

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



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

Copyright © 2003 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 29 September 2003. Printed in the United States of America.

IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

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

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

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

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.

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

Introduction

This introduction is not part of IEEE Std 802.15.3-2003, 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 1802TM, has been established to identify the conformance test methodology documents for the IEEE 802[®] family of standards. Thus the conformance test documents for IEEE 802.3TM are numbered 1802.3TM, the conformance test documents for IEEE 802.5TM will be 1802.5TM, 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 Alfvén
Arun Arunachalam
Naiel Askar
Venkat Bahl
Anuj Batra
Timothy J. Blaney
Stan Bottoms
Monique Bourgeois
Chuck Brabenec
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.

James D. Allen	Robert F. Heile	Roger Pandanda
Eladio Arvelo	Jin-Meng Ho	Subbu Ponnuswamy
Daniel Bailey	Andreas Hutter	Hugo Pues
Jay Bain	Tony Jefree	Vikram Punj
John Barnett	Bobby Jose	Richard Roberts
John R. Barr	Jeyhan Karaoguz	Jon Rosdahl
Catherine Berger	Kevin Karcz	Mike Rudnick
Keith Bernhardt	James Kemerling	John Sarallo
Maurice Bizzarri	Stuart J. Kerry	Durga Satapathy
Mitchell Buchman	Myoung Kim	Mark Schrader
Ed Callaway	Yongsuk Kim	Neil Shipp
Yawgeng Chau	Patrick Kinney	William Shvodian
Steven Crowley	Gunter Kleindl	Kevin Smart
Guru Dutt Dhingra	John Kowalski	Amjad Soomro
Thomas Dineen	Pi-Cheng Law	Kenneth Stanwood
Mary DuVal	Randolph Little	Carl Stevenson
Vern Dubendorf	Gregory Luri	Rene Struik
Dr. Sourav Dutta	Gerard Lynch	Lawrence Telle
Avraham Freedman	Roger Marks	Jerry Thrasher
Ernesto Garcia	Colleen McGinn	Anthony Tintera
Ian Gifford	Michael McNinn	Joan Viaplana
James P. K. Gilb	George Miao	Hung-yu Wei
Tim Godfrey	Timothy Miller	Matthew Welborn
Rajugopal Gubbi	Charles Ngethe	Edward Woodrow
Jose Gutierrez	Erwin Noble	Harry Worstell
Steven Halford	Bob O'Hara	Forrest Wright
Simon Harrison	Knut Odman	Jung Yee
Allen Heberling	Richard Paine	

Major technical contributions were received from the following individuals:

Masa Akahane	Keith Holt	Mike Rudnick
James D. Allen	Robert Huang	Mike Ryan
Rick Alfvin	Jeyhan Karaoguz	John Sarallo
Daniel Bailey	Patrick Kinney	Mark Schrader
Jay Bain	Philip Kossin	Tim Schmidl
John R. Barr	Stanley Ling	Bill Shvodian
Anand Dabak	Wayne Music	Ari Singer
Mary DuVal	Mohammed Nafie	René Struik
James P. K. Gilb	Knut Odman	Shige Sugaya
Rajugopal Gubbi	Greg Parks	Mitsuhiro Suzuki
Tom Guerena	Douglas J. Percival	Kazuhisa Takamura
Allen Heberling	Gregg Rasor	Chandra Vaidyanathan
Jin-Meng Ho	Richard Roberts	

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 Nikolic
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.

¹ANSI 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>).

²IEEE 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/>).

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

⁴ISO/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 802.15.3TM 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 piconet: 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. 802.15.3TM-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 802.15.3TM piconet.

3.13 device-host: The equipment that incorporates an 802.15.3TM 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.

⁵NIST 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>).

⁶The 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.

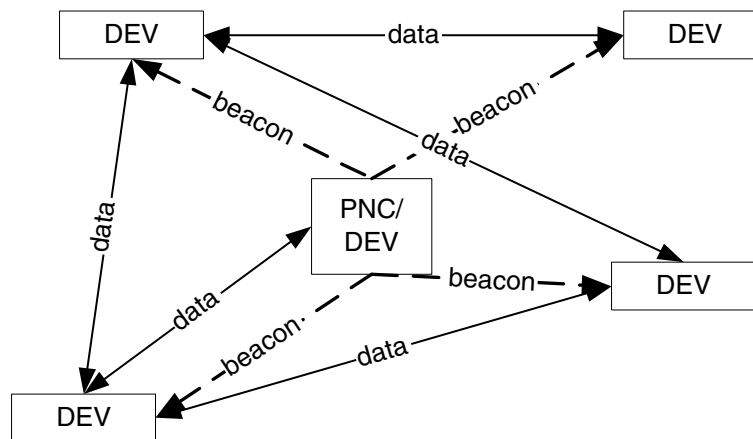


Figure 1—802.15.3 piconet elements

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 independent 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 management 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.

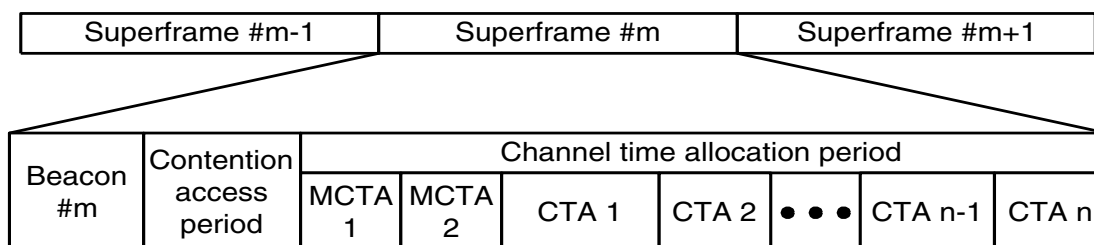


Figure 2—802.15.3 piconet superframe

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:

- Sending asynchronous data in the CAP, if present, as described in 8.4.2.
- Allocating channel time for isochronous streams in the CTAP, as described in 8.5.1.
- 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 mMaxLost-Beacons 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 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 (DPS) 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 asynchronous 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-1999TM [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.

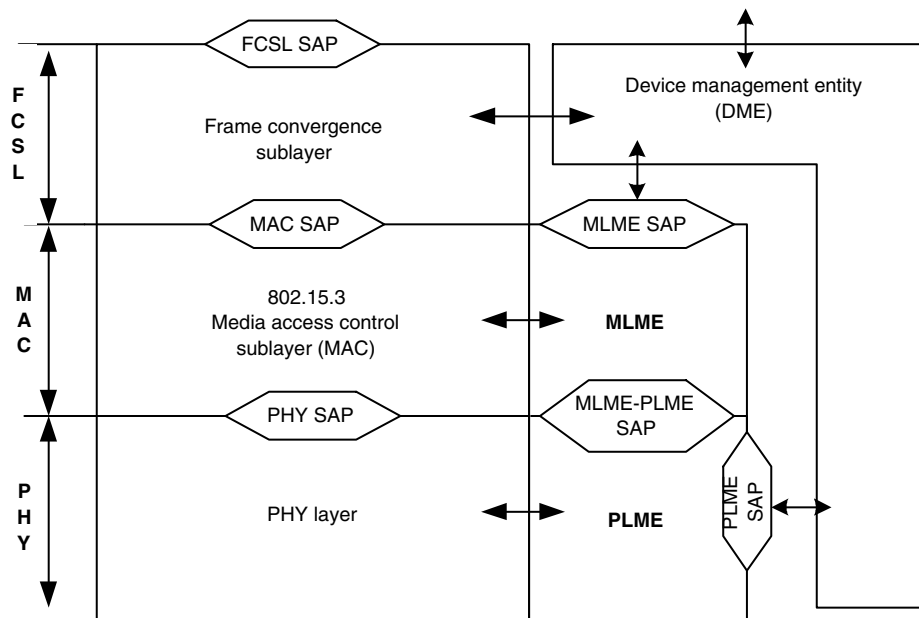


Figure 3—The reference model used in this standard

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 Piconet-Description	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
(
    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.
DistributeKeyTime-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, 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.indication 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.