after encryption
X_i	Intermediate value of CBC-MAC	16 octets	

11. PHY specification for the 2.4 GHz band

11.1 Overview of the 2.4 GHz PHY

This clause specifies the PHY for a single carrier system that supports up to five modulation formats with coding at 11 Mbaud to achieve scalable data rates. The formats, coding and data rates are given in Table 69.

Table 69—Modulation, coding and data rates for 2.4 GHz PHY

Modulation type	Coding	Data rate
QPSK	8-state TCM	11 Mb/s
DQPSK	none	22 Mb/s
16-QAM	8-state TCM	33 Mb/s
32-QAM	8-state TCM	44 Mb/s
64-QAM	8-state TCM	55 Mb/s

This standard is based on the established regulations for Europe, Japan, Canada and the United States. The regulatory documents listed below are for information only and are subject to change or revision at any time. The regulatory domains are enumerated in a vector called PHYPIB_RegDomainSupported, specified in 11.7, and are indicated by the parameter PHYPIB_CurrentRegDomain.

Europe (except France and Spain):

Approval standards: European Telecommunications Standards Institute (ETSI)
Documents: ETS 300-328 [B1], ETS 300-826
Approval authority: National type approval authorities

Japan:

Approval standards: Association of Radio Industries and Businesses (ARIB)
Document: ARIB STD-T66
Approval authority: Ministry of Post and Telecommunications (MPT)

United States:

Approval standards: Federal Communications Commission (FCC), USA
Documents: 47 CFR, Part 15, Sections 15.205, 15.209, 15.249

Canada:

Approval standards: Industry Canada, IC, Canada
Document: GL36

11.2 General requirements

11.2.1 Operating frequency range

This PHY operates in the 2.4–2.4835 GHz frequency range as allocated by the regulatory agencies in Europe, Japan, Canada and the United States as well as any other areas where the regulatory bodies have allocated this band.

11.2.2 RF power measurements

Unless otherwise stated, all RF power measurements for the purposes of this standard, either transmit or receive, shall be made at the appropriate transceiver to antenna connector. The measurements shall be made with equipment that is either matched to the impedance of the antenna connector or is corrected for any mismatch. For DEVs without an antenna connector, the measurements shall be interpreted as EIRP (i.e., a 0 dBi gain antenna) and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

11.2.3 Channel assignments

A total of 5 channels in two sets are assigned for operation. The first set is the high-density mode which allocates 4 channels while the second is an IEEE Std 802.11b™-1999 [B3] co-existence mode which allocates 3 channels. Since the two outer channels of the sets overlap, there are a total of 5 channels allowed for operation. The assigned channels are shown in Table 70. A compliant 802.15.3 implementation shall support all 5 channels.

Table 70—2.4 GHz PHY channel plan

CHNL_ID	Center frequency	High-density	802.11b coexistence
1	2.412 GHz	X	X
2	2.428 GHz	X	
3	2.437 GHz		X
4	2.445 GHz	X	
5	2.462 GHz	X	X

The PHYPIB_Current Channel is the CHNL_ID of the current channel. For the purpose of the Remote Scan Request and Remote Scan Response commands, as described in 7.5.7.3 and 7.5.7.4, respectively, the Channel Index field is the CHNL_ID in Table 70.

11.2.4 Scanning channels

A DEV may, in the course of a scan, change to an 802.11b channel for the purpose of detecting the presence of 802.11b networks.

When a DEV is scanning to start a piconet, it should scan all 5 channels to decrease the probability of chosing an occupied channel.

If a DEV is capable of identifying an 802.11b network and it does identify an 802.11b network while scanning, it should use the 802.11b coexistence channel set. It should also rate the channels where 802.11b networks were identified as the worst channels. If multiple 802.11b networks are detected, the DEV should order them based on an estimate of the amount of traffic and the power level in the channel.

11.2.5 Unwanted emissions

Conformant implementations shall comply with the in-band and out-of-band emissions for all operational modes as set by the applicable regulatory bodies**.**

11.2.6 Operating temperature range

A conformant implementation shall meet all of the specifications in this standard for ambient temperatures from 0 to 40 C°.

11.2.7 PHY layer timing

The values for the PHY layer timing parameters are defined Table 71.

Table 71—PHY layer timing parameters

PHY parameter	Value	Subclause
pPHYMIFSTime	2 μs	11.2.7.4
pPHYSIFSTime	10 μs	11.2.7.2
pCCADetectTime	5*16/11 μs	11.6.5
pPHYChannelSwitchTime	500 μs	11.2.7.5

11.2.7.1 Interframe space

A conformant implementation shall support the IFS parameters, as described in 8.4.1, given in Table 72.

Table 72—IFS parameters

802.15.3 MAC parameter	Corresponding PHY parameter	Definition
MIFS	pPHYMIFSTime	11.2.7.4
SIFS	pPHYSIFSTime	11.2.7.2
pBackoffSlot	pPHYSIFSTime + pCCADetectTime	11.6.5
BIFS	pPHYSIFSTime + pCCADetectTime	11.2.7.2, 11.6.5
RIFS	2*pPHYSIFSTime + pCCADetectTime	11.2.7.2, 11.6.5

11.2.7.2 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be pPHYSIFSTime, including the power-up ramp specified in 11.5.7.

The RX-to-TX turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

11.2.7.3 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than pPHYSIFSTime, including the power-down ramp specified in 11.5.7.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol until the receiver is ready to begin the reception of the next PHY frame.

11.2.7.4 Time between successive transmissions

The time between successive transmissions shall be pPHYMIFSTime, including the power-up ramp specified in 11.5.7.

The pPHYMIFSTime shall be measured at the air interface from the trailing edge of the last symbol transmitted until the first symbol of the PHY preamble is present at the air interface.

11.2.7.5 Channel switch time

The channel switch time is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than pPHYChannelSwitchTime.

11.2.8 Data size restrictions

The PHY definitions creates restrictions on the maximum frame size, maximum transfer unit size and minimum fragmentation size that will be supported. These parameters are defined in this subclause.

11.2.8.1 Maximum frame length

The maximum frame length allowed, pMaxFrameBodySize, shall be 2048 octets. This total includes the frame payload and FCS but not the PHY preamble, PHY header, MAC header or HCS. The maximum frame length also does not include the tail symbols, as described in 11.4.7, or the stuff bits, as described in 11.4.6.

11.2.8.2 Maximum transfer unit size

The maximum size data frame passed from the upper layers, pMaxTransferUnitSize, shall be 2044 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2044 octets minus the security overhead as defined in 7.3.4.2.

11.2.8.3 Minimum fragment size

The minimum fragment size, pMinFragmentSize, that is allowed with the 2.4 GHz PHY shall be 64 octets.

11.2.9 Header check sequence

The combined PHY and MAC headers shall be protected with a CCITT CRC-16 header check sequence (HCS). The MAC parameter, pLengthHCS shall be 2 for this PHY. The CCITT CRC-16 HCS shall be the ones complement of the remainder generated by the modulo-2 division of the protected combined PHY and MAC headers by the polynomial

$$x^{16}+x^{12}+x^5+1 \quad (4)$$

The protected bits shall be processed in transmit order. All HCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 166.

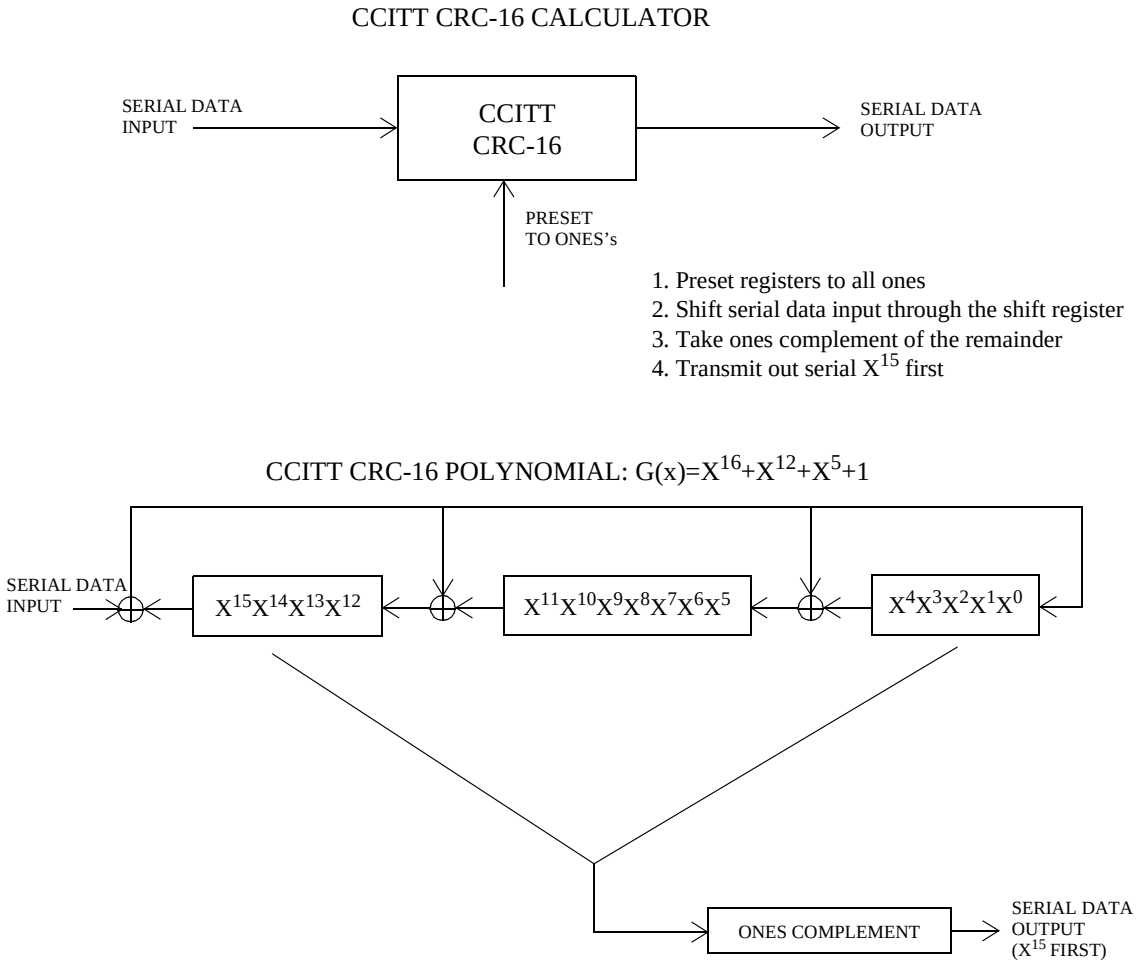

Figure 166—CCITT CRC-16 Implementation

As an example, consider the following 32-bit length sequence to be protected by the CRC-16

0101 0000 0000 0000 0000 0011 0000 0000
b0.....b31

The leftmost bit (b0) is transmitted first in time.

The ones complement HCS for this sequence would be the following:

0101 1011 0101 0111

b0.....b15

The leftmost bit (b0) is transmitted first in time. Bit b0 corresponds to X15 in the Figure 166.

An illustrative example of the CCITT CRC-16 HCS using the information from Figure 166 is shown in Figure 167.

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	

Figure 167—Example of CRC calculation

The CRC-16 described in this subclause is the same one used in IEEE Std 802.11b™-1999 [B3].

11.2.10 Channel access methods

A PNC-capable DEV compliant to this standard shall allow the use of the CAP for contention based access for association, data and commands, as described in 7.3.1, when using the 2.4 GHz PHY. A DEV compliant to this standard shall support the use of the CAP when using the 2.4 GHz PHY.

11.3 Modulation and coding

The IEEE 802.15.3, 2.4-GHz physical layer standard specifies uncoded DQPSK modulation as well as QPSK, 16/32/64-QAM with trellis coding (see Ungerboeck [B14]). An 802.15.3-compliant DEV shall, at a minimum, support DQPSK modulation. In addition, if an 802.15.3 DEV supports a given modulation format other than DQPSK, it shall also support all of the lower modulation formats. For example, if an 802.15.3 implementation supports 32-QAM, it shall also support 16-QAM and QPSK-TCM as well as the DQPSK modulation formats.

The symbol rate for all modulations shall be 11 Mbaud. Based on this symbol rate and the coding, the raw physical layer data rates supported are 11, 22, 33, 44, 55 Mb/s (QPSK-TCM, DQPSK, 16/32/64-QAM – TCM, respectively) as shown in Table 69. The data rates are respectively the entries to the PHYPIB_DataRateVector, as described in 11.7.

11.3.1 Base data rate

The base data rate of the 802.15.3, 2.4-GHz PHY shall be 22 Mb/s operating in the uncoded DQPSK mode.

The DQPSK mode is used as a base rate instead of the 11 Mb/s QPSK-TCM mode to reduce the overhead due to the duration of the PHY and MAC headers. Also, DQPSK capability is necessary to implement the PHY preamble, 11.4.2.

The QPSK-TCM mode is implemented in assigned CTAs to help maintain connections of DEVs that are in range of the PNC, but which may be more distant from each other.

11.3.2 Signal constellations

Figure 168 illustrates the signal constellations used in encoding bit streams into discrete signal levels sent through the common air interface.

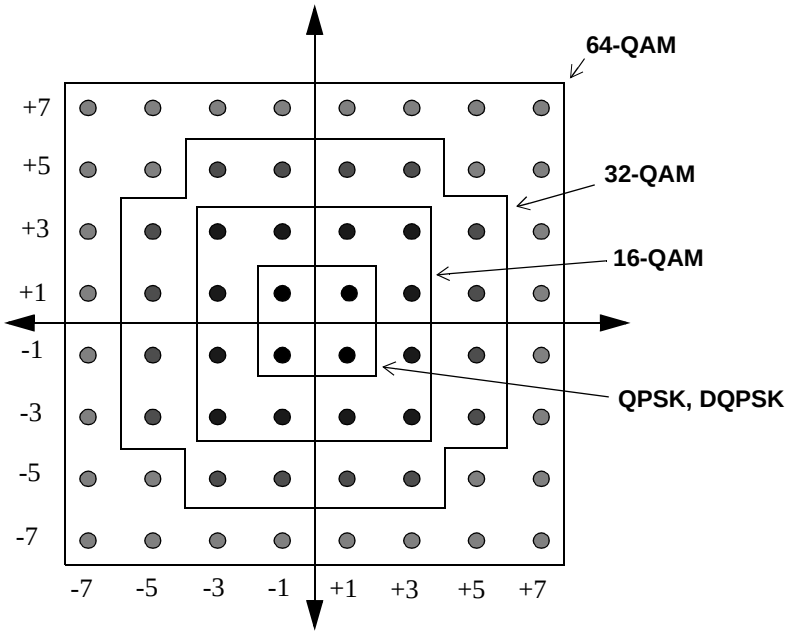

Figure 168—DQPSK, QPSK, 16/32/64-QAM signal constellations

The average power within a frame including the PHY preamble and header is required to be a constant, regardless of the modulation format. Thus, a conformant implementation shall scale the constellation such that the PHY header and the MPDU have the same average power. One method to calculate the normalization is as follows: The output values, d, are formed by multiplying the resulting (I+jQ) value by a normalization factor K_{MOD}, i.e.

$$d = (I + jQ) \times K_{MOD} \tag{5}$$

The normalization factor, K_{MOD}, depends on the base modulation mode and is given in for each of the modulation formats in Table 73. 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 may be used, as long as the DEV conforms with the modulation accuracy requirements described in 11.5.2.

Table 73—Normalization factor for PHY modulation formats

Modulation	K_{MOD}
DQPSK	1
QPSK-TCM	1
16-QAM-TCM	$1/(\sqrt{5})$
32-QAM-TCM	$1/(\sqrt{10})$
64-QAM-TCM	$1/(\sqrt{21})$

11.3.3 DQPSK modulation

No coding shall be applied to the DQPSK modulation. The mapping of the bit pairs to DQPSK symbols shall be implemented as specified in Table 74. In Table 74, a "+jω" phase change shall be defined as a counterclockwise rotation. The differential encoding shall apply only to the DQPSK mode. In this mode the entire frame, with the exception of the PHY preamble, shall be encoded differentially. The phase change of the first symbol is determined relative to the phase of the last symbol in the CAZAC sequence, as described in 11.4.2.

The differential encoding is provided to allow for non-coherent receiver implementations.

Table 74—DQPSK encoding table

Bit pattern (d0,d1) d0 is first in time	Phase change (+jω)
0,0	0
0,1	π/2
1,1	π
1,0	3π/2 (-π/2)

11.3.4 QPSK and 16/32/64-QAM with trellis coding

The QPSK and 16/32/64-QAM formats shall use an 8-State 2-D trellis code. The encoder shown in Figure 169 shall be used in implementing the 8-State 2-D trellis code.

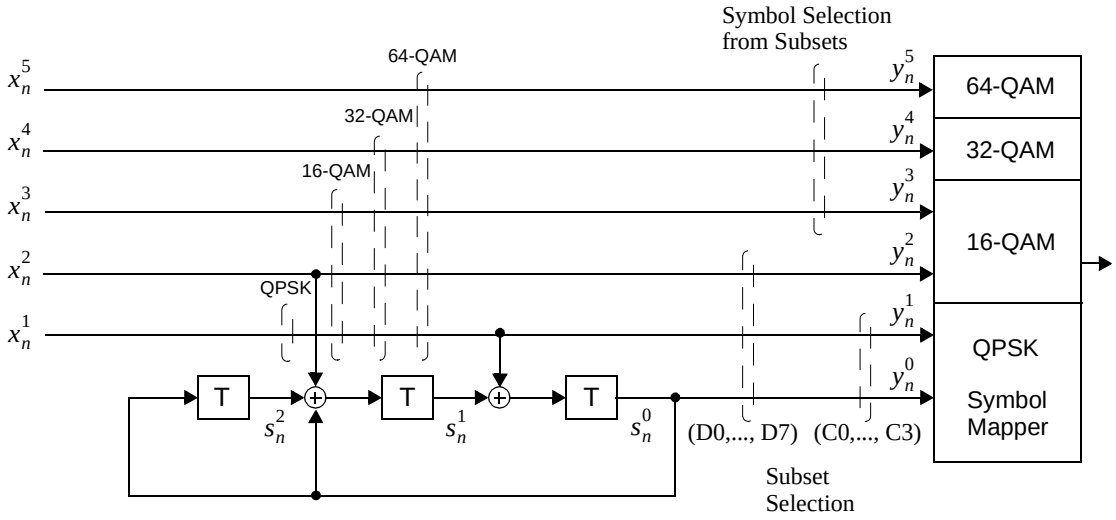

Figure 169—QPSK, 16/32/64-QAM 8-state trellis encoder

The pair $([x_n^5, x_n^4, x_n^3, x_n^2, x_n^1],[y_n^5, y_n^4, y_n^3, y_n^2, y_n^1, y_n^0])$ represents the input and output relationship of the trellis encoder, where 5 input bits per symbol interval are encoded into 6 output bits for 64-QAM symbol mapping. Likewise, the input/output pair $([x_n^4, x_n^3, x_n^2, x_n^1],[y_n^4, y_n^3, y_n^2, y_n^1, y_n^0])$ represents the 32-QAM symbol mapping case, where 4 input bits per symbol interval are encoded into 5 output bits. Similarly, for the case of 16-QAM symbol mapping, 3 input bits per symbol interval are encoded into 4 output bits as denoted by the input/output pair $([x_n^3, x_n^2, x_n^1],[y_n^3, y_n^2, y_n^1, y_n^0])$. Finally, for the QPSK symbol mapping, 1 input bit per symbol interval is encoded into 2 output bits as denoted by the input/output pair $([x_n^1],[y_n^1, y_n^0])$. The 8 states generated by the trellis encoder shown in Figure 169 are denoted as S0, S1, S2,..., S7, which correspond to the binary representations of $s_n^2 s_n^1 s_n^0$. The input-output relations and state transitions of the trellis encoder shown above is given in Table 75 for the QPSK modulation format and in Table 76 for the 16/32/64-QAM formats. Trellis symbol subsets C0, C1, C2, C3 for the QPSK constellation and D0, D1, D2,...,D7 for the 16/32/64-QAM constellations will be described next in the context of "mapping-by-set-partitioning" concept.

Table 75—Input-output relations and state transitions of QPSK trellis encoder

Current state $s_n^2\ s_n^1\ s_n^0$	**Input bit** x_n^1	**Output bits**		**Next state** $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^1\ y_n^0$	**Subset number**	
0 0 0 (**S0**)	0	0 0	**C0**	0 0 0 (**S0**)
	1	1 0	**C2**	0 0 1 (**S1**)
0 0 1 (**S1**)	0	0 1	**C1**	1 1 0 (**S6**)
	1	1 1	**C3**	1 1 1 (**S7**)

Table 75—Input-output relations and state transitions of QPSK trellis encoder (*Continued*)

Current state $s_n^2\ s_n^1\ s_n^0$	Input bit x_n^1	Output bits		Next state $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^1\ y_n^0$	Subset number	
0 10 (**S2**)	0	0 0	**C0**	0 0 1 (**S1**)
	1	1 0	**C2**	0 0 0 (**S0**)
0 1 1 (**S3**)	0	0 1	**C1**	1 1 1 (**S7**)
	1	1 1	**C3**	1 1 0 (**S6**)
1 0 0 (**S4**)	0	0 0	**C0**	0 1 0 (**S2**)
	1	1 0	**C2**	0 1 1 (**S3**)
1 0 1 (**S5**)	0	0 1	**C1**	1 0 0 (**S4**)
	1	1 1	**C3**	1 0 1 (**S5**)
1 1 0 (**S6**)	0	0 0	**C0**	0 1 1 (**S3**)
	1	1 0	**C2**	0 1 0 (**S2**)
1 1 1 (**S7**)	0	0 1	**C1**	1 0 1 (**S5**)
	1	1 1	**C3**	1 0 0 (**S4**)

The symbol mapper shown in Figure 169 provides a one-to-one mapping between an output bit vector of the trellis encoder and a two-dimensional signal point of the signal constellation given in Figure 168. For a given output bit vector of the trellis encoder, a QPSK or 16/32/64-QAM constellation point is selected based on the set partitioning rule illustrated in Figure 170 and Figure 171, respectively. The lower order output bits y_n^1, y_n^0 and y_n^2, y_n^1, y_n^0 are used in determining the symbol subsets C0, C1, C2, C3 and D0, D1, D2,...,D7 for QPSK and 16/32/64-QAM constellations, respectively. The subsets C0, C1, C2, C3 each contain 1 symbol for the QPSK modulation, whereas the subsets D0, D1, D2,...,D7 each contain 2, 4, and 8 symbols for 16-QAM, 32-QAM, and 64-QAM cases, respectively. Therefore, as shown in Figure 169, the remaining output bits y_n^5, y_n^4, y_n^3 select one of the 8 symbols from the subsets D0,...,D7 in the 64-QAM case, and the output bits y_n^4, y_n^3 select one of the 4 symbols from the subsets D0,...,D7 in the 32-QAM case, and finally the output bit y_n^3 selects one of the 2 symbols from the subsets D0,...,D7 in the 16-QAM case. Figure 172 and Figure 173 show the assignment of signal subsets to the QPSK and 16/32/64-QAM constellations, respectively. Furthermore, specific bit mappings to constellation points are given in Figure 174 and Figure 175 for the respective constellations. Binary representations below the subset numbers correspond to $y_n^1 y_n^0$ in the QPSK case and to $y_n^5 y_n^4 y_n^3 y_n^2 y_n^1 y_n^0$ in the 16/32/64-QAM cases. The lower order 2 bits correspond to the subset numbers for the QPSK modulation. Likewise, the lower order 3 bits correspond to the subset numbers for the 16/32/64-QAM. The higher order 3 bits for the 16/32/64-QAM cases are assigned within each signal subset (D0,...,D7) such that decimal representation of the bit mapping goes from low to high as the constellation points are traced from center outward. This rule ensures that the decimal representations of the bit mappings from 0 to 15 belong to 16-QAM constellation, and 0 to 31 belong to 32-QAM constellation, and 0 to 63 belong to 64-QAM constellation.

Table 76—Input-output relations and state transitions of 16/32/64-QAM trellis encoder

Current state $s_n^2\ s_n^1\ s_n^0$	Input bits $x_n^2\ x_n^1$	Output bits		Next state $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^2\ y_n^1\ y_n^0$	Subset number	
0 0 0 (**S0**)	0 0	0 0 0	**D0**	0 0 0 (**S0**)
	0 1	0 1 0	**D2**	0 0 1 (**S1**)
	1 0	1 0 0	**D4**	0 1 0 (**S2**)
	1 1	1 1 0	**D6**	0 1 1 (**S3**)
0 0 1 (**S1**)	0 0	0 0 1	**D1**	1 1 0 (**S6**)
	0 1	0 1 1	**D3**	1 1 1 (**S7**)
	1 0	1 0 1	**D5**	1 0 0 (**S4**)
	1 1	1 1 1	**D7**	1 0 1 (**S5**)
0 1 0 (**S2**)	0 0	0 0 0	**D0**	0 0 1 (**S1**)
	0 1	0 1 0	**D2**	0 0 0 (**S0**)
	1 0	1 0 0	**D4**	0 1 1 (**S3**)
	1 1	1 1 0	**D6**	0 1 0 (**S2**)
0 1 1 (**S3**)	0 0	0 0 1	**D1**	1 1 1 (**S7**)
	0 1	0 1 1	**D3**	1 1 0 (**S6**)
	1 0	1 0 1	**D5**	1 0 1 (**S5**)
	1 1	1 1 1	**D7**	1 0 0 (**S4**)
1 0 0 (**S4**)	0 0	0 0 0	**D0**	0 1 0 (**S2**)
	0 1	0 1 0	**D2**	0 1 1 (**S3**)
	1 0	1 0 0	**D4**	0 0 0 (**S0**)
	1 1	1 1 0	**D6**	0 0 1 (**S1**)
1 0 1 (**S5**)	0 0	0 0 1	**D1**	1 0 0 (**S4**)
	0 1	0 1 1	**D3**	1 0 1 (**S5**)
	1 0	1 0 1	**D5**	1 1 0 (**S6**)
	1 1	1 1 1	**D7**	1 1 1 (**S7**)
1 1 0 (**S6**)	0 0	0 0 0	**D0**	0 1 1 (**S3**)
	0 1	0 1 0	**D2**	0 1 0 (**S2**)
	1 0	1 0 0	**D4**	0 0 1 (**S1**)
	1 1	1 1 0	**D6**	0 0 0 (**S0**)
1 1 1 (**S7**)	0 0	0 0 1	**D1**	1 0 1 (**S5**)
	0 1	0 1 1	**D3**	1 0 0 (**S4**)
	1 0	1 0 1	**D5**	1 1 1 (**S7**)
	1 1	1 1 1	**D7**	1 1 0 (**S6**)

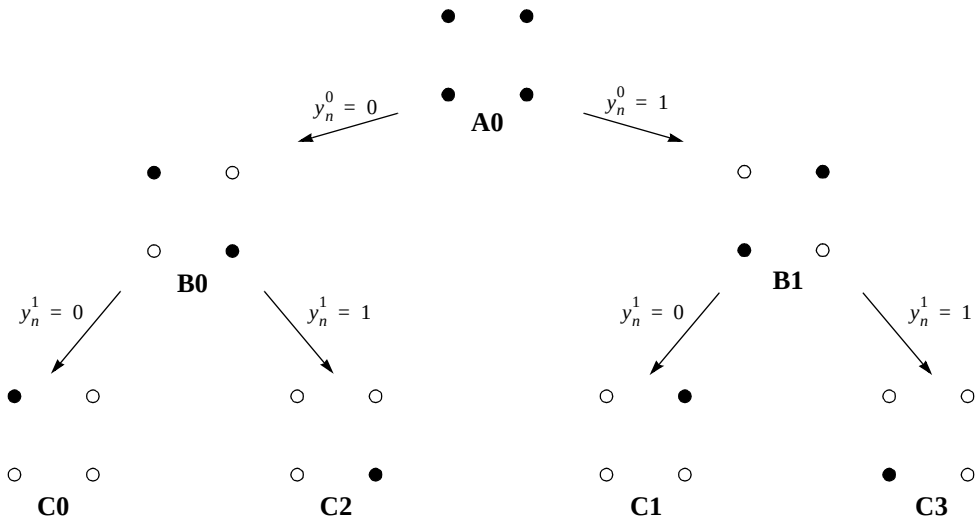

Figure 170—QPSK set partitioning

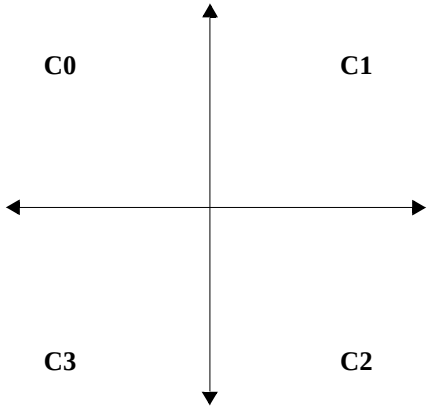

Figure 172—Assignment of subsets to QPSK constellation symbols

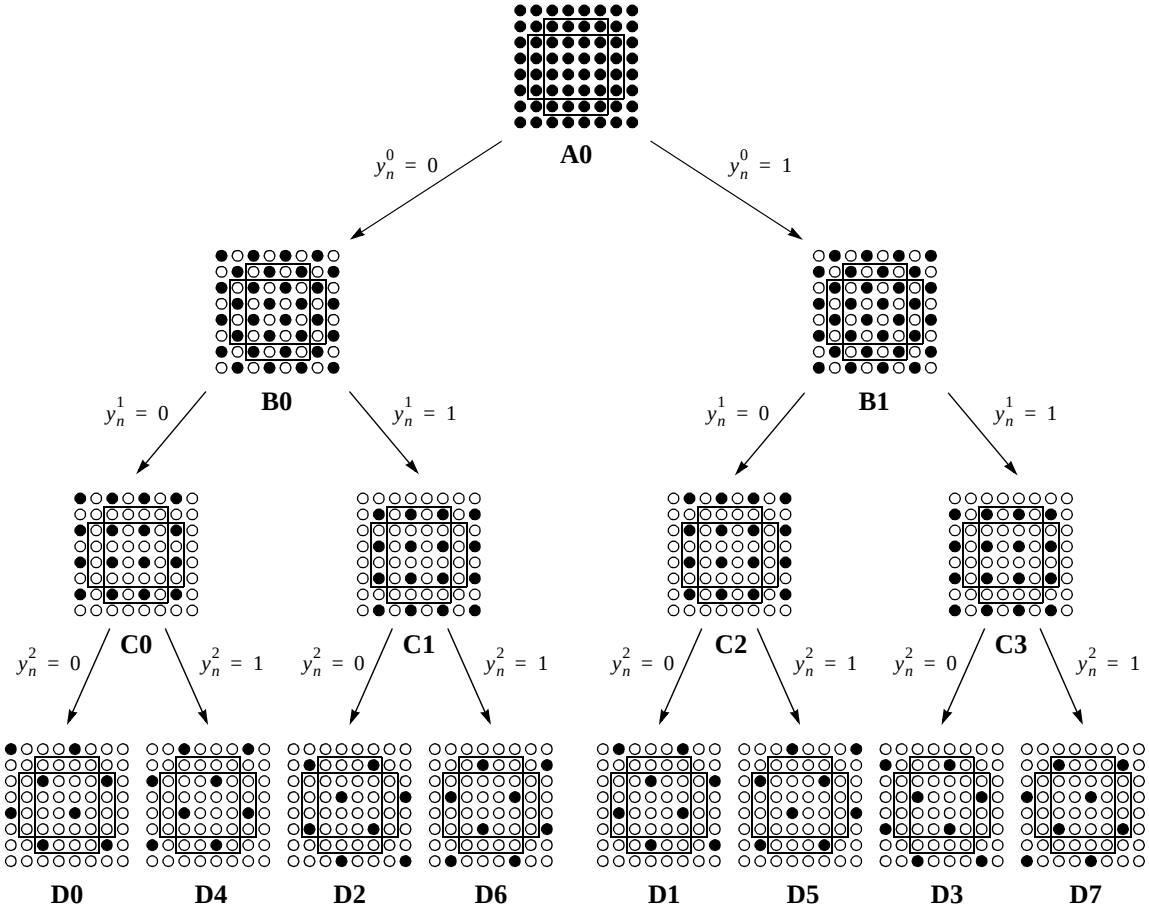

Figure 171—16/32/64-QAM set partitioning

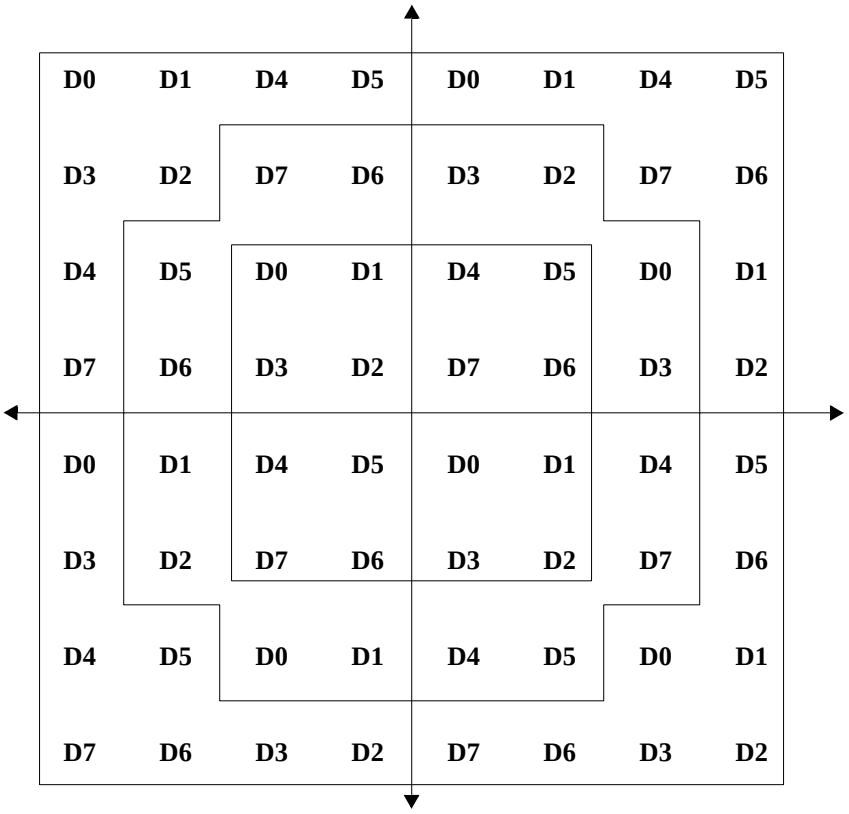

Figure 173—Assignment of subsets to 16/32/64-QAM constellation symbols

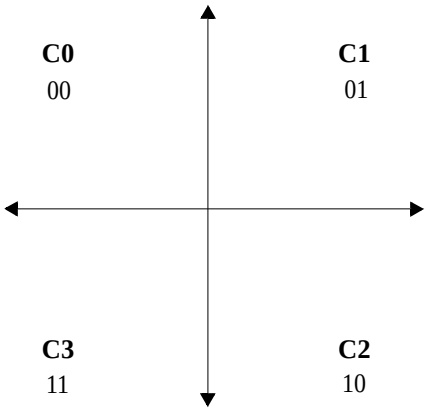

Figure 174—QPSK constellation bit mappings

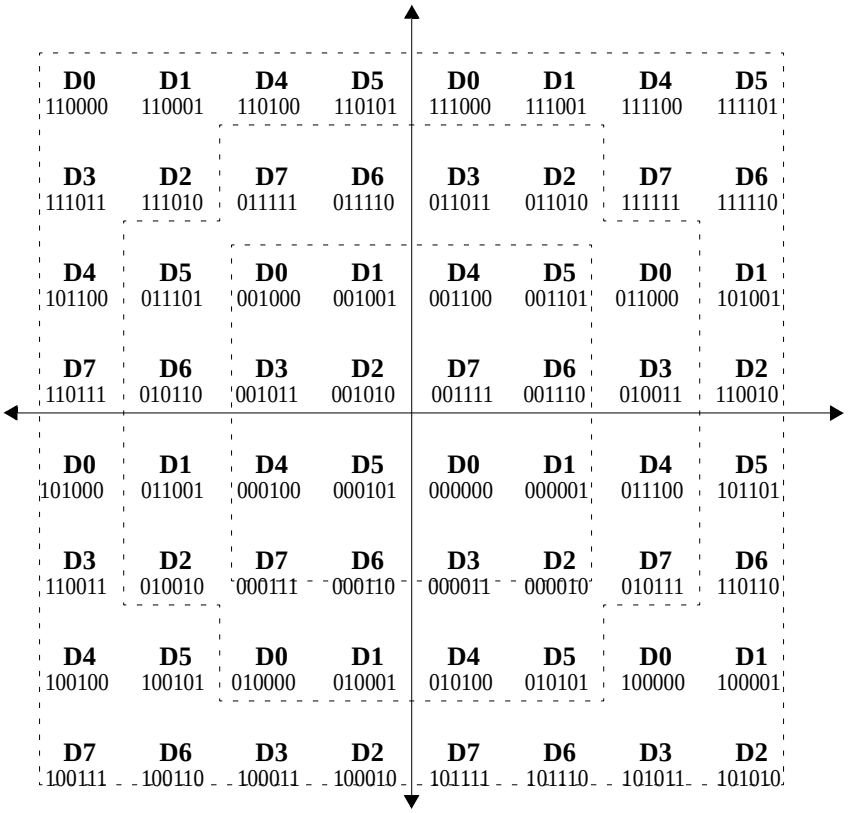

Figure 175—16/32/64-QAM constellation bit mappings

Finally, based on Table 75 and Table 76, the state-transition diagram of the 8-state trellis code is shown for QPSK modulation in Figure 176 and for 16/32/64-QAM in Figure 177.

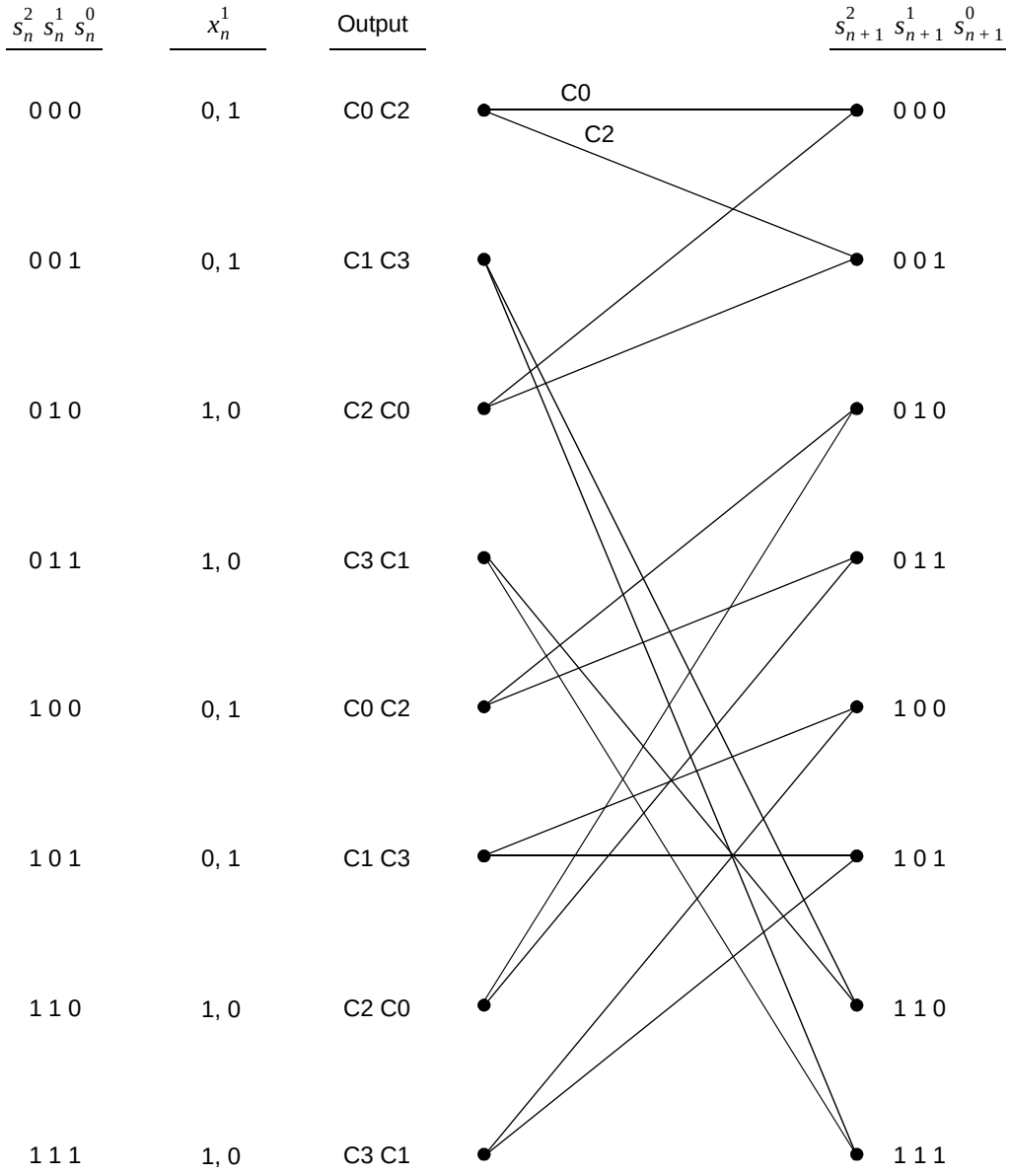

Figure 176—State transition diagram of 8-state QPSK trellis code

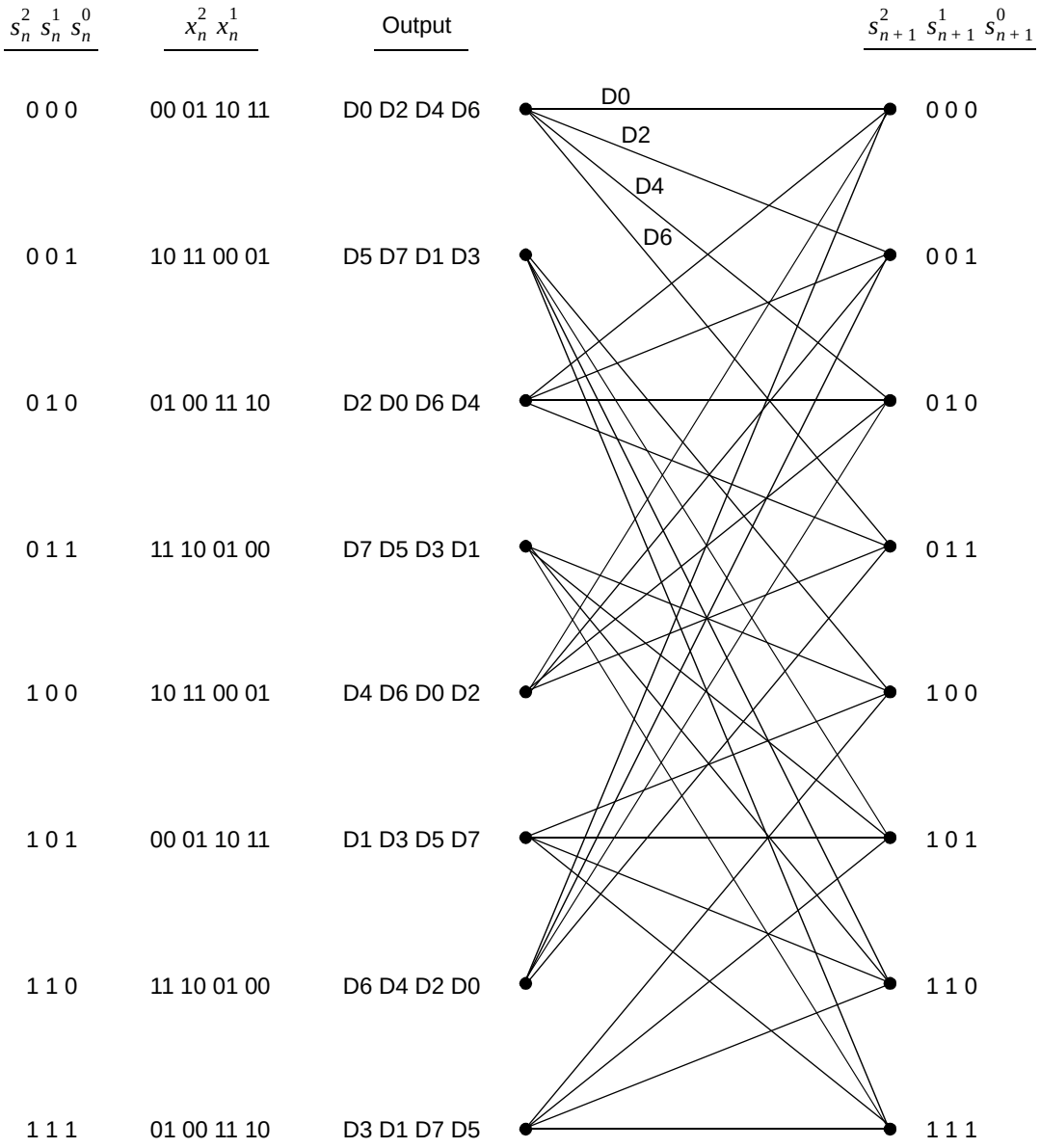

Figure 177—State transition diagram of 8-state 16/32/64-QAM trellis code

11.4 PHY frame format

11.4.1 Frame format

The PHY frame format for the 22, 33, 44 and 55 Mb/s modes is illustrated in Figure 178. The PHY layer prepends the PHY header, as described in 11.4.5, to the MAC header, as described in 7.2, calculates the HCS, as described in 11.2.9, over the combined PHY and MAC headers, and appends the HCS to the end of the MAC header. If the size of the MAC frame body (i.e. the frame payload plus FCS) in bits, is not an

integer multiple of the bits/symbol, then stuff bits (SB) are added following the MAC frame body, as described in 11.4.6. The PHY preamble, as described in 11.4.2, is sent first, followed by the PHY header, MAC header and HCS, followed by the frame payload, the FCS, the stuff bits (SB), if necessary, and finally the tail symbols (TS), as described in 11.4.7. As shown in Figure 178, for the 22, 33, 44, and 55 Mb/s transmission modes, the PHY preamble, as described in 11.4.2, is modulated with the 22 Mb/s QPSK mode. The PHY header, MAC header and the HCS, is modulated in the 22 Mb/s DQPSK mode. Finally, the frame payload, FCS, stuff bits (if necessary) and the tail symbols are modulated at the desired rates of either 22, 33, 44, or 55 Mb/s.

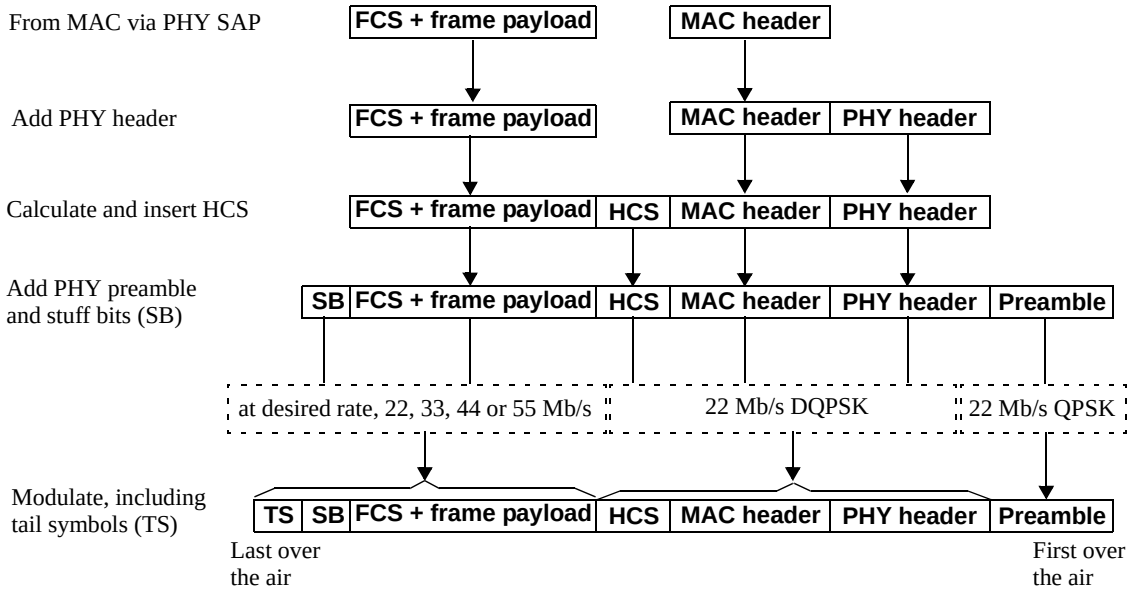

Figure 178—PHY frame formatting for 22, 33, 44 and 55 Mb/s modes

The PHY frame format for the 11 Mb/s mode is slightly different from the other modes and is illustrated in Figure 179. As in the other modes, the PHY layer prepends the PHY header, as described in 11.4.5, to the MAC header, as described in 7.2, calculates the HCS, as described in 11.2.9, over the combined PHY and MAC headers, and appends the HCS to the end of the MAC header. Since the 11 Mb/s mode is modulated at 1 bit per symbol, stuff bits are not needed for the MAC frame body in this mode. The concatenation of the PHY header, MAC header and HCS is repeated in the 11 Mb/s transmission mode. That is, the PHY preamble, as described in 11.4.2, is sent first in the frame, followed by two repetitions of the combined PHY header, MAC header and HCS, followed by the frame payload, the FCS and finally the tail symbols, as described in 11.4.7.

As illustrated in Figure 179, for the 11 Mb/s transmission mode, the PHY preamble, as described in 11.4.2, is modulated in the 22 Mb/s QPSK mode since the CAZAC sequence used, as described in 11.4.2, is based on 4-phase symbols. The first repetition of the combined PHY header, MAC header and the HCS, is modulated in the 22 Mb/s DQPSK mode. The second repetition of the combined PHY header, MAC header and HCS, is modulated in the 11 Mb/s QPSK-TCM mode. The repetition process ensures that the header error rate is significantly lower than the MAC frame body error rate. Finally, the frame payload, FCS, and the tail symbols are all modulated with the 11 Mb/s QPSK-TCM mode. Note that although the concatenation of the PHY header, MAC header and HCS is repeated twice in the PHY frame format, the PHY layer shall only provide one copy of the correctly decoded MAC header to the MAC sublayer at the receiver side.

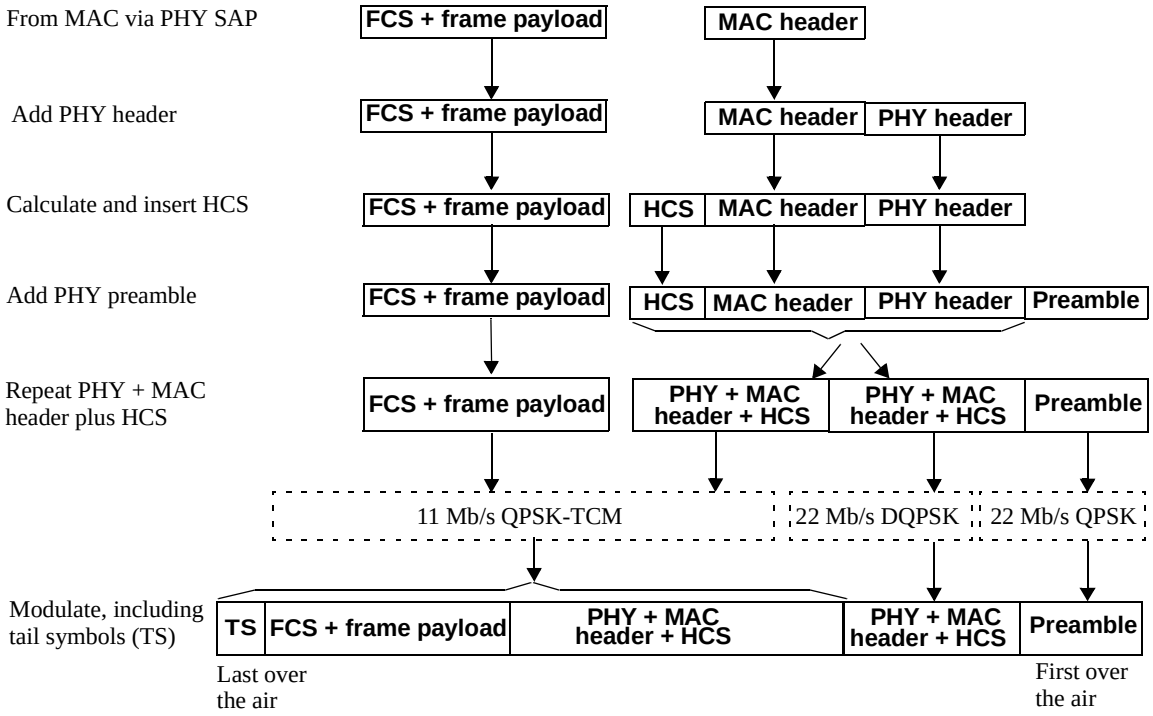

Figure 179—PHY frame formatting for 11 Mb/s mode

11.4.2 PHY preamble

A PHY preamble shall be added prior to the PHY header to aid receiver algorithms related to synchronization, carrier-offset recovery, and signal equalization. The preamble shall consist of multiple periods of a special sequence of 16 QPSK symbols called a CAZAC sequence (see Milewski [B8]), which demonstrates a constant amplitude zero auto-correlation property. The CAZAC sequence shall be denoted as $\{C_0, C_1, C_2, ..., C_{15}\}$. Each element, C_i, of the CAZAC sequence shall have a complex value representing the in-phase and quadrature components of a QPSK-type sequence, as shown in Table 77.

The PHY preamble shall be constructed by successively appending 12 periods, denoted as $\{P_0, P_1, P_2, ..., P_{10}, E\}$, of the CAZAC sequence defined in Table 77, except for the 12^{th} period where each element of the CAZAC sequence shall be negated, or equivalently, rotated by 180 degrees. $P_0, P_1, ..., P_{10}$ are all identical vectors containing the CAZAC symbols defined in Table 77, and the vector E denotes the end-of-preamble delimiter that is the 180 degrees rotated version of the CAZAC sequence given in Table 77. The complete physical layer preamble is shown in Figure 180.

Table 77—CAZAC sequence

CAZAC sequence element	Value
C_0	$1 + j$
C_1	$1 + j$
C_2	$1 + j$
C_3	$1 + j$
C_4	$-1 + j$
C_5	$-1 - j$
C_6	$1 - j$
C_7	$1 + j$
C_8	$-1 - j$
C_9	$1 + j$
C_{10}	$-1 - j$
C_{11}	$1 + j$
C_{12}	$1 - j$
C_{13}	$-1 - j$
C_{14}	$-1 + j$
C_{15}	$1 + j$

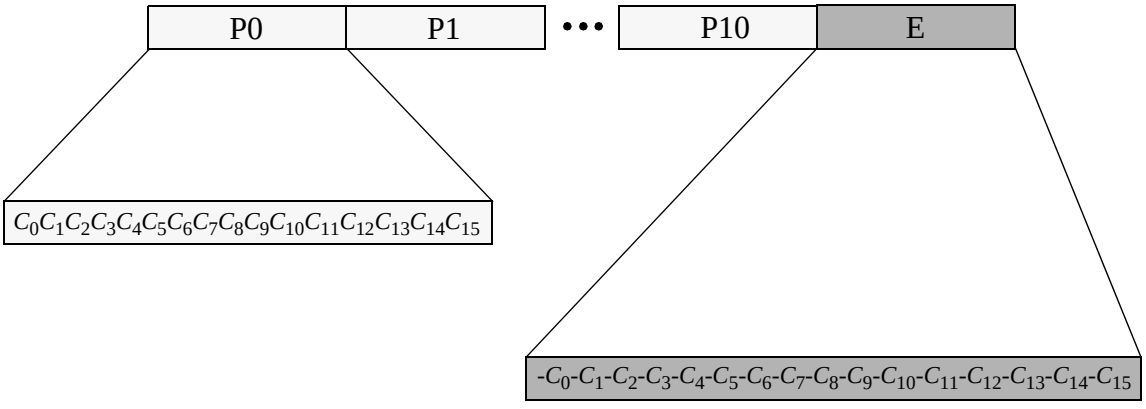

Figure 180—Physical layer preamble format

11.4.3 Header modulation

The PHY header and MAC header shall be modulated using DQPSK modulation for all modulation formats except the 11-Mb/s QPSK-TCM mode.

The header is usually much shorter than the MAC frame body. Consequently, it is more probable to correctly receive the header than the MAC frame body even without a FEC for the header that is modulated in the DQPSK format. This is true for all modulation formats other than the 11-Mb/s QPSK-TCM.

For the QPSK-TCM mode, the combined PHY header, MAC header and the HCS shall be repeated twice as shown in Figure 179 to ensure that the header error rate is lower than the frame payload error rate. The first occurrence of the combined PHY header, MAC header and HCS shall be encoded using the DQPSK modulation. The second occurrence of the combined PHY header, MAC header and HCS shall be encoded using the 11-Mb/s QPSK-TCM format.

11.4.4 Scrambling

A side-stream scrambler shall be used for the MAC header, HCS, MAC frame body and, if present, the stuff bits. The PHY preamble, PHY header and tail symbols shall not be scrambled. The polynomial generator, *g(D)*, for the pseudo-random binary sequence (PRBS) generator shall be

$$g(D) = 1 + D^{14} + D^{15} \tag{6}$$

where *D* is a single bit delay element. The polynomial forms not only a maximal length sequence, but also is a primitive polynomial (see Peterson, et al [B10]). By the given generator polynomial, the corresponding PRBS, x_n, is generated as

$$x_n = x_{n-14} \oplus x_{n-15} \tag{7}$$

where $"" \oplus ""$ denotes modulo-2 addition.

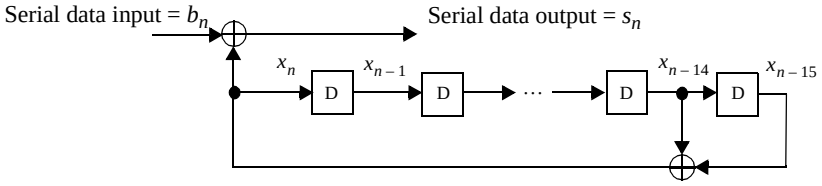

Figure 181—Realization of side-stream scrambler by linear feedback shift registers

The following sequence defines the initialization sequence, x_{init}, which is specified by the parameter “seed value” in Table 78

$$x_{\text{init}} = \left[x^i_{n-1}\ x^i_{n-2}\ x^i_{n-3}\ x^i_{n-4}\ x^i_{n-5}\ x^i_{n-6}\ x^i_{n-7}\ x^i_{n-8}\ x^i_{n-9}\ x^i_{n-10}\ x^i_{n-11}\ x^i_{n-12}\ x^i_{n-13}\ x^i_{n-14}\ x^i_{n-15} \right] \tag{8}$$

where x^i_{n-k} represents the binary initial value at the output of the k^{th} delay element.

The scrambled data bits, s_n, are obtained as follows

$$s_n = b_n \oplus x_n \quad (9)$$

where b_n represents the unscrambled data bits. The side-stream de-scrambler at the receiver shall be initialized with the same initialization vector, x_{init}, used in the transmitter scrambler. The initialization vector is determined from the seed identifier contained in the PHY header of the received frame.

The 15-bit seed value chosen shall correspond to the seed identifier, as described in 11.4.5, and as shown in Table 78. The seed identifier value is set to 00 when the PHY is initialized and is incremented in a 2-bit roll-over counter for each frame that is sent by the PHY. The value of the seed identifier that is used for the frame is sent in the PHY header, as described in 11.4.5.

Table 78—Scrambler seed selection

Seed identifier b1, b0	Seed value $x_{14} \ldots x_0$
0, 0	0011 1111 1111 111
0, 1	0111 1111 1111 111
1, 0	1011 1111 1111 111
1, 1	1111 1111 1111 111

The 15-bit seed value is configured as follows. At the beginning of each PHY frame, the register is cleared, the seed value is loaded in using x_{init}, where n in Equation (8) is set to 15, and the first scrambler bit is calculated. The PHY preamble and PHY header shall not be scrambled. The first bit of data of the MAC header, is modulo-2 added with the first scrambler bit, followed by the rest of the bits in the MAC header and the MAC frame body. In the 11 Mb/s mode, the PHY header of the second repetition of the PHY + MAC header + HCS shall not be scrambled. In this mode, the scrambler shall be re-initialized with the same seed used for the first header when it begins scrambling the second header. The scrambler shall continue its operation as normal for the MAC frame body following the second header structure.

11.4.5 PHY header

The PHY header consists of two octets that identify the number of octets in the frame payload (which does not include the FCS, as described in 7.2), the data rate of the MAC frame body and seed identifier for the data scrambler. The frame payload length does not include the tail symbols, as described in 11.4.7, or the stuff bits, as described in 11.4.6. The fields for the PHY header are shown in Table 79. Bit b0 is sent over the air first and the other bits follow sequentially.

The encoding for the MAC frame body data rate is defined in Table 80.

11.4.6 Stuff bits

If the total length of the MPDU is not an integer multiple of the number of bits/symbol that will be used to modulate the MPDU, then stuff bits shall be added to the end of the MPDU prior to modulation. The stuff bits may be set to either 0 or 1 and shall be ignored when the frame is received. Note that the stuff bits are not a part of either the HCS or FCS calculation. A compliant PHY shall add enough stuff bits so that the MPDU plus stuff bits is an integer multiple of the bits/symbol that is to be used to modulate the MPDU. A compliant PHY shall add less than the number of bits than are contained in a single symbol, i.e. less than 3 for the 33-Mb/s mode and less than 5 for the 55-Mb/s mode.

Table 79—PHY header

Bits	Content	Description
b1–b0	Seed identifier	2-bit field that selects the seed for the data scrambler, defined in Table 78.
b4–b2	MAC frame body data rate	3-bit field that indicates the data rate at which the MAC frame body is sent. The data rate encodings are defined in Table 80.
b15–b5	Payload length	An 11-bit field that contains the length of the frame payload, in octets, msb is b15, lsb is b5, e.g. 5 octets of data, is encoded as 0b00000000101 (msb on left, lsb on right). A zero-length frame payload is encoded as 0b00000000000 and there is no FCS for this frame.

Table 80—MAC frame body data rate encoding

Modulation	Data rate	b4	b3	b2
QPSK-TCM	11 Mb/s	0	0	0
DQPSK	22 Mb/s	0	0	1
16-QAM-TCM	33 Mb/s	0	1	0
32-QAM-TCM	44 Mb/s	0	1	1
64-QAM-TCM	55 Mb/s	1	0	0

Since the MPDU is an integer number of octets, this requirement applies only to the 33- and 55-Mb/s modes. For the 11-, 22-, and 44-Mb/s modes, the bits/symbol (1, 2, and 4, respectively) allow the MPDU to be directly mapped into an integer number of symbols.

11.4.7 Tail symbols

Tail symbols shall be added to the end of the MAC frame body, i.e. after either the FCS or the stuff bits, if they are present, for all modulation formats. The tail symbols are used for trellis coded modulation formats in order to terminate the encoded trellis sequence in a known state to aid the decoding process.

For 11-Mb/s QPSK-TCM format, 3 tail symbols, each containing 1 bit, shall be appended to the end of the MAC frame body. As shown in Figure 176, the lowest order input bit x_n^1 solely determines the state transitions. Encoded trellis sequences shall be terminated in state 0, i.e., S0 at the end of each transmission frame as illustrated in Figure 182. Table 81 shows the assignment of trellis bits based on the last state visited at the end of the frame body.

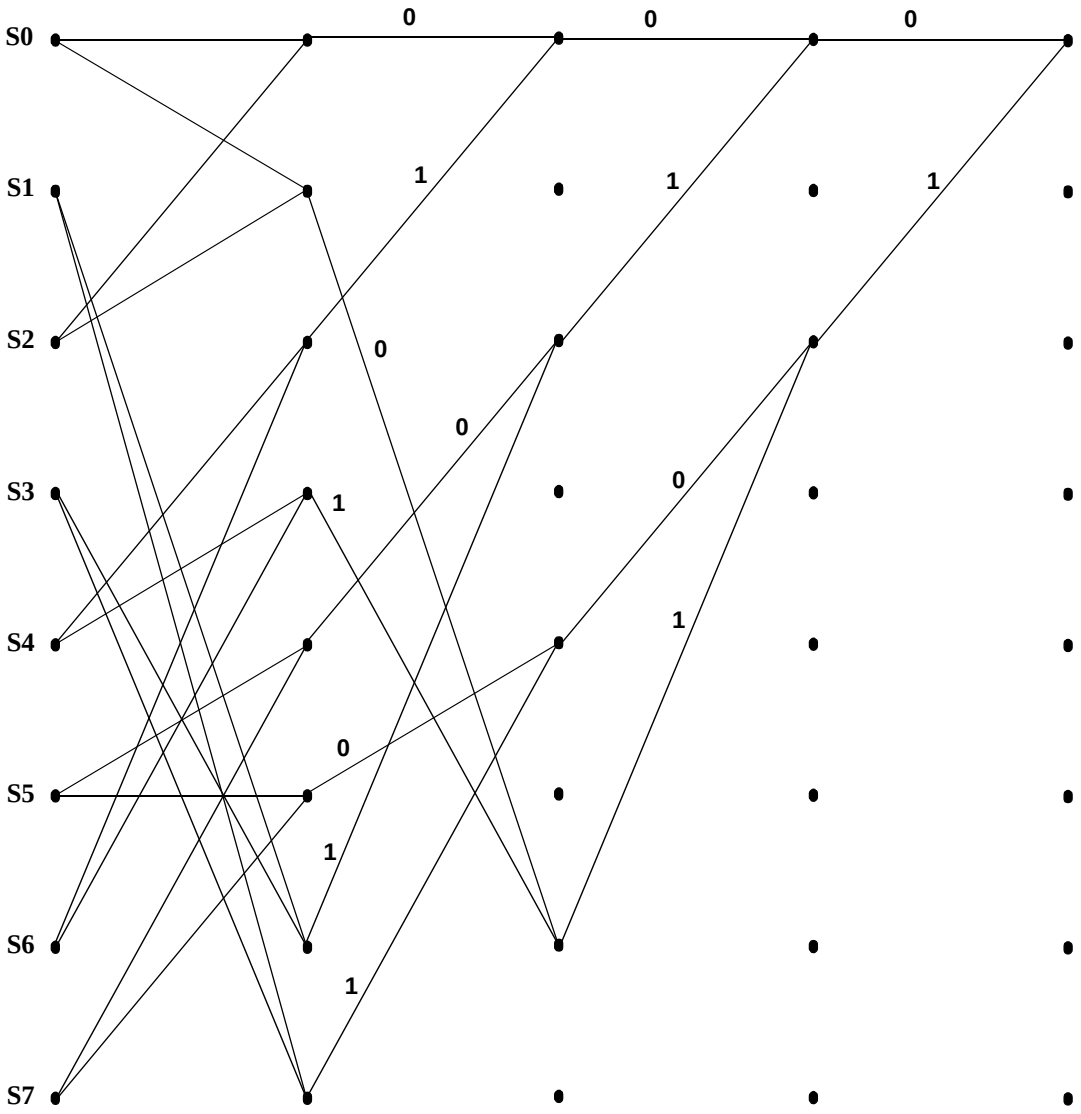

Figure 182—Assignment of tail symbols for QPSK-TCM

For the 16/32/64-QAM formats, the lower order two input bits x_n^2, x_n^1 to the trellis encoder determine the trellis code state transitions as illustrated in Figure 177. In this case, only 2 trellis symbols shall be appended, as shown in Figure 183, to the end of the frame body in order to terminate the trellis sequence in state 0, i.e., S0. Consequently, the one higher order bit, x_n^3, for 16-QAM does not affect the outcome. Likewise, the higher order bits, x_n^4, x_n^3 and x_n^5, ,x_n^4, x_n^3, for 32-QAM and 64-QAM, respectively, are irrelevant in determining the final state. The resultant bit assignments to trellis tail symbols shall take on the form given in Table 82, for which x_n^3, x_n^2, x_n^1 refer to 16-QAM, x_n^4, x_n^3, x_n^2, x_n^1 refer to 32-QAM, and x_n^5, x_n^4, x_n^3, x_n^2, x_n^1 refer to 64-QAM.

Table 81—Assignment of trellis tail symbol (1 bit) for QPSK-TCM

Last State	1st tail bit x_n^1	2nd tail bit x_n^1	3nd tail bit x_n^1
S0	0	0	0
S1	0	1	1
S2	1	0	0
S3	1	1	1
S4	0	1	0
S5	0	0	1
S6	1	1	0
S7	1	0	1

Table 82—Assignment of trellis tail symbols for 16/32/64-QAM

Last State	1st Tail Symbol $x_n^5, x_n^4, x_n^3, x_n^2, x_n^1$	2nd Tail Symbol $x_n^5, x_n^4, x_n^3, x_n^2, x_n^1$
S0	XXX00	XXX00
S1	XXX00	XXX11
S2	XXX01	XXX00
S3	XXX01	XXX11
S4	XXX10	XXX00
S5	XXX10	XXX11
S6	XXX11	XXX00
S7	XXX11	XXX11

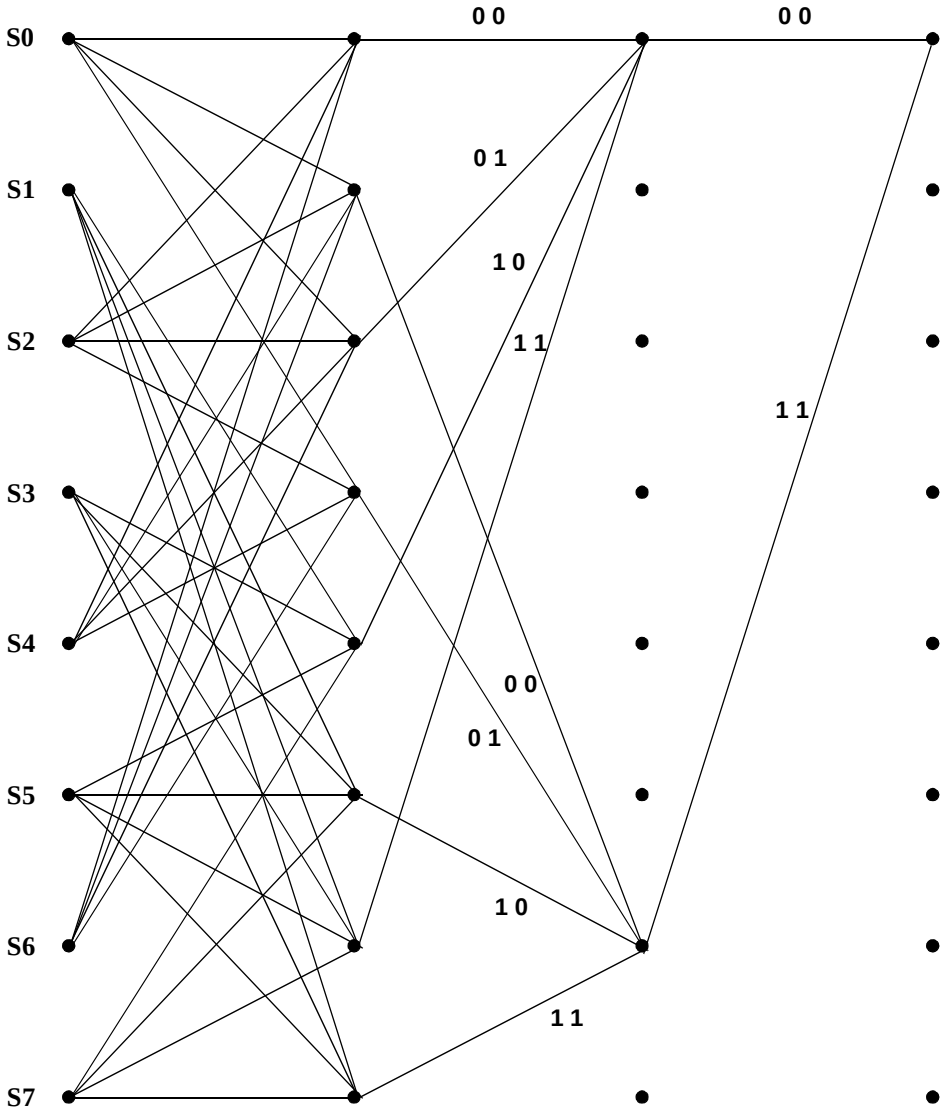

Figure 183—Assignment of tail symbols for 16/32/64-QAM-TCM

For the DQPSK modulation format, the two tail symbols shall be constructed from the binary sequence 0101 where 0 is the first bit of the bit pair.

11.5 Transmitter specifications

11.5.1 Error vector magnitude definition

The modulation accuracy of a compliant IEEE 802.15.3 transmitter is determined with an error-vector magnitude (EVM) measurement. In order to calculate this measurement, a time record of N received signal coordinate pairs $(\tilde{I}_j, \tilde{Q}_j)$ is captured. For each received symbol, a decision is made as to which symbol was transmitted. The ideal position of the chosen symbol (the center of the decision box) is represented by the

vector (I_j, Q_j) . The error vector $(\delta I_j, \delta Q_j)$ is defined as the distance from this ideal position to the actual position of the received symbol.

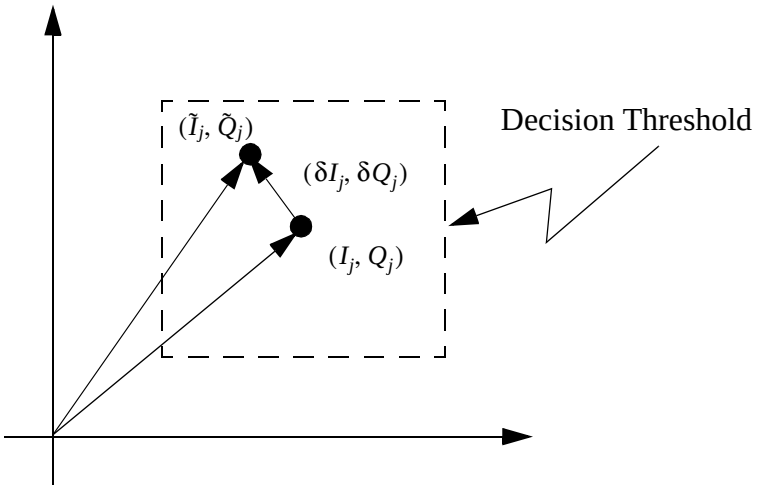

Figure 184—Error vector calculation

Thus, the received vector is the sum of the ideal vector and the error vector.

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q_j) \tag{10}$$

The EVM for this standard is defined as

$$EVM \equiv \sqrt{\frac{\frac{1}{N}\sum_{j=1}^{N}(\delta I_j^{\,2} + \delta Q_j^{\,2})}{S^2_{max}}} \quad \text{X } 100\% \tag{11}$$

where S_{max} is the magnitude of the vector to the outermost constellation point and $(\delta I_j, \delta Q_j)$ is the error vector.

11.5.2 EVM calculated values

A compliant transmitter shall have EVM values of less than those given in Table 83 for all of the modulation levels supported by the PHY when measured for 1000 symbols. The error vector measurement shall be made on baseband I and Q data after recovery through a ideal reference receiver system. The ideal reference receiver shall perform carrier lock, symbol timing recovery and amplitude adjustment while making the measurements. The ideal reference receiver shall have a data filter impulse response that approximates that of an ideal root raised cosine filter with 30% excess bandwidth.

11.5.3 Transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 84. The power shall be measured in a 100 kHz bandwidth relative to the highest average power in a 100 kHz bandwidth measured within +/– 6 MHz of the center frequency.

Table 83—EVM values for various modulations

Modulation	EVM (%)
64 QAM	3.3
32 QAM	4.8
16 QAM	7.5
DQPSK	9.2
QPSK-TCM	20.0

Table 84—Transmit PSD limits

Frequency	Relative limit
7.5 MHz < $\|f–f_c\|$ < 15MHz	–30 dBr
15 MHz < $\|f–f_c\|$ < 22MHz	–10/7[$\|f–f_c$ (MHz)| + 13] dBr
22 MHz < $\|f–f_c\|$	–50 dBr

The transmitter may also have one in-band image, i.e. within 2.4–2.4835 GHz, with a relaxed transmit PSD requirement of –40 dBr over a 15-MHz bandwidth.

A graphical (informative) representation of the transmit PSD is shown in Figure 185.

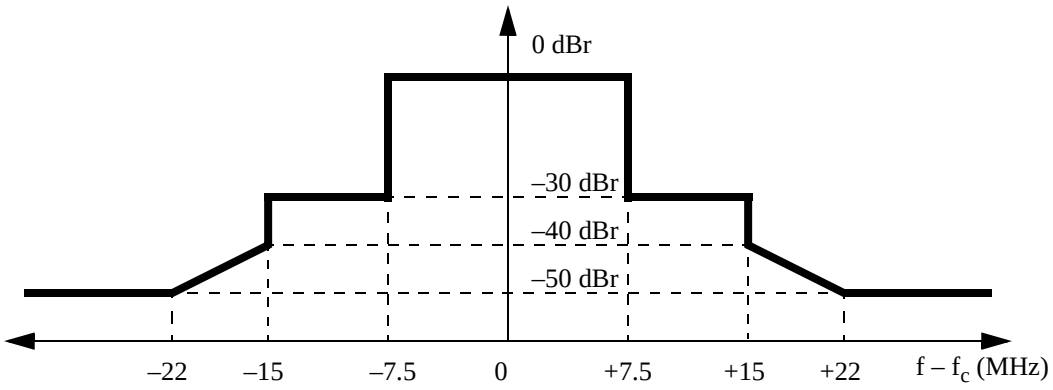

Figure 185—Transmit power spectral density mask

11.5.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ±25 ppm maximum.

11.5.5 Symbol rate

The PHY shall be capable of transmitting at a symbol rate of 11-Mbaud ±25 ppm.

The MAC parameter pPHYClockAccuracy shall be ±25 ppm.

11.5.6 Clock synchronization

The transmit center frequency and the symbol rate shall be derived from the same reference oscillator.

11.5.7 Transmit power-on and power-down ramp

The transmit power-on ramp is defined as the time it takes for the RF power emitted by the compliant DEV to rise from less than 10% to greater than 90% of the maximum power to be transmitted in the frame.

The transmit power-on ramp shall be less than 2 µs.

The transmit power-down ramp is defined as the time it takes for the RF power emitted by the compliant DEV to fall from greater than 90% to less than 10% of the maximum power to be transmitted in the frame.

The transmit power-down ramp shall be less than 2 µs.

The transmit power ramps shall be constructed such that the emissions conform with the unwanted emissions specification defined in 11.2.5.

11.5.8 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.

11.5.9 Transmit power

The maximum allowable output power, as measured in accordance with practices specified by the appropriate regulatory bodies, is shown in Table 85. A compliant DEV may use any transmit power level up to the applicable limits in the geographical region. In the USA, the radiated emissions should also conform to the ANSI uncontrolled radiation emission standards (see IEEE Std C95.1-1999 [B5]). In other geographical regions, compliant DEVs shall also conform to any applicable radiation emission standards.

Table 85—Maximum transmit power levels

Geographical Region	Power limit	Regulatory document
Japan	10 mW	ARIB STD-T66
Europe (except Spain and France)	100 mW EIRP 10 mW/MHz peak power density	ETS 300-328 [B1]
USA	50 mV/m at 3 m in at least a 1 MHz resolution bandwidth*	47 CFR 15.249

*Electric field strength measurement rather than conducted power measurement.

A compliant transmitter that is capable of transmitting more than 0 dBm shall be capable of reducing its power to less than 0 dBm in monotonic steps no smaller than 3 dB and no larger than 5 dB. The steps shall form a monotonically decreasing sequence of transmit power levels. A compliant DEV shall have its supported power levels indicated in its PHY PIB based on its maximum transmit power and power level step size.

The minimum TX power level required to support TPC, pMinTPCLevel, shall be 0 dBm.

11.6 Receiver specifications

11.6.1 Error rate criterion

The error rate criterion shall be a frame error rate (FER) of less than 8% with a frame payload length of 1024 octets of pseudo-random data generated with a PN23 sequence as defined by $x_{n+1} = x_n^{23} + x_n^5 + 1$.

Note that the frames used for measuring the error rate criterion include not only the frame payload of 1024 octets, but also the PHY preamble, PHY header, MAC header, HCS and the FCS.

11.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 11.6.1 is met. The error ratio shall be determined after any error correction has been applied. Compliant systems may have a lower actual sensitivity than the reference sensitivity. A compliant DEV shall achieve at least the reference sensitivity listed in Table 86 for each of the modulation formats that the DEV supports.

Table 86—Reference sensitivity levels for modulation formats

Modulation	Reference sensitivity
QPSK-TCM	–82 dBm
DQPSK	–75 dBm
16-QAM-TCM	–74 dBm
32-QAM-TCM	–71 dBm
64-QAM-TCM	–68 dBm

11.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 11.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

11.6.4 Receiver jamming resistance

The jamming resistance levels are given in Table 87. The high-density channel plan shall be used for this test. The adjacent channel is the one on either side of the desired channel that is closest in frequency to the desired channel for the high-density channel mode defined in 11.2.3. The alternate channel is one removed from the adjacent channel in that channel mode. For example, when channel 1 is the desired channel, channel 2 is the adjacent channel and channels 4 and 5 are the alternate channels. When channel 2 is the desired channel, channels 1 and 4 are the adjacent channels and channel 5 is the alternate channel. Channel 3 is not used for the purposes of this test.

The list of adjacent and alternate channels is given in Table 88.

Table 87—Receiver jamming resistance requirements

Modulation format	Adjacent channel rejection	Alternate channel rejection
QPSK-TCM	33 dB	48 dB
DQPSK	26 dB	41 dB
16-QAM-TCM	25 dB	40 dB
32-QAM-TCM	22 dB	37 dB
64-QAM-TCM	19 dB	34 dB

Table 88—Adjacent and alternate channels for receiver jamming resistance test

Desired channel number	Adjacent channel number	Alternate channel number
1	2	4, 5
2	1, 4	5
4	2, 5	1
5	4	1, 2

The adjacent channel rejection shall be measured as follows. The desired signal shall be a conformant 802.15.3 signal of pseudo-random data modulated with one of the five modulation types. The desired signal is input to the receiver at a level 6 dB above the reference sensitivity for that modulation as given in Table 86. In either the adjacent or alternate channel a conformant 802.15.3 signal is input at the level specified in Table 87 relative to the reference sensitivity for that modulation as given in Table 86. For example, for QPSK-TCM the desired signal is input at –76 dBm while the adjacent channel interferer would be input at a level of –49 dBm. The interfering signal shall be DQPSK modulated with pseudo-random data uncorrelated in time with the desired signal. The receiver shall meet the error rate criteria defined in 11.6.1 under these conditions.

A compliant implementation shall satisfy the receiver jamming test for all of the modulations types supported by the DEV.

The desired and interfering signals shall conform to the transmit PSD mask specified in 11.5.3. In addition, the test shall be performed for only one interfering signal at a time.

11.6.5 Receiver CCA performance

A compliant receiver provides CCA capability by performing energy detection in the received signal bandwidth. The start of a valid preamble sequence at a receive level equal to or greater than the minimum sensitivity for the DQPSK base rate, as described in 11.6.2, shall cause CCA to indicate medium busy with a probability of >90% within five CAZAC periods (approximately 7.3 μs), as described in 11.4.2. The receiver CCA function shall in all circumstances report medium busy with any signal 20 dB above the minimum sensitivity for the DQPSK base rate.

The CCA detection time shall be equal pCCADetectTime, which is five CAZAC periods. The CCA shall be maintained as busy until the end of the frame for which the inverted CAZAC sequence was detected.

11.6.6 Receiver RSSI

RSSI is defined as the power relative to the maximum receiver input power level, as described in 11.6.3, in 8 steps of 8 dB with +/– 4 dB step size accuracy. The range covered shall be a minimum of 40 dB. The steps shall be monotonic. The RSSI power shall be the average power measured in the last CAZAC sequence of the PHY preamble, as described in 11.4.2. This number is reported via the PHY-RX-START.indication, as described in 6.7.4.3, if that particular PHY interface is has been implemented.

11.6.7 Link quality indication

The link quality indication (LQI) shall be reported for the TCM coded QAM modes using an SNR estimation. The SNR shall be measured at the decision point in the receiver. The SNR includes the thermal noise, distortion, uncorrected interference and other signal impairments at the decision point in the receiver. The receiver shall report the SNR as a 5-bit number that covers a range from 6 dB to 21.5 dB of SNR. The value 0x00000 shall correspond to less than or equal to 6 dB SNR and 0x11111 shall correspond to more than or equal to 21.5 dB SNR with equal steps in between. The LQI SNR shall be measured during the TCM MAC frame body and shall be reported after the last FCS symbol. This number shall be reported via the PHY-RX-END indication, as described in 6.7.4.6.

11.7 PHY management

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the PHY layer of a DEV. The encoding of the PHY data rates used in the Supported Data Rates field in the Capability IE, as described in 7.4.11, is given in Table 89.

Table 89—2.4 GHz PHY supported data rate encoding

Rates supported (Mb/s)	b0	b1	b2	b3	b4
22	0	0	0	0	0
11, 22	1	0	0	0	0
11, 22, 33	0	1	0	0	0
11, 22, 33, 44	1	1	0	0	0
11, 22, 33, 44, 55	0	0	1	0	0

The encoding of the supported PHY data rates into an octet number is accomplished by adding bits b5-b7, all set to zero, to the encoding given in Table 89. Bit b0 is the lsb while bit 7 is the msb. Thus a DEV that supports 11, 22 and 33 Mb/s would have a Supported Data Rates field 01000 (lsb to msb) and an octet encoding of 0x2.

The encoding of the preferred fragment size used in the Capability IE, as described in 7.4.11, is given in Table 90.

Table 90—2.4 GHz PHY preferred fragment size encoding

Field value	Preferred fragment size (octets)
0	pMaxFrameBodySize
1	1792
2	1536
3	1280
4	1024
5	512
6	256
7	pMinFragmentSize

The encoding of the DataRate parameter used in the PLME SAP, as described in 6.4, and the encodings of the TXDataRate and RXDataRate parameters used in the PHY SAP, as described in 6.7, are based on the value for the data rate sent in the PHY header, Table 80 and is given in Table 91.

Table 91—Encoding of the 2.4 GHz PHY data rates for the PHY SAP

Modulation	Data rate	TXDataRate/ RXDataRate value
QPSK TCM	11 Mb/s	0x00
DQPSK	22 Mb/s	0x01
16-QAM TCM	33 Mb/s	0x02
32-QAM TCM	44 Mb/s	0x03
64-QAM TCM	55 Mb/s	0x04

The PHY dependent PIB values for the 2.4 GHz PHY are given in Table 92 and Table 93. The PHY PIB characteristics group, Table 92, contains information that is common to most 2.4-GHz implementations.

Table 92—PHY PIB characterisitcs group parameters

Managed object	Octets	Definition	Access
PHYPIB_Type	1	0x00=2.4 GHz	Read only
PHYPIB_RegDomainsSupported	Variable	One octet for each regulatory domain supported, as defined for PHYPIB_CurrentRegDomain	Read only
PHYPIB_CurrentRegDomain	1	0x00 = European Telecommunications Standards Institute (ETSI) 0x01 = Federal Communications Commission (FCC) 0x02 = Industry Canada (IC) 0x03 = Association of Radio Industries and Businesses (ARIB)	Read only
PHYPIB_DataRateVector	1	Encodes the data rates, defined in Table 89 and 11.7.	Read only
PHYPIB_NumChannelsSupported	1	Value = 0x05, see 11.2.3.	Read only
PHYPIB_CurrentChannel	1	Indicates the channel that is currently being used, see 11.2.3.	Read only
PHYPIB_CCAThreshold	1	The CCA threshold in dBm, encoded in 2's complement format. The value is implementation dependent but no larger than the value listed in 11.6.5.	Read only
PHYPIB_FrameLengthMax	2	pMaxFrameBodySize, see 11.2.8.1.	Read only

The PHY PIB implementation group, Table 93, contains information that is more characteristic of a particular PHY implementation than of the PHY as a whole.

Table 93—PHY PIB implementation group parameters

Managed object	Octets	Definition	Access
PHYPIB_DiversitySupported	1	Numeric entry that indicates the number of antennas that are available.	Read only
PHYPIB_MaxTXPower	1	The maximum TX power that the DEV is capable of using, 7.4.11, implementation dependent.	Read only
PHYPIB_TXPowerStepSize	1	The step size for power control supported by the DEV, 7.4.12, implementation dependent.	Read only
PHYPIB_NumPMLevels	1	Number of power management levels supported. The range is 1to 255 and the value is implementation dependent.	Read only
PHYPIB_PMLevelReturn	Variable	Table of vectors with number of entries given by PHYPIB_NumPMLevels. Each vector is the time required to change between power saving states of the PHY. Vector number 0 is the time required to change the PHY from the off state to a state where it is ready to receive commands. Other values are implementation dependent.	Read only

Annex A

(normative)

Frame convergence sublayer

A.1 Generic convergence layer

The frame convergence sublayer (FCSL) may contain one or more convergence sublayers [i.e. IEEE 802.2™, IEEE 1394™ [7], Universal Serial Bus (USB), etc.] as illustrated in Figure A.1. The FCSL provides these functions:

a) receiving PDUs from upper protocol layers via the appropriate FCSL-SAP
b) classifying PDUs received from upper protocol layers according to a classification rule set.
c) delivering each classified PDU to the MAC-SAP
d) receiving PDUs from peer FCSLs via the MAC-SAP
e) delivering received PDUs to the upper protocol layers via the appropriate FCSL-SAP.

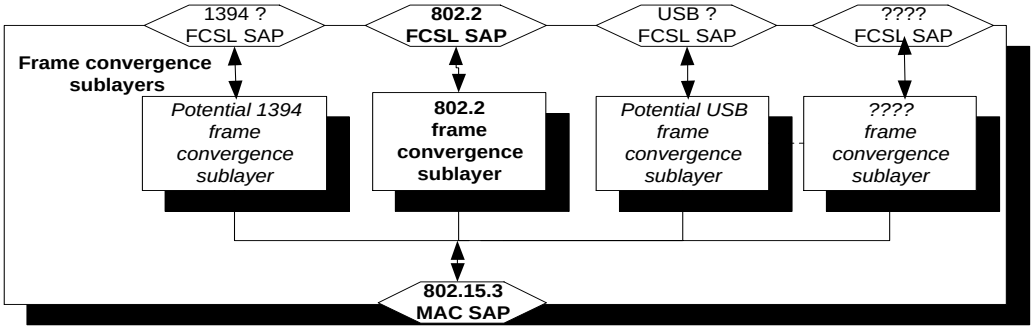

Figure A.1—802.15.3 Frame convergence sublayer model

An implementer is allowed to send IntServ packages and define IntServ policy functions in the FCSL, but these capabilities are outside of the scope of this standard. The FCSL also allows other QoS services to be supported, e.g. IEEE 1394™ and others, but these are also outside of the scope of this standard.

[7]The IEEE product referred to is a trademark belonging to the Institute of Electrical and Electronics Engineers, Inc.

Figure A.2 illustrates the relationship between an FCSL and the rest of the 802.15.3 protrocol entities.

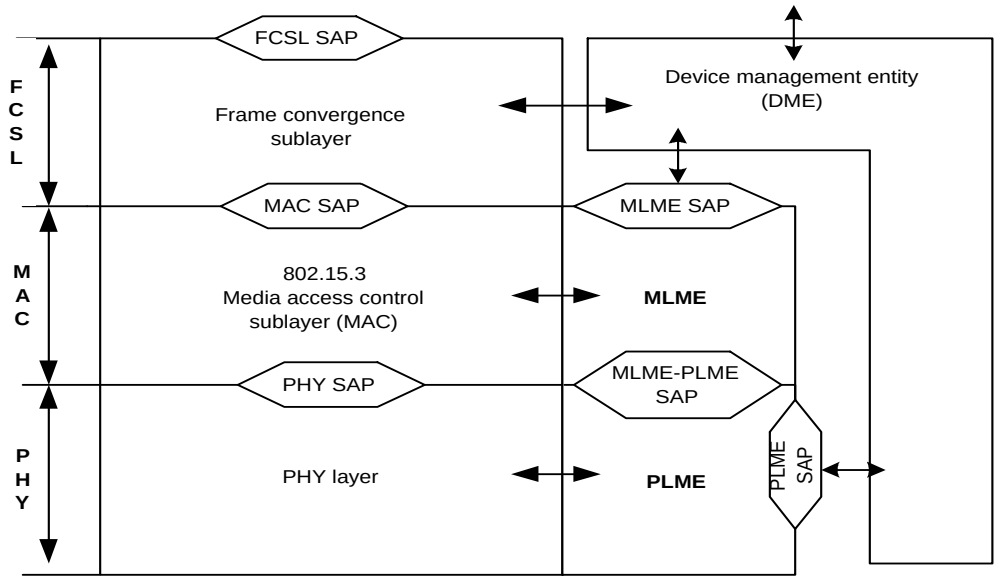

Figure A.2—802.15.3 protocol layer model

A.1.1 FCSL PDU classification

The FCSL PDU classification process maps each FCSL PDU to a specific stream index. Each stream index has associated with it a set of QoS characteristics. Consequently, after classification, each FCSL PDU will be delivered using the QoS parameters specified for the stream index. This includes asynchronous data which is mapped to the asynchronous stream index.

The classification process uses one or more classification parameter sets to analyze each frame entering the FCSL. In the case of an 802.2 classifier, the classification set includes a classification priority, stream index, and protocol specific parameters such as destination MAC address, source MAC address, and optionally a priority parameter. If an FCSL PDU, received from an upper layer protocol, matches the specified protocol specific parameters, it is then sent to the MAC-SAP for delivery using the stream indicated by the stream index. If the FCSL PDU does not match the specified protocol parameters, the frame may be delivered using either a default stream index (i.e. asynchronous stream index) or the frame may be discarded. The policy for deciding the method used to handle a frame in this instance is outside of the scope of this standard. Figure A.3 provides a graphical representation of the entities involved.

In the case where more than one classification parameters set is available, the classification process first shall use the classification parameters set containing the highest valued classification priority parameter. If no match is found with the first classification parameters set, the next highest priority classification parameters set will be applied. This process will repeat itself until either the incoming frame is properly matched and assigned to a specific stream index for subsequent delivery, or there are no more classification parameters sets available and the incoming frame is either discarded or delivered with a default delivery QoS (i.e. Best Effort).

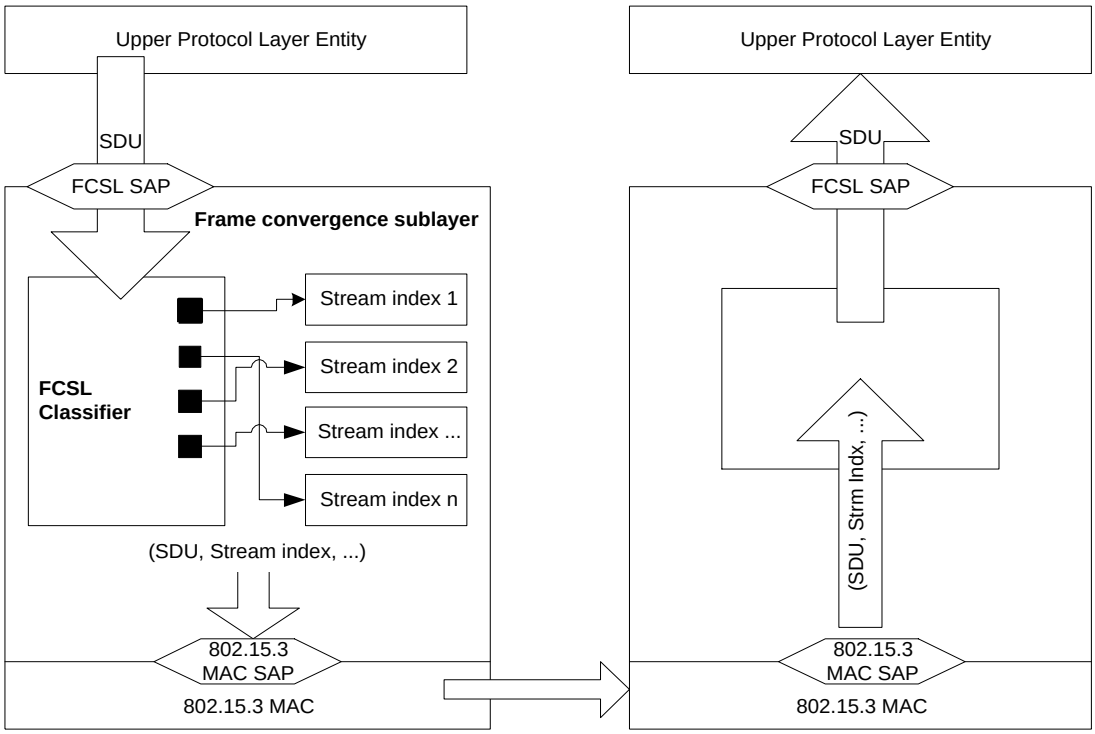

Figure A.3—Classifications and stream index mapping

A.1.2 IEEE 802.2 FCSL

The IEEE 802.2 FCSL shall:

- Receive upper layer 802.2 frame PDUs via the 802.2 FCSL SAP,
- Classify each received PDU according to these attributes:
  1) Destination address
  2) Source address
  3) Priority, an 802.1D™ hierarchical QoS scheme
- Map each received PDU to a specific StreamIndex according to the rules of the 802.2 FCSL classifier.
- Map each received PDU source and destination address to a corresponding 802.15.3 SrcID and DestID, by communicating in an unspecified manner with the DME which maintains this information.
- Deliver each valid frame SDU to the MAC-SAP.
- Receive frame SDUs from the MAC-SAP.
- Deliver received frame SDUs to the upper layer 802.2 via the 802.2 FCSL SAP.

A.1.2.1 IEEE 802.2 FCSL QoS support

The 802.2 FCSL shall support one or both of these QoS schemes:

a) Best Effort: This is the default QoS that shall be supported. All 802.2 PDUs are handled the same, i.e. no QoS guarantees are provided regarding delivery of the received PDU.
b) Hierarchical IEEE 802.1D QoS priority scheme.

The 802.1D priority scheme, which describes up to eight (0–7) differenct QoS levels, may be included in the rule set for the 802.2 FCSL classifier. Each of the 8 different QoS priorities are described in Table A.1.

Table A.1—IEEE 802.1p traffic types

User priority	Traffic type	Used for:	Comments
0 (default)	Best effort (BE)	Asynchronous data	Default piconet traffic
1	Background (BK)	Asynchronous data	
2		A spare	Currently not assigned
3	Excellent effort (EE)	Isochronous	For valued customers
4	Controlled load (CL)	Isochronous	Traffic will have to conform to some higher protocol layer admission control
5	Video (VI)	Isochronous	< 100 ms delay and jitter
6	Voice (VO)	Isochronous	< 10 ms delay and jitter
7	Network control (NC)		

A.1.2.2 Data entity inter-relationships

Figure A.4 illustrates the relationship among the SDU, classification parameters set, connection and PDU entities. These entities and the underlying protocol mechanisms that establish these entities and their relationship with each other are key to enabling support for QoS delivery of PDUs from one MAC entity to another.

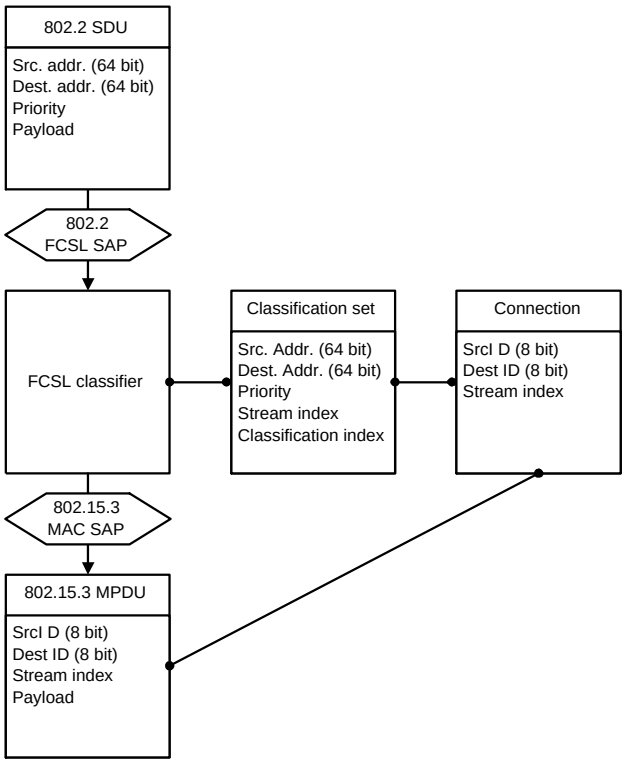

Figure A.4—Data entity inter-relationship model

A.2 802.2 FCSL SAP

The IEEE 802.15.3 MAC supports the following service primitives as defined in ISO/IEC 8802-2:

— MA-UNITDATA.request
— MA-UNITDATA.indication
— MA-UNITDATA-STATUS.indication

The LLC definitions of the primitives and the specific parameter value restrictions imposed by IEEE 802.15.3 are given in the following subclauses.

A.2.1 MA-UNITDATA.request

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 broadcast or multicast addresses. The parameters of the primitive are as follows:

```
MA-UNITDATA.request         (
                            SourceAddress,
                            DestinationAddress,
                            RoutingInformation,
                            Data,
                            Priority,
                            ServiceClass
                            )
```

The SourceAddress parameter specifies an individual MAC sublayer address of the sublayer entity from which the MSDU is being transferred.

The DestinationAddress parameter specifies either an individual, broadcast or multicast MAC sublayer entity address.

The RoutingInformation parameter specifies the route desired for the data transfer (a null value indicates that source routing is not to be used). For IEEE 802.15.3, the RoutingInformation parameter shall be null.

The Data parameter specifies the MSDU to be transmitted by the MAC sublayer entity. For the IEEE 802.15.3, the length of the MSDU shall be less than or equal to pMaxTransferUnitSize.

The Priority parameter specifies a prioritized QoS request for the MSDU transfer. IEEE 802.15.3 allows 8 integer values between 0 and 7 inclusive for directly indicating prioritized QoS.

The ServiceClass parameter specifies the service class desired for the data unit transfer. For 802.15.3 the ServiceClass parameter shall be null.

A.2.1.1 When generated

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

A.2.1.2 Effect of receipt

On receipt of this primitive the FCSL entity determines whether it is able to fulfill the request according to the requested parameters.

If the FCSL entity cannot fulfill the request according to the requested parameters, it discards the request and indicates the action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive which describes the reason for its inability to fulfill the request.

If the FCSL entity is able to fulfill the request according to the requested parameters, it appends all MAC specified fields that are unique to IEEE 802.15.3 to the data parameter, passes the properly formatted frame to the lower layers for transfer to peer FCSL entity or entities, and indicates the action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive with TransmissionStatus set to “successful,” as described in A.2.3.

A.2.2 MA-UNITDATA.indication

This primitive indicates the transfer of an MSDU from the FCSL entity to the LLC sublayer entity. 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. The semantics of the primitive are as follows:

```
MA-UNITDATA.indication      (
                            SourceAddress,
                            DestinationAddress,
                            RoutingInformation,
                            Data,
                            ReceptionStatus,
                            ServiceClass
                            )
```


The SourceAddress parameter is an individual address as specified by the mapping of the SrcID field of the incoming frame to the corresponding DEV address.

The DestinationAddress parameter is either an individual, broadcast or multicast address as specified by the mapping of the DestID field of the incoming frame to the corresponding DEV address.

The RoutingInformation parameter specifies the route that was used for the data transfer. For IEEE 802.15.3 this field shall be set to null.

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

The ReceptionStatus parameter indicates the success or failure of the received frame for those frames that IEEE 802.15.3 reports via an MA-UNITDATA.indication. This FCSL only reports success as all failures of reception are discarded without generating MA-UNITDATA.indication.

The ServiceClass parameter specifies the receive service class that was used for the data unit transfer. For 802.15.3 the ServiceClass parameter shall be null.

A.2.2.1 When generated

The MA-UNITDATA.indication primitive is passed from the FCSL entity to the LLC sublayer entity to indicate the arrival of a frame at the local FCSL entity. Frames are reported only if they are validly formatted, received without error, received with valid security properties according to the security policy at the local FCSL entity, and their destination address designates the local FCSL entity.

A.2.2.2 Effect of receipt

The effect of receipt of this primitive by the LLC sublayer is dependent on the validity and content of the frame.

A.2.3 MA-UNITDATA-STATUS.indication

This primitive has local significance and provides the LLC sublayer with status information for the immediately preceding MA-UNITDATA.request primitive. The semantics of the primitive are as follows:

```
MA-UNITDATA-STATUS.indication   (
                                SourceAddress,
                                DestinationAddress,
                                TransmissionStatus,
                                ProvidedPriority,
                                ProvidedServiceClass
                                )
```

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

The DestinationAddress is either an individual, broadcast or multicast MAC sublayer entity address as specified in the associated MA-UNITDATA.request primitive.

The TransmissionStatus parameter is used to pass status information back to the local requesting LLC sublayer entity. IEEE 802.15.3 specifies the following values for TransmissionStatus when delivery of the MSDU is attempted:

0—Successful
1—Excessive data length
2—Non-null source routing
3—Unsupported priority (for priorities other than an integer value between 0 and 7 inclusive)
4—Undeliverable (no piconet available)
5—Undeliverable (the local MAC sublayer entity does not have the required credentials or other security data to transmit the frame)
6—Undeliverable (channel conditions are too severe)

The ProvidedPriority parameter specifies the priority that was used for the associated data unit transfer as defined in A.2.1.

The ProvidedServiceClass shall be null for 802.15.3.

A.2.3.1 When generated

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

A.2.3.2 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.

Annex B

(informative)

Security considerations

B.1 Background assumptions

All security solutions rely on assumptions about DEVs and the capabilities of potential attackers to thwart possible threats. The goals of mode 1 security are that only authorized DEVs will be able to join a secure piconet and that communication is restricted to authorized DEVs.

B.1.1 Physical assumptions

The assumptions below are made about the physical environment for the piconet. The physical constraints help to determine the security architecture.

- **Open communications medium:** Since the data being transmitted will be able to be received by any other entity that is sufficiently close and has a sufficiently good receiver, it is assumed that transmissions are be heard by entities that are not part of the piconet.
- **Low cost:** Like all other components of a DEV, security is provided with careful attention to cost.
- **Dynamic group membership:** DEVs are expected to be mobile and it is therefore assumed that the DEVs enter or exit the network at any time.
- **No access to external networks:** Security solutions need to be effective without access to external networks.
- **Bandwidth:** Since 802.15.3 piconets provide high data rates, reasonable amounts of bandwidth overhead due to security are acceptable.
- **Computational power:** The DEVs are assumed to have very little computational power with only a small portion of that available for cryptographic computations.
- **Memory:** It is assumed that the low end DEVs implementing 802.15.3 will have little memory available for security.

B.1.2 Network assumptions

The assumptions below are made about the network structure of the piconet. The network constraints help to determine the security architecture.

- **Network size:** Although there is a fixed upper bound of fewer than 255 DEVs in a piconet, the security solution might need to scale to arbitrary sets of DEVs, rather than to a fixed set of limited size. DEVs join and leave the network in an ad-hoc fashion and in some cases will not have previously communicated with the other DEV(s).
- **Controller:** One DEV, the PNC, has the role of managing message control and entry into the piconet.
- **Dynamic controller:** The PNC is assumed to have the ability to leave the network or hand over the PNC role to other DEVs.
- **Device relationships:** The wide array of use cases describe multiple models for the pre-existing relationship of DEVs in the piconet. It is assumed that DEVs could have pre-existing security relationships or that they have never met and that both types of relationship could exist within a single piconet.

B.1.3 Attack model assumptions

In order to make statements about the effectiveness of security measures, it is necessary to describe the capabilities of the attackers and the nature of the attackers.

— **Computational capabilities:** It is assumed that the attacker has state of the art technologies to perform rapid computations.
— **Listening capabilities:** It is assumed that the attacker is within listening range of the DEVs in the piconet and understands the communication mechanism.
— **Broadcast capabilities:** It is assumed that the attacker has sophisticated broadcasting equipment that is able to synchronize with the piconet and transmit data for the DEVs in the piconet at the appropriate time.
— **Security setup:** The security setup for the DEVs occurs either before entry into the piconet or after the piconet has been established. No assumptions are made about the presence of attackers during security setup.

B.1.4 Security key lifecycle issues

B.1.4.1 Key lifecycle

In order to maintain security, care needs to be taken to protect keys from exposure for their entire lifetime. This standard provides the DEV-host the necessary methods for good key life cycle management. The requirements for key life cycle management depend on the type of application.

B.1.4.2 Membership lifecycle

The PNC or another DEV is able to require that each DEV with which it has a secure relationship periodically transmit a secure frame using the management key to be certain that the DEV is still in the piconet. If no secure frames are being transmitted by the target DEV, the PNC or requesting DEV is able send a secure Probe Request command requesting an IE from the target DEV. If the target DEV does not respond with a secure frame within a period of time determined by the PNC or requesting DEV, the PNC or requesting DEV will assume that the target DEV is no longer present and disassociate or terminate the secure relationship with the target DEV.

B.1.4.3 Group membership change rekey

Only DEVs that are currently members of the piconet are allowed to generate, read or modify piconet data. This implies that when a DEV joins or leaves the piconet, the currently active group keys need to be changed. Changes in the group membership key are described in 9.3.2.

B.2 Claimed security services

Each of the protocols defined in Clause 9 are designed to offer specific security services. These security services are consistent with the security services required by the 802.15.3 security model. Subclauses B.2.1 through B.2.4 describe the security services provided by each protocol and the method implemented to provide the security service.

B.2.1 Beacon protection protocol

Table B.1 specifies the security services provided by the beacon protection protocol specified in 9.4.2 along with a description of the method employed to provide the security service.

Table B.1—Beacon protection security services

Security service	Method provided
Communication of current time token to the DEVs in the piconet.	The PNC increments the time token for each superframe and protects it using the current group key. The integrity protection on the beacon and the storage of the previous time token allows each DEV to determine that the time token is fresh.
Indication of the identity of the PNC to the DEVs in the piconet.	If PNC handover has not occurred, the DEV address of the current PNC appears in the beacon. If PNC handover has occurred, the DEV address of the new PNC appears in the beacon. The integrity protection on the beacon and the freshness from the time token allow each DEV to determine the identity of the current PNC.

B.2.2 Distribute key protocol

Table B.2 specifies the security services provided by the distribute key protocol specified in 9.4.2 along with a description of the method employed to provide the security service.

Table B.2—Key distribution security services

Security service	Method provided
Privacy protection on distributed key.	The encryption of the key with the shared key encryption key ensures that the key remains private.
Integrity protection on the distributed key.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.
Verification by the key originator that the DEV received the key.	The key originator verifies that the integrity code verifies properly and that the freshness checks succeed.

B.2.3 Key request protocol

Table B.3 specifies the security services provided by the key request protocol specified in 9.4.3 along with a description of the method employed to provide the security service.

Table B.3—Key request security services

Security service	Method provided
Privacy protection on requested key.	The encryption of the key with the shared key encryption key ensures that the key remains private.
Integrity protection on the requested key.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.

B.2.4 Data protection protocol

Table B.4 specifies the security services provided by the data protection protocol specified in 10.2.2, along with a description of the method employed to provide the security service.

Table B.4—Data protection security services

Security service	Method provided
Privacy protection on the data.	The encryption of the data with the shared encryption key ensures that the key remains private.
Integrity protection on the data.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.

Annex C

(informative)

Coexistence, interoperability, and interference

C.1 Interoperability

The 802.15.3 standard does not require interoperability with any other IEEE 802® wireless standards or other wireless specifications. However, choices were made at the PHY layer to make it easier for implementers to be able to make low-cost, dual-mode radios. The IEEE 802 wireless protocols where dual-mode solutions are more easily created are:

— 802.11 DSSS and 802.11b
— 802.11 FHSS
— 802.15.1

For other protocols, no specific facilities were included to enhance interoperability. However, interoperability is not precluded and an implementor could make dual-mode radios with the following protocols:

— 802.11a
— 802.11 IR
— 802.16™

While it is possible to implement these interoperable radio modules, the details of the techniques used to accomplish this are out of the scope of this standarad and so are left to the implementer.

C.1.1 Interoperability with 802.11 DSSS and 802.11b

Since 802.11b [B3] is a superset of 802.11 DSSS, both will be referred to in this discussion as simply 802.11b. 802.11b and 802.15.3 share the same frequency band, which makes interoperability of radio modules much simpler. Also, the 802.15.3 PHY layer uses 11 Mbaud, DQPSK modulation for the base rate, which is the same as the chip rate and modulation for 802.11b. However, 802.11b uses either a Barker code, CCK or PBCC as a spreading code, which is not a part of the 802.15.3 standard.

The 802.15.3 PHY was also chosen with the same frequency accuracy, allowing the reuse of reference frequency source and frequency synthesizers. While the 802.11b and 802.15.3 frequency plans are slightly different, the synthesizers that would normally be used in either radio would be capable of 1 MHz frequency step size and so would be capable of supporting either frequency plan. The RX/TX turnaround time is also the same for both protocols. However, the TX/RX turnaround for 802.11b is 5 μs vs. 10 μs for 802.15.3, which could have an impact on the architecture of a dual-mode radio.

To summarize, the similarities between 802.15.3 and 802.11b are:

— DQPSK modulation
— 11 Mbaud symbol (chip) rate.
— Frequency and symbol timing accuracy of +/- 25 ppm.
— RX/TX turnaround time
— Power ramp up/down

Some of the differences include

— Barker, CCK or PBCC spreading code
— Power spectral density
— Frequency plan
— Performance criteria (e.g. sensitivity, jamming resistance, etc.)
— TX/RX turnaround time.
— PHY preamble, header, frame structure
— MAC

C.1.2 802.11 FHSS and 802.15.1

A narrow band, frequency hopping radio, as defined in 802.11 FHSS and 802.15.1, has many differences with the 802.15.3 radio. However, since these protocols share the same frequency, it is possible to architect dual-mode radios that would support a combination of these protocols. The design of dual-mode radios is outside of the scope of this standard.

C.2 Coexistence

The 802.15.3 standard provides many facilities that allow it to coexist with other wireless protocols. This subclause provides an overview of the methods that are defined in the standard.

C.2.1 Coexistence with 802.11b

The 802.15.3 PHY presents two challenges in coexisting with 802.11b

a) Both use the same frequency range
b) 802.11b uses CSMA/CA and a polling method with the point coordination function while 802.15.3 uses a hybrid CSMA/CA and TDMA.

802.15.3 piconets use two access methods in the superframe; CSMA/CA during the CAP and TDMA during the CTAP. The CAP provides the best method of coexistence with 802.11b networks, since the CSMA/CA algorithm used in the CAP is similar to the CSMA/CA algorithm used in 802.11b, i.e. the transmitter uses a listen-before-talk mechanism. In the case of 802.11, there is more than one CCA method allowed and some of them would not recognize an 802.15.3 frame. In this case, the 802.11b transmission might collide with 802.15.3 frames. However, an 802.11b station which implemented ‘energy above threshold’ for CCA, i.e. CCA mode 1 or CCA mode 5 (see IEEE Std 802.11b-1999 [B3]), would signal that the medium is busy when a sufficiently strong 802.15.3 signal is present. The 2.4 GHz PHY of 802.15.3 requires energy detection as a part of the CCA process. A sufficiently strong 802.11b signal would result in the 802.15.3 DEV signaling that the medium is busy, which would improve the coexistence performance.

CTAs provide the best QoS for 802.15.3 connections, but potentially will also cause the most coexistence problems with 802.11b products. This is because once a DEV has a CTA, the DEV transmits without using a listen-before-talk mechanism.

To address this issue, the 802.15.3 standard provides the following techniques to handle coexistence with 802.11b:

— passive scanning
— dynamic channel selection
— the ability to request channel quality information

— link quality and RSSI
— a channel plan that minimizes channel overlap
— lower transmit power
— transmit power control
— neighbor piconet capability

C.2.1.1 Passive scanning

All 802.15.3 PNC capable DEVs (i.e. ACs) are required to passively scan, as described in 8.2.1, a potential channel before attempting to start a piconet, as described in 8.2.2. While detecting an 802.11b WLAN is not required, the PNC capable DEV will, at a minimum, be looking for a channel that is relatgively quiet. Passive scanning implies that the PNC capable DEV, when starting a piconet, or other DEVs that wish to join an existing piconet will not cause inteference while searching the channels.

C.2.1.2 Dynamic channel selection

The PNC will periodically request channel status information, as described in 8.9.4, from the DEVs in the piconet via the Channel Status Request command, as described in 7.5.7.1. If the PNC determines, from the number of lost frames, that the channel is having problems, as it would when an 802.11b network is present, then it would search for a new channel, as described in 8.11.1, that had a lower level of interference. If the PNC finds a channel with less interference then the PNC uses the Piconet Parameter Change IE in the beacon, as described in 7.4.6, to move the piconet to a quieter channel.

Thus, if an 802.11b network is present, the 802.15.3 piconet would change channels to avoid interfering with 802.11b.

C.2.1.3 The ability to request channel quality information

Dynamic channel selection, as described in 8.11.1, requires the ability to obtain an estimate of the interference in a channel. In the case of 802.15.3, not only does the DEV sense the channel in its area, but it is also capable of asking any other DEV to respond with its own estimate of the channel status, as described in 8.9.4. These commands indicate the frame error rate at a remote DEV. This command is useful for detecting coexistence problems in remote DEVs by the PNC or other DEVs that are unable to detect an interference environment (for example during a passive scan).

C.2.1.4 Link quality and RSSI

The 2.4 GHz PHY specifies that a DEV returns the RSSI, as described in 11.6.6, and for the higher speed modulations, an estimate of the link quality, as described in 11.6.7. The RSSI provides an estimate of the strength of the received signal, which is useful for transmit power control. The RSSI combined with the link quality indication, LQI, provides a method to differentiate between low signal power and interference causing the loss of frames. For example, if the RSSI is low and frames are being lost, then the cause is low receive power. On the other hand, if the RSSI is relatively high, but the LQI is low, that would indicate the possibility of interference in the channel.

C.2.1.5 Channel plan that minimizes channel overlap

The channel plan for the 2.4 GHz PHY, as described in 11.2.3, balances the requirement of four simultaneous piconets with the desire to coexist with other wireless standards, such as 802.11b. To do this, two channel plans are available. If there are no 802.11b networks detected, then the high density channel plan would be used. On the other hand, if 802.11b networks are detected, then the PNC would want to choose the 802.11b coexistence channel plan. The reason for this is that each of the two center channels in the high-density channel plan would overlap 2 802.11b channels. The 802.11b coexistence channel plan roughly aligns

the channels so that 802.15.3 operation in one of these channels would only affect a single 802.11b channel. By choosing the “quietest” one, the 802.15.3 piconet would minimize its impact on the 802.11b networks.

If an 802.15.3 PNC has selected the center channel of the 802.11b coexistence channel plan, then other 802.15.3 piconets that entered or started in the same operational area would also adopt the 802.11b coexistence channel plan, i.e. the piconets would use one of the two remaining outer channels, either channel 1 or channel 5. This would only occur when the center channel, channel 3, is occupied first. Otherwise, subsequent piconets in the same operational area would be able to use the high density channel plan.

C.2.1.6 Lower transmit power

The 802.15.3 standard operates in the United States under the 47 CFR 15.249 rules, as described in 11.1, often called the “low power” rules. Under this section of Part 15, the maximum allowed transmit power is approximately 8 dBm EIRP for the 802.15.3, 2.4 GHz PHY. This measurement includes the antenna gain, so a 1 dB increase antenna gain requires a 1 dB decrease in transmit power. 802.11b WLANs, on the other hand, operate under 47 CFR 15.247, which allow up to 1 W of transmit power with as much as 6 dB of antenna gain at that power level. Most 802.11b products on the market at this time operate with between 12 and 18 dBm of transmit power, with unspecified antenna gain. Access points often have antennas with moderate gain, 3–6 dB.

This implies that a high power 802.15.3 implementation would operate with 7 to 13 dB less effective transmit power than a typcial 802.11b implementation.

C.2.1.7 Transmit power control

The 802.15.3 standard provides three methods for controlling transmit power. The first is that the PNC is able to set a maximum power level for the beacon, CAP and directed MCTAs. For the 2.4 GHz PHY, the lowest setting possible is 0 dBm. This allows 802.15.3 piconets to reduce the interference it creates for other networks while maintaining the operation of the piconet.

Individual DEVs in a CTA are able to request a change in the transmit power of the remote DEV for that link, as described in 8.11.2.2. The originating DEV sends a message to the target DEV that tells the target DEV the amount to change its power up or down, depending on the status of the link. Thus, two DEVs that are relatively close to each other are able to both save power and reduce the inteference to other networks while maintaining a high quality link.

DEVs are also able to change their transmit power based on their own estimation of the channel. The Probe Request command, as described in 8.9.2, allows DEVs to request information from other DEVs in the piconet to assist in getting this information.

C.2.1.8 Neighbor piconet capability

The neighbor piconet capability, as described in 8.2.6, allows a DEV, which may not be fully 802.15.3 compliant, to request time to operate a network that is co-located in frequency with the 802.15.3 network. While current 802.11b radios do not implement this functionality, it is relatively easy to build 802.11b radios that could support enough of a subset of 802.15.3 to request the neighbor piconet capability. One reason that it is possible to do this is that the 802.15.3 PHY has characteristics that make it easier to build dual-mode radios (see C.1.1 for more information).

Once a dual-mode 802.11b/802.15.3 access point has requested and received a CTA for a neighbor network, it would use the PCF and NAV to set aside time for the operation of the 802.15.3 piconet, while maintaining clear operation for part of the time for the 802.11b WLAN.

C.2.2 Coexistence with 802.15.1 and 802.11 FHSS

Narrow band FHSS systems have very different coexistence requirements than 802.11b, which has a PHY similar to 802.15.3. For these systems, there are methods in the standard as well as methods that are implementation dependent, which enhance the coexistence peformance. Within the standard, the following methods are defined which improve the coexistence with FHSS systems:

a) lower transmit power, as described in C.2.1.6

b) transmit power control, as described in C.2.1.7

Techniques to manage coexistence between 802.11b and narrowband interference such as 802.15.1 are addressed by IEEE Std 802.15.2™-2003 [B4], which is a recommended practice for coexistence. While some sections of the document are only relevant for 802.15.1 systems, portions of the 802.15.2 recommended practice are directly relevant to the 802.15.3 PHY, and some of the recommended practices for the MAC will also improve performance in a coexistence environment. It is up to the implementer to decide the methods that are included in their implementation.

The terminology in 802.15.2 describes both collaborative and non-collaborative coexistence mechanisms. Collaborative mechanisms are those that would require direct communication between the 802.15.3 system and the FHSS system it desires to coexist with. By the definition in 802.15.2, collaborative systems are collocated within 0.5 m of each other. Non-collaborative systems, which do not have to be collocated, are those that modify their PHY or MAC behavior by inferring the interference environment, not by direct communication between the systems.

C.2.2.1 PHY colloaborative coexistence enhancements

In addition, certain PHY mechanisms, as described in 802.15.2, are available to improve coexistence. In particular, deterministic frequency excision (variable notch filter) is a collaborative mechanism that is able to excise 802.15.1 signals from the 802.15.3 signal before detection. In this case, the nulling of the narrowband signal is done at either IF or complex baseband. In the case of 802.15.1 interference, the jamming tones occur at predetermined channels and times since the 802.15.3 DEV is assumed to have knowledge of the 802.15.1 piconet's future behavior. This mechanism and the algorithms needed to implement it are described in the 802.15.2 recommended practice.

C.2.2.2 MAC coexistence via collaboration

IEEE Std 802.15.2-2003 [B4] also describes two collaborative techniques that involve MAC sublayer coordination:

a) The packet traffic arbitrator method uses shared knowledge of the current traffic in the 802.15.3 and the 802.15.1 piconets as well as future 802.15.1 frequency utilization, frame type, and frame priority to arbitrate access to the medium. This arbitration occurs on a frame-by-frame basis.

b) The alternating wireless medium access method alternates access to the medium by 802.15.1 and 802.15.3 members based on a timing criterion; the 802.15.3 piconet is restricted to activity during a portion of the beacon interval, while the 802.15.1 piconet is active during the remainder.

Either of these optional techniques would enhance the coexistence of collocated 802.15.3 and 802.15.1 systems. The impact of either method on overall system performance is outside of the scope of this standard.

C.2.2.3 Other techniques

Due to the popularity of 802.11b and 802.15.1, many methods have been proposed to improve the coexistence of these two systems. As with the methods published in the 802.15.2 recommended practice, some of these methods are able to dramatically improve the coexistence of the two piconets. Since these methods are outside of the scope of this standard, they are not listed in this document. However, the existence of these techniques shows that similar innovative techniques might be used in the case of 802.15.1 and 802.15.3.

C.3 Coexistence performance

C.3.1 Allowed operation

The following IEEE wireless protocols are always allowed to operate in an operational area that overlaps with the operational area of an 802.15.3 piconet due to the fact that they use different frequency bands:

a) 802.11a
b) 802.11 IR
c) 802.16

The following IEEE wireless protocols are allowed to operate in an operational area that overlaps with the operational area of an 802.15.3 piconet, but could experience reduced throughput:

a) 802.11 DSSS
b) 802.11 FHSS
c) 802.11b
d) 802.15.1

The 802.15.3 network has mechanisms that allow coexistence with other overlapping IEEE wireless networks and so overlapping operation with any of the IEEE wireless networks is not prohibited. However, as noted above, with some networks, the throughput could be reduced and so under certain conditions, overlapping operation might be undesireable.

C.3.2 Assumptions for coexistence calculations

For the calculations used to determine the level of coexistence, the following assumptions have been made:

a) The sensitivity for each of the receivers is the reference sensitivity given in each of the standards, i.e.
   1) –70 dBm for 802.15.1
   2) –76 dBm for 802.11b 11 Mb/s CCK
   3) –75 dBm for 802.15.3 22 Mb/s DQPSK
b) The received power at the desired receiver is 10 dB above the receiver sensitivity. The level of 10dB was selected because a 10 dB margin results in 10% FER in a Raleigh fading channel. Reliable communications without interference would require at least this margin. The distance between the desired transmitter and receiver is not directly specified by this requirement, instead the transmitter power and the channel model listed below would be used to determine the resulting distance.
c) The transmitter power for each of the protocols is
   1) 0 dBm for 802.15.1
   2) +14 dBm for 802.11b
   3) +8 dBm for 802.15.3
d) The channel model is one that was proposed for 802.11 and used by 802.15.2.

$$d = 10^{\frac{(Pt - Pr - 40.2)}{20}} \quad \text{for d < 8 m} \qquad \text{(C1)}$$

$$d = 8 \times 10^{\frac{(Pt - Pr - 58.5)}{33}} \quad \text{for d > 8 m} \qquad \text{(C2)}$$

e) The receiver bandwidths are based on the requirements in the standard
   1) 1 MHz for 802.15.1
   2) 22 MHz for 802.11b
   3) 15 MHz for 802.15.3

f) The transmitter spectral masks are the maximum allowed in the standards. This is a very pessimistic assumption since the transmitter spectrum will generally be significantly lower than the spectral mask over most of the frequencies. There are usually only narrow peaks that come close to the required limits. The subclauses that define the transmitter spectral mask for the three standards are:
   1) Subclause 7.2.3.1 for 802.15.1
   2) Subclause 18.4.7.3 for 802.11b
   3) Subclause 11.5.3 for 802.15.3

g) The energy from the interfering signal affects the desired signal in a manner equivalent to additive white Gaussian noise (AWGN) in the same bandwidth.

h) The 802.15.3 piconet operates with the 802.11b coexistence channel plan as described in 11.2.3.

The BER calculations were calculated using the analytical model from IEEE Std 802.15.2-2003 [B4]. The calculation follows the approach outlined in C.3.2 of IEEE Std 802.15.2-2003, and the conversion from SNR to BER uses the formulas in C.3.6 of IEEE Std 802.15.2-2003. Figure C.1 illustrates the relationship between BER and SNR for 802.11b, 802.15.3 base rate and 802.15.1.

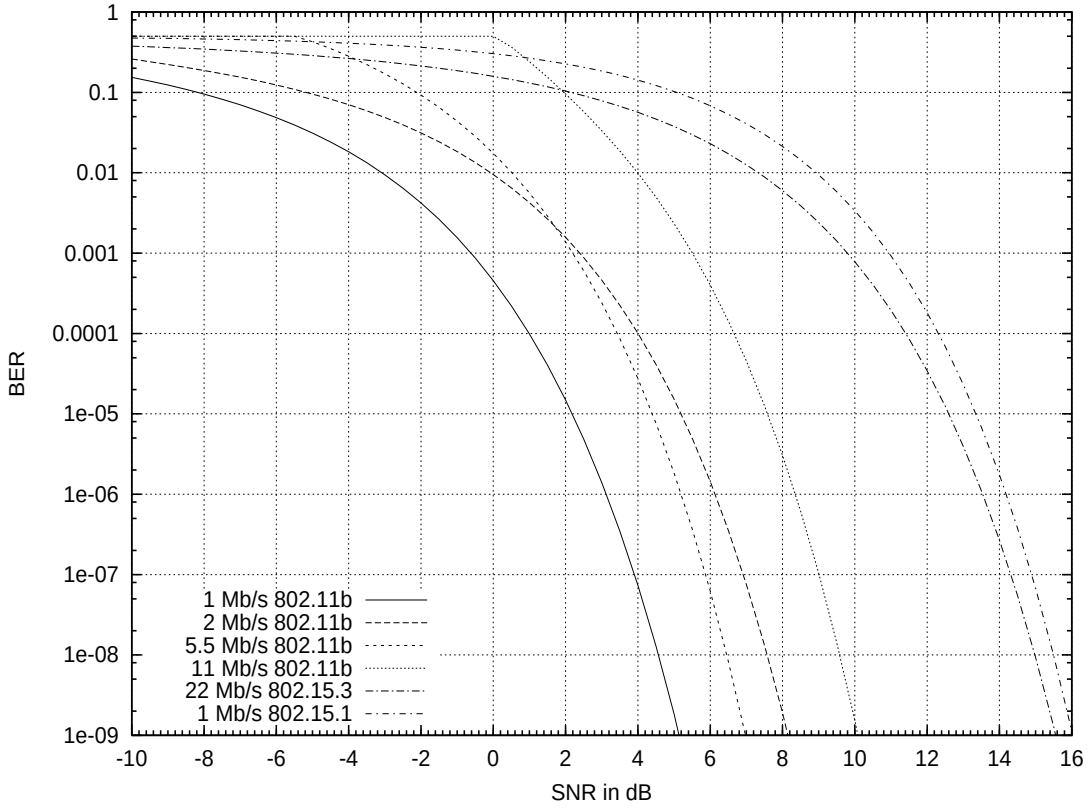

Figure C.1—BER results for 802.15.3, 802.11b and 802.15.1

For this analysis, the SNR is used instead of the signal to interference ratio. One reason is that 802.11b and 802.15.3 only specify the sensitivity with respect to noise. 802.15.1 specifies the receiver performance with respect to an interferer, but only for an 802.15.1 interferer. The performance of one of these system with respect to a specific interference source depends on the actual implementation of the receiver. The approximation that the interferring signal is equivalent to AWGN is sufficient for the purposes of this analysis.

C.3.3 Peformance impact on 802.15.3 piconets

This subclause provides an estimate of the performance impact of other IEEE 802 wireless networks on the operation 802.15.3 piconets. Any evaluation of the performance impact depends greatly on the assumed channel model, the physical distribution of DEVs in the network and the traffic pattern. Because of this, the IEEE 802.15.3 working group adopted a simple model to provide an estimate of the performance impacts. This model only takes into account the PHY parameters of the system to estimate the reduction in the FER for overlapping systems.

There is no peformance degredation anticipated from the overlapping operation of 802.11 IR, 802.11a, or 802.16.

The peformance degredation of overlapping operation with 802.11b, 802.11 FHSS, and 802.15.1 is addressed in C.3.3.1 and C.3.3.2.

C.3.3.1 802.11b overlapping with 802.15.3

When an 802.11b network operational area overlaps with the operational area of an 802.15.3 piconet, the 802.15.3 piconet will experience reduced throughput. When the PNC notices this throughput reduction by the responses to the Channel Status Request command, as described in 8.9.4, it would then search for a better channel to operate on. The PHY layer rates the channels, best to worst (from lowest to highest interference, as described in 8.2.1), and would choose a new channel to operate on that has the least interference.

If the 802.15.3 DEV is able to postively detect the presence of 802.11b networks, it will adopt the 802.11b coexistence channel plan, as described in 11.2.3, to minimize the overlap of selected channels with 802.11b channels. It would also then rate as worst any channels that it finds contains 802.11b networks, as described in 11.2.4. Thus, the PNC would choose a new channel that is not used by the overlapping 802.11b network.

Note that if the 802.11b AP supported both the ability to request neighbor piconet status, as described in 8.2.6, and the ability to peform the PCF function, then that AP would be able to negotiate with the 802.15.3 piconet to share time in the overlapping medium. This capability is not specified in the 802.11 standard and so providing this capability would require an AP that had additional functionality that is outside of the current 802.11 standard. IEEE Std 802.15.2-2003 [B4] has proposed an information element and extra functionality that, if added to the 802.11 standard, would make it possible to build a standards-compliant AP that could then support the 802.15.3 neighbor piconet capability.

Using the modeling assumptions stated in C.3.2, the degradation of the FER of an 802.15.3 piconet in the presence of an 802.11b WLAN is illustrated in Figure C.2. In the graph, the adjacent channel is 25 MHz away while the alternate channel is 49 MHz separation.

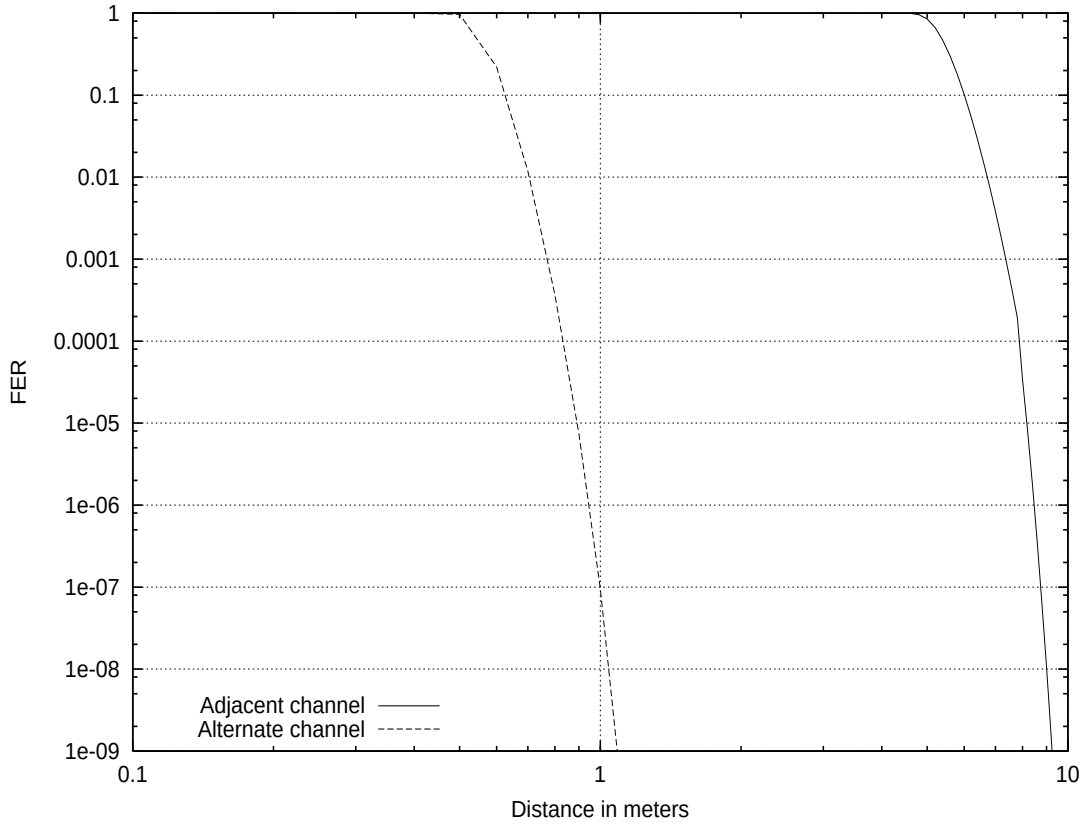

Figure C.2—FER results for 802.15.3 with 802.11b as the interferer

The adjacent and alternate channels for 802.15.3 as the receiver with 802.11b as the interferer are given in Table C.1

Table C.1—Adjacent and alternate channels for 802.15.3 as receiver

802.15.3 channel	Adjacent 802.11b channel	Alternate 802.11b channel
1	6	11
3	1, 11	none
5	6	1

In the graph of Figure C.2, there is almost no effect when the 802.11b network is in the alternate channel. When the 802.11b network is in the adjacent channel, however, the performance is noticeably impacted up to a distance of 6 m. There are two reasons for this. The first reason that 802.11b uses higher transmit power than 802.15.3 and so it affects the 802.15.3 network to a greater degree. The second reason is that the 802.11b spectral mask is wider than 802.15.3, so that the impact in the adjacent channel is much greater.

The impact of a 802.15.3 network on the FER of an 802.11b network is illustrated in Figure C.3 for the adjacent channel and in Figure C.4 for the alternate channel.

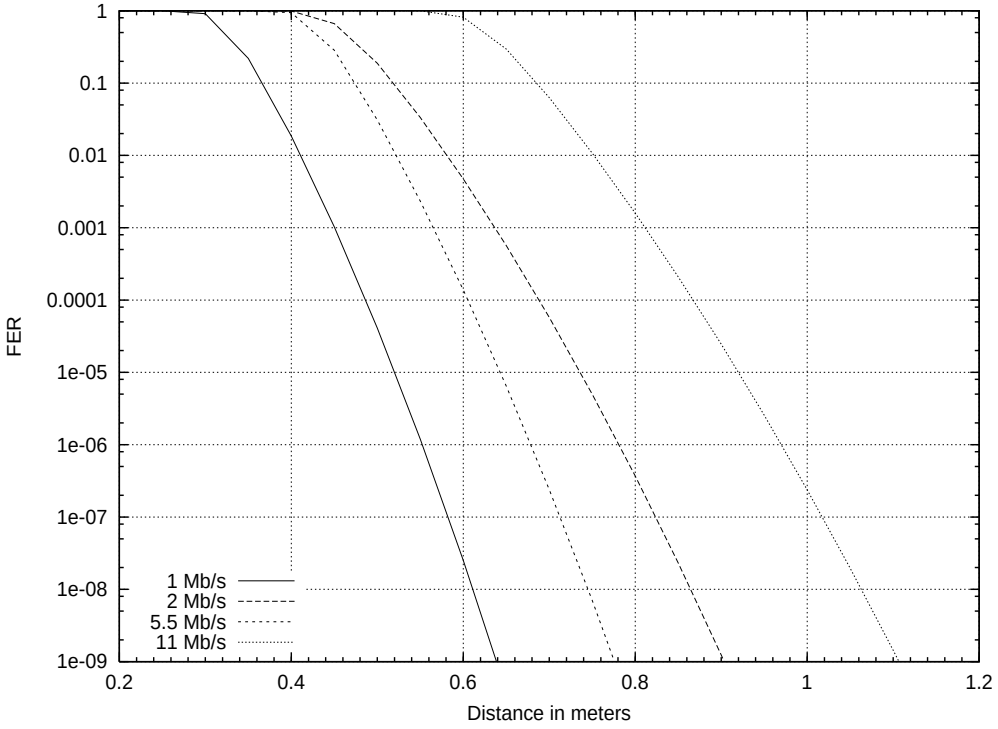

Figure C.3—FER results for 802.11b with 802.15.3 as the interferer in the adjacent channel

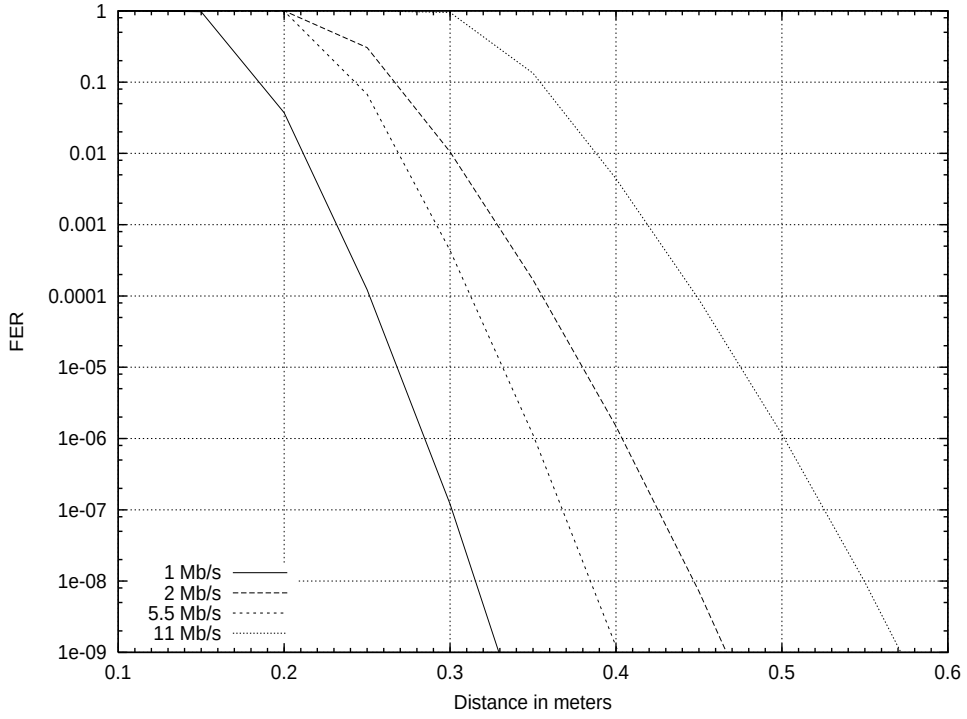

Figure C.4—FER results for 802.11b with 802.15.3 as the interferer in the alternate channel

The adjacent and alternate channel numbers for 802.11b as the receiver with 802.15.3 as the interferer are given in Table C.2.

Table C.2—Adjacent and alternate channels for 802.11b as receiver

802.11b channel	Adjacent 802.15.3 channel	Alternate 802.15.3 channel
1	3	5
6	1, 5	none
11	3	1

The 802.15.3 network has very little effect on the 802.11b network. The 802.15.3 piconet causes less than 10% FER at a distance of just 0.7 m even when it is in the adjacent channel. Part of the reason for this is that the 802.15.3 piconet operates at a lower transmitter power. Also, there is not much difference between the FER results for the adjacent and alternate channels. This is due to the much more stringent transmitter mask that is required for 802.15.3 than for 802.11b.

If the 802.15.3 piconet is occupies the same channel as the 802.11b network, the peformance degradation is much worse since the interference is within the passband of either radio. The results for cochannel interference are show in Figure C.5.

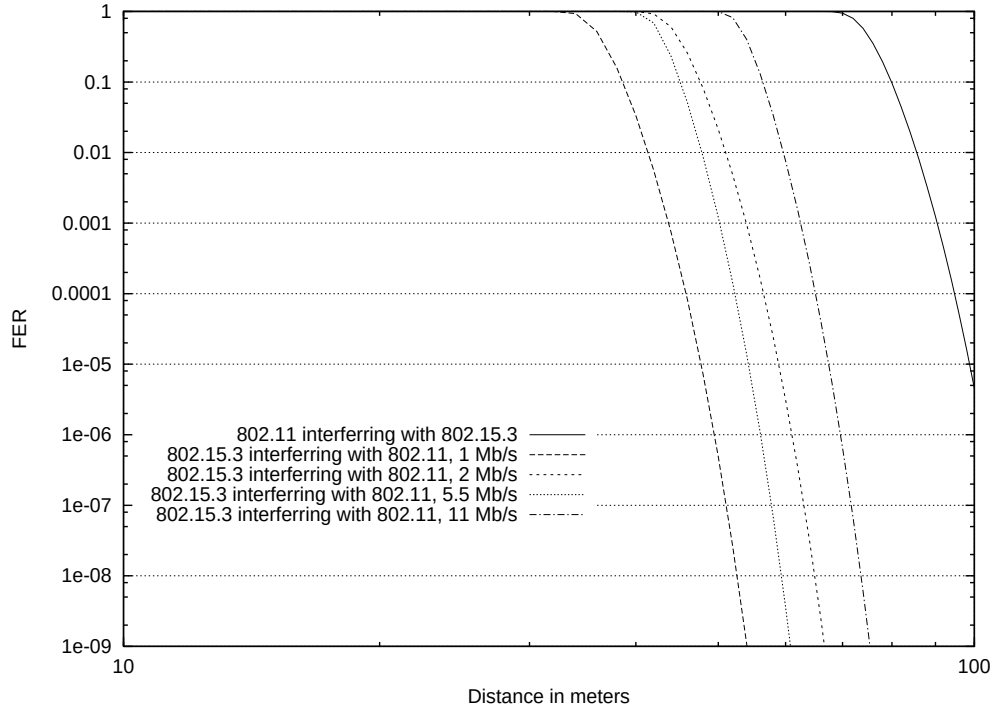

Figure C.5—FER results for 802.11b and 802.15.3 co-channel interference

In this case, both systems suffer significant interference at shorter distance and only achieve reasonable throughputs when the separatation exceeds 40 m for 802.11 and 80 m for 802.15.3. Similar results are obtained for 802.11/802.11 and 802.15.3/802.15.3 co-channel interference. These results show the importance of dynamic channel selection capability in the 802.15.3 protocol.

C.3.3.2 802.15.1 and 802.11 FHSS overlapping with 802.15.3

In the case of an 802.15.1 piconet sharing the operational space with an 802.15.3 piconet, the analysis is a little different. Since the 802.15.1 piconet hops frequencies, the impact on the 802.15.3 FER will depend on the current hop frequency. Overall, the average FER for either network will be an average of the FER for each of the channels.

802.11 FHSS systems will have a similar peformance impact as 802.15.1 piconets on 802.15.3 piconets, so analysis in this section will apply to that situation as well. Thus, in this subclause, only 802.15.1 piconets are discussed.

The average FER with 802.15.1 as the receiver and 802.15.3 as the interferer ("802.15.1 receiving") and with 802.15.3 as the receiver and 802.15.1 as the interferer ("802.15.3 receiving") is shown in Figure C.6.

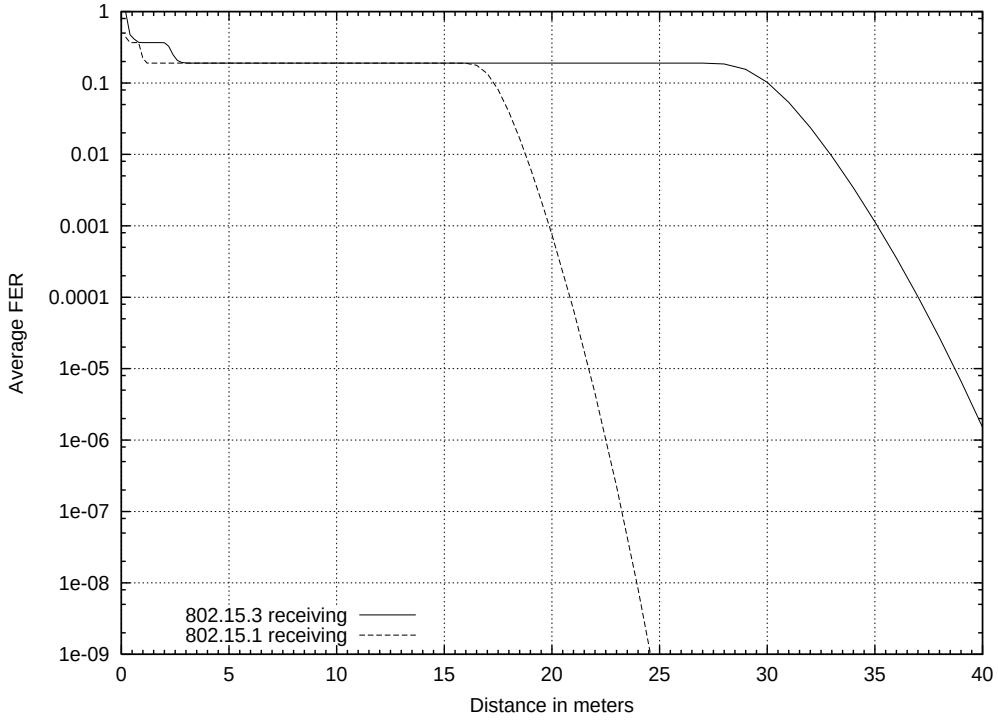

Figure C.6—Average FER results for 802.15.1 and 802.15.3

The flat part of the FER curve is due to the times when the 802.15.1 hop frequency is in the 802.15.3 pass band. Since there is no attenuation of the signal due to channel filtering, these frequencies have an FER of essentially 1. The average FER then works out to be 15/79 = 0.19, as shown in Figure C.6.

The analysis in this section assumes that the 802.15.3 receiver does not use notch filtering or other equalization techniques to minimize the impact of narrow band interferers. If the implementation uses any of the recommended practices listed in C.2.2.1, the performance of the 802.15.3 piconet in the presence of an 802.15.1 interferer would see a large improvement.

C.4 Notes on the calculations

The calculations for this annex are based on the formulas and descriptions from IEEE Std 802.15.2-2003 [B4].

All source files, including spreadsheets and graphs, will be archived in the IEEE Standards Code and Electronic Forms Index at http://standards.ieee.org/reading/ieee/std/downloads/index.html under “Calculations and Simulations.”

Annex D

(normative)

Protocol implementation conformance statement (PICS) proforma

D.1 Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given standard. Such a statement is called a protocol implementation conformance statement (PICS).

D.1.1 Scope

This annex provides the PICS proforma for IEEE Std 802.15.3-2003™ in compliance with the relevant requirements, and in accordance with the relevant guidance, given in ISO/IEC 9646-7 [B6].

D.1.2 Purpose

The supplier of a protocol implementation claiming to conform to this standard shall complete the following PICS proforma and accompanies it with the information necessary to identify fully both the supplier and the implementation.

The PICS of a protocol implementation is a statement of which capabilities and options of the protocol have been implemented. The statement is in the form of answers to a set of questions in the PICS proforma. The questions in a proforma consist of a systematic list of protocol capabilities and options as well as their implementation requirements. The implementation requirement indicates whether implementation of a capability is mandatory, optional, or conditional, depending on options selected. When a protocol implementor answers questions in a PICS proforma, the implementor would indicate whether an item is implemented or not, and provide explanations if an item is not implemented.

D.2 Abbreviations and special symbols

Notations for requirement status:

M	Mandatory
O	Optional
O.n	Optional, but support of at least one of the group of options labeled O.n is required.
N/A	Not applicable
“item”	Conditional, status dependent upon the support marked for the “item.”

For example, FD1: M indicates that the status is mandatory if the protocol feature item, FD1, is implemented.

D.3 Instructions for completing the PICS proforma

If it is claimed to conform to this standard, the actual PICS proforma to be filled in by a supplier shall be technically equivalent to the text of the PICS proforma in this annex, and shall preserve the numbering and naming and the ordering of the PICS proforma.

A PICS which conforms to this annex shall be a conforming PICS proforma completed in accordance with the instructions for completion given in this annex.

The main part of the PICS is a fixed-format questionnaire, divided into five tables. Answers to the questionnaire are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (such as Yes or No), or by entering a value or a set or range of values.

D.4 Identification of the implementation

*Implementation Under Test (IUT) Identification

IUT Name: _____

IUT Version: _____

_____

*System Under Test (SUT) Identification

SUT Name: _____

Hardware Configuration: _____

_____

_____

Operating System: _____

*Product Supplier

Name: _____

Address: _____

_____

Telephone Number: _____

Facsimile Number: _____

Email Address: _____

Additional Information: _____

*Client

Name: _____

Address: _____

_____

Telephone Number: _____

Facsimile Number: _____

Email Address: _____

Additional Information: _____

*PICS Contact Person

Name: _____

Address: _____

_____

Telephone Number: _____

Facsimile Number: _____

Email Address: _____

Additional Information: _____

*PICS/System Conformance Statement

Provide the relationship of the PICS with the System Conformance Statement for the system: _____

_____

_____

D.5 Identification of the protocol

This PICS proforma applies to IEEE Std 802.15.3-2003.

D.6 Global statement of conformance

The implementation described in this PICS proforma meets all of the mandatory requirements of the referenced standard.

[ ] Yes

[ ] No

NOTE—Answering "No" indicates non-conformance to the specified protocol standard. Non-supported mandatory capabilities are to be identified in Tables D.1 through D.5, with an explanation by the implementor on why the implementation is non-conforming.

The supplier will have fully complied with the requirements for a statement of conformance by completing the statement contained in this subclause. However, the supplier may find it helpful to continue to complete the detailed tabulations in D.7.

D.7 PICS proforma—IEEE Std. 802.15.3-2003[12]

Tables D.1 through D.5 are composed of the detailed questions to be answered, which make up the PICS proforma. Subclause D.7.1 contains the major roles for an IEEE Std 802.15.3 DEV. Subclause D.7.2 contains the major capabilities for the physical layer and radio frequencies. Subclause D.7.3 contains the major capabilities for the MAC sublayer. Subclause D.7.4 indicates which level and type of security is supported in the implementation.

D.7.1 Major roles for IEEE 802.15.3 DEVs

Table D.1—Functional DEV types

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
FD1	Is this entity DEV capable?		M			
FD2	Is this DEV PNC capable?	8.2.8	O			

[12]Copyright release for PICS proforma: Users of this standard may freely reproduce the PICS proforma in this annex to use it for its intended purpose and may further publish the completed PICS.

D.7.2 PHY functions

Table D.2—PHY functions

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
PLF1	Conforms to general requirements (i.e. timing, freqency, etc.)	11.2	M			
PLF1.1	Able to dectect 802.11b networks	11.2.4	O			
PLF2	Supports 22 Mb/s DQPSK modulation	11.3.1, 11.3.2, 11.3.3	M			
PLF2.1	Supports 33, 22 and 11 Mb/s modulations	11.3.2, 11.3.4	O			
PLF2.2	Supports 44, 33, 22 and 11 Mb/s modulations	11.3.2, 11.3.4	O			
PLF2.3	Supports 55, 44, 33, 22 and 11 Mb/s modulations	11.3.2, 11.3.4	O			
PLF3	Encodes and decodes PHY frame format	11.4	M			
PLF4	Conforms to transmitter requirments	11.5	M			
PLF5	Conforms to receiver requirements	11.6	M			
PLF5.1	Supports link quality assessment	11.6.7	PLF2.1: M, PLF2.2: M, PLF2.3: M,			
PLF6	PHY PIB values supported	11.7	M			
PLF7	CAP mandatory	11.2.10	M			

D.7.3 Major capabilities for the MAC sublayer

D.7.3.1 MAC frames

Table D.3—MAC frames

Item Number	Item Description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF1	General Frame Format	7.2	M		M	
MF2	**Frame Types**					
MF2.1	Non-secure beacon	7.3.1.1	FD2: M		M	
MF2.3	Secure beacon	7.3.1.2	FD2: O S2 & FD2: M		O S2: M	
MF2.4	Imm-ACK	7.3.2.1	M		M	
MF2.5	Dly-ACK	7.3.2.2	O		O	
MF2.6	Non-secure command	7.3.3.1	M		M	
MF2.7	Secure command	7.3.3.2	O S2: M		O S2: M	
MF2.8	Non-secure data	7.3.4.1	M		M	
MF2.9	Secure data	7.3.4.2	O S2: M		O S2: M	
MF3	**Information Elements**					
MF3.1	Channel time allocation	7.4.1	FD2: M		M	
MF3.2	BSID	7.4.2	FD2: M		M	
MF3.3	Parent piconet	7.4.3	FD2: O		O	
MF3.4	DEV association	7.4.4	FD2: M		M	
MF3.5	PNC shutdown	7.4.5	FD2: M		M	
MF3.6	Piconet parameter change	7.4.6	FD2: M		M	
MF3.7	Application specific	7.4.7	O		O	
MF3.8	Pending channel time map (PCTM)	7.4.8	FD2: O		O	
MF3.9	PNC handover	7.4.9	FD2: M		M	
MF3.10	CTA status	7.4.10	FD2: M		M	
MF3.11	Capability	7.4.11	M		M	
MF3.12	Transmit power parameters	7.4.12, 8.11.2.2	M		O	
MF3.13	PS status	7.4.13	FD2: M		O	

Table D.3—MAC frames (*Continued*)

Item Number	Item Description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF3.14	Continued wake beacon (CWB)	7.4.14	O		O	
MF3.15	Overlapping PNID	7.4.15	S2: M		S2: M	
MF3.16	Piconet services	7.4.16	O		O	
MF3.17	Vendor specific	7.4.17	O		O	
MF4	**Command Types**					
MF4.1	AssociationRequest	7.5.1.1	M		FD2: M	
MF4.2	Association Response	7.5.1.2	FD2: M		M	
MF4.3	Disassociation Request	7.5.1.3	M		M	
MF4.4	Request Key	7.5.2.1	S2: M		S3: M	
MF4.5	Request Key Response	7.5.2.2	S3: M		S2: M	
MF4.6	Distribute key request	7.5.2.3	S3: M		S2: M	
MF4.7	Distribute key response	7.5.2.4	S2: M		S3: M	
MF4.8	PNC handover request	7.5.3.1	FD2: M		FD2: M	
MF4.9	PNC handover response	7.5.3.2	FD2: M		FD2: M	
MF4.10	PNC handover information	7.5.3.3	FD2: M		FD2: M	
MF4.11	PNC information request	7.5.4.1	M		FD2: M	
MF4.12	PNC information	7.5.4.2	FD2: M		M	
MF4.13	Security information request	7.5.4.3	O		O	
MF4.14	Security information	7.5.4.4	O		O	
MF4.15	Probe request	7.5.4.5	M		M	
MF4.16	Probe response	7.5.4.6	M		M	
MF4.17	Piconet services	7.5.5.1	FD2: O		O	
MF4.18	Announce	7.5.5.2	FD2: O		O	
MF4.19	Channel time request	7.5.6.1	M		FD2: M	
MF4.20	Channel time response	7.5.6.2	FD2: M		M	
MF4.21	Channel status request	7.5.7.1	O		M	
MF4.22	Channel status response	7.5.7.2	M		O	
MF4.23	Remote scan request	7.5.7.3	FD2: O		M	
MF4.24	Remote scan response	7.5.7.4	M		FD2: O	
MF4.25	Transmit power change	7.5.7.5	O		M	

Table D.3—MAC frames (*Continued*)

Item Number	Item Description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF4.26	PM mode change	7.5.8.5	O		FD2: M	
MF4.27	SPS configuration request	7.5.8.3	O		FD2: M	
MF4.28	SPS configuration response	7.5.8.4	FD2: M		O	
MF4.29	PS set information request	7.5.8.1	O		FD2: M	
MF4.30	PS set information response	7.5.8.2	FD2: M		O	
MF4.31	Security message	7.5.9.1	O		O	
MF4.32	Vendor specific	7.5.9.2	O		O	

D.7.3.2 MAC sublayer functions

Table D.4—MAC sublayer functions

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
MLF1	Scanning capable	8.2.1	M			
MLF2	Starting capable	8.2.2	FD2: M			
MLF3	PNC handover capable	8.2.3	FD2: M			
MLF4	**Child piconet support**					
MLF4.1	Parent PNC supports request mechansim for creating child piconet	8.2.5	FD2: M			
MLF4.2	PNC capable DEV support of becoming a child PNC	8.2.5	FD2: O			
MLF5	**Neighbor piconet support**					
MLF5.1	Parent PNC supports request mechansim for creating neighbor piconet	7.5.1.2, 8.2.6	FD2: O			
MLF5.2	PNC capable DEV support of becoming a neighbor PNC	8.2.6	FD2: O			
MLF6	Stopping piconet operations	8.2.7.1	FD2: M			

Table D.4—MAC sublayer functions (*Continued*)

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
MLF7	Association	8.3.1	M			
MLF8	Piconet services support	8.3.2	O			
MLF9	Broadcasting of piconet information	8.3.3	FD2: M			
MLF10	DEV disassociation	8.3.4	M			
MLF11	PNC disassociation	8.3.4	FD2: M			
MLF12	**Contention access methods**					
MLF12.1	CAP channel access during piconet operations	8.4.2	O.1 PLF7: M			
MLF12.2	Open and association MCTA operations	8.4.3.3, 8.4.3.4	O.1			
MLF12.3	Regular MCTA operations	8.4.3.3	M			
MLF13	Asynchronous channel time reservation	8.5.2	FD2: M			
MLF14	Synchronization	8.6	M			
MLF14.1	Beacon generation	8.6.2	FD2: M			
MLF14.2	Extended beacon support, reception	8.6.2	M			
MLF14.3	Extended beacon support, generation	8.6.2	FD2: O			
MLF15	Fragmentation and defragmentation	8.7	M			
MLF16	**Acknowledgement and retransmissions**					
MLF16.1	No acknowledement	8.8.1	M			
MLF16.2	Immediate acknowledgement	8.8.2	M			
MLF16.3	Delayed acknowledgement	8.8.3	O			
MLF16.4	Retransmissions	8.8.4	M			
MLF16.5	Duplicate detection	8.8.5	M			
MLF17	**Peer discovery**					
MLF17.1	PNC information request	8.9.1	M			
MLF17.2	Probe request and response	8.9.2	M			
MLF17.3	Announce	8.9.3	M			

Table D.4—MAC sublayer functions (*Continued*)

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
MLF17.4	Channel status request	8.9.4	M			
MLF17.5	Remote Scan	8.9.5	O			
MLF18	**Changing piconet parameters**					
MLF18.1	Moving beacon	8.10.1	FD1: M, FD2: O, MLF4.2: M, MLF5.2: M			
MLF18.2	Changing superframe duration	8.10.2	FD1: M, FD2: O, MLF4.2: M, MLF5.2: M			
MLF18.3	Setting the BSID and PNID	8.2.8, 8.10.3	FD1: M, FD2: M			
MLF18.4	Maintaining synchroniza-tion in dependent piconets	8.10.4	MLF4.2: M, MLF5.2: M			
MLF19	Multi-rate support	8.12	O			
MLF20	Dynamic channel selection	8.11.1	O			
MLF21	**Power management**					
MLF21.1	PSPS	8.13.1	FD1: O FD2: M			
MLF21.2	SPS (support for at least one SPS set)	8.13.2	FD1: O FD2: M			
MLF21.3	APS	8.13.3	FD1: O FD2: M			
MLF22	**Transmit power control**					
MLF22.1	Fixed maximum transmit power	8.11.2.1	FD1: M, FD2: O			
MLF22.2	Request transmitter power adjustment	8.11.2.2	O			
MLF22.3	Adjust transmitter power as requested	8.11.2.2	M			

D.7.4 Security support

Table D.5—Security capabilities

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
S1	Mode 0 support	9.2.1	M			
S2	Mode 1 support	9.2.2, 10.2, 10.3	O			
S3	Supports acting as a key originator.	9.3	FD2 & S2: M, S2: O			
S4	Security info handover	9.4.1	O			

Annex E

(informative)

Bibliography

[B1] ETS 300-328: 1996, Radio Equipment and System (RES); Wideband Data Transmission Systems Technical Characteristics and Test Conditions for Data Transmission Equipment Operating in the 2.4 ISM Band and USing Spread Spectrum Modulation Techniques.

[B2] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.[13]

[B3] IEEE Std 802.11b™-1999 (Supplement to ANSI/IEEE Std 802.11, 1999 Edition), Supplement to 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: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[B4] IEEE Std 802.15.2™-2003, IEEE Recommended Practice for Information technology – Telecommunications and Information exchange between systems–Local and metropolitan area networks—Specific Requirements—Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands.

[B5] IEEE Std C95.1, 1999 Edition, IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 kHz to 300 GHz.

[B6] ISO/IEC 9646-7: 1995, Information Technology—Open Systems Interconnection—Conformance Testing Methodology and Framework—Part 7: Implementation Conformance Statements.

[B7] ITU-T X.200-1994, Information Technology—Open Systems Interconnection—Basic Reference Model.[14]

[B8] Milewski, A., "Periodic Sequences with Optimal Properties for Channel Estimation and Fast Start-Up Equalization," *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 426–431, Sept. 1983.

[B9] Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1965.

[B10] Peterson, William Wesley, and Weldon, E. J., *Error Correcting Codes*, Second Edition, 1972.

[B11] Proakis, *Digital Communications*, Third Edition, McGraw-Hill, 1995.

[B12] RFC 1157:1990, A Simple Network Management Protocol (SNMP).[15]

[B13] Stallings, *Data and Computer Communications*, Second Edition, MacMillan, 1988, pp. 300–302.

[B14] Ungerboeck, G., "Channel Coding with Multilevel/Phase and Signals," *IEEE Transactions on Information Theory*, vol. 28, Jan. 1982.

[13]The IEEE products referred to in this annex are trademarks belonging to the Institute of Electrical and Electronics Engineers, Inc.

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

[15]Requests for Comments (RFCs) are available from the Internet Engineering Task Force (IETF) at http://www.ietf.org/rfc.

[B15] Xylomenos, G., and Polyzos, G. C., “Link Layer Support for Quality of Service on Wireless Internet Links,” *IEEE Personal Communications*, vol. 6, no. 5, pp. 52–60, Oct. 1999.