INTERNATIONAL STANDARD

ISO/IEC/ IEEE 8802-15-4

First edition
2010-10-15

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

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

Part 15-4:

Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Réseaux locaux et métropolitains — Exigences spécifiques —

Partie 15-4: Spécifications du contrôle d'accès du milieu sans fil (MAC) et de la couche physique (PHY) pour les réseaux personnels sans fil de faible débit (WPAN)

Reference number
ISO/IEC/IEEE 8802-15-4:2010(E)

© IEEE 2010

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. Neither the ISO Central Secretariat nor IEEE accepts any liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies and IEEE members. In the unlikely event that a problem relating to it is found, please inform the ISO Central Secretariat or IEEE at the address given below.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

COPYRIGHT PROTECTED DOCUMENT

© IEEE 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO or IEEE at the respective address below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York • NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published in Switzerland

© IEEE 2010 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the 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. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

ISO/IEC/IEEE 8802-15-4 was prepared by the LAN/MAN Standards Committee of the IEEE Computer Society (as IEEE Std 802.15.4-2006). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*, in parallel with its approval by the ISO/IEC national bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. IEEE is responsible for the maintenance of this document with participation and input from ISO/IEC national bodies.

ISO/IEC/IEEE 8802 consists of the following parts, under the general title *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements*:

— *Part 15-4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

802.15.4™

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

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

8 September 2006

IEEE Std 802.15.4™-2006
(Revision of
IEEE Std 802.15.4-2003)

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

IEEE Std 802.15.4™-2006
(Revision of
IEEE Std 802.15.4-2003)

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

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 7 June 2006

IEEE-SA Standards Board

Abstract: IEEE Std 802.15.4-2003 defined the protocol and compatible interconnection for data communication devices using low-data-rate, low-power, and low-complexity short-range radio frequency (RF) transmissions in a wireless personal area network (WPAN). This revision extends the market applicability of IEEE Std 802.15.4, removes ambiguities in the standard, and makes improvements revealed by implementations of IEEE Std 802.15.4-2003.
Keywords: ad hoc network, low data rate, low power, LR-WPAN, mobility, PAN, personal area network, radio frequency, RF, short range, wireless, wireless personal area network, WPAN

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

Copyright © 2006 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 8 September 2006. 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 0-7381-4996-9 SH95552
PDF: ISBN 0-7381-4997-7 SS95552

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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

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

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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Introduction

This introduction is not part of IEEE Std 802.15.4-2006, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).

This standard defines the protocol and interconnection of devices via radio communication in a personal area network (PAN). The standard uses carrier sense multiple access with collision avoidance (CSMA-CA) medium access mechanism and supports star as well as peer-to-peer topologies. The media access is contention based; however, using the optional superframe structure, time slots can be allocated by the PAN coordinator to devices with time critical data. Connectivity to higher performance networks is provided through a PAN coordinator.

This revision was initiated to incorporate additional features and enhancements as well as some simplifications to the 2003 edition of this standard. The standard now includes two optional physical layers (PHYs) yielding higher data rates in the lower frequency bands and, therefore, specifies the following four PHYs:

- An 868/915 MHz direct sequence spread spectrum (DSSS) PHY employing binary phase-shift keying (BPSK) modulation
- An 868/915 MHz DSSS PHY employing offset quadrature phase-shift keying (O-QPSK) modulation
- An 868/915 MHz parallel sequence spread spectrum (PSSS) PHY employing BPSK and amplitude shift keying (ASK) modulation
- A 2450 MHz DSSS PHY employing O-QPSK modulation

The 868/915 MHz PHYs support over-the-air data rates of 20 kb/s, 40 kb/s, and optionally 100kb/s and 250kb/s. The 2450 MHz PHY supports an over-the-air data rate of 250 kb/s. The PHY chosen depends on local regulations and user preference.

This revision also incorporates the following additions and enhancements to the 2003 edition:

- Adds support for a shared time base through the addition of a data time stamping mechanism
- Adds extensions of the 2.4GHz derivative modulation yielding higher data rates at the lower frequency bands
- Incorporates a mechanism for communicating the revision level on a frame-by-frame basis
- Adds support for beacon scheduling
- Allows synchronization of broadcast messages in beacon-enabled PANs
- Improves usage of security suite

Also, this revision incorporates the following changes and simplifications:

- Makes GTS support optional
- Removes restrictions for manually enabling the receiver
- Simplifies passive and active scan procedures
- Allows for more flexibility in the CSMA-CA algorithm
- Reduces association time in nonbeacon networks

This revision is backward-compatible to the 2003 edition; in other words, devices conforming to this standard are capable of joining and functioning in a PAN composed of devices conforming to IEEE Std 802.15.4-2003.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Notice to users

Errata

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

Interpretations

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

Patents

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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Contents

1. Overview 1

1.1 General 1
1.2 Scope 1
1.3 Purpose 2

2. Normative references 3

3. Definitions 5

4. Acronyms and abbreviations 9

5. General description 13

5.1 Introduction 13
5.2 Components of the IEEE 802.15.4 WPAN 13
5.3 Network topologies 14
5.3.1 Star network formation 14
5.3.2 Peer-to-peer network formation 15
5.4 Architecture 15
5.4.1 Physical layer (PHY) 17
5.4.2 MAC sublayer 17
5.5 Functional overview 17
5.5.1 Superframe structure 17
5.5.2 Data transfer model 18
5.5.2.1 Data transfer to a coordinator 19
5.5.2.2 Data transfer from a coordinator 20
5.5.2.3 Peer-to-peer data transfers 21
5.5.3 Frame structure 21
5.5.3.1 Beacon frame 21
5.5.3.2 Data frame 22
5.5.3.3 Acknowledgment frame 22
5.5.3.4 MAC command frame 23
5.5.4 Improving probability of successful delivery 23
5.5.4.1 CSMA-CA mechanism 23
5.5.4.2 Frame acknowledgment 24
5.5.4.3 Data verification 24
5.5.5 Power consumption considerations 24
5.5.6 Security 24
5.6 Concept of primitives 25

6. PHY specification 27

6.1 General requirements and definitions 27
6.1.1 Operating frequency range 27
6.1.2 Channel assignments 28
6.1.2.1 Channel numbering 29
6.1.2.2 Channel pages 29
6.1.3 Minimum long interframe spacing (LIFS) and short interframe spacing (SIFS) periods 30

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.1.4 RF power measurement 31
6.1.5 Transmit power 31
6.1.6 Out-of-band spurious emission 31
6.1.7 Receiver sensitivity definitions 31
6.2 PHY service specifications 31
6.2.1 PHY data service 32
6.2.1.1 PD-DATA.request 32
6.2.1.2 PD-DATA.confirm 33
6.2.1.3 PD-DATA.indication 34
6.2.2 PHY management service 34
6.2.2.1 PLME-CCA.request 35
6.2.2.2 PLME-CCA.confirm 35
6.2.2.3 PLME-ED.request 36
6.2.2.4 PLME-ED.confirm 36
6.2.2.5 PLME-GET.request 37
6.2.2.6 PLME-GET.confirm 38
6.2.2.7 PLME-SET-TRX-STATE.request 39
6.2.2.8 PLME-SET-TRX-STATE.confirm 40
6.2.2.9 PLME-SET.request 40
6.2.2.10 PLME-SET.confirm 41
6.2.3 PHY enumerations description 42
6.3 PPDU format 43
6.3.1 Preamble field 43
6.3.2 SFD field 44
6.3.3 Frame Length field 45
6.3.4 PSDU field 45
6.4 PHY constants and PIB attributes 45
6.4.1 PHY constants 45
6.4.2 PHY PIB attributes 45
6.5 2450 MHz PHY specifications 47
6.5.1 Data rate 47
6.5.2 Modulation and spreading 47
6.5.2.1 Reference modulator diagram 47
6.5.2.2 Bit-to-symbol mapping 47
6.5.2.3 Symbol-to-chip mapping 47
6.5.2.4 O-QPSK modulation 48
6.5.2.5 Pulse shape 49
6.5.2.6 Chip transmission order 49
6.5.3 2450 MHz band radio specification 49
6.5.3.1 Transmit power spectral density (PSD) mask 49
6.5.3.2 Symbol rate 49
6.5.3.3 Receiver sensitivity 49
6.5.3.4 Receiver jamming resistance 50
6.6 868/915 MHz band binary phase-shift keying (BPSK) PHY specifications 50
6.6.1 868/915 MHz band data rates 50
6.6.2 Modulation and spreading 50
6.6.2.1 Reference modulator diagram 50
6.6.2.2 Differential encoding 51
6.6.2.3 Bit-to-chip mapping 51
6.6.2.4 BPSK modulation 51
6.6.3 868/915 MHz band radio specification 52
6.6.3.1 Operating frequency range 52
6.6.3.2 915 MHz band transmit PSD mask 52
6.6.3.3 Symbol rate 52

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.6.3.4 Receiver sensitivity 52
6.6.3.5 Receiver jamming resistance 52
6.7 868/915 MHz band (optional) amplitude shift keying (ASK) PHY specifications 53
6.7.1 868/915 MHz band data rates 53
6.7.2 Modulation and spreading 53
6.7.2.1 Reference modulator diagram 53
6.7.2.2 Bit-to-symbol mapping 54
6.7.2.3 Symbol-to-chip mapping 54
6.7.2.4 ASK modulation 55
6.7.3 868/915 MHz band radio specification for the ASK PHY 57
6.7.3.1 Operating frequency range 57
6.7.3.2 915 MHz band transmit PSD mask 57
6.7.3.3 Symbol rate 57
6.7.3.4 Receiver sensitivity 57
6.7.3.5 Receiver jamming resistance 57
6.7.4 SHR for ASK PHY 58
6.7.4.1 Preamble for ASK PHY 58
6.7.4.2 SFD for ASK PHY 58
6.7.4.3 Example of PSSS encoding 58
6.8 868/915 MHz band (optional) O-QPSK PHY specifications 60
6.8.1 868/915 MHz band data rates 60
6.8.2 Modulation and spreading 60
6.8.2.1 Reference modulator diagram 60
6.8.2.2 Bit-to-symbol mapping 60
6.8.2.3 Symbol-to-chip mapping 60
6.8.2.4 O-QPSK modulation 61
6.8.2.5 Pulse shape 62
6.8.2.6 Chip transmission order 62
6.8.3 868/915 MHz band radio specification 62
6.8.3.1 Operating frequency range 62
6.8.3.2 Transmit PSD mask 62
6.8.3.3 Symbol rate 63
6.8.3.4 Receiver sensitivity 63
6.8.3.5 Receiver jamming resistance 63
6.9 General radio specifications 63
6.9.1 TX-to-RX turnaround time 63
6.9.2 RX-to-TX turnaround time 64
6.9.3 Error-vector magnitude (EVM) definition 64
6.9.4 Transmit center frequency tolerance 65
6.9.5 Transmit power 65
6.9.6 Receiver maximum input level of desired signal 65
6.9.7 Receiver ED 65
6.9.8 Link quality indicator (LQI) 65
6.9.9 Clear channel assessment (CCA) 66

7. MAC sublayer specification 67

7.1 MAC sublayer service specification 67
7.1.1 MAC data service 68
7.1.1.1 MCPS-DATA.request 68
7.1.1.2 MCPS-DATA.confirm 71
7.1.1.3 MCPS-DATA.indication 72
7.1.1.4 MCPS-PURGE.request 75
7.1.1.5 MCPS-PURGE.confirm 75

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.1.6 Data service message sequence chart ... 76
7.1.2 MAC management service ... 76
7.1.3 Association primitives ... 77
7.1.3.1 MLME-ASSOCIATE.request ... 78
7.1.3.2 MLME-ASSOCIATE.indication ... 80
7.1.3.3 MLME-ASSOCIATE.response ... 81
7.1.3.4 MLME-ASSOCIATE.confirm ... 83
7.1.3.5 Association message sequence charts ... 85
7.1.4 Disassociation primitives ... 86
7.1.4.1 MLME-DISASSOCIATE.request ... 86
7.1.4.2 MLME-DISASSOCIATE.indication ... 89
7.1.4.3 MLME-DISASSOCIATE.confirm ... 90
7.1.4.4 Disassociation message sequence charts ... 91
7.1.5 Beacon notification primitive ... 92
7.1.5.1 MLME-BEACON-NOTIFY.indication ... 92
7.1.6 Primitives for reading PIB attributes ... 94
7.1.6.1 MLME-GET.request ... 95
7.1.6.2 MLME-GET.confirm ... 96
7.1.7 GTS management primitives ... 97
7.1.7.1 MLME-GTS.request ... 97
7.1.7.2 MLME-GTS.confirm ... 99
7.1.7.3 MLME-GTS.indication ... 100
7.1.7.4 GTS management message sequence charts ... 102
7.1.8 Primitives for orphan notification ... 103
7.1.8.1 MLME-ORPHAN.indication ... 103
7.1.8.2 MLME-ORPHAN.response ... 104
7.1.8.3 Orphan notification message sequence chart ... 106
7.1.9 Primitives for resetting the MAC sublayer ... 106
7.1.9.1 MLME-RESET.request ... 106
7.1.9.2 MLME-RESET.confirm ... 107
7.1.10 Primitives for specifying the receiver enable time ... 108
7.1.10.1 MLME-RX-ENABLE.request ... 108
7.1.10.2 MLME-RX-ENABLE.confirm ... 110
7.1.10.3 Message sequence chart for changing the state of the receiver ... 110
7.1.11 Primitives for channel scanning ... 111
7.1.11.1 MLME-SCAN.request ... 111
7.1.11.2 MLME-SCAN.confirm ... 114
7.1.11.3 Channel scan message sequence charts ... 116
7.1.12 Communication status primitive ... 116
7.1.12.1 MLME-COMM-STATUS.indication ... 116
7.1.13 Primitives for writing PIB attributes ... 119
7.1.13.1 MLME-SET.request ... 119
7.1.13.2 MLME-SET.confirm ... 121
7.1.14 Primitives for updating the superframe configuration ... 122
7.1.14.1 MLME-START.request ... 122
7.1.14.2 MLME-START.confirm ... 126
7.1.14.3 Message sequence chart for updating the superframe configuration ... 126
7.1.15 Primitives for synchronizing with a coordinator ... 127
7.1.15.1 MLME-SYNC.request ... 127
7.1.15.2 MLME-SYNC-LOSS.indication ... 128
7.1.15.3 Message sequence chart for synchronizing with a coordinator ... 131
7.1.16 Primitives for requesting data from a coordinator ... 131
7.1.16.1 MLME-POLL.request ... 132
7.1.16.2 MLME-POLL.confirm ... 133

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.16.3 Message sequence chart for requesting data from a coordinator 134
7.1.17 MAC enumeration description 135
7.2 MAC frame formats 137
7.2.1 General MAC frame format 138
7.2.1.1 Frame Control field 138
7.2.1.2 Sequence Number field 140
7.2.1.3 Destination PAN Identifier field 140
7.2.1.4 Destination Address field 140
7.2.1.5 Source PAN Identifier field 141
7.2.1.6 Source Address field 141
7.2.1.7 Auxiliary Security Header field 141
7.2.1.8 Frame Payload field 141
7.2.1.9 FCS field 141
7.2.2 Format of individual frame types 142
7.2.2.1 Beacon frame format 142
7.2.2.2 Data frame format 146
7.2.2.3 Acknowledgment frame format 147
7.2.2.4 MAC command frame format 147
7.2.3 Frame compatibility 148
7.3 MAC command frames 149
7.3.1 Association request command 150
7.3.1.1 MHR fields 150
7.3.1.2 Capability Information field 150
7.3.2 Association response command 151
7.3.2.1 MHR fields 151
7.3.2.2 Short Address field 152
7.3.2.3 Association Status field 152
7.3.3 Disassociation notification command 152
7.3.3.1 MHR fields 153
7.3.3.2 Disassociation Reason field 153
7.3.4 Data request command 153
7.3.5 PAN ID conflict notification command 154
7.3.6 Orphan notification command 155
7.3.7 Beacon request command 156
7.3.8 Coordinator realignment command 156
7.3.8.1 MHR fields 157
7.3.8.2 PAN Identifier field 157
7.3.8.3 Coordinator Short Address field 157
7.3.8.4 Logical Channel field 157
7.3.8.5 Short Address field 157
7.3.8.6 Channel Page field 157
7.3.9 GTS request command 158
7.3.9.1 MHR fields 158
7.3.9.2 GTS Characteristics field 158
7.4 MAC constants and PIB attributes 159
7.4.1 MAC constants 159
7.4.2 MAC PIB attributes 160
7.5 MAC functional description 166
7.5.1 Channel access 167
7.5.1.1 Superframe structure 167
7.5.1.2 Incoming and outgoing superframe timing 169
7.5.1.3 Interframe spacing (IFS) 169
7.5.1.4 CSMA-CA algorithm 170
7.5.2 Starting and maintaining PANs 172

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.2.1 Scanning through channels ... 172
7.5.2.2 PAN identifier conflict resolution ... 176
7.5.2.3 Starting and realigning a PAN ... 177
7.5.2.4 Beacon generation ... 178
7.5.2.5 Device discovery ... 179
7.5.3 Association and disassociation ... 179
7.5.3.1 Association ... 179
7.5.3.2 Disassociation ... 181
7.5.4 Synchronization ... 182
7.5.4.1 Synchronization with beacons ... 182
7.5.4.2 Synchronization without beacons ... 183
7.5.4.3 Orphaned device realignment ... 183
7.5.5 Transaction handling ... 183
7.5.6 Transmission, reception, and acknowledgment ... 185
7.5.6.1 Transmission ... 185
7.5.6.2 Reception and rejection ... 186
7.5.6.3 Extracting pending data from a coordinator ... 187
7.5.6.4 Use of acknowledgments and retransmissions ... 189
7.5.6.5 Promiscuous mode ... 190
7.5.6.6 Transmission scenarios ... 191
7.5.7 GTS allocation and management ... 192
7.5.7.1 CAP maintenance ... 193
7.5.7.2 GTS allocation ... 193
7.5.7.3 GTS usage ... 194
7.5.7.4 GTS deallocation ... 195
7.5.7.5 GTS reallocation ... 196
7.5.7.6 GTS expiration ... 197
7.5.8 Frame security ... 197
7.5.8.1 Security-related MAC PIB attributes ... 197
7.5.8.2 Functional description ... 199
7.6 Security suite specifications ... 206
7.6.1 PIB security material ... 206
7.6.2 Auxiliary security header ... 210
7.6.2.1 Integer and octet representation ... 210
7.6.2.2 Security Control field ... 210
7.6.2.3 Frame Counter field ... 212
7.6.2.4 Key Identifier field ... 212
7.6.3 Security operations ... 213
7.6.3.1 Integer and octet representation ... 213
7.6.3.2 CCM* Nonce ... 213
7.6.3.3 CCM* prerequisites ... 213
7.6.3.4 CCM* transformation data representation ... 214
7.6.3.5 CCM* inverse transformation data representation ... 215
7.7 Message sequence charts illustrating MAC-PHY interaction ... 216

Annex A (normative) Service-specific convergence sublayer (SSCS) ... 227

A.1 IEEE 802.2 convergence sublayer ... 227
A.1.1 MA-UNITDATA.request ... 227
A.1.1.1 Semantics of the service primitive ... 227
A.1.1.2 Appropriate usage ... 228
A.1.1.3 Effect on receipt ... 228
A.1.2 MA-UNITDATA.indication ... 228
A.1.2.1 Semantics of the service primitive ... 228

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

A.1.2.2 When generated ... 228
A.1.2.3 Appropriate usage ... 228
A.1.3 MA-UNITDATA-STATUS.indication ... 229
A.1.3.1 Semantics of the service primitive ... 229
A.1.3.2 When generated ... 229
A.1.3.3 Appropriate usage ... 229

Annex B (normative) CCM* mode of operation ... 231

B.1 Introduction ... 231
B.2 Notation and representation ... 231
B.2.1 Strings and string operations ... 231
B.2.2 Integers, octets, and their representation ... 231
B.3 Symmetric-key cryptographic building blocks ... 231
B.3.1 Block cipher ... 231
B.3.2 Mode of operation ... 232
B.4 Specification of generic CCM* mode of operation ... 232
B.4.1 CCM* mode encryption and authentication transformation ... 232
B.4.1.1 Input transformation ... 233
B.4.1.2 Authentication transformation ... 233
B.4.1.3 Encryption transformation ... 234
B.4.2 CCM* mode decryption and authentication checking transformation ... 234
B.4.2.1 Decryption transformation ... 235
B.4.2.2 Authentication checking transformation ... 235
B.4.3 Restrictions ... 235

Annex C (informative) Test vectors for cryptographic building blocks ... 237

C.1 AES block cipher ... 237
C.2 Mode of operation ... 237
C.2.1 MAC beacon frame ... 237
C.2.1.1 Description ... 237
C.2.1.2 CCM* mode encryption and authentication transformation ... 238
C.2.1.3 CCM* mode decryption and authentication checking transformation ... 240
C.2.2 MAC data frame ... 241
C.2.2.1 Description ... 241
C.2.2.2 CCM* mode encryption and authentication transformation ... 241
C.2.2.3 CCM* mode decryption and authentication checking transformation ... 243
C.2.3 MAC command frame ... 245
C.2.3.1 Description ... 245
C.2.3.2 CCM* mode encryption and authentication transformation ... 245
C.2.3.3 CCM* mode decryption and authentication checking transformation ... 247

Annex D (normative) Protocol implementation conformance statement (PICS) proforma ... 251

D.1 Introduction ... 251
D.1.1 Scope ... 251
D.1.2 Purpose ... 251
D.2 Abbreviations and special symbols ... 251
D.3 Instructions for completing the PICS proforma ... 252
D.4 Identification of the implementation ... 252
D.5 Identification of the protocol ... 253
D.6 Global statement of conformance ... 253
D.7 PICS proforma tables ... 254

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

D.7.1 Major roles for devices compliant with IEEE Std 802.15.4-2006 ... 254
D.7.2 Major capabilities for the PHY ... 255
D.7.2.1 PHY functions ... 255
D.7.2.2 PHY packet ... 255
D.7.2.3 Radio frequency (RF) ... 256
D.7.3 Major capabilities for the MAC sublayer ... 256
D.7.3.1 MAC sublayer functions ... 256
D.7.3.2 MAC frames ... 258

Annex E (informative) Coexistence with other IEEE standards and proposed standards ... 261

E.1 Introduction ... 261
E.2 Standards and proposed standards characterized for coexistence ... 261
E.3 General coexistence issues ... 261
E.3.1 Clear channel assessment (CCA) ... 262
E.3.2 Modulation ... 262
E.3.2.1 2400 MHz band PHY ... 262
E.3.2.2 800/900 MHz band PHYs ... 262
E.3.3 ED and LQI ... 263
E.3.4 Low duty cycle ... 263
E.3.5 Low transmit power ... 263
E.3.5.1 2400 MHz band PHY ... 263
E.3.5.2 800 MHz band PHYs ... 263
E.3.5.3 900 MHz band PHYs ... 264
E.3.6 Channel alignment ... 264
E.3.7 Dynamic channel selection ... 264
E.3.8 Neighbor piconet capability ... 264
E.4 2400 MHz band coexistence performance ... 265
E.4.1 Assumptions for coexistence quantification ... 265
E.4.1.1 Channel model ... 265
E.4.1.2 Receiver sensitivity ... 266
E.4.1.3 Transmit power ... 266
E.4.1.4 Receiver bandwidth ... 266
E.4.1.5 Transmit spectral masks ... 266
E.4.1.6 IEEE 802.11b transmit PSD ... 267
E.4.1.7 Interference characteristics ... 267
E.4.1.8 Bit error rate (BER) calculations ... 267
E.4.1.9 Packet error rate (PER) ... 268
E.4.2 BER model ... 268
E.4.3 Coexistence simulation results ... 268
E.5 800/900 MHz bands coexistence performance ... 273
E.5.1 Victims and assailants ... 273
E.5.2 Bandwidth ... 273
E.5.3 Path loss model ... 273
E.5.4 Temporal model ... 274
E.5.5 Coexistence assurance results ... 275
E.5.5.1 868 MHz BPSK PHY ... 275
E.5.5.2 868 MHz O-QPSK PHY ... 276
E.5.5.3 868 MHz PSSS PHY ... 278
E.5.5.4 915 MHz BPSK PHY ... 279
E.5.5.5 915 MHz O-QPSK PHY ... 280
E.5.5.6 915 MHz PSSS PHY ... 281
E.6 Notes on the calculations ... 282

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Annex F (informative) IEEE 802.15.4 regulatory requirements 283

F.1 Introduction 283
F.2 Applicable U.S. (FCC) rules 285
F.2.1 Section 15.35 of FCC CFR47 285
F.2.2 Section 15.209 of FCC CFR47 287
F.2.3 Section 15.205 of FCC CFR47 287
F.2.4 Section 15.247 of FCC CFR47 288
F.2.5 Section 15.249 of FCC CFR47 289
F.3 Applicable European rules 290
F.3.1 European 2400 MHz band rules 291
F.3.2 European 868–870 MHz band rules 292
F.4 Applicable Japanese rules 294
F.5 Emissions specification analysis with respect to known worldwide regulations 295
F.5.1 General analysis and impact of detector bandwidth and averaging rules 295
F.5.2 Frequency spreading and averaging effects specific to IEEE Std 802.15.4 297
F.6 Summary of out-of-band spurious emissions limits 299
F.7 Phase noise requirements inferred from regulatory limits 300
F.8 Summary of transmission power levels 301

Annex G (informative) Bibliography 303

G.1 General 303
G.2 Regulatory documents 304

Annex H (informative) IEEE list of participants 306

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

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

Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

1. Overview

1.1 General

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

This document defines a standard for a low-rate WPAN (LR-WPAN).

1.2 Scope

The scope of this revision is to produce specific enhancements and corrections to IEEE Std 802.15.4, all of which will be backwards compatible with IEEE Std 802.15.4-2003. These enhancements and corrections include resolving ambiguities, reducing unnecessary complexity, increasing flexibility in security key usage, considerations for newly available frequency allocations, and others.

IEEE Std 802.15.4 defines the physical layer (PHY) and medium access control (MAC) sublayer specifications for low-data-rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements typically operating in the personal operating space (POS) of 10 m. It is foreseen that, depending on the application, a longer range at a lower data rate may be an acceptable tradeoff.

© IEEE 2010 – All rights reserved

It is the intent of this revision to work toward a level of coexistence with other wireless devices in conjunction with Coexistence Task Groups, such as IEEE 802.15.2 and IEEE 802.11/ETSI-BRAN/MMAC 5GSG.

1.3 Purpose

The purpose of this revision is to extend the market applicability of IEEE Std 802.15.4 and to remove ambiguities in the standard. Implementations of the 2003 edition of this standard have revealed potential areas of improvements. Additional frequency bands are being made available in various countries that are attractive for this application space.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

2. Normative references

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

FIPS Pub 197, Advanced Encryption Standard (AES).[1]

IEEE Std 802®, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[2]

ISO/IEC 8802-2 (IEEE Std 802.2™), Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control.[3]

ISO/IEC 9646-7 (ITU-T Rec. X.296), Information technology — Open systems interconnection — Conformance testing methodology and framework — Part 7: Implementation conformance statements.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

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

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

[3]ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

3. Definitions

For the purposes of this standard, the following terms and definitions apply. Terms not defined in this clause can be found in the *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition [B3].[4]

3.1 alternate personal area network (PAN) coordinator: A coordinator that is capable of replacing the PAN coordinator, if the PAN coordinator leaves the network for any reason. A PAN can have zero or more alternate PAN coordinators.

3.2 association: The service used to establish membership for a device in a wireless personal area network (WPAN).

3.3 authentication tag: Information that allows the verification of authenticity of a message.

3.4 beacon-enabled personal area network (PAN): A PAN in which all coordinators emit regular beacons, i.e., have beacon order < 0x0F.

3.5 block cipher: A cryptographic function that operates on strings of fixed size.

3.6 block size: The size, in bits, of the strings on which a block cipher operates.

3.7 contention access period (CAP): The period of time immediately following a beacon frame during which devices wishing to transmit will compete for channel access using a slotted carrier sense multiple access with collision avoidance (CSMA-CA) mechanism.

3.8 contention access period (CAP) symbol: A symbol period occurring during the CAP.

3.9 coordinator: A full-function device (FFD) capable of relaying messages. If a coordinator is the principal controller of a personal area network (PAN), it is called the PAN coordinator.

3.10 data authentication: The process whereby an entity receiving a message corroborates evidence about the true source of the information in the message and, thereby, evidence that the message has not been modified in transit.

3.11 data authenticity: Assurance about the source of information.

3.12 device: Any entity containing an implementation of the IEEE 802.15.4 medium access control (MAC) and physical interface to the wireless medium. A device may be a reduced-function device (RFD) or a full-function device (FFD).

3.13 encryption: The transformation of a message into a new representation so that privileged information is required to recover the original representation.

3.14 frame: The format of aggregated bits from a medium access control (MAC) sublayer entity that are transmitted together in time.

3.15 full-function device (FFD): A device capable of operating as a coordinator.

3.16 group key: A key that is known only to a set of devices.

3.17 idle period: A duration of time where no transceiver activity is scheduled to take place.

[4]The numbers in brackets correspond to the numbers of the bibliography in Annex G.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

3.18 key: Privileged information that may be used, for example, to protect information from disclosure to, and/or undetectable modification by, parties that do not have access to this privileged information.

3.19 key establishment: A process whereby two or more parties establish a key.

3.20 keying material: The combination of a key and associated security information (e.g., a nonce value).

3.21 key management: The collection of processes for the establishment and maintenance of keying relationships over a system's lifetime.

3.22 key sharing group: A set of devices that share a key.

3.23 local clock: The symbol clock internal to a device.

3.24 link key: A secret key that is shared between precisely two devices.

3.25 minimum security level: Indication of minimum protection required on information in transit.

3.26 mobile device: A device whose logical location in the network may change during use.

3.27 m-sequence: Maximal length linear feedback shift register sequence.

3.28 nonbeacon-enabled personal area network (PAN): A PAN in which coordinators do not emit regular beacons, i.e., have beacon order = 0x0F.

3.29 nonce: A nonrepeating value, such as an increasing counter, a sufficiently long random string, or a timestamp.

3.30 orphaned device: A device that has lost contact with its associated coordinator.

3.31 packet: The formatted, aggregated bits that are transmitted together in time across the physical medium.

3.32 payload data: The contents of a data message that is being transmitted.

3.33 personal area network (PAN) coordinator: A coordinator that is the principal controller of a PAN. An IEEE 802.15.4 network has exactly one PAN coordinator.

3.34 personal operating space (POS): The space about a person or object that is typically about 10 m in all directions and envelops the person or object whether stationary or in motion.

3.35 plain text: A string of unscrambled information.

3.36 protection: The combination of security services provided for information in transit, such as confidentiality, data authenticity, and/or replay protection.

3.37 radio sphere of influence: The region of space throughout which a radio may successfully communicate with other like radios.

3.38 reduced-function device (RFD): A device that is not capable of operating as a coordinator.

3.39 security level: Indication of purported protection applied to information in transit.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

3.40 self-healing: The ability of the network to detect, and recover from, faults appearing in either network nodes or communication links, without human intervention.

3.41 self-organizing: The ability of network nodes to detect the presence of other nodes and to organize into a structured, functioning network without human intervention.

3.42 symmetric key: A secret key that is shared between two or more parties that may be used for encryption/decryption or integrity protection/integrity verification, depending on its intended use.

3.43 transaction: The exchange of related, consecutive frames between two peer medium access control (MAC) entities, required for a successful transmission of a MAC command or data frame.

3.44 transaction queue: A list of the pending transactions, which are to be sent using indirect transmission, that are initiated by the medium access control (MAC) sublayer of a given coordinator. The transaction queue is maintained by that coordinator while the transactions are in progress, and its length is implementation-dependent but must be at least one.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

4. Acronyms and abbreviations

AES	advanced encryption standard
ASK	amplitude shift keying
AWGN	additive white Gaussian noise
AWN	affected wireless network
BE	backoff exponent
BER	bit error rate
BI	beacon interval
BLE	battery life extension
BO	beacon order
BPSK	binary phase-shift keying
BSN	beacon sequence number
CAP	contention access period
CBC-MAC	cipher block chaining message authentication code
CCA	clear channel assessment
CCM	counter with CBC-MAC (mode of operation)
CCM*	extension of CCM
CFP	contention-free period
CRC	cyclic redundancy check
CSMA-CA	carrier sense multiple access with collision avoidance
CTR	counter mode
CW	contention window (length)
DSN	data sequence number
DSSS	direct sequence spread spectrum
ED	energy detection
EIRP	effective isotropic radiated power
EMC	electromagnetic compatibility
ERP	effective radiated power
EVM	error vector magnitude
FCS	frame check sequence
FFD	full-function device
FH	frequency hopping
FHSS	frequency hopping spread spectrum
GTS	guaranteed time slot
IFS	interframe space or spacing
ISM	industrial, scientific, and medical
IUT	implementation under test
IWN	interfering wireless network
LIFS	long interframe spacing
LLC	logical link control
LQI	link quality indication
LPDU	LLC protocol data unit
LR-WPAN	low-rate wireless personal area network
LSB	least significant bit
MAC	medium access control

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

MCPS	MAC common part sublayer
MCPS-SAP	MAC common part sublayer service access point
MFR	MAC footer
MHR	MAC header
MIC	message integrity code
MLME	MAC sublayer management entity
MLME-SAP	MAC sublayer management entity service access point
MSB	most significant bit
MPDU	MAC protocol data unit
MSDU	MAC service data unit
NB	number of backoff (periods)
OCDM	orthogonal code division multiplexing
O-QPSK	offset quadrature phase-shift keying
OSI	open systems interconnection
PAN	personal area network
PC	personal computer
PD	PHY data
PD-SAP	PHY data service access point
PER	packet error rate
PHR	PHY header
PHY	physical layer
PIB	PAN information base
PICS	protocol implementation conformance statement
PLME	physical layer management entity
PLME-SAP	physical layer management entity service access point
PN	pseudo-random noise
POS	personal operating space
PPDU	PHY protocol data unit
PSD	power spectral density
PSDU	PHY service data unit
PSSS	parallel sequence spread spectrum
RF	radio frequency
RFD	reduced-function device
RX	receive or receiver
SD	superframe duration
SER	symbol error rate
SFD	start-of-frame delimiter
SHR	synchronization header
SIFS	short interframe spacing
SIR	signal-to-interference ratio
SNR	sygnal-to-noise ratio
SO	superframe order
SPDU	SSCS protocol data units
SRD	short-range device
SSCS	service-specific convergence sublayer
SUT	system under test

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

TRX	transceiver
TX	transmit or transmitter
WLAN	wireless local area network
WPAN	wireless personal area network

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5. General description

5.1 Introduction

An LR-WPAN is a simple, low-cost communication network that allows wireless connectivity in applications with limited power and relaxed throughput requirements. The main objectives of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol.

Some of the characteristics of an LR-WPAN are as follows:

- Over-the-air data rates of 250 kb/s, 100kb/s, 40 kb/s, and 20 kb/s
- Star or peer-to-peer operation
- Allocated 16-bit short or 64-bit extended addresses
- Optional allocation of guaranteed time slots (GTSs)
- Carrier sense multiple access with collision avoidance (CSMA-CA) channel access
- Fully acknowledged protocol for transfer reliability
- Low power consumption
- Energy detection (ED)
- Link quality indication (LQI)
- 16 channels in the 2450 MHz band, 30 channels in the 915 MHz band, and 3 channels in the 868 MHz band

Two different device types can participate in an IEEE 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). The FFD can operate in three modes serving as a personal area network (PAN) coordinator, a coordinator, or a device. An FFD can talk to RFDs or other FFDs, while an RFD can talk only to an FFD. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only associate with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

This standard is backward-compatible to the 2003 edition; in other words, devices conforming to this standard are capable of joining and functioning in a PAN composed of devices conforming to IEEE Std 802.15.4-2003.

5.2 Components of the IEEE 802.15.4 WPAN

A system conforming to this standard consists of several components. The most basic is the device. A device may be an RFD or an FFD. Two or more devices within a POS communicating on the same physical channel constitute a WPAN. However, this WPAN shall include at least one FFD, operating as the PAN coordinator.

An IEEE 802.15.4 network is part of the WPAN family of standards although the coverage of the network may extend beyond the POS, which typically defines the WPAN.

A well-defined coverage area does not exist for wireless media because propagation characteristics are dynamic and uncertain. Small changes in position or direction may result in drastic differences in the signal strength or quality of the communication link. These effects occur whether a device is stationary or mobile, as moving objects may impact station-to-station propagation.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.3 Network topologies

Depending on the application requirements, an IEEE 802.15.4 LR-WPAN may operate in either of two topologies: the star topology or the peer-to-peer topology. Both are shown in Figure 1. In the star topology the communication is established between devices and a single central controller, called the PAN coordinator. A device typically has some associated application and is either the initiation point or the termination point for network communications. A PAN coordinator may also have a specific application, but it can be used to initiate, terminate, or route communication around the network. The PAN coordinator is the primary controller of the PAN. All devices operating on a network of either topology shall have unique 64-bit addresses. This address may be used for direct communication within the PAN, or a short address may be allocated by the PAN coordinator when the device associates and used instead. The PAN coordinator might often be mains powered, while the devices will most likely be battery powered. Applications that benefit from a star topology include home automation, personal computer (PC) peripherals, toys and games, and personal health care.

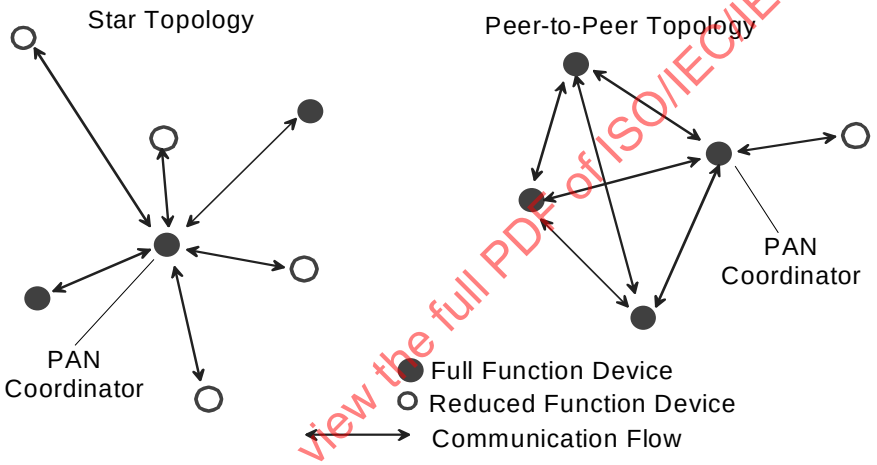


Figure 1—Star and peer-to-peer topology examples

The peer-to-peer topology also has a PAN coordinator; however, it differs from the star topology in that any device may communicate with any other device as long as they are in range of one another. Peer-to-peer topology allows more complex network formations to be implemented, such as mesh networking topology. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, intelligent agriculture, and security would benefit from such a network topology. A peer-to-peer network can be ad hoc, self-organizing, and self-healing. It may also allow multiple hops to route messages from any device to any other device on the network. Such functions can be added at the higher layer, but are not part of this standard.

Each independent PAN selects a unique identifier. This PAN identifier allows communication between devices within a network using short addresses and enables transmissions between devices across independent networks. The mechanism by which identifiers are chosen is outside the scope of this standard.

The network formation is performed by the higher layer, which is not part of this standard. However, 5.3.1 and 5.3.2 provide a brief overview on how each supported topology can be formed.

5.3.1 Star network formation

The basic structure of a star network is illustrated in Figure 1. After an FFD is activated, it can establish its own network and become the PAN coordinator. All star networks operate independently from all other star networks currently in operation. This is achieved by choosing a PAN identifier that is not currently used by any other network within the radio sphere of influence. Once the PAN identifier is chosen, the PAN

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

coordinator allows other devices, potentially both FFDs and RFDs, to join its network. The higher layer can use the procedures described in 7.5.2 and 7.5.3 to form a star network.

5.3.2 Peer-to-peer network formation

In a peer-to-peer topology, each device is capable of communicating with any other device within its radio sphere of influence. One device is nominated as the PAN coordinator, for instance, by virtue of being the first device to communicate on the channel. Further network structures are constructed out of the peer-to-peer topology and it is possible to impose topological restrictions on the formation of the network.

An example of the use of the peer-to-peer communications topology is the cluster tree. The cluster tree network is a special case of a peer-to-peer network in which most devices are FFDs. An RFD connects to a cluster tree network as a leaf device at the end of a branch because RFDs do not allow other devices to associate. Any of the FFDs may act as a coordinator and provide synchronization services to other devices or other coordinators. Only one of these coordinators can be the overall PAN coordinator, which may have greater computational resources than any other device(s) in the PAN. The PAN coordinator forms the first cluster by choosing an unused PAN identifier and broadcasting beacon frames to neighboring devices. A contention resolution mechanism is required if two or more FFDs simultaneously attempt to establish themselves as PAN coordinators; however, such a mechanism is outside the scope of this standard. A candidate device receiving a beacon frame may request to join the network at the PAN coordinator. If the PAN coordinator permits the device to join, it adds the new device as a child device in its neighbor list. Then the newly joined device adds the PAN coordinator as its parent in its neighbor list and begins transmitting periodic beacons; other candidate devices may then join the network at that device. If the original candidate device is not able to join the network at the PAN coordinator, it will search for another parent device. The detailed procedures describing how a PAN is started and how devices join a PAN are found in 7.5.2 and 7.5.3.

The simplest form of a cluster tree network is a single cluster network, but larger networks are possible by forming a mesh of multiple neighboring clusters. Once predetermined application or network requirements are met, the first PAN coordinator may instruct a device to become the PAN coordinator of a new cluster adjacent to the first one. Other devices gradually connect and form a multicluster network structure, such as the one seen in Figure 2. The lines in Figure 2 represent the parent-child relationships of the devices and not the communication flow. The advantage of a multicluster structure is increased coverage area, while the disadvantage is an increase in message latency.

5.4 Architecture

The IEEE 802.15.4 architecture is defined in terms of a number of blocks in order to simplify the standard. These blocks are called layers. Each layer is responsible for one part of the standard and offers services to the higher layers. The layout of the blocks is based on the open systems interconnection (OSI) seven-layer model (see ISO/IEC 7498-1:1994 [B12]).

The interfaces between the layers serve to define the logical links that are described in this standard.

An LR-WPAN device comprises a PHY, which contains the radio frequency (RF) transceiver along with its low-level control mechanism, and a MAC sublayer that provides access to the physical channel for all types of transfer. Figure 3 shows these blocks in a graphical representation, which are described in more detail in 5.4.1 and 5.4.2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

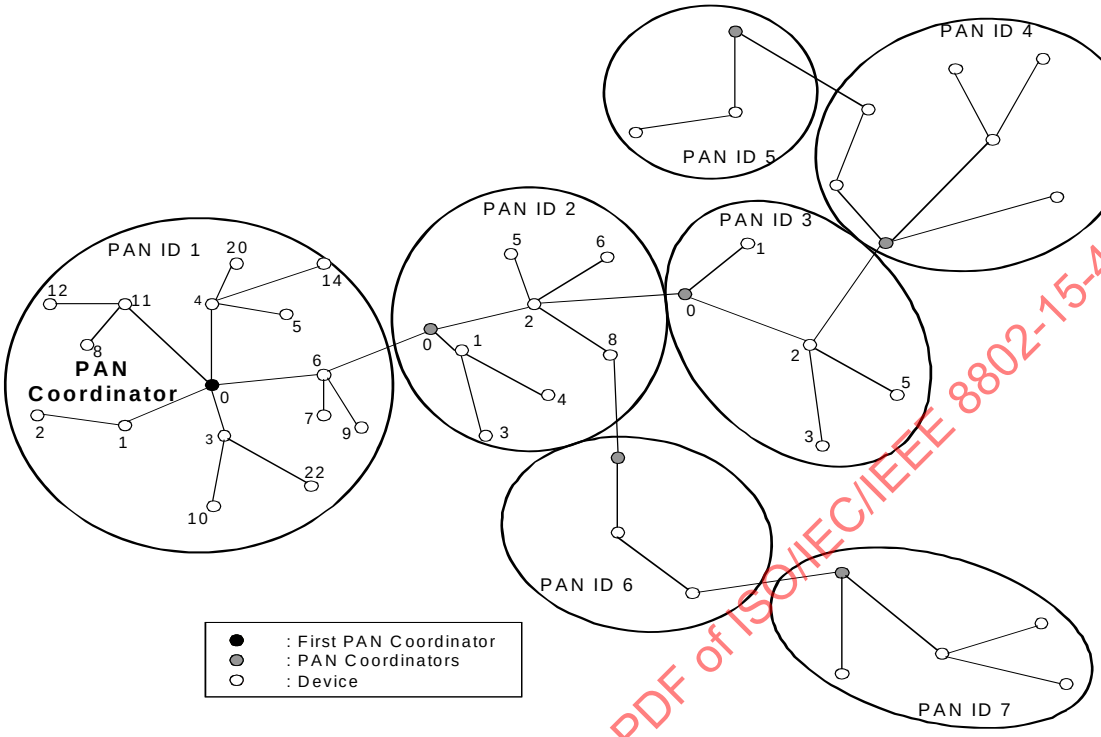
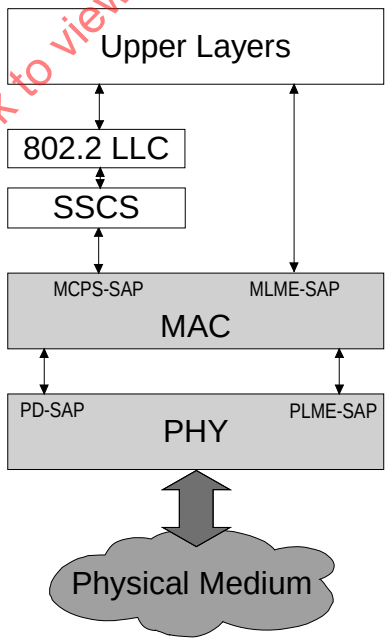


Figure 2—Cluster tree network



NOTE—For MCPS-SAP, see 7.1; for MLME-SAP, see 5.4.2; for PD-SAP, see 6.2; and for PLME-SAP, see 5.4.1.

Figure 3—LR-WPAN device architecture

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The upper layers, shown in Figure 3, consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of the device. The definition of these upper layers is outside the scope of this standard. An IEEE 802.2 Type 1 logical link control (LLC) can access the MAC sublayer through the service-specific convergence sublayer (SSCS), defined in Annex A. The LR-WPAN architecture can be implemented either as embedded devices or as devices requiring the support of an external device such as a PC.

5.4.1 Physical layer (PHY)

The PHY provides two services: the PHY data service and the PHY management service interfacing to the physical layer management entity (PLME) service access point (SAP) (known as the PLME-SAP). The PHY data service enables the transmission and reception of PHY protocol data units (PPDUs) across the physical radio channel.

Clause 6 contains the specifications for the PHY.

The features of the PHY are activation and deactivation of the radio transceiver, ED, LQI, channel selection, clear channel assessment (CCA), and transmitting as well as receiving packets across the physical medium. The radio operates at one or more of the following unlicensed bands:

— 868–868.6 MHz (e.g., Europe)
— 902–928 MHz (e.g., North America)
— 2400–2483.5 MHz (worldwide)

Refer to Annex F for an informative summary of regulatory requirements.

5.4.2 MAC sublayer

The MAC sublayer provides two services: the MAC data service and the MAC management service interfacing to the MAC sublayer management entity (MLME) service access point (SAP) (known as MLME-SAP). The MAC data service enables the transmission and reception of MAC protocol data units (MPDUs) across the PHY data service.

The features of the MAC sublayer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation. In addition, the MAC sublayer provides hooks for implementing application-appropriate security mechanisms.

Clause 7 contains the specifications for the MAC sublayer.

5.5 Functional overview

A brief overview of the general functions of a LR-WPAN is given in 5.5.1 through 5.5.6 and includes information on the superfame structure, the data transfer model, the frame structure, improving probability of successful delivery, power consumption considerations, and security.

5.5.1 Superframe structure

This standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. The superframe is bounded by network beacons sent by the coordinator (see Figure 4a) and is divided into 16 equally sized slots. Optionally, the superframe can have an active and an inactive portion (see Figure 4b). During the inactive portion, the coordinator may enter a low-power mode. The beacon frame is transmitted in the first slot of each superframe. If a coordinator does not wish to use a superframe structure, it will turn off the beacon transmissions. The beacons are used to synchronize the attached

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

devices, to identify the PAN, and to describe the structure of the superframes. Any device wishing to communicate during the contention access period (CAP) between two beacons competes with other devices using a slotted CSMA-CA mechanism. All transactions are completed by the time of the next network beacon.

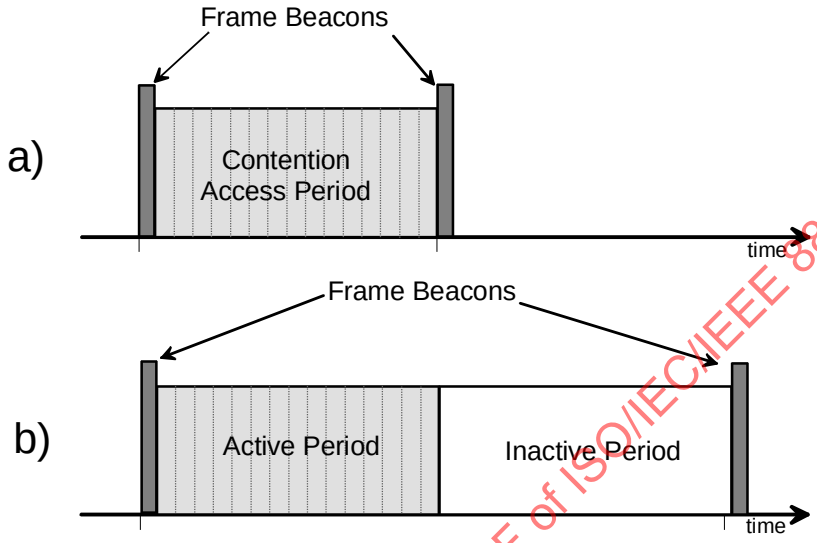


Figure 4—Superframe structure without GTSs

For low-latency applications or applications requiring specific data bandwidth, the PAN coordinator may dedicate portions of the active superframe to that application. These portions are called guaranteed time slots (GTSs). The GTSs form the contention-free period (CFP), which always appears at the end of the active superframe starting at a slot boundary immediately following the CAP, as shown in Figure 5. The PAN coordinator may allocate up to seven of these GTSs, and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP remains for contention-based access of other networked devices or new devices wishing to join the network. All contention-based transactions is completed before the CFP begins. Also each device transmitting in a GTS ensures that its transaction is complete before the time of the next GTS or the end of the CFP. More information on the superframe structure can be found in 7.5.1.1.

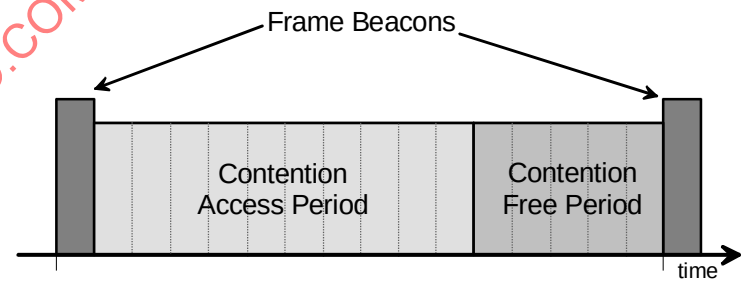


Figure 5—Superframe structure with GTSs

5.5.2 Data transfer model

Three types of data transfer transactions exist. The first one is the data transfer to a coordinator in which a device transmits the data. The second transaction is the data transfer from a coordinator in which the device receives the data. The third transaction is the data transfer between two peer devices. In star topology, only two of these transactions are used because data may be exchanged only between the coordinator and a device. In a peer-to-peer topology, data may be exchanged between any two devices on the network; consequently all three transactions may be used in this topology.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The mechanisms for each transfer type depend on whether the network supports the transmission of beacons. A beacon-enabled PAN is used in networks that either require synchronization or support for low-latency devices, such as PC peripherals. If the network does not need synchronization or support for low-latency devices, it can elect not to use the beacon for normal transfers. However, the beacon is still required for network discovery. The structure of the frames used for the data transfer is specified in 7.2.

5.5.2.1 Data transfer to a coordinator

When a device wishes to transfer data to a coordinator in a beacon-enabled PAN, it first listens for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate time, the device transmits its data frame, using slotted CSMA-CA, to the coordinator. The coordinator may acknowledge the successful reception of the data by transmitting an optional acknowledgment frame. This sequence is summarized in Figure 6.

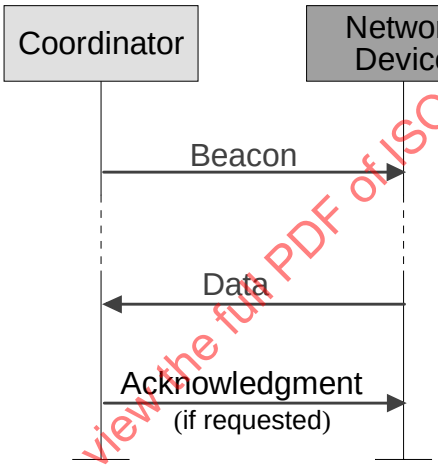


Figure 6—Communication to a coordinator in a beacon-enabled PAN

When a device wishes to transfer data in a nonbeacon-enabled PAN, it simply transmits its data frame, using unslotted CSMA-CA, to the coordinator. The coordinator acknowledges the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete. This sequence is summarized in Figure 7.

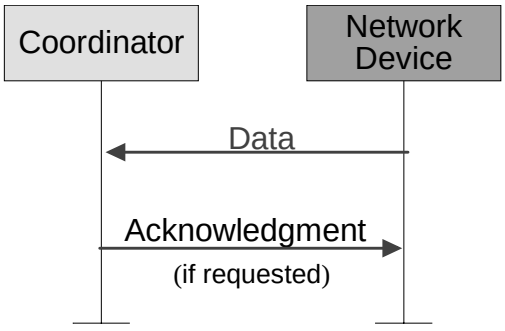


Figure 7—Communication to a coordinator in a nonbeacon-enabled PAN

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.5.2.2 Data transfer from a coordinator

When the coordinator wishes to transfer data to a device in a beacon-enabled PAN, it indicates in the network beacon that the data message is pending. The device periodically listens to the network beacon and, if a message is pending, transmits a MAC command requesting the data, using slotted CSMA-CA. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. The pending data frame is then sent using slotted CSMA-CA or, if possible, immediately after the acknowledgment (see 7.5.6.3). The device may acknowledge the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete. Upon successful completion of the data transaction, the message is removed from the list of pending messages in the beacon. This sequence is summarized in Figure 8.

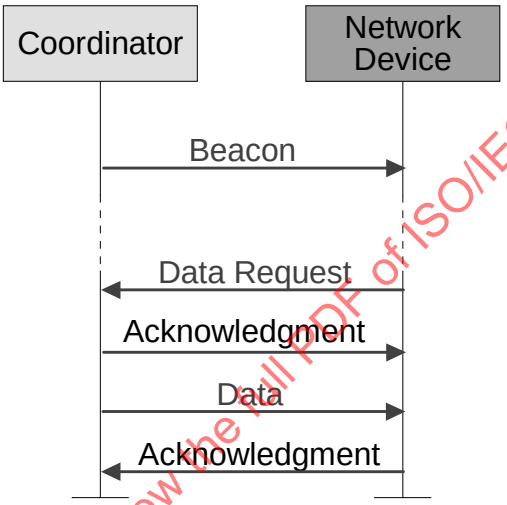


Figure 8—Communication from a coordinator a beacon-enabled PAN

When a coordinator wishes to transfer data to a device in a nonbeacon-enabled PAN, it stores the data for the appropriate device to make contact and request the data. A device may make contact by transmitting a MAC command requesting the data, using unslotted CSMA-CA, to its coordinator at an application-defined rate. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. If a data frame is pending, the coordinator transmits the data frame, using unslotted CSMA-CA, to the device. If a data frame is not pending, the coordinator indicates this fact either in the acknowledgment frame following the data request or in a data frame with a zero-length payload (see 7.5.6.3). If requested, the device acknowledges the successful reception of the data frame by transmitting an acknowledgment frame. This sequence is summarized in Figure 9.

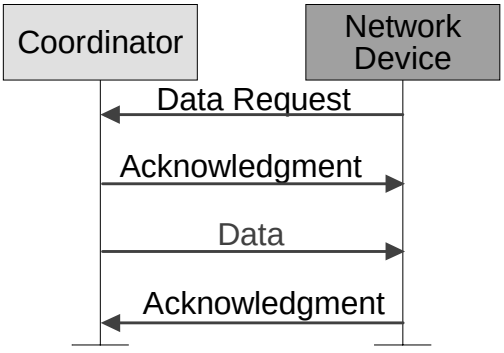


Figure 9—Communication from a coordinator in a nonbeacon-enabled PAN

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.5.2.3 Peer-to-peer data transfers

In a peer-to-peer PAN, every device may communicate with every other device in its radio sphere of influence. In order to do this effectively, the devices wishing to communicate will need to either receive constantly or synchronize with each other. In the former case, the device can simply transmit its data using unslotted CSMA-CA. In the latter case, other measures need to be taken in order to achieve synchronization. Such measures are beyond the scope of this standard.

5.5.3 Frame structure

The frame structures have been designed to keep the complexity to a minimum while at the same time making them sufficiently robust for transmission on a noisy channel. Each successive protocol layer adds to the structure with layer-specific headers and footers. This standard defines four frame structures:

— A beacon frame, used by a coordinator to transmit beacons

— A data frame, used for all transfers of data

— An acknowledgment frame, used for confirming successful frame reception

— A MAC command frame, used for handling all MAC peer entity control transfers

The structure of each of the four frame types is described in 5.5.3.1 through 5.5.3.4. The diagrams in these subclauses illustrate the fields that are added by each layer of the protocol.

5.5.3.1 Beacon frame

Figure 10 shows the structure of the beacon frame, which originates from within the MAC sublayer. A coordinator can transmit network beacons in a beacon-enabled PAN. The MAC payload contains the superframe specification, GTS fields, pending address fields, and beacon payload (see 7.2.2.1). The MAC payload is prefixed with a MAC header (MHR) and appended with a MAC footer (MFR). The MHR contains the MAC Frame Control field, beacon sequence number (BSN), addressing fields, and optionally the auxiliary security header. The MFR contains a 16-bit frame check sequence (FCS). The MHR, MAC payload, and MFR together form the MAC beacon frame (i.e., MPDU).

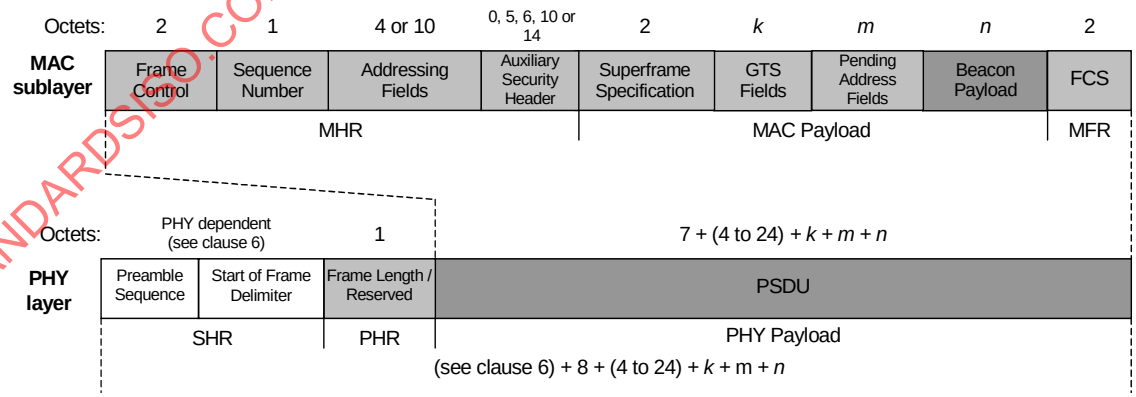


Figure 10—Schematic view of the beacon frame and the PHY packet

The MAC beacon frame is then passed to the PHY as the PHY service data unit (PSDU), which becomes the PHY payload. The PHY payload is prefixed with a synchronization header (SHR), containing the Preamble Sequence and Start-of-Frame Delimiter (SFD) fields, and a PHY header (PHR) containing the length of the PHY payload in octets. The SHR, PHR, and PHY payload together form the PHY packet (i.e., PPDU).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.5.3.2 Data frame

Figure 11 shows the structure of the data frame, which originates from the upper layers.

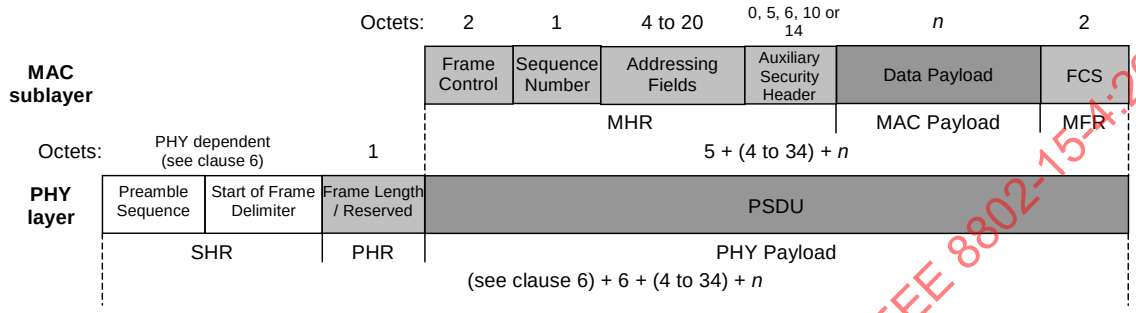


Figure 11—Schematic view of the data frame and the PHY packet

The data payload is passed to the MAC sublayer and is referred to as the MAC service data unit (MSDU). The MAC payload is prefixed with an MHR and appended with an MFR. The MHR contains the Frame Control field, data sequence number (DSN), addressing fields, and optionally the auxiliary security header. The MFR is composed of a 16-bit FCS. The MHR, MAC payload, and MFR together form the MAC data frame, (i.e., MPDU).

The MPDU is passed to the PHY as the PSDU, which becomes the PHY payload. The PHY payload is prefixed with an SHR, containing the Preamble Sequence and SFD fields, and a PHR containing the length of the PHY payload in octets. The preamble sequence and the data SFD enable the receiver to achieve symbol synchronization. The SHR, PHR, and PHY payload together form the PHY packet, (i.e., PPDU).

5.5.3.3 Acknowledgment frame

Figure 12 shows the structure of the acknowledgment frame, which originates from within the MAC sublayer. The MAC acknowledgment frame is constructed from an MHR and an MFR; it has no MAC payload. The MHR contains the MAC Frame Control field and DSN. The MFR is composed of a 16-bit FCS. The MHR and MFR together form the MAC acknowledgment frame (i.e., MPDU).

The MPDU is passed to the PHY as the PSDU, which becomes the PHY payload. The PHY payload is prefixed with the SHR, containing the Preamble Sequence and SFD fields, and the PHR containing the length of the PHY payload in octets. The SHR, PHR, and PHY payload together form the PHY packet, (i.e., PPDU).

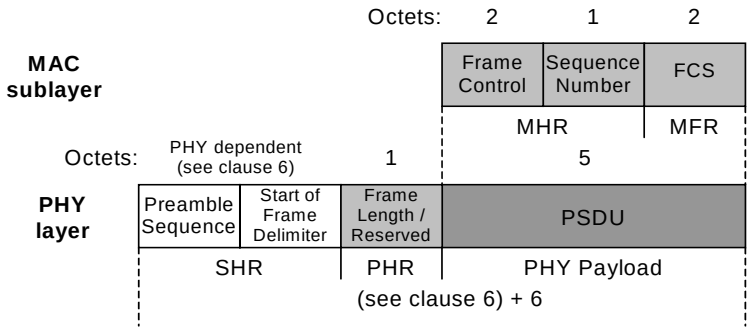


Figure 12—Schematic view of the acknowledgment frame and the PHY packet

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.5.3.4 MAC command frame

Figure 13 shows the structure of the MAC command frame, which originates from within the MAC sublayer. The MAC payload contains the Command Type field and the command payload (see 7.2.2.4). The MAC payload is prefixed with an MHR and appended with an MFR. The MHR contains the MAC Frame Control field, DSN, addressing fields, and optionally the auxiliary security header. The MFR contains a 16-bit FCS. The MHR, MAC payload, and MFR together form the MAC command frame, (i.e., MPDU).

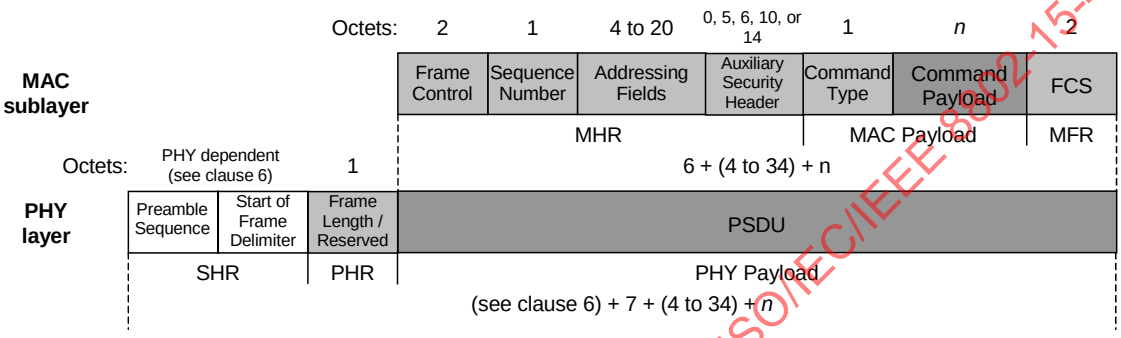


Figure 13—Schematic view of the MAC command frame and the PHY packet

The MPDU is then passed to the PHY as the PSDU, which becomes the PHY payload. The PHY payload is prefixed with an SHR, containing the Preamble Sequence and SFD fields, and a PHR containing the length of the PHY payload in octets. The preamble sequence enables the receiver to achieve symbol synchronization. The SHR, PHR, and PHY payload together form the PHY packet, (i.e., PPDU).

5.5.4 Improving probability of successful delivery

The IEEE 802.15.4 LR-WPAN employs various mechanisms to improve the probability of successful data transmission. These mechanisms are the CSMA-CA mechanism, frame acknowledgment, and data verification and are briefly discussed in 5.5.4.1 through 5.5.4.3.

5.5.4.1 CSMA-CA mechanism

The IEEE 802.15.4 LR-WPAN uses two types of channel access mechanism, depending on the network configuration. Nonbeacon-enabled PANs use an unslotted CSMA-CA channel access mechanism, as described in 7.5.1. Each time a device wishes to transmit data frames or MAC commands, it waits for a random period. If the channel is found to be idle, following the random backoff, the device transmits its data. If the channel is found to be busy following the random backoff, the device waits for another random period before trying to access the channel again. Acknowledgment frames are sent without using a CSMA-CA mechanism.

Beacon-enabled PANs use a slotted CSMA-CA channel access mechanism, where the backoff slots are aligned with the start of the beacon transmission. The backoff slots of all devices within one PAN are aligned to the PAN coordinator. Each time a device wishes to transmit data frames during the CAP, it locates the boundary of the next backoff slot and then waits for a random number of backoff slots. If the channel is busy, following this random backoff, the device waits for another random number of backoff slots before trying to access the channel again. If the channel is idle, the device begins transmitting on the next available backoff slot boundary. Acknowledgment and beacon frames are sent without using a CSMA-CA mechanism.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

5.5.4.2 Frame acknowledgment

A successful reception and validation of a data or MAC command frame can be optionally confirmed with an acknowledgment, as described in 7.5.6.4. If the receiving device is unable to handle the received data frame for any reason, the message is not acknowledged.

If the originator does not receive an acknowledgment after some period, it assumes that the transmission was unsuccessful and retries the frame transmission. If an acknowledgment is still not received after several retries, the originator can choose either to terminate the transaction or to try again. When the acknowledgment is not required, the originator assumes the transmission was successful.

5.5.4.3 Data verification

In order to detect bit errors, an FCS mechanism employing a 16-bit International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) cyclic redundancy check (CRC) is used to detect errors in every frame.

The FCS mechanism is discussed in 7.2.1.9.

5.5.5 Power consumption considerations

In many applications that use this standard, devices will be battery powered, and battery replacement or recharging in relatively short intervals is impractical. Therefore, the power consumption is of significant concern. This standard was developed with limited power supply availability in mind. However, the physical implementation of this standard will require additional power management considerations that are beyond the scope of this standard.

The protocol has been developed to favor battery-powered devices. However, in certain applications, some of these devices could potentially be mains powered. Battery-powered devices will require duty-cycling to reduce power consumption. These devices will spend most of their operational life in a sleep state; however, each device periodically listens to the RF channel in order to determine whether a message is pending. This mechanism allows the application designer to decide on the balance between battery consumption and message latency. Higher powered devices have the option of listening to the RF channel continuously.

5.5.6 Security

From a security perspective, wireless ad hoc networks are no different from any other wireless network. They are vulnerable to passive eavesdropping attacks and potentially even active tampering because physical access to the wire is not required to participate in communications. The very nature of ad hoc networks and their cost objectives impose additional security constraints, which perhaps make these networks the most difficult environments to secure. Devices are low-cost and have limited capabilities in terms of computing power, available storage, and power drain; and it cannot always be assumed they have a trusted computing base nor a high-quality random number generator aboard. Communications cannot rely on the online availability of a fixed infrastructure and might involve short-term relationships between devices that may never have communicated before. These constraints might severely limit the choice of cryptographic algorithms and protocols and would influence the design of the security architecture because the establishment and maintenance of trust relationships between devices need to be addressed with care. In addition, battery lifetime and cost constraints put severe limits on the security overhead these networks can tolerate, something that is of far less concern with higher bandwidth networks. Most of these security architectural elements can be implemented at higher layers and may, therefore, be considered to be outside the scope of this standard.

The cryptographic mechanism in this standard is based on symmetric-key cryptography and uses keys that are provided by higher layer processes. The establishment and maintenance of these keys are outside the

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

scope of this standard. The mechanism assumes a secure implementation of cryptographic operations and secure and authentic storage of keying material.

The cryptographic mechanism provides particular combinations of the following security services:

— *Data confidentiality:* Assurance that transmitted information is only disclosed to parties for which it is intended.

— *Data authenticity:* Assurance of the source of transmitted information (and, hereby, that information was not modified in transit).

— *Replay protection:* Assurance that duplicate information is detected.

The actual frame protection provided can be adapted on a frame-by-frame basis and allows for varying levels of data authenticity (to minimize security overhead in transmitted frames where required) and for optional data confidentiality. When nontrivial protection is required, replay protection is always provided.

Cryptographic frame protection may use a key shared between two peer devices (link key) or a key shared among a group of devices (group key), thus allowing some flexibility and application-specific tradeoffs between key storage and key maintenance costs versus the cryptographic protection provided. If a group key is used for peer-to-peer communication, protection is provided only against outsider devices and not against potential malicious devices in the key-sharing group.

For more detailed information on the cryptographic security mechanisms used for protected MAC frames following this standard, refer to Clause 7.

5.6 Concept of primitives

This subclause provides a brief overview of the concept of service primitives. Refer to ISO/IEC 8802.2[5] for more detailed information.

The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 14, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

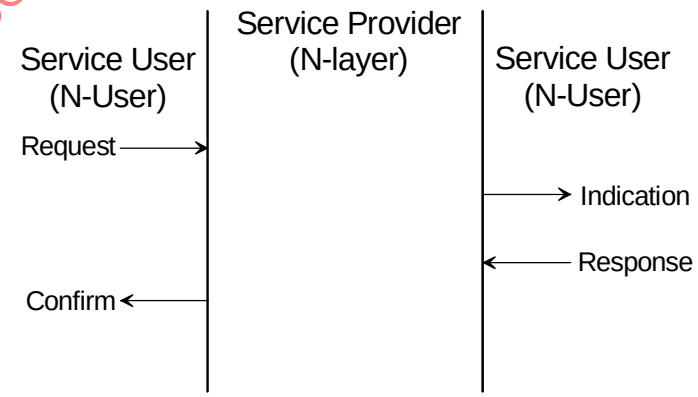


Figure 14—Service primitives

[5]For information on references, see Clause 2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The services are specified by describing the information flow between the N-user and the N-layer. This information flow is modeled by discrete, instantaneous events, which characterize the provision of a service. Each event consists of passing a service primitive from one layer to the other through a layer SAP associated with an N-user. Service primitives convey the required information by providing a particular service. These service primitives are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation.

A service is specified by describing the service primitives and parameters that characterize it. A service may have one or more related primitives that constitute the activity that is related to that particular service. Each service primitive may have zero or more parameters that convey the information required to provide the service.

A primitive can be one of four generic types:

— *Request:* The request primitive is passed from the N-user to the N-layer to request that a service is initiated.

— *Indication:* The indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event.

— *Response:* The response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive.

— *Confirm:* The confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6. PHY specification

6.1 General requirements and definitions

The PHY is responsible for the following tasks:

- Activation and deactivation of the radio transceiver
- Energy detection (ED) within the current channel
- Link quality indicator (LQI) for received packets
- Clear channel assessment (CCA) for carrier sense multiple access with collision avoidance (CSMA-CA)
- Channel frequency selection
- Data transmission and reception

Constants and attributes that are specified and maintained by the PHY are written in the text of this clause in italics. Constants have a general prefix of "a", e.g., *aMaxPHYPacketSize*, and are listed in Table 22 (in 6.4.1). Attributes have a general prefix of "phy", e.g., *phyCurrentChannel*, and are listed in Table 23 (in 6.4.2).

This subclause specifies requirements that are common to all of the PHYs that conform to this standard.

The standard specifies the following four PHYs:

- An 868/915 MHz direct sequence spread spectrum (DSSS) PHY employing binary phase-shift keying (BPSK) modulation
- An 868/915 MHz DSSS PHY employing offset quadrature phase-shift keying (O-QPSK) modulation
- An 868/915 MHz parallel sequence spread spectrum (PSSS) PHY employing BPSK and amplitude shift keying (ASK) modulation
- A 2450 MHz DSSS PHY employing O-QPSK modulation

In addition to the 868/915 MHz BPSK PHY, which was originally specified in the 2003 edition of this standard, two optional high-data-rate PHYs are specified for the 868/915 MHz bands, offering a tradeoff between complexity and data rate. Both optional PHYs offer a data rate much higher than that of the 868/915 MHz BPSK PHY, which provides for 20 kb/s in the 868 MHz band and 40 kb/s in the 915 MHz band. The ASK[6] PHY offers data rates of 250 kb/s in both the 868 MHz and 915 MHz bands, which is equal to that of the 2.4 GHz band PHY. The O-QPSK PHY, which offers a signaling scheme identical to that of the 2.4 GHz band PHY, offers a data rate in the 915 MHz band equal to that of the 2.4 GHz band PHY and a data rate of 100 kb/s in the 868 MHz band.

6.1.1 Operating frequency range

A compliant device shall operate in one or several frequency bands using the modulation and spreading formats summarized in Table 1.

Devices shall start in the mode (PHY) they are instructed to do so. If the device is capable of operating in the 868/915 MHz bands using one of the optional PHYs, it shall be able to switch dynamically between the optional 868/915 MHz band PHY and the mandatory 868/915 MHz BPSK PHYs when instructed to do so.

[6]The 868/915 MHz band ASK PHYs use BPSK modulation for the SHR and ASK modulation for the remainder of the PPDU.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 1—Frequency bands and data rates

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
868/915 (optional)	868–868.6	400	ASK	250	12.5	20-bit PSSS
	902–928	1600	ASK	250	50	5-bit PSSS
868/915 (optional)	868–868.6	400	O-QPSK	100	25	16-ary Orthogonal
	902–928	1000	O-QPSK	250	62.5	16-ary Orthogonal
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

This standard is intended to conform with established regulations in Europe, Japan, Canada, and the United States. The regulatory documents listed below are for information only and are subject to change and revisions at any time. Devices conforming to this standard shall also comply with specific regional legislation. Additional regulatory information is provided in Annex F.

Europe:

- Approval standards: European Telecommunications Standards Institute (ETSI)
- Documents: ETSI EN 300 328-1 [B26], ETSI EN 300 328-2 [B27], ETSI EN 300 220-1 [B25], ERC Recommendation 70-03 [B24]
- Approval authority: National type approval authorities

Japan:

- Approval standards: Association of Radio Industries and Businesses (ARIB)
- Document: ARIB STD-T66 [B22]
- Approval authority: Ministry of Public Management, Home Affairs, Posts and Telecommunications (MPHPT)

United States:

- Approval standards: Federal Communications Commission (FCC), United States
- Document: FCC CFR47, Section 15.247 [B29]

Canada:

- Approval standards: Industry Canada (IC), Canada
- Document: GL36 [B32]

6.1.2 Channel assignments

The introduction of the “868/915 MHz band (optional) amplitude shift keying (ASK) PHY specifications” and “868/915 MHz band (optional) O-QPSK PHY specifications” results in the total number of channel assignments exceeding the channel numbering capability of 32 channel numbers that was defined in the 2003 edition of this standard.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

To support the growing number of channels, channel assignments shall be defined through a combination of channel numbers and channels pages.

The upper 5 most significant bits (MSBs) of the 32-bit channel bitmaps in *phyChannelsSupported* shall be used as an integer value to specify 32 possible channel pages. The lower 27 bits of the channel bit map shall be used as a bit mask to specify channel numbers within a channel page.

6.1.2.1 Channel numbering

A total of 27 channels numbered 0 to 26 are available per channel page.

For channel page 0, 27 channels numbered 0 to 26 are available across the three frequency bands. Sixteen channels are available in the 2450 MHz band, 10 in the 915 MHz band, and 1 in the 868 MHz band. This channel page supports the channels defined in the 2003 edition of this standard. The center frequency of these channels is defined as follows:

F_c = 868.3 in megahertz, for k = 0

F_c = 906 + 2 (k – 1) in megahertz, for k = 1, 2, …, 10

and F_c = 2405 + 5 (k – 11) in megahertz, for k = 11, 12, …, 26

where
k is the channel number.

For channel pages 1 and 2, 11 channels numbered 0 to 10 are available across the two frequency bands to support the 868/915 MHz ASK and O-QPSK PHYs, respectively. Ten channels are available in the 915 MHz band and 1 in the 868 MHz band. The center frequency of these channels is defined as follows:

F_c = 868.3 in megahertz, for k = 0

and F_c = 906 + 2 (k – 1) in megahertz, for k = 1, 2, …, 10

where
k is the channel number.

For each PHY supported, a compliant device shall support all channels allowed by regulations for the region in which the device operates.

6.1.2.2 Channel pages

A total of 32 channel pages are available with channel pages 3 to 31 being reserved for future use. The *phyPagesSupported* PHY PAN information base (PIB) attribute indicates which channel pages are supported by the current PHY, while the *phyCurrentPage* PHY PIB attribute identifies the channel page that is currently used. The PHY PIB attributes are described in 6.4.2.

If the requested PHY PIB attribute is the *phyCurrentPage*, the attribute was successfully set to a different value from the current value, and the channel is no longer valid, then the PHY shall also set the *phyCurrentChannel* to the lowest valid channel for the requested page.

The channel pages and associated channel numbers are shown in Table 2.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 2—Channel page and channel number

Channel page (decimal)	Channel page (binary) (b_{31}, b_{30}, b_{29}, b_{28}, b_{27})	Channel number(s) (decimal)	Channel number description
0	0 0 0 0 0	0	Channel 0 is in 868 MHz band using BPSK
		1–10	Channels 1 to 10 are in 915 MHz band using BPSK
		11–26	Channels 11 to 26 are in 2.4 GHz band using O-QPSK
1	0 0 0 0 1	0	Channel 0 is in 868 MHz band using ASK
		1–10	Channels 1 to 10 are in 915 MHz band using ASK
		11–26	Reserved
2	0 0 0 1 0	0	Channel 0 is in 868 MHz band using O-QPSK
		1–10	Channels 1 to 10 are in 915 MHz band using O-QPSK
		11–26	Reserved
3–31	0 0 0 1 1 - 1 1 1 1	reserved	Reserved

For example, the bitmap for channel 4 of channel page 2 would be as follows:

0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

6.1.3 Minimum long interframe spacing (LIFS) and short interframe spacing (SIFS) periods

The minimum LIFS and SIFS periods for each of the PHYs are shown in Table 3. The description, use, and illustration of LIFS and SIFS is shown in Figure 68 (in 7.5.1.3).

Table 3—Minimum LIFS and SIFS period

PHY	*macMinLIFSPeriod*	*macMinSIFSPeriod*	Units
868–868.6 MHz BPSK	40	12	Symbols
902–928 MHz BPSK	40	12	Symbols
868–868.6 MHz ASK	40	12	Symbols
902–928 MHz ASK	40	12	Symbols
868–868.6 MHz O-QPSK	40	12	Symbols
902–928 MHz O-QPSK	40	12	Symbols
2400–2483.5 MHz O-QPSK	40	12	Symbols

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.1.4 RF power measurement

Unless otherwise stated, all RF power measurements, either transmit or receive, shall be made at the appropriate transceiver to antenna connector. The measurements shall be made with equipment that is either matched to the impedance of the antenna connector or corrected for any mismatch. For devices without an antenna connector, the measurements shall be interpreted as effective isotropic radiated power (EIRP) (i.e., a 0 dBi gain antenna), and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

6.1.5 Transmit power

The maximum transmit power shall conform with local regulations. Refer to Annex F for additional information on regulatory limits. A compliant device shall have its nominal transmit power level indicated by its PHY parameter, *phyTransmitPower* (see 6.4).

6.1.6 Out-of-band spurious emission

The out-of-band spurious emissions shall conform with local regulations. Refer to Annex F for additional information on regulatory limits on out-of-band emissions.

6.1.7 Receiver sensitivity definitions

The receiver sensitivity definitions used throughout this standard are defined in Table 4.

Table 4—Receiver sensitivity definitions

Term	Definition of term	Conditions
Packet error rate (PER)	Average fraction of transmitted packets that are not correctly received.	– Average measured over random PSDU data.
Receiver sensitivity	Threshold input signal power that yields a specified PER.	– PSDU length = 20 octets. – PER < 1%. – Power measured at antenna terminals. – Interference not present.

6.2 PHY service specifications

The PHY provides an interface between the MAC sublayer and the physical radio channel, via the RF firmware and RF hardware. The PHY conceptually includes a management entity called the PLME. This entity provides the layer management service interfaces through which layer management functions may be invoked. The PLME is also responsible for maintaining a database of managed objects pertaining to the PHY. This database is referred to as the PHY PAN information base (PIB).

Figure 15 depicts the components and interfaces of the PHY.

The PHY provides two services, accessed through two SAPs: the PHY data service, accessed through the PHY data SAP (PD-SAP), and the PHY management service, accessed through the PLME-SAP.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

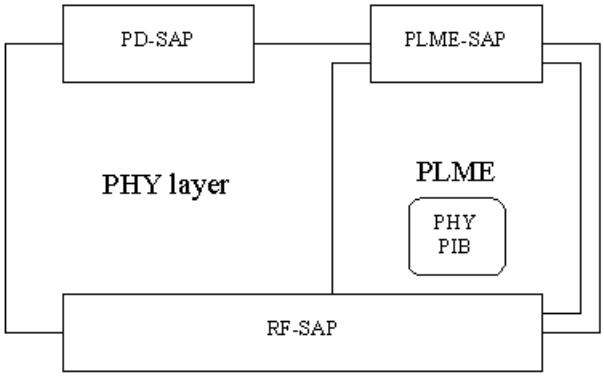


Figure 15—The PHY reference model

6.2.1 PHY data service

The PD-SAP supports the transport of MPDUs between peer MAC sublayer entities. Table 5 lists the primitives supported by the PD-SAP. These primitives are discussed in the subclauses referenced in the table.

Table 5—PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	6.2.1.1	6.2.1.2	6.2.1.3

6.2.1.1 PD-DATA.request

The PD-DATA.request primitive requests the transfer of an MPDU (i.e., PSDU) from the MAC sublayer to the local PHY entity.

6.2.1.1.1 Semantics of the service primitive

The semantics of the PD-DATA.request primitive is as follows:

```
PD-DATA.request          (
                         psduLength,
                         psdu
                         )
```

Table 6 specifies the parameters for the PD-DATA.request primitive.

Table 6—PD-DATA.request parameters

Name	Type	Valid range	Description
psduLength	Unsigned integer	≤ *aMaxPHYPacketSize*	The number of octets contained in the PSDU to be transmitted by the PHY entity.
psdu	Set of octets	—	The set of octets forming the PSDU to be transmitted by the PHY entity.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.2.1.1.2 Appropriate usage

The PD-DATA.request primitive is generated by a local MAC sublayer entity and issued to its PHY entity to request the transmission of an MPDU.

6.2.1.1.3 Effect on receipt

The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PSDU to be attempted. Provided the transmitter is enabled (TX_ON state), the PHY will first construct a PPDU, containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission, it will issue the PD-DATA.confirm primitive with a status of SUCCESS.

If the PD-DATA.request primitive is received while the receiver is enabled (RX_ON state), the PHY entity will discard the PSDU and issue the PD-DATA.confirm primitive with a status of RX_ON. If the PD-DATA.request primitive is received while the transceiver is disabled (TRX_OFF state), the PHY entity will discard the PSDU and issue the PD-DATA.confirm primitive with a status of TRX_OFF. If the PD-DATA.request primitive is received while the transmitter is already busy transmitting (BUSY_TX state), the PHY entity will discard the PSDU and issue the PD-DATA.confirm primitive with a status of BUSY_TX.

6.2.1.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of an MPDU (i.e., PSDU) from a local PHY entity to a peer PHY entity.

6.2.1.2.1 Semantics of the service primitive

The semantics of the PD-DATA.confirm primitive is as follows:

```
PD-DATA.confirm         (
                        status
                        )
```

Table 7 specifies the parameters for the PD-DATA.confirm primitive.

Table 7—PD-DATA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF, or BUSY_TX	The result of the request to transmit a packet.

6.2.1.2.2 When generated

The PD-DATA.confirm primitive is generated by the PHY entity and issued to its MAC sublayer entity in response to a PD-DATA.request primitive. The PD-DATA.confirm primitive will return a status of either SUCCESS, indicating that the request to transmit was successful, or an error code of RX_ON, TRX_OFF, or BUSY_TX. The reasons for these status values are fully described in 6.2.1.1.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.2.1.2.3 Appropriate usage

On receipt of the PD-DATA.confirm primitive, the MAC sublayer entity is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.1.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of an MPDU (i.e., PSDU) from the PHY to the local MAC sublayer entity.

6.2.1.3.1 Semantics of the service primitive

The semantics of the PD-DATA.indication primitive is as follows:

```
PD-DATA.indication      (
                        psduLength,
                        psdu,
                        ppduLinkQuality
                        )
```

Table 8 specifies the parameters for the PD-DATA.indication primitive.

Table 8—PD-DATA.indication parameters

Name	Type	Valid range	Description
psduLength	Unsigned Integer	≤ *aMaxPHYPacketSize*	The number of octets contained in the PSDU received by the PHY entity.
psdu	Set of octets	—	The set of octets forming the PSDU received by the PHY entity.
ppduLinkQuality	Integer	0x00–0xff	Link quality (LQI) value measured during reception of the PPDU (see 6.9.8).

6.2.1.3.2 When generated

The PD-DATA.indication primitive is generated by the PHY entity and issued to its MAC sublayer entity to transfer a received PSDU. This primitive will not be generated if the received psduLength field is zero or greater than *aMaxPHYPacketSize*.

6.2.1.3.3 Appropriate usage

On receipt of the PD-DATA.indication primitive, the MAC sublayer is notified of the arrival of an MPDU across the PHY data service.

6.2.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME and the PLME. Table 9 lists the primitives supported by the PLME-SAP. These primitives are discussed in the clauses referenced in the table.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 9—PLME-SAP primitives

PLME-SAP primitive	Request	Confirm
PLME-CCA	6.2.2.1	6.2.2.2
PLME-ED	6.2.2.3	6.2.2.4
PLME-GET	6.2.2.5	6.2.2.6
PLME-SET-TRX-STATE	6.2.2.7	6.2.2.8
PLME-SET	6.2.2.9	6.2.2.10

6.2.2.1 PLME-CCA.request

The PLME-CCA.request primitive requests that the PLME perform a CCA as defined in 6.9.9.

6.2.2.1.1 Semantics of the service primitive

The semantics of the PLME-CCA.request primitive is as follows:

```
PLME-CCA.request        ()
```

There are no parameters associated with the PLME-CCA.request primitive.

6.2.2.1.2 Appropriate usage

The PLME-CCA.request primitive is generated by the MLME and issued to its PLME whenever the CSMA-CA algorithm requires an assessment of the channel.

6.2.2.1.3 Effect on receipt

If the receiver is enabled on receipt of the PLME-CCA.request primitive, the PLME will cause the PHY to perform a CCA. When the PHY has completed the CCA, the PLME will issue the PLME-CCA.confirm primitive with a status of either BUSY or IDLE, depending on the result of the CCA.

If the PLME-CCA.request primitive is received while the transceiver is disabled (TRX_OFF state) or if the transmitter is enabled (TX_ON state), the PLME will issue the PLME-CCA.confirm primitive with a status of TRX_OFF or BUSY, respectively.

6.2.2.2 PLME-CCA.confirm

The PLME-CCA.confirm primitive reports the results of a CCA as defined in 6.9.9.

6.2.2.2.1 Semantics of the service primitive

The semantics of the PLME-CCA.confirm primitive is as follows:

```
PLME-CCA.confirm        (
                        status
                        )
```

Table 10 specifies the parameters for the PLME-CCA.confirm primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 10—PLME-CCA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	TRX_OFF, BUSY, or IDLE	The result of the request to perform a CCA.

6.2.2.2.2 When generated

The PLME-CCA.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-CCA.request primitive. The PLME-CCA.confirm primitive will return a status of either BUSY or IDLE, indicating a successful CCA, or an error code of TRX_OFF. The reasons for these status values are fully described in 6.2.2.1.3.

6.2.2.2.3 Appropriate usage

On receipt of the PLME-CCA.confirm primitive, the MLME is notified of the results of the CCA. If the CCA attempt was successful, the status parameter is set to either BUSY or IDLE. Otherwise, the status parameter will indicate the error.

6.2.2.3 PLME-ED.request

The PLME-ED.request primitive requests that the PLME perform an ED measurement (see 6.9.7).

6.2.2.3.1 Semantics of the service primitive

The semantics of the PLME-ED.request primitive is as follows:

PLME-ED.request ()

There are no parameters associated with the PLME-ED.request primitive.

6.2.2.3.2 Appropriate usage

The PLME-ED.request primitive is generated by the MLME and issued to its PLME to request an ED measurement.

6.2.2.3.3 Effect on receipt

If the receiver is enabled on receipt of the PLME-ED.request primitive, the PLME will cause the PHY to perform an ED measurement. When the PHY has completed the ED measurement, the PLME will issue the PLME-ED.confirm primitive with a status of SUCCESS.

If the PLME-ED.request primitive is received while the transceiver is disabled (TRX_OFF state) or if the transmitter is enabled (TX_ON state), the PLME will issue the PLME-ED.confirm primitive with a status of TRX_OFF or TX_ON, respectively.

6.2.2.4 PLME-ED.confirm

The PLME-ED.confirm primitive reports the results of the ED measurement as defined in 6.9.7.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.2.2.4.1 Semantics of the service primitive

The semantics of the PLME-ED.confirm primitive is as follows:

```
PLME-ED.confirm          (
                         status,
                         EnergyLevel
                         )
```

Table 11 specifies the parameters for the PLME-ED.confirm primitive.

Table 11—PLME-ED.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, TRX_OFF, or TX_ON	The result of the request to perform an ED measurement.
EnergyLevel	Integer	0x00–0xff	ED level for the current channel. If status is set to SUCCESS, this is the ED level for the current channel. Otherwise, the value of this parameter will be ignored.

6.2.2.4.2 When generated

The PLME-ED.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-ED.request primitive. The PLME-ED.confirm primitive will return a status of SUCCESS, indicating a successful ED measurement, or an error code of TRX_OFF or TX_ON. The reasons for these status values are fully described in 6.2.2.3.3.

6.2.2.4.3 Appropriate usage

On receipt of the PLME-ED.confirm primitive, the MLME is notified of the results of the ED measurement. If the ED measurement attempt was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.2.5 PLME-GET.request

The PLME-GET.request primitive requests information about a given PHY PIB attribute.

6.2.2.5.1 Semantics of the service primitive

The semantics of the PLME-GET.request primitive is as follows:

```
PLME-GET.request         (
                         PIBAttribute
                         )
```

Table 12 specifies the parameters for the PLME-GET.request primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 12—PLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	See Table 23 (in 6.4.2)	The identifier of the PHY PIB attribute to get.

6.2.2.5.2 Appropriate usage

The PLME-GET.request primitive is generated by the MLME and issued to its PLME to obtain information from the PHY PIB.

6.2.2.5.3 Effect on receipt

On receipt of the PLME-GET.request primitive, the PLME will attempt to retrieve the requested PHY PIB attribute from its database. If the identifier of the PIB attribute is not found in the database, the PLME will issue the PLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested PHY PIB attribute is successfully retrieved, the PLME will issue the PLME-GET.confirm primitive with a status of SUCCESS.

6.2.2.6 PLME-GET.confirm

The PLME-GET.confirm primitive reports the results of an information request from the PHY PIB.

6.2.2.6.1 Semantics of the service primitive

The semantics of the PLME-GET.confirm primitive is as follows:

```
PLME-GET.confirm        (
                        status,
                        PIBAttribute,
                        PIBAttributeValue
                        )
```

Table 13 specifies the parameters for the PLME-GET.confirm primitive.

Table 13—PLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the request for PHY PIB attribute information.
PIBAttribute	Enumeration	See Table 23 (in 6.4.2)	The identifier of the PHY PIB attribute that was requested.
PIBAttributeValue	Various	Attribute specific	The value of the indicated PHY PIB attribute that was requested. This parameter has zero length when the status parameter is set to UNSUPPORTED_ATTRIBUTE.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.2.2.6.2 When generated

The PLME-GET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-GET.request primitive. The PLME-GET.confirm primitive will return a status of either SUCCESS, indicating that the request to read a PHY PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. The reasons for these status values are fully described in 6.2.2.5.3.

6.2.2.6.3 Appropriate usage

On receipt of the PLME-GET.confirm primitive, the MLME is notified of the results of its request to read a PHY PIB attribute. If the request to read a PHY PIB attribute was successful, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.2.7 PLME-SET-TRX-STATE.request

The PLME-SET-TRX-STATE.request primitive requests that the PHY entity change the internal operating state of the transceiver. The transceiver will have three main states:

— Transceiver disabled (TRX_OFF)
— Transmitter enabled (TX_ON)
— Receiver enabled (RX_ON)

6.2.2.7.1 Semantics of the service primitive

The semantics of the PLME-SET-TRX-STATE.request primitive is as follows:

```
PLME-SET-TRX-STATE.request      (
                                    state
                                )
```

Table 14 specifies the parameters for the PLME-SET-TRX-STATE.request primitive.

Table 14—PLME-SET-TRX-STATE.request parameters

Name	Type	Valid range	Description
state	Enumeration	RX_ON, TRX_OFF, FORCE_TRX_OFF, or TX_ON	The new state in which to configure the transceiver.

6.2.2.7.2 Appropriate usage

The PLME-SET-TRX-STATE.request primitive is generated by the MLME and issued to its PLME when the current operational state of the receiver needs to be changed.

6.2.2.7.3 Effect on receipt

On receipt of the PLME-SET-TRX-STATE.request primitive, the PLME will cause the PHY to change to the requested state. If the state change is accepted, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status of SUCCESS. If this primitive requests a state that the transceiver is already configured, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status indicating the current state, i.e., RX_ON, TRX_OFF, or TX_ON. If this primitive is issued with an RX_ON

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

or TRX_OFF argument and the PHY is busy transmitting a PPDU, at the end of transmission the state change will occur, and then the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status of SUCCESS. If this primitive is issued with TRX_OFF and the PHY is in RX_ON state and has already received a valid SFD, at the end of reception of the PPDU the state change will occur, and then the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status SUCCESS. If this primitive is issued with TX_ON, the PHY will cause the PHY to go to the TX_ON state irrespective of the state the PHY is in. The PHY will then issue the PLME-SET-TRX-STATE.confirm primitive with a status SUCCESS. If this primitive is issued with FORCE_TRX_OFF, the PHY will cause the PHY to go to the TRX_OFF state irrespective of the state the PHY is in. The PHY will then issue the PLME-SET-TRX-STATE.confirm primitive with a status SUCCESS.

6.2.2.8 PLME-SET-TRX-STATE.confirm

The PLME-SET-TRX-STATE.confirm primitive reports the result of a request to change the internal operating state of the transceiver.

6.2.2.8.1 Semantics of the service primitive

The semantics of the PLME-SET-TRX-STATE.confirm primitive is as follows:

```
PLME-SET-TRX-STATE.confirm      (
                                status
                                )
```

Table 15 specifies the parameters for the PLME-SET-TRX-STATE.confirm primitive.

Table 15—PLME-SET-TRX-STATE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF, or TX_ON	The result of the request to change the state of the transceiver.

6.2.2.8.2 When generated

The PLME-SET-TRX-STATE.confirm primitive is generated by the PLME and issued to its MLME after attempting to change the internal operating state of the transceiver.

6.2.2.8.3 Appropriate usage

On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MLME is notified of the result of its request to change the internal operating state of the transceiver. A status value of SUCCESS indicates that the internal operating state of the transceiver was accepted. A status value of RX_ON, TRX_OFF, or TX_ON indicates that the transceiver is already in the requested internal operating state.

6.2.2.9 PLME-SET.request

The PLME-SET.request primitive attempts to set the indicated PHY PIB attribute to the given value.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.2.2.9.1 Semantics of the service primitive

The semantics of the PLME-SET.request primitive is as follows:

```
PLME-SET.request        (
                        PIBAttribute,
                        PIBAttributeValue
                        )
```

Table 16 specifies the parameters for the PLME-SET.request primitive.

Table 16—PLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	See Table 23 (in 6.4.2)	The identifier of the PIB attribute to set.
PIBAttributeValue	Various	Attribute specific	The value of the indicated PIB attribute to set.

6.2.2.9.2 Appropriate usage

The PLME-SET.request primitive is generated by the MLME and issued to its PLME to write the indicated PHY PIB attribute.

6.2.2.9.3 Effect on receipt

On receipt of the PLME-SET.request primitive, the PLME will attempt to write the given value to the indicated PHY PIB attribute in its database. Not all PIB values are settable. If the PIBAttribute parameter specifies an attribute that is not found in the database (see Table 23 in 6.4.2), the PLME will issue the PLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttribute parameter specifies an attribute whose value is read-only, the PLME will issue the PLME-SET.confirm primitive with a status of READ_ONLY. If the PIBAttibuteValue parameter specifies a value that is out of the valid range for the given attribute, the PLME will issue the PLME-SET.confirm primitive with a status of INVALID_PARAMETER.

If the requested PHY PIB attribute is successfully written, the PLME will issue the PLME-SET.confirm primitive with a status of SUCCESS.

6.2.2.10 PLME-SET.confirm

The PLME-SET.confirm primitive reports the results of the attempt to set a PIB attribute.

6.2.2.10.1 Semantics of the service primitive

The semantics of the PLME-SET.confirm primitive is as follows:

```
PLME-SET.confirm        (
                        status,
                        PIBAttribute
                        )
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 17 specifies the parameters for the PLME-SET.confirm primitive.

Table 17—PLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER, or READ_ONLY	The status of the attempt to set the requested PIB attribute.
PIBAttribute	Enumeration	See Table 23 (in 6.4.2)	The identifier of the PIB attribute being confirmed.

6.2.2.10.2 When generated

The PLME-SET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-SET.request primitive. The PLME-SET.confirm primitive will return a status of either SUCCESS, indicating that the requested value was written to the indicated PHY PIB attribute, or an error code of UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER, or READ_ONLY. The reasons for these status values are fully described in 6.2.2.9.3.

6.2.2.10.3 Appropriate usage

On receipt of the PLME-SET.confirm primitive, the MLME is notified of the result of its request to set the value of a PHY PIB attribute. If the requested value was written to the indicated PHY PIB attribute, the status parameter is set to SUCCESS. Otherwise, the status parameter will indicate the error.

6.2.3 PHY enumerations description

Table 18 shows a description of the PHY enumeration values defined in the PHY specification.

Table 18—PHY enumerations description

Enumeration	Value	Description
BUSY	0x00	The CCA attempt has detected a busy channel.
BUSY_RX	0x01	The transceiver is asked to change its state while receiving.
BUSY_TX	0x02	The transceiver is asked to change its state while transmitting.
FORCE_TRX_OFF	0x03	The transceiver is to be switched off immediately.
IDLE	0x04	The CCA attempt has detected an idle channel.
INVALID_PARAMETER	0x05	A SET/GET request was issued with a parameter in the primitive that is out of the valid range.
RX_ON	0x06	The transceiver is in or is to be configured into the receiver enabled state.
SUCCESS	0x07	A SET/GET, an ED operation, or a transceiver state change was successful.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 18—PHY enumerations description *(continued)*

Enumeration	Value	Description
TRX_OFF	0x08	The transceiver is in or is to be configured into the transceiver disabled state.
TX_ON	0x09	The transceiver is in or is to be configured into the transmitter enabled state.
UNSUPPORTED_ATTRIBUTE	0x0a	A SET/GET request was issued with the identifier of an attribute that is not supported.
READ_ONLY	0x0b	A SET/GET request was issued with the identifier of an attribute that is read-only.

6.3 PPDU format

This subclause specifies the format of the PPDU packet.

For convenience, the PPDU packet structure is presented so that the leftmost field as written in this standard shall be transmitted or received first. All multiple octet fields shall be transmitted or received least significant octet first and each octet shall be transmitted or received least significant bit (LSB) first. The same transmission order should apply to data fields transferred between the PHY and MAC sublayer.

Each PPDU packet consists of the following basic components:

— A synchronization header (SHR), which allows a receiving device to synchronize and lock onto the bit stream

— A PHY header (PHR), which contains frame length information

— A variable length payload, which carries the MAC sublayer frame

The PPDU packet structure shall be formatted as illustrated in Figure 16.

		Octets		
		1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Figure 16—Format of the PPDU

6.3.1 Preamble field

The Preamble field is used by the transceiver to obtain chip and symbol synchronization with an incoming message. The length of the preamble for the different PHYs is shown in Table 19.

Preamble lengths for ASK are expressed in equivalent octet times as the preamble for ASK is defined using a special symbol. For all PHYs except the ASK PHY, the bits in the Preamble field shall be binary zeros. The ASK preamble format is described in 6.7.4.1.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 19—Preamble field length

PHY	Length		Duration (uS)
868–868.6 MHz BPSK	4 octets	32 symbols	1600
902–928 MHz BPSK	4 octets	32 symbols	800
868–868.6 MHz ASK	5 octets	2 symbols	160
902–928 MHz ASK	3.75 octets	6 symbols	120
868–868.6 MHz O-QPSK	4 octets	8 symbols	320
902–928 MHz O-QPSK	4 octets	8 symbols	128
2400–2483.5 MHz O-QPSK	4 octets	8 symbols	128

6.3.2 SFD field

The SFD is a field indicating the end of the SHR and the start of the packet data. The length of the SFD for the different PHYs is shown in Table 20.

Table 20—SFD field length

PHY	Length	
868–868.6 MHz BPSK	1 octet	8 symbols
902–928 MHz BPSK	1 octet	8 symbols
868–868.6 MHz ASK	2.5 octets	1 symbol
902–928 MHz ASK	0.625 octets	1 symbol
868–868.6 MHz O-QPSK	1 octet	2 symbols
902–928 MHz O-QPSK	1 octet	2 symbols
2400–2483.5 MHz O-QPSK	1 octet	2 symbols

For all PHYs, except for the ASK PHY, the SFD is an 8-bit field. For the ASK PHY, the SFD is defined using a special symbol. The lengths of the SFD for the ASK PHY are expressed in equivalent octet times. The SFD for all PHYs except the ASK PHY shall be formatted as illustrated in Figure 17. The SFD for the ASK PHY is defined in 6.7.4.2.

Bits: 0	1	2	3	4	5	6	7
1	1	1	0	0	1	0	1

Figure 17—Format of the SFD field (except for ASK)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.3.3 Frame Length field

The Frame Length field is 7 bits in length and specifies the total number of octets contained in the PSDU (i.e., PHY payload). It is a value between 0 and *aMaxPHYPacketSize* (see 6.4). Table 21 summarizes the type of payload versus the frame length value.

Table 21—Frame length values

Frame length values	Payload
0–4	Reserved
5	MPDU (Acknowledgment)
6–8	Reserved
9 to *aMaxPHYPacketSize*	MPDU

6.3.4 PSDU field

The PSDU field has a variable length and carries the data of the PHY packet.

6.4 PHY constants and PIB attributes

This subclause specifies the constants and attributes required by the PHY.

6.4.1 PHY constants

The constants that define the characteristics of the PHY are presented in Table 22. These constants are hardware dependent and cannot be changed during operation.

Table 22—PHY constants

Constant	Description	Value
aMaxPHYPacketSize	The maximum PSDU size (in octets) the PHY shall be able to receive.	127
aTurnaroundTime	RX-to-TX or TX-to-RX maximum turnaround time (in symbol periods) (see 6.9.1 and 6.9.2)	12

6.4.2 PHY PIB attributes

The PHY PIB comprises the attributes required to manage the PHY of a device. The attributes contained in the PHY PIB are presented in Table 23. Attributes marked with a dagger (†) are read-only attributes (i.e., attribute can only be set by the PHY), which can be read by the next higher layer using the PLME-GET.request primitive. Attributes marked with an asterisk (*) have specific bits that are read-only attributes (i.e., attribute can only be set by the PHY), which can be read by the next higher layer using the PLME-GET.request primitive and other bits that can be read or written by the next higher layer using the PLME-GET.request or PLME-SET.request primitives, respectively. All other attributes can be read or written by the next higher layer using the PLME-GET.request or PLME-SET.request primitives, respectively.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 23—PHY PIB attributes

Attribute	Identifier	Type	Range	Description
phyCurrentChannel	0x00	Integer	0–26	The RF channel to use for all following transmissions and receptions (see 6.1.2).
phyChannelsSupported†	0x01	Array	An R x 32 bit array, where R ranges from 1 to 32	The array is composed of R rows, each of which is a bit string with the following properties: The 5 MSBs (b_{27}, …, b_{31}) indicate the channel page, and the 27 LSBs (b_0, b_1, …, b_{26}) indicate the status (1=available, 0=unavailable) for each of the up to 27 valid channels (b_k shall indicate the status of channel k as in 6.1.2) supported by that channel page. The device only needs to add the rows (channel pages) for the PHY(s) it supports.
*phyTransmitPower**	0x02	Bitmap	0x00–0xbf	The 2 MSBs represent the tolerance on the transmit power: 00 = ± 1 dB 01 = ± 3 dB 10 = ± 6 dB and shall be read-only. The 6 LSBs, which may be written to, represent a signed integer in twos-complement format, corresponding to the nominal transmit power of the device in decibels relative to 1 mW. The lowest value of *phyTransmitPower* is interpreted as less than or equal to –32 dBm.
phyCCAMode	0x03	Integer	1–3	The CCA mode (see 6.9.9).
phyCurrentPage	0x04	Integer	0–31	This is the current PHY channel page. This is used in conjunction with *phyCurrentChannel* to uniquely identify the channel currently being used.
phyMaxFrameDuration†	0x05	Integer	55, 212, 266, 1064	The maximum number of symbols in a frame: = *phySHRDuration* + *ceiling([aMaxPHYPacketSize + 1] x phySymbolsPerOctet)*
phySHRDuration†	0x06	Integer	3, 7, 10, 40	The duration of the synchronization header (SHR) in symbols for the current PHY.
phySymbolsPerOctet†	0x07	Float	0.4, 1.6, 2, 8	The number of symbols per octet for the current PHY.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.5 2450 MHz PHY specifications

6.5.1 Data rate

The data rate of the IEEE 802.15.4 (2450 MHz) PHY shall be 250 kb/s.

6.5.2 Modulation and spreading

The 2450 MHz PHY employs a 16-ary quasi-orthogonal modulation technique. During each data symbol period, four information bits are used to select one of 16 nearly orthogonal pseudo-random noise (PN) sequences to be transmitted. The PN sequences for successive data symbols are concatenated, and the aggregate chip sequence is modulated onto the carrier using offset quadrature phase-shift keying (O-QPSK).

6.5.2.1 Reference modulator diagram

The functional block diagram in Figure 18 is provided as a reference for specifying the 2450 MHz PHY modulation and spreading functions. The number in each block refers to the subclause that describes that function.

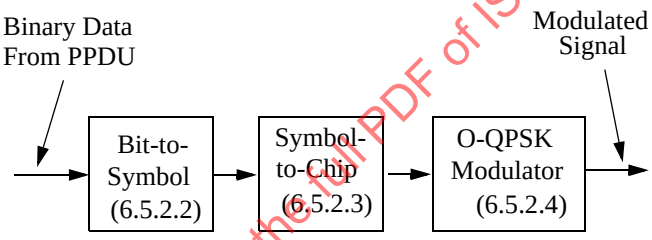


Figure 18—Modulation and spreading functions

6.5.2.2 Bit-to-symbol mapping

All binary data contained in the PPDU shall be encoded using the modulation and spreading functions shown in Figure 18. This subclause describes how binary information is mapped into data symbols.

The 4 LSBs (b_0, b_1, b_2, b_3) of each octet shall map into one data symbol, and the 4 MSBs (b_4, b_5, b_6, b_7) of each octet shall map into the next data symbol. Each octet of the PPDU is processed through the modulation and spreading functions (see Figure 18) sequentially, beginning with the Preamble field and ending with the last octet of the PSDU.

6.5.2.3 Symbol-to-chip mapping

Each data symbol shall be mapped into a 32-chip PN sequence as specified in Table 24. The PN sequences are related to each other through cyclic shifts and/or conjugation (i.e., inversion of odd-indexed chip values).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 24—Symbol-to-chip mapping

Data symbol (decimal)	Data symbol (binary) (b_0 b_1 b_2 b_3)	Chip values (c_0 c_1 … c_{30} c_{31})
0	0 0 0 0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 0 0 0	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 1 0 0	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	1 1 0 0	0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 0 1 0	0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 0 1 1
5	1 0 1 0	0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	0 1 1 0	1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 1 1 0	1 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1
8	0 0 0 1	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 0 1	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 0 1	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	1 1 0 1	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 1 1	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	1 0 1 1	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	0 1 1 1	1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 1 1	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

6.5.2.4 O-QPSK modulation

The chip sequences representing each data symbol are modulated onto the carrier using O-QPSK with half-sine pulse shaping. Even-indexed chips are modulated onto the in-phase (I) carrier and odd-indexed chips are modulated onto the quadrature-phase (Q) carrier. Because each data symbol is represented by a 32-chip sequence, the chip rate (nominally 2.0 Mchip/s) is 32 times the symbol rate. To form the offset between I-phase and Q-phase chip modulation, the Q-phase chips shall be delayed by T_c with respect to the I-phase chips (see Figure 19), where T_c is the inverse of the chip rate.

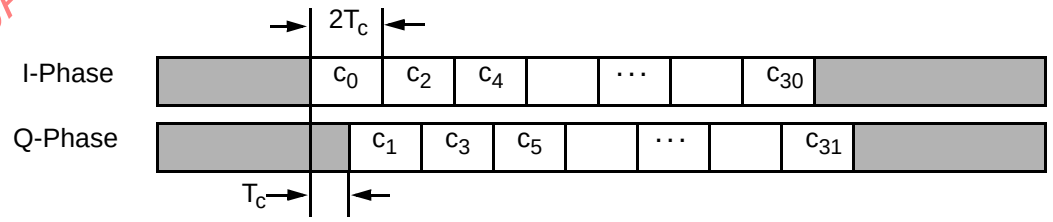


Figure 19—O-QPSK chip offsets

© IEEE 2010 – All rights reserved

6.5.2.5 Pulse shape

The half-sine pulse shape used to represent each baseband chip is described by Equation (1):

$$p(t) = \begin{cases} \sin\left(\pi \frac{t}{2T_c}\right), 0 \le t \le 2T_c \\ 0, otherwise \end{cases} \quad (1)$$

Figure 20 shows a sample baseband chip sequence (the zero sequence) with half-sine pulse shaping.

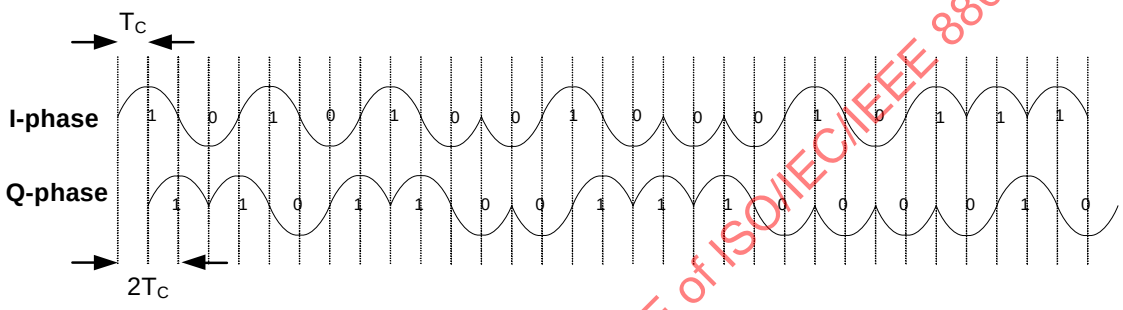


Figure 20—Sample baseband chip sequences with pulse shaping

6.5.2.6 Chip transmission order

During each symbol period, the least significant chip, c_0, is transmitted first and the most significant chip, c_{31}, is transmitted last.

6.5.3 2450 MHz band radio specification

6.5.3.1 Transmit power spectral density (PSD) mask

The transmitted spectral products shall be less than the limits specified in Table 25. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 1 MHz of the carrier frequency.

Table 25—Transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 3.5$ MHz	–20 dB	–30 dBm

6.5.3.2 Symbol rate

The 2450 MHz PHY symbol rate shall be 62.5 ksymbol/s ± 40 ppm.

6.5.3.3 Receiver sensitivity

Under the conditions specified in 6.1.7, a compliant device shall be capable of achieving a sensitivity of –85 dBm or better.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.5.3.4 Receiver jamming resistance

The minimum jamming resistance levels are given in Table 26. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 13 is the desired channel, channel 12 and channel 14 are the adjacent channels, and channel 11 and channel 15 are the alternate channels.

Table 26—Minimum receiver jamming resistance requirements for 2450 MHz PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant 2450 MHz IEEE 802.15.4 O-QPSK PHY signal, as defined by 6.5.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.5.3.3.

In either the adjacent or the alternate channel, an IEEE 802.15.4 signal, as defined by 6.5.2, is input at the relative level specified in Table 26. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.7 under these conditions.

6.6 868/915 MHz band binary phase-shift keying (BPSK) PHY specifications

6.6.1 868/915 MHz band data rates

The data rate of the 868/915 MHz band BPSK PHY shall be 20 kb/s when operating in the 868 MHz band and 40 kb/s when operating in the 915 MHz band.

6.6.2 Modulation and spreading

The 868/915 MHz BPSK PHY shall employ direct sequence spread spectrum (DSSS) with BPSK used for chip modulation and differential encoding used for data symbol encoding.

6.6.2.1 Reference modulator diagram

The functional block diagram in Figure 21 is provided as a reference for specifying the 868/915 MHz band BPSK PHY modulation and spreading functions. The number in each block refers to the subclause that describes that function. Each bit in the PPDU shall be processed through the differential encoding, bit-to-chip mapping and modulation functions in octet-wise order, beginning with the Preamble field and ending with the last octet of the PSDU. Within each octet, the LSB, b_0, is processed first and the MSB, b_7, is processed last.

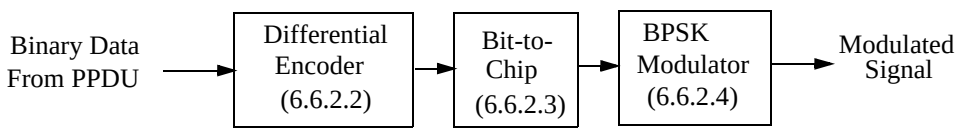


Figure 21—Modulation and spreading functions

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.6.2.2 Differential encoding

Differential encoding is the modulo-2 addition (exclusive or) of a raw data bit with the previous encoded bit. This is performed by the transmitter and can be described by Equation (2):

$$E_n = R_n \oplus E_{n-1} \tag{2}$$

where

R_n is the raw data bit being encoded

E_n is the corresponding differentially encoded bit

E_{n-1} is the previous differentially encoded bit

For each packet transmitted, R_1 is the first raw data bit to be encoded and E_0 is assumed to be zero.

Conversely, the decoding process, as performed at the receiver, can be described by Equation (3):

$$R_n = E_n \oplus E_{n-1} \tag{3}$$

For each packet received, E_1 is the first bit to be decoded, and E_0 is assumed to be zero.

6.6.2.3 Bit-to-chip mapping

Each input bit shall be mapped into a 15-chip PN sequence as specified in Table 27.

Table 27—Symbol-to-chip mapping

Input bits	Chip values (c_0 c_1 … c_{14})
0	1 1 1 1 0 1 0 1 1 0 0 1 0 0 0
1	0 0 0 0 1 0 1 0 0 1 1 0 1 1 1

6.6.2.4 BPSK modulation

The chip sequences are modulated onto the carrier using BPSK with raised cosine pulse shaping (roll-off factor = 1) where a chip value of one corresponds to a positive pulse and a chip value of zero corresponds to a negative pulse. The chip rate is 300 kchip/s for the 868 MHz band and 600 kchip/s in the 915 MHz band.

6.6.2.4.1 Pulse shape

The raised cosine pulse shape (roll-off factor = 1) used to represent each baseband chip is described by Equation (4):

$$p(t) = \begin{cases} \dfrac{\sin \pi t / T_c}{\pi t / T_c} \dfrac{\cos \pi t / T_c}{1 - 4t^2 / T_c^2}, & t \neq 0 \\ 1, & t = 0 \end{cases} \tag{4}$$

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.6.2.4.2 Chip transmission order

During each symbol period, the least significant chip, c_0, is transmitted first, and the most significant chip, c_{15}, is transmitted last.

6.6.3 868/915 MHz band radio specification

6.6.3.1 Operating frequency range

The 868/915 MHz BPSK PHY operates in the 868.0–868.6 MHz frequency band and in the 902–928 MHz frequency band.

6.6.3.2 915 MHz band transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 28. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 600 kHz of the carrier frequency.

Table 28—915 MHz band transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 1.2$ MHz	–20 dB	–20 dBm

6.6.3.3 Symbol rate

The symbol rate of an 868/915 MHz BPSK PHY conforming to this standard shall be 20 ksymbol/s when operating in the 868 MHz band and 40 ksymbol/s when operating in the 915 MHz band with an accuracy of ± 40 ppm.

6.6.3.4 Receiver sensitivity

Under the conditions specified in 6.1.7, a compliant device shall be capable of achieving a sensitivity of –92 dBm or better.

6.6.3.5 Receiver jamming resistance

This subclause applies only to the 902–928 MHz band as there is only one channel available in the 868.0–868.6 MHz band.

The minimum jamming resistance levels are given in Table 29. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 5 is the desired channel, channel 4 and channel 6 are the adjacent channels, and channel 3 and channel 7 are the alternate channels.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 29—Minimum receiver jamming resistance requirements for 915 MHz PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant 915 MHz IEEE 802.15.4 BPSK PHY signal, as defined by 6.6.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.6.3.4.

In either the adjacent or the alternate channel, a compliant IEEE 802.15.4 signal, as defined by 6.6.2, is input at the relative level specified in Table 29. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.7 under these conditions.

6.7 868/915 MHz band (optional) amplitude shift keying (ASK) PHY specifications

6.7.1 868/915 MHz band data rates

The data rate of the ASK PHY shall be 250 kb/s when operating in the 868 MHz band and in the 915 MHz band. The ASK PHY is not mandatory in the 868/915 MHz band. If the ASK PHY is used in the 868/915 MHz band, then the same device shall be capable of signaling using the 868/915 MHz BPSK PHY as well.

6.7.2 Modulation and spreading

The ASK PHY employs a multi-code modulation technique named parallel sequence spread spectrum (PSSS), which is also more generically known as orthogonal code division multiplexing (OCDM). During each data symbol period, 20 information bits for 868 MHz and 5 information bits for 915 MHz are separately modulated onto 20 and 5 nearly orthogonal PN sequences, respectively. The 20 for 868 MHz and 5 for 915 MHz PN sequences are linearly summed to create a multi-level 32-chip symbol, equal to a 64-half-chip symbol for 868 MHz and a multi-level 32-chip symbol for 915 MHz. A simple precoding is then executed per symbol, and the resulting multi-level 64-half-chip sequence for 868 MHz and multi-level 32-chip sequence for 915 MHz are modulated onto the carrier using ASK.

6.7.2.1 Reference modulator diagram

The functional block diagram in Figure 22 is provided as a reference for specifying the PSSS modulation and spreading functions. The number in each block refers to the subclause that describes that function.

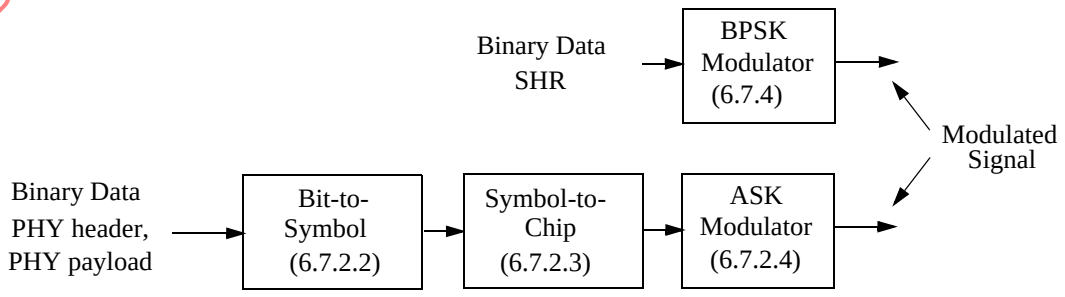


Figure 22—Modulation and spreading functions

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Each octet of the PPDU is sequentially processed through the spreading and modulation functions. The synchronization header shall be encoded using the BPSK modulator, and the PHY header and PHY payload shall be processed using the bit-to-symbol mapping, and symbol-to-chip spreading and the ASK modulator as shown in Figure 22.

6.7.2.2 Bit-to-symbol mapping

The first 20 bits for 868 MHz and the first 5 bits for 915 MHz, starting with the LSB (b_0) of the first octet of the PHY header and continuing with the subsequent octet of the PHY header, shall be mapped into the first data symbol. Further 20 bits for 868 MHz and further 5 bits for 915 MHz continuing until the end of the PPDU shall be mapped sequentially to each subsequent data symbol until all octets of the PHY header and PHY payload have been mapped into symbols, always mapping the LSBs of any octet first. For each symbol, the least significant chip shall be the first chip transmitted over the air. The last input bits shall be followed by padding, with "0" bits, of its high order bits to fill out the symbol. These zero pad bits will also be spread via the PSSS encoding to keep the over the air signaling as random as possible.

6.7.2.3 Symbol-to-chip mapping

Each data symbol shall be mapped into a multi-level 64-half-chip symbol for 868 MHz and a multi-level 32-chip symbol for 915 MHz as described in this subclause. Figure 23 provides an overview of the symbol-to-chip mapping.

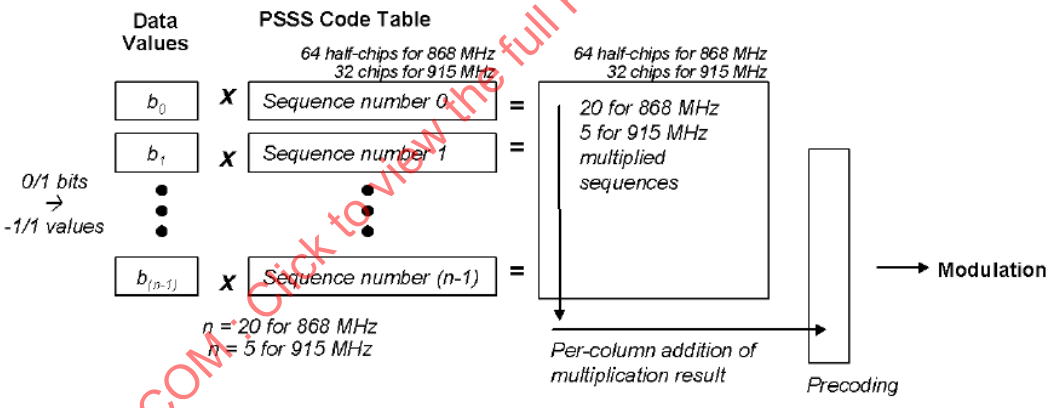


Figure 23—Symbol-to-chip mapping for PHY header and PHY payload

Each bit of the data stream is multiplied with its corresponding sequence of the code table defined in Table 30 for 868 MHz and Table 31 for 915 MHz. The PSSS code tables were generated by selecting 20 for 868 MHz and 5 for 915 MHz cyclically shifted sequences of a 31-chip base sequence and then adding a 1-bit cyclic extension to each sequence. For 868 MHz PSSS, the 31-chip base sequence is cyclically shifted by 1.5 chips to generate the next sequence in the table. For 915 MHz PSSS, the 31-chip base sequence is cyclically shifted by 6 chips to generate the next sequence in the table.

The vector of bits comprising the data symbol is multiplied with the PSSS code table. In other words, bit b_0 of the data symbol is multiplied with sequence number "0", bit b_1 is multiplied with sequence number "1", etc. Prior to multiplication, the data bits are converted to bipolar levels: data bit "1" becomes +1 and data bit "0" becomes –1. The result of multiplication is a modulated code table, which is similar to Table 30 or Table 31, but with each row either inverted or not according to the data bits.

Subsequently, all rows of the modulated code table are linearly summed to create a multi-level 64-half-chip symbol for 868 MHz and multi-level 32-chip symbol for 915 MHz. For example, chip 0 of the multi-level

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

sequence is produced by linearly summing chip 0 from each of the modulated sequences. The per-column results are 32 chips $c_0 \ldots c_{31}$ for 915 MHz and 64 half-chips $c_0 \ldots c_{63}$ for 868 MHz.

Next, a precoding operation is applied to the multi-level 32-chip symbol for 915 MHz and the multi-level 64-half-chip symbol for 868 MHz. The precoding is independent from one symbol to the next, and is performed in two steps. In the first step, a constant value is added to each of the 32 chips for 915 MHz and 64 half-chips for 868 MHz. The constant is selected so that the minimum and maximum values of the resulting sequence are symmetric about zero. Representing the original multi-level 32-chip symbol sequence by p(m), then the modified sequence p'(m) after step one of precoding is as shown in Equation (5).

$$p'(m) = p(m) - \frac{(Max + Min)}{2} \tag{5}$$

In the second step, a scaling constant is multiplied by each chip for 915 MHz and half-chip for 868 MHz in $p'(m)$. The scaling constant is selected such that the resulting sequence has a maximum amplitude of one. Letting $p''(m)$ be the output of the second precoding step, then

$$p''(m) = \frac{p'(m)}{Max'} \tag{6}$$

where Max' is the maximum within $p'(m)$. See also example in 6.7.4.3.

The precoded sequence of 32 multi-level chips for 915 MHz and 64 multi-value half-chips for 868 MHz is modulated onto the carrier as described in 6.7.2.4.

6.7.2.4 ASK modulation

The chip sequences representing each data symbol are modulated onto the carrier using ASK with square root raised cosine pulse shaping. The chip rate is 400 kchip/s, equal to 800 khalf-chip/s for 868 MHz. The chip rate is 1600 kchip/s for 915 MHz.

6.7.2.4.1 Pulse shape

The root-raised-cosine pulse shape used to represent each baseband chip is described by Equation (7).

$$h(t) = \begin{cases} \dfrac{\left\{\pi(r+1)\cdot\sin\left(\dfrac{\pi}{4}\dfrac{(r+1)}{r}\right) + \pi(r-1)\cdot\cos\left(\dfrac{\pi}{4}\dfrac{(r-1)}{r}\right) - 4r\cdot\sin\left(\dfrac{\pi}{4}\dfrac{(r-1)}{r}\right)\right\}}{2\pi\sqrt{T_c}}, & t = (\pm T_c/(4r)) \\ \dfrac{4r}{\pi\sqrt{T_c}} - \dfrac{(r-1)}{\sqrt{T_c}}, & t = 0 \\ 4r\dfrac{\cos((1+r)\pi t/T_c) + \sin((1-r)\pi t/T_c)/(4rt/T_c)}{\pi\sqrt{T_c}(1-(4rt/T_c)^2)}, & t \neq 0 \text{ and } t \neq (\pm T_c/(4r)) \end{cases} \tag{7}$$

with a rolloff factor $r = 0.2$ for 868 MHz and 915 MHz. The pulse for stimulating the pulse shaping filter will be generated at chip rate/half chip rate for 915/868 MHz. T_c in Equation (7) is the chip duration (not half chip) for 868 MHz and 915 MHz.

6.7.2.4.2 Chip transmission order

During each symbol period, the least significant chip/half-chip, c_0 / hc_0, is transmitted first, and the most significant chip, c_{31} / hc_{63}, is transmitted last.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Table 30—PSSS code table used in symbol-to-chip mapping for 868 MHz

Sequence number	Chip number																															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1
1	-1	1	-1	-1	-1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1
2	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	-1	1	-1
3	1	-1	1	-1	1	-1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	-1
4	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1
5	-1	1	1	1	-1	1	-1	1	-1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1
6	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	1	1	1	-1	1	1	-1	-1	-1	1	1
7	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	1	-1	-1	1
8	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1
9	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1
10	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	1	1	1	1
11	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1
12	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1
13	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	-1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1
14	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	-1	1	-1	1	-1	-1	-1	1	-1	1	-1	-1	1	1	1	1
15	-1	1	1	-1	-1	1	1	1	1	-1	-1	-1	1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1
16	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1
17	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1
18	1	-1	-1	1	-1	1	1	-1	1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1
19	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	half-chip number																															

Table 31—PSSS code table used in symbol-to-chip mapping for 915 MHz

Sequence number	Chip number																															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1
1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	1	1	-1	1	1	1
2	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1
3	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1
4	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1

© IEEE 2010 – All rights reserved

6.7.3 868/915 MHz band radio specification for the ASK PHY

6.7.3.1 Operating frequency range

The 868/915 MHz ASK PHY operates in the 868.0–868.6 MHz frequency band and in the 902–928 MHz frequency band.

6.7.3.2 915 MHz band transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 32. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 600 kHz of the carrier frequency.

Table 32—915 MHz band ASK PHY transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 1.2$ MHz	–20 dB	–20 dBm

6.7.3.3 Symbol rate

The ASK PHY symbol rate shall be 12.5 ksymbol/s ± 40 ppm for 868 MHz and 50 ksymbol/s ± 40 ppm for 915 MHz.

6.7.3.4 Receiver sensitivity

Under the conditions specified in 6.1.7, a compliant device shall be capable of achieving a sensitivity of –85 dBm or better.

6.7.3.5 Receiver jamming resistance

This subclause applies only to the 902–928 MHz band as there is only one channel available in the 868.0–868.6 MHz band.

The minimum jamming resistance levels are given in Table 33. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 5 is the desired channel, channel 4 and channel 6 are the adjacent channels, and channel 3 and channel 7 are the alternate channels.

Table 33—Minimum receiver jamming resistance requirements for 915 MHz ASK PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant 915 MHz IEEE 802.15.4 ASK PHY signal, as defined by 6.7.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.7.3.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

In either the adjacent or the alternate channel, a compliant IEEE 802.15.4 signal, as defined by 6.7.2, is input at the relative level specified in Table 33. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.7 under these conditions.

6.7.4 SHR for ASK PHY

The SHR uses a subset of the PSSS coding for the PHR and PSDU. The SHR chips shall be transmitted with BPSK modulation using the same chip rates (see 6.7.2.4) and pulse shaping (see 6.7.2.4.1) that are used for the PHR and PSDU. However, the symbol-to-chip mapping for the synchronization header is different, as described in 6.7.4.1 and 6.7.4.2.

6.7.4.1 Preamble for ASK PHY

The preamble is generated by repeating 2 times for 868 MHz and repeating 6 times for 915 MHz sequence number 0 from Table 30 and Table 31, respectively. The resulting preamble duration is 160 µs for 868 MHz and 120 µs for 915 MHz. The leftmost chip number “0” in the diagram, with a value of “–1”, is transmitted first. The preamble will be BPSK modulated as shown in Figure 22. The pulse shaping is the same as for the PHY payload as defined in Equation (7).

6.7.4.2 SFD for ASK PHY

The SFD for 868 MHz and 915 MHz is the inverted sequence 0 from Table 30 and Table 31, respectively. The SFD will be BPSK modulated as shown in Figure 22. The pulse shaping is the same as for the PHY payload as defined in Equation (7).

6.7.4.3 Example of PSSS encoding

Table 34 shows the example 5-bit symbol that will be encoded using Table 31, the code table for 915 MHz.

Table 34—Example 5 bit symbol

Data bits	
Bit	**Bit value**
b_0	1
b_1	–1
b_2	1
b_3	–1
b_4	–1

By weighting sequence 0 with b_0, sequence 1 with b_1 … sequence 4 with b_4, the weighted sequences in Table 35 are calculated. The PSSS symbol $p(m)$ is calculated by adding the columns of weighted sequences. For 868 MHz, the half-chips are added column-wise.

For the example $p(m)$ in Table 35 the Max$\{p(m)\}$ = 5 and the Min$\{p(m)\}$ = –5. The precoding of one symbol is executed independently of the precoding of any other symbol with the two steps described in Equation (5) and Equation (6). The resulting $p'(m)$ and $p''(m)$ are shown in Table 36.

A graph illustrating the example $p''(m)$ chip weights that will be applied to the pulse shaping filter in Equation (7) is shown in Figure 24.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

© IEEE 2010 – All rights reserved

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Table 35—Weighted sequences

Weighted sequence	Chip number																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1
1	-1	-1	1	-1	1	-1	1	1	1	1	-1	1	1	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	-1	-1
2	-1	-1	1	1	-1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	1	1	1	1	-1	-1
3	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	-1	-1	-1	1	-1	1	-1	1	1	1	1	-1	1	1	-1	1	-1	-1	1	1	-1
4	-1	1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	-1	-1	-1	1	-1	1	-1	1	1	1	1	-1	1	1	-1
p(m)	-5	-3	-1	-3	1	1	1	3	-3	1	-1	1	-1	-1	1	-3	3	-1	-1	-1	-1	1	1	-3	3	3	5	-1	-1	3	1	-5

Table 36—*p*'(*m*), aligned symmetric to zero PSSS symbol, and *p*"(*m*), precoded PSSS symbol

p'(m)	-5	-3	-1	-3	1	1	1	3	-3	1	-1	1	-1	-1	1	-3	3	-1	-1	-1	-1	1	1	-3	3	3	5	-1	-1	3	1	-5
p"(m)	-1.0	-0.6	-0.2	-0.6	0.2	0.2	0.2	0.6	-0.6	0.2	-0.2	0.2	-0.2	-0.2	0.2	-0.6	0.6	-0.2	-0.2	-0.2	-0.2	0.2	0.2	-0.6	0.6	0.6	1.0	-0.2	-0.2	0.6	0.2	-1.0

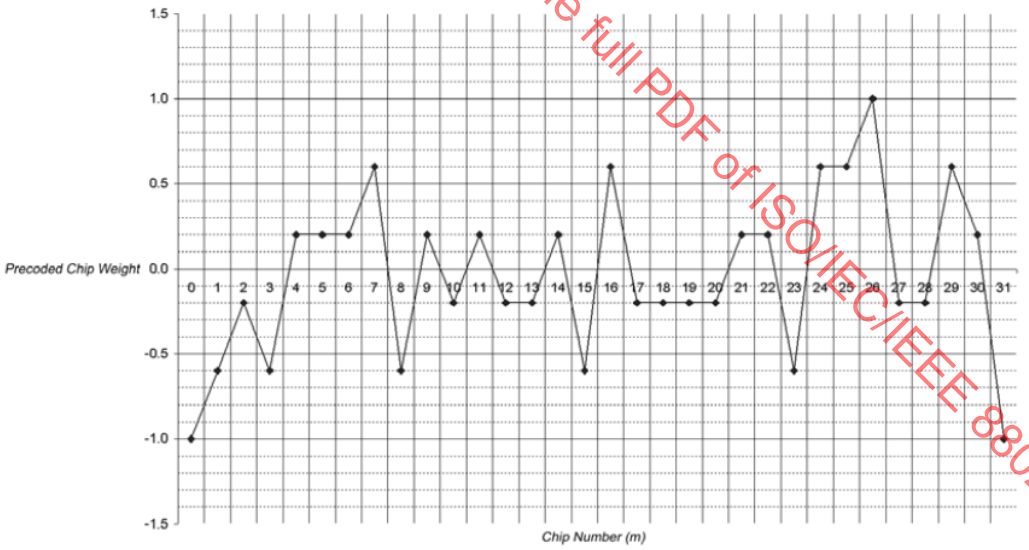


Figure 24—Precoded PSSS symbol *p*"(*m*)

6.8 868/915 MHz band (optional) O-QPSK PHY specifications

6.8.1 868/915 MHz band data rates

The data rate of the O-QPSK PHY shall be 100 kb/s when operating in the 868 MHz band and 250 kb/s when operating in the 915 MHz band. The O-QPSK PHY is not mandatory in the 868/915 MHz band. If the O-QPSK PHY is used in the 868/915 MHz band then the same device shall be capable of signaling using the 868/915 MHz BPSK PHY as well.

6.8.2 Modulation and spreading

The O-QPSK PHY employs a 16-ary quasi-orthogonal modulation technique. During each data symbol period, four information bits are used to select one of 16 nearly orthogonal PN sequences to be transmitted. The PN sequences for successive data symbols are concatenated, and the aggregate chip sequence is modulated onto the carrier using O-QPSK.

6.8.2.1 Reference modulator diagram

The functional block diagram in Figure 25 is provided as a reference for specifying the 868/915 MHz band PHY modulation and spreading functions. The number in each block refers to the subclause that describes that function. Each bit in the PPDU shall be processed through the bit-to-symbol mapping, symbol-to-chip mapping, and modulation functions in octet-wise order, beginning with the Preamble field and ending with the last octet of the PSDU. Within each octet, the LSB, b_0, is processed first and the MSB, b_7, is processed last.

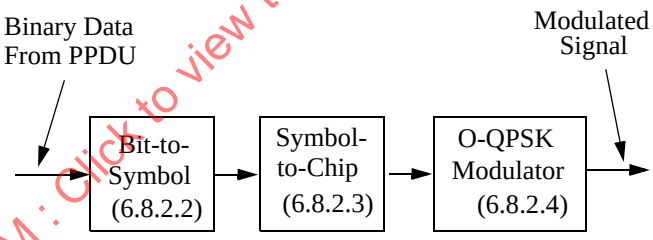


Figure 25—Modulation and spreading functions

6.8.2.2 Bit-to-symbol mapping

All binary data contained in the PPDU shall be encoded using the modulation and spreading functions shown in Figure 25. This subclause describes how binary information is mapped into data symbols.

The 4 LSBs (b_0, b_1, b_2, b_3) of each octet shall map into one data symbol, and the 4 MSBs (b_4, b_5, b_6, b_7) of each octet shall map into the next data symbol. Each octet of the PPDU is processed through the modulation and spreading functions (see Figure 25) sequentially, beginning with the Preamble field and ending with the last octet of the PSDU. Within each octet, the least significant symbol (b_0, b_1, b_2, b_3) is processed first, and the most significant symbol (b_4, b_5, b_6, b_7) is processed second.

6.8.2.3 Symbol-to-chip mapping

Each data symbol shall be mapped into a 16-chip PN sequence as specified in Table 37.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 37—Symbol-to-chip mapping for O-QPSK

Data symbol (decimal)	Data symbol (binary) (b_0 b_1 b_2 b_3)	Chip values (c_0 c_1 … c_{14} c_{15})
0	0 0 0 0	0 0 1 1 1 1 1 0 0 0 1 0 0 1 0 1
1	1 0 0 0	0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 1
2	0 1 0 0	0 1 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0
3	1 1 0 0	1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0
4	0 0 1 0	0 0 1 0 0 0 1 0 1 0 0 1 1 1 1 1 0
5	1 0 1 0	1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1
6	0 1 1 0	1 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1
7	1 1 1 0	1 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 0
8	0 0 0 1	0 1 1 0 1 0 1 1 0 1 1 1 0 0 0 0
9	1 0 0 1	0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0
10	0 1 0 1	0 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1
11	1 1 0 1	1 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1
12	0 0 1 1	0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1
13	1 0 1 1	1 1 0 1 1 1 0 0 0 0 0 1 1 0 1 0
14	0 1 1 1	1 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0
15	1 1 1 1	1 0 1 0 1 1 0 1 1 1 0 0 0 0 0 1

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

6.8.2.4 O-QPSK modulation

The chip sequences representing each data symbol are modulated onto the carrier using O-QPSK with half-sine pulse shaping. Even-indexed chips are modulated onto the in-phase (I) carrier and odd-indexed chips are modulated onto the quadrature-phase (Q) carrier. Because each data symbol is represented by a 16-chip sequence, the chip rate (nominally 400 kchip/s and 1.0 Mchip/s for the 868 MHz and 915 MHz bands, respectively) is 16 times the symbol rate. To form the offset between I-phase and Q-phase chip modulation, the Q-phase chips shall be delayed by T_c with respect to the I-phase chips (see Figure 26), where T_c is the inverse of the chip rate.

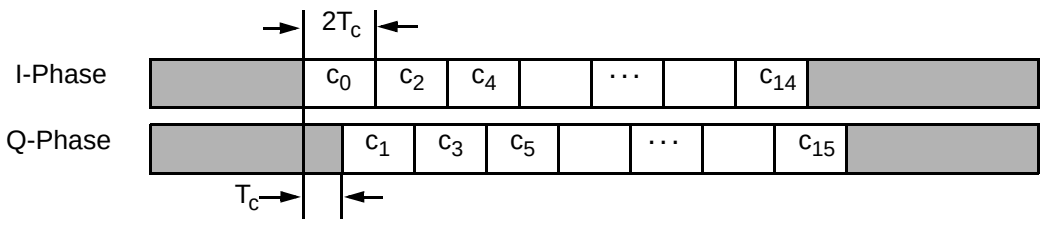


Figure 26—O-QPSK chip offsets

© IEEE 2010 – All rights reserved

6.8.2.5 Pulse shape

The half-sine pulse shape used to represent each baseband chip is described by Equation (8).

$$p(t) = \begin{cases} \sin\left(\pi \frac{t}{2T_c}\right), 0 \le t \le 2T_c \\ 0, otherwise \end{cases} \quad (8)$$

Figure 27 shows a sample baseband chip sequence (the zero sequence) with half-sine pulse shaping.

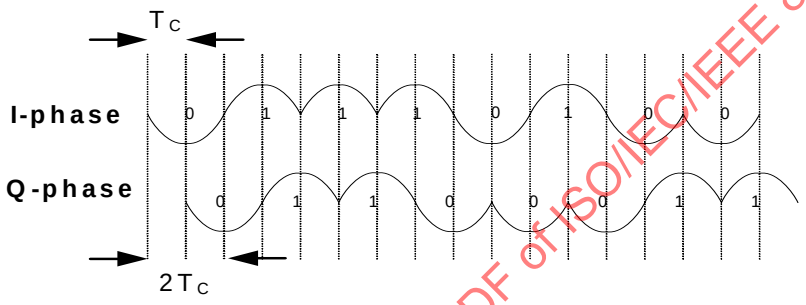


Figure 27—Sample baseband chip sequences with pulse shaping

6.8.2.6 Chip transmission order

During each symbol period, the least significant chip, c_0, is transmitted first, and the most significant chip, c_{15}, is transmitted last.

6.8.3 868/915 MHz band radio specification

6.8.3.1 Operating frequency range

The 868/915 MHz O-QPSK PHY operates in the 868.0–868.6 MHz frequency band and in the 902–928 MHz frequency band.

6.8.3.2 Transmit PSD mask

When operating in the 868 MHz band, the signal shall be filtered before transmission to regulate the transmit PSD. The filter shall approximate an ideal raised cosine filter with a roll-off factor r = 0.2, which is specified in Equation (9).

$$p(t) = \begin{cases} \frac{\sin \pi t/T_c}{\pi t/T_c} \frac{\cos r\pi t/T_c}{1 - 4r^2 t^2/T_c^2}, t \ne 0 \\ 1, t = 0 \end{cases} \quad (9)$$

When operating in the 915 MHz band, the transmitted spectral products shall be less than the limits specified in Table 38. For both relative and absolute limits, average spectral power shall be measured using a 100 kHz resolution bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within ± 600 kHz of the carrier frequency f_c.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 38—915 MHz band O-QPSK PHY transmit PSD limits

Frequency	Relative limit	Absolute limit
$\|f - f_c\| > 1.2$ MHz	–20 dB	–20 dBm

6.8.3.3 Symbol rate

The O-QPSK PHY symbol rate shall be 25 ksymbol/s when operating in the 868 MHz band and 62.5 ksymbol/s when operating in the 915 MHz band with an accuracy of ± 40 ppm.

6.8.3.4 Receiver sensitivity

Under the conditions specified in 6.1.7, a compliant device shall be capable of achieving a sensitivity of –85 dBm or better.

6.8.3.5 Receiver jamming resistance

This subclause applies only to the 902–928 MHz band as there is only one channel available in the 868.0–868.6 MHz band.

The minimum jamming resistance levels are given in Table 39. The adjacent channel is one on either side of the desired channel that is closest in frequency to the desired channel, and the alternate channel is one more removed from the adjacent channel. For example, when channel 5 is the desired channel, channel 4 and channel 6 are the adjacent channels, and channel 3 and channel 7 are the alternate channels.

Table 39—Minimum receiver jamming resistance requirements for 915 MHz O-QPSK PHY

Adjacent channel rejection	Alternate channel rejection
0 dB	30 dB

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant 915 MHz IEEE 802.15.4 O-QPSK PHY signal, as defined by 6.8.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 6.8.3.4.

In either the adjacent or the alternate channel, a compliant signal, as defined by 6.8.2, is input at the relative level specified in Table 39. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.7 under these conditions.

6.9 General radio specifications

The specifications in 6.9.1 through 6.9.9 apply to both the 2450 MHz PHY and the 868/915 MHz PHYs and, with the exception of 6.9.3, apply to all PHY implementations including the alternate PHYs.

6.9.1 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be less than or equal to *aTurnaroundTime* (see 6.4.1).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The TX-to-RX turnaround time is defined as the shortest time possible at the air interface from the trailing edge of the last chip (of the last symbol) of a transmitted PPDU to the leading edge of the first chip (of the first symbol) of the next received PPDU.

The TX-to-RX turnaround time shall be less than or equal to the RX-to-TX turnaround time.

6.9.2 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be less than or equal to *aTurnaroundTime* (see 6.4.1).

The RX-to-TX turnaround time is defined as the shortest time possible at the air interface from the trailing edge of the last chip (of the last symbol) of a received PPDU to the leading edge of the first chip (of the first symbol) of the next transmitted PPDU.

6.9.3 Error-vector magnitude (EVM) definition

The modulation accuracy of the transmitter is determined with an EVM measurement. In order to calculate the EVM measurement, a time record of N received complex chip values $(\tilde{I}_j, \tilde{Q}_j)$ is captured. For each received complex chip, a decision is made about which complex chip value was transmitted. The ideal position of the chosen complex chip (the center of the decision box) is represented by the vector (I_j, Q_j). The error vector $(\delta I_j, \delta Q_j)$ is defined as the distance from this ideal position to the actual position of the received point (see Figure 28).

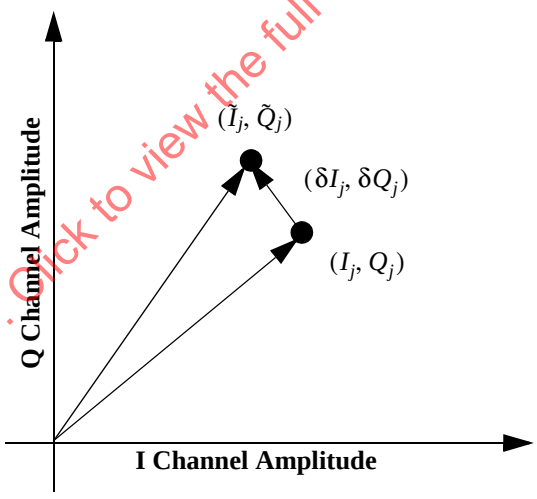


Figure 28—Error vector calculation

Thus, the received vector is the sum of the ideal vector and the error vector as shown in Equation (10).

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q_j) \tag{10}$$

The EVM is defined as shown in Equation (11).

$$EVM \equiv \sqrt{\frac{\frac{1}{N}\sum_{j=1}^{N}(\delta I_j^{\,2} + \delta Q_j^{\,2})}{S^2}} \times 100\% \tag{11}$$

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

where

S	is the magnitude of the vector to the ideal constellation point (for PSSS in 915/868 MHz S is the ASK step size, and the PHR and PHY payload should be set to 0 for testing)
$(\delta I_j, \delta Q_j)$	is the error vector

A transmitter shall have EVM values of less than 35% when measured for 1000 chips. The error-vector measurement shall be made on baseband I and Q chips after recovery through a reference receiver system. The reference receiver shall perform carrier lock, symbol timing recovery, and amplitude adjustment while making the measurements.

6.9.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ± 40 ppm maximum.

6.9.5 Transmit power

A transmitter shall be capable of transmitting at least –3 dBm. Devices should transmit lower power when possible in order to reduce interference to other devices and systems.

The maximum transmit power is limited by local regulatory bodies.

6.9.6 Receiver maximum input level of desired signal

The receiver maximum input level is the maximum power level of the desired signal present at the input of the receiver for which the error rate criterion in 6.1.7 is met. A receiver shall have a receiver maximum input level greater than or equal to –20 dBm.

6.9.7 Receiver ED

The receiver ED measurement is intended for use by a network layer as part of a channel selection algorithm. It is an estimate of the received signal power within the bandwidth of the channel. No attempt is made to identify or decode signals on the channel. The ED measurement time, to average over, shall be equal to 8 symbol periods.

The ED result shall be reported to the MLME using PLME-ED.confirm (see 6.2.2.4) as an 8 bit integer ranging from 0x00 to 0xff. The minimum ED value (zero) shall indicate received power less than 10 dB above the specified receiver sensitivity (see 6.5.3.3 and 6.6.3.4), and the range of received power spanned by the ED values shall be at least 40 dB. Within this range, the mapping from the received power in decibels to ED value shall be linear with an accuracy of ± 6 dB.

6.9.8 Link quality indicator (LQI)

The LQI measurement is a characterization of the strength and/or quality of a received packet. The measurement may be implemented using receiver ED, a signal-to-noise ratio estimation, or a combination of these methods. The use of the LQI result by the network or application layers is not specified in this standard.

The LQI measurement shall be performed for each received packet, and the result shall be reported to the MAC sublayer using PD-DATA.indication (see 6.2.1.3) as an integer ranging from 0x00 to 0xff. The minimum and maximum LQI values (0x00 and 0xff) should be associated with the lowest and highest quality compliant signals detectable by the receiver, and LQI values in between should be uniformly distributed between these two limits. At least eight unique values of LQI shall be used.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

6.9.9 Clear channel assessment (CCA)

The PHY shall provide the capability to perform CCA according to at least one of the following three methods:

- *CCA Mode 1: Energy above threshold.* CCA shall report a busy medium upon detecting any energy above the ED threshold.
- *CCA Mode 2: Carrier sense only.* CCA shall report a busy medium only upon the detection of a signal compliant with this standard with the same modulation and spreading characteristics of the PHY that is currently in use by the device. This signal may be above or below the ED threshold.
- *CCA Mode 3: Carrier sense with energy above threshold.* CCA shall report a busy medium using a logical combination of
  - Detection of a signal with the modulation and spreading characteristics of this standard and
  - Energy above the ED threshold, where the logical operator may be AND or OR.

For any of the CCA modes, if the PLME-CCA.request primitive (see 6.2.2.1) is received by the PHY during reception of a PPDU, CCA shall report a busy medium. PPDU reception is considered to be in progress following detection of the SFD, and it remains in progress until the number of octets specified by the decoded PHR has been received.

A busy channel shall be indicated by the PLME-CCA.confirm primitive (see 6.2.2.2) with a status of BUSY.

A clear channel shall be indicated by the PLME-CCA.confirm primitive with a status of IDLE.

The PHY PIB attribute *phyCCAMode* (see 6.4) shall indicate the appropriate operation mode. The CCA parameters are subject to the following criteria:

a) The ED threshold shall correspond to a received signal power of at most 10 dB above the specified receiver sensitivity (see 6.5.3.3, 6.6.3.4, 6.7.3.4, and 6.8.3.4).

b) The CCA detection time shall be equal to 8 symbol periods.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7. MAC sublayer specification

This clause specifies the MAC sublayer of this standard. The MAC sublayer handles all access to the physical radio channel and is responsible for the following tasks:

- Generating network beacons if the device is a coordinator
- Synchronizing to network beacons
- Supporting PAN association and disassociation
- Supporting device security
- Employing the CSMA-CA mechanism for channel access
- Handling and maintaining the GTS mechanism
- Providing a reliable link between two peer MAC entities

Constants and attributes that are specified and maintained by the MAC sublayer are written in the text of this clause in italics. Constants have a general prefix of "a", e.g., *aBaseSlotDuration*, and are listed in Table 85 (see 7.4.1). Attributes have a general prefix of "mac", e.g., *macAckWaitDuration*, and are listed in Table 86 (see 7.4.2), while the security attributes are listed in Table 88 (see 7.6.1).

7.1 MAC sublayer service specification

The MAC sublayer provides an interface between the SSCS and the PHY. The MAC sublayer conceptually includes a management entity called the MLME. This entity provides the service interfaces through which layer management functions may be invoked. The MLME is also responsible for maintaining a database of managed objects pertaining to the MAC sublayer. This database is referred to as the MAC sublayer PIB.

Figure 29 depicts the components and interfaces of the MAC sublayer.

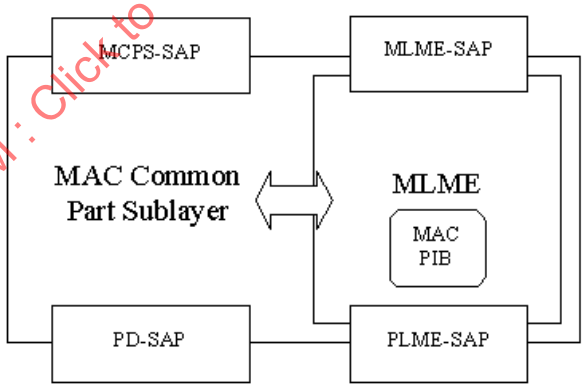


Figure 29—The MAC sublayer reference model

The MAC sublayer provides two services, accessed through two SAPs:

- The MAC data service, accessed through the MAC common part sublayer (MCPS) data SAP (MCPS-SAP), and
- The MAC management service, accessed through the MLME-SAP.

These two services provide the interface between the SSCS and the PHY, via the PD-SAP and PLME-SAP interfaces (see 6.2). In addition to these external interfaces, an implicit interface also exists between the MLME and the MCPS that allows the MLME to use the MAC data service.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.1 MAC data service

The MCPS-SAP supports the transport of SSCS protocol data units (SPDUs) between peer SSCS entities. Table 40 lists the primitives supported by the MCPS-SAP. Primitives marked with a diamond (♦) are optional for an RFD. These primitives are discussed in the subclauses referenced in the table.

Table 40—MCPS-SAP primitives

MCPS-SAP primitive	Request	Confirm	Indication
MCPS-DATA	7.1.1.1	7.1.1.2	7.1.1.3
MCPS-PURGE	7.1.1.4♦	7.1.1.5♦	—

7.1.1.1 MCPS-DATA.request

The MCPS-DATA.request primitive requests the transfer of a data SPDU (i.e., MSDU) from a local SSCS entity to a single peer SSCS entity.

7.1.1.1.1 Semantics of the service primitive

The semantics of the MCPS-DATA.request primitive are as follows:

```
MCPS-DATA.request    (
                     SrcAddrMode,
                     DstAddrMode,
                     DstPANId,
                     DstAddr,
                     msduLength,
                     msdu,
                     msduHandle,
                     TxOptions,
                     SecurityLevel,
                     KeyIdMode,
                     KeySource,
                     KeyIndex
                     )
```

Table 41 specifies the parameters for the MCPS-DATA.request primitive.

Table 41—MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 7.2.1.1.8). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 41—MCPS-DATA.request parameters *(continued)*

Name	Type	Valid range	Description
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 7.2.1.1.6). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstPANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	≤ *aMaxMACPayloadSize*	The number of octets contained in the MSDU to be transmitted by the MAC sublayer entity.
msdu	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
msduHandle	Integer	0x00–0xff	The handle associated with the MSDU to be transmitted by the MAC sublayer entity.
TxOptions	Bitmap	3-bit field	The 3 bits (b_0, b_1, b_2) indicate the transmission options for this MSDU. For b_0, 1 = acknowledged transmission, 0 = unacknowledged transmission. For b_1, 1 = GTS transmission, 0 = CAP transmission for a beacon-enabled PAN. For b_2, 1 = indirect transmission, 0 = direct transmission. For a nonbeacon-enabled PAN, bit b_1 should always be set to 0.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.1.1.1.2 Appropriate usage

The MCPS-DATA.request primitive is generated by a local SSCS entity when a data SPDU (i.e., MSDU) is to be transferred to a peer SSCS entity.

© IEEE 2010 – All rights reserved

7.1.1.1.3 Effect on receipt

On receipt of the MCPS-DATA.request primitive, the MAC sublayer entity begins the transmission of the supplied MSDU.

The MAC sublayer builds an MPDU to transmit from the supplied arguments. The flags in the SrcAddrMode and DstAddrMode parameters correspond to the addressing subfields in the Frame Control field (see 7.2.1.1) and are used to construct both the Frame Control and addressing fields of the MHR. If both the SrcAddrMode and the DstAddrMode parameters are set to 0x00 (i.e., addressing fields omitted), the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_ADDRESS.

If the msduLength parameter is greater than *aMaxMACSafePayloadSize*, the MAC sublayer will set the Frame Version subfield of the Frame Control field to one.

The TxOptions parameter indicates how the MAC sublayer data service transmits the supplied MSDU. If the TxOptions parameter specifies that an acknowledged transmission is required, the Acknowledgment Request subfield of the Frame Control field will be set to one (see 7.5.6.4).

If the TxOptions parameter specifies that a GTS transmission is required, the MAC sublayer will determine whether it has a valid GTS (for GTS usage rules, see 7.5.7.3). If a valid GTS could not be found, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_GTS. If a valid GTS was found, the MAC sublayer will defer, if necessary, until the GTS. If the TxOptions parameter specifies that a GTS transmission is not required, the MAC sublayer will transmit the MSDU using either slotted CSMA-CA in the CAP for a beacon-enabled PAN or unslotted CSMA-CA for a nonbeacon-enabled PAN. Specifying a GTS transmission in the TxOptions parameter overrides an indirect transmission request.

If the TxOptions parameter specifies that an indirect transmission is required and this primitive is received by the MAC sublayer of a coordinator, the data frame is sent using indirect transmission, i.e., the data frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3. Transactions with a broadcast destination address will be transmitted using the mechanism described in 7.2.1.1.3. Transactions with a unicast destination address can then be extracted at the discretion of each device concerned using the method described in 7.5.6.3. If there is no capacity to store the transaction, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5. If the TxOptions parameter specifies that an indirect transmission is required and if the device receiving this primitive is not a coordinator, the destination address is not present, or the TxOptions parameter also specifies a GTS transmission, the indirect transmission option will be ignored.

If the TxOptions parameter specifies that an indirect transmission is not required, the MAC sublayer will transmit the MSDU using CSMA-CA either in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN. If the TxOptions parameter specifies that a direct transmission is required and the MAC sublayer does not receive an acknowledgment from the recipient after *macMaxFrameRetries* retransmissions (see 7.5.6.4), it will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of NO_ACK.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MAC sublayer will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on the DstAddr, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MAC sublayer will discard the frame and issue the MCPS-DATA.confirm primitive with the error status returned by outgoing frame processing.

If the requested transaction is too large to fit in the CAP or GTS, as appropriate, the MAC sublayer shall discard the frame and issue the MCPS-DATA.confirm primitive with a status of FRAME_TOO_LONG.

If the transmission uses CSMA-CA and the CSMA-CA algorithm failed due to adverse conditions on the channel, and the TxOptions parameter specifies that a direct transmission is required, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MPDU was successfully transmitted and, if requested, an acknowledgment was received, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of SUCCESS.

If any parameter in the MCPS-DATA.request primitive is not supported or is out of range, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_PARAMETER.

7.1.1.2 MCPS-DATA.confirm

The MCPS-DATA.confirm primitive reports the results of a request to transfer a data SPDU (MSDU) from a local SSCS entity to a single peer SSCS entity.

7.1.1.2.1 Semantics of the service primitive

The semantics of the MCPS-DATA.confirm primitive are as follows:

```
MCPS-DATA.confirm       (
                        msduHandle,
                        status,
                        Timestamp
                        )
```

Table 42 specifies the parameters for the MCPS-DATA.confirm primitive.

7.1.1.2.2 When generated

The MCPS-DATA.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-DATA.request primitive. The MCPS-DATA.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or the appropriate error code. The status values are fully described in 7.1.1.1.3 and subclauses referenced by 7.1.1.1.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 42—MCPS-DATA.confirm parameters

Name	Type	Valid range	Description
msduHandle	Integer	0x00–0xff	The handle associated with the MSDU being confirmed.
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the last MSDU transmission.
Timestamp	Integer	0x000000–0xffffff	Optional. The time, in symbols, at which the data were transmitted (see 7.5.4.1). The value of this parameter will be considered valid only if the value of the status parameter is SUCCESS; if the status parameter is not equal to SUCCESS, the value of the Timestamp parameter shall not be used for any other purpose. The symbol boundary is described by *macSyncSymbolOffset* (see Table 86 in 7.4.1). This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.

7.1.1.2.3 Appropriate usage

On receipt of the MCPS-DATA.confirm primitive, the SSCS of the initiating device is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

7.1.1.3 MCPS-DATA.indication

The MCPS-DATA.indication primitive indicates the transfer of a data SPDU (i.e., MSDU) from the MAC sublayer to the local SSCS entity.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.1.3.1 Semantics of the service primitive

The semantics of the MCPS-DATA.indication primitive are as follows:

```
MCPS-DATA.indication        (
                            SrcAddrMode,
                            SrcPANId,
                            SrcAddr,
                            DstAddrMode,
                            DstPANId
                            DstAddr,
                            msduLength,
                            msdu,
                            mpduLinkQuality,
                            DSN,
                            Timestamp,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 43 specifies the parameters for the MCPS-DATA.indication primitive.

Table 43—MCPS-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcPANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the entity from which the MSDU was received.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the MSDU was received.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short device address. 0x03 = 64-bit extended device address.
DstPANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the entity to which the MSDU is being transferred.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 43—MCPS-DATA.indication parameters *(continued)*

Name	Type	Valid range	Description
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	Integer	≤ *aMaxMACFrame-Size*	The number of octets contained in the MSDU being indicated by the MAC sublayer entity.
msdu	Set of octets	—	The set of octets forming the MSDU being indicated by the MAC sublayer entity.
mpduLinkQuality	Integer	0x00–0xff	LQI value measured during reception of the MPDU. Lower values represent lower LQI (see 6.9.8).
DSN	Integer	0x00–0xff	The DSN of the received data frame.
Timestamp	Integer	0x000000–0xffffff	Optional. The time, in symbols, at which the data were received (see 7.5.4.1). The symbol boundary is described by *macSyncSymbolOffset* (see Table 86 in 7.4.1). This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received data frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.1.3.2 When generated

The MCPS-DATA.indication primitive is generated by the MAC sublayer and issued to the SSCS on receipt of a data frame at the local MAC sublayer entity that passes the appropriate message filtering operations as described in 7.5.6.2.

7.1.1.3.3 Appropriate usage

On receipt of the MCPS-DATA.indication primitive, the SSCS is notified of the arrival of data at the device. If the primitive is received while the device is in promiscuous mode, the parameters will be set as specified in 7.5.6.5.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.1.4 MCPS-PURGE.request

The MCPS-PURGE.request primitive allows the next higher layer to purge an MSDU from the transaction queue.

This primitive is optional for an RFD.

7.1.1.4.1 Semantics of the service primitive

The semantics of the MCPS-PURGE.request primitive are as follows:

```
MCPS-PURGE.request          (
                            msduHandle
                            )
```

Table 44 specifies the parameters for the MCPS-PURGE.request primitive.

Table 44—MCPS-PURGE.request parameters

Name	Type	Valid range	Description
msduHandle	Integer	0x00–0xff	The handle of the MSDU to be purged from the transaction queue.

7.1.1.4.2 Appropriate usage

The MCPS-PURGE.request primitive is generated by the next higher layer whenever a MSDU is to be purged from the transaction queue.

7.1.1.4.3 Effect on receipt

On receipt of the MCPS-PURGE.request primitive, the MAC sublayer attempts to find in its transaction queue the MSDU indicated by the msduHandle parameter. If an MSDU has left the transaction queue, the handle will not be found, and the MSDU can no longer be purged. If an MSDU matching the given handle is found, the MSDU is discarded from the transaction queue, and the MAC sublayer issues the MCPS-PURGE.confirm primitive with a status of SUCCESS. If an MSDU matching the given handle is not found, the MAC sublayer issues the MCPS-PURGE.confirm primitive with a status of INVALID_HANDLE.

7.1.1.5 MCPS-PURGE.confirm

The MCPS-PURGE.confirm primitive allows the MAC sublayer to notify the next higher layer of the success of its request to purge an MSDU from the transaction queue.

This primitive is optional for an RFD.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.1.5.1 Semantics of the service primitive

The semantics of the MCPS-PURGE.confirm primitive are as follows:

```
MCPS-PURGE.confirm      (
                        msduHandle,
                        status
                        )
```

Table 45 specifies the parameters for the MCPS-PURGE.confirm primitive.

Table 45—MCPS-PURGE.confirm parameters

Name	Type	Valid range	Description
msduHandle	Integer	0x00–0xff	The handle of the MSDU requested to be purge from the transaction queue.
status	Enumeration	SUCCESS or INVALID_HANDLE	The status of the request to be purged an MSDU from the transaction queue.

7.1.1.5.2 When generated

The MCPS-PURGE.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-PURGE.request primitive. The MCPS-PURGE.confirm primitive returns a status of either SUCCESS, indicating that the purge request was successful, or INVALID_HANDLE, indicating an error. The status values are fully described in 7.1.1.4.3.

7.1.1.5.3 Appropriate usage

On receipt of the MCPS-PURGE.confirm primitive, the next higher layer is notified of the result of its request to purge an MSDU from the transaction queue. If the purge request was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error.

7.1.1.6 Data service message sequence chart

Figure 30 illustrates a sequence of messages necessary for a successful data transfer between two devices. Figure 84 and Figure 85 (see 7.7) also illustrate this, including the steps taken by the PHY.

7.1.2 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 46 summarizes the primitives supported by the MLME through the MLME-SAP interface. Primitives marked with a diamond (♦) are optional for an RFD. Primitives marked with an asterisk (*) are optional for both device types (i.e., RFD and FFD). The primitives are discussed in the subclauses referenced in the table.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

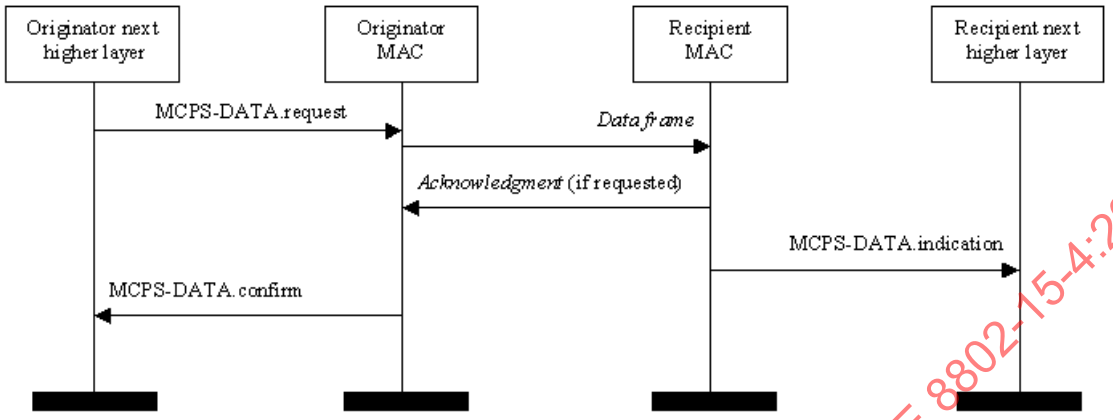


Figure 30—Message sequence chart describing the MAC data service

Table 46—Summary of the primitives accessed through the MLME-SAP

Name	Request	Indication	Response	Confirm
MLME-ASSOCIATE	7.1.3.1	7.1.3.2♦	7.1.3.3♦	7.1.3.4
MLME-DISASSOCIATE	7.1.4.1	7.1.4.2		7.1.4.3
MLME-BEACON-NOTIFY		7.1.5.1		
MLME-GET	7.1.6.1			7.1.6.2
MLME-GTS	7.1.7.1*	7.1.7.3*		7.1.7.2*
MLME-ORPHAN		7.1.8.1♦	7.1.8.2♦	
MLME-RESET	7.1.9.1			7.1.9.2
MLME-RX-ENABLE	7.1.10.1*			7.1.10.2*
MLME-SCAN	7.1.11.1			7.1.11.2
MLME-COMM-STATUS		7.1.12.1		
MLME-SET	7.1.13.1			7.1.13.2
MLME-START	7.1.14.1♦			7.1.14.2♦
MLME-SYNC	7.1.15.1*			
MLME-SYNC-LOSS		7.1.15.2		
MLME-POLL	7.1.16.1			7.1.16.2

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.1.3 Association primitives

MLME-SAP association primitives define how a device becomes associated with a PAN.

All devices shall provide an interface for the request and confirm association primitives. The indication and response association primitives are optional for an RFD.

© IEEE 2010 – All rights reserved

7.1.3.1 MLME-ASSOCIATE.request

The MLME-ASSOCIATE.request primitive allows a device to request an association with a coordinator.

7.1.3.1.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.request primitive are as follows:

```
MLME-ASSOCIATE.request          (
                                LogicalChannel,
                                ChannelPage,
                                CoordAddrMode,
                                CoordPANId,
                                CoordAddress,
                                CapabilityInformation,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex
                                )
```

Table 47 specifies the parameters for the MLME-ASSOCIATE.request primitive.

Table 47—MLME-ASSOCIATE.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see 6.1.2).	The logical channel on which to attempt association.
ChannelPage	Integer	Selected from the available channel pages supported by the PHY (see 6.1.2).	The channel page on which to attempt association.
CoordAddrMode	Integer	0x02–0x03	The coordinator addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 2=16-bit short address. 3=64-bit extended address.
CoordPANId	Integer	0x0000–0xffff	The identifier of the PAN with which to associate.
CoordAddress	Device address	As specified by the CoordAddrMode parameter.	The address of the coordinator with which to associate.
CapabilityInformation	Bitmap	See 7.3.1.2	Specifies the operational capabilities of the associating device.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 47—MLME-ASSOCIATE.request parameters *(continued)*

Name	Type	Valid range	Description
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.3.1.2 Appropriate usage

The MLME-ASSOCIATE.request primitive is generated by the next higher layer of an unassociated device and issued to its MLME to request an association with a PAN through a coordinator. If the device wishes to associate through a coordinator on a beacon-enabled PAN, the MLME may optionally track the beacon of that coordinator prior to issuing this primitive.

7.1.3.1.3 Effect on receipt

On receipt of the MLME-ASSOCIATE.request primitive, the MLME of an unassociated device first updates the appropriate PHY and MAC PIB attributes and then generates an association request command (see 7.3.1), as dictated by the association procedure described in 7.5.3.1.

The SecurityLevel parameter specifies the level of security to be applied to the association request command frame. Typically, the association request command should not be implemented using security. However, if the device requesting association shares a key with the coordinator, then security may be specified.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on the CoordAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-ASSOCIATE.confirm primitive with the error status returned by outgoing frame processing.

If the association request command cannot be sent to the coordinator due to the CSMA-CA algorithm indicating a busy channel, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits an association request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_ACK (see 7.5.6.4).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the MLME of an unassociated device successfully receives an acknowledgment to its association request command, the MLME will wait for a response to the request (see 7.5.3.1). If the MLME of the device does not receive a response, it will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA.

If the MLME of the device extracts an association response command frame from the coordinator, it will then issue the MLME-ASSOCIATE.confirm primitive with a status equal to the contents of the Association Status field in the association response command (see 7.3.2.3).

On receipt of the association request command, the MLME of the coordinator issues the MLME-ASSOCIATE.indication primitive.

If any parameter in the MLME-ASSOCIATE.request primitive is either not supported or out of range, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of INVALID_PARAMETER.

7.1.3.2 MLME-ASSOCIATE.indication

The MLME-ASSOCIATE.indication primitive is used to indicate the reception of an association request command.

7.1.3.2.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.indication primitive are as follows:

```
MLME-ASSOCIATE.indication    (
                             DeviceAddress,
                             CapabilityInformation,
                             SecurityLevel,
                             KeyIdMode,
                             KeySource,
                             KeyIndex
                             )
```

Table 48 specifies the parameters for the MLME-ASSOCIATE.indication primitive.

Table 48—MLME-ASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address.	The address of the device requesting association.
CapabilityInformation	Bitmap	See 7.3.1.2	The operational capabilities of the device requesting association.
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received MAC command frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 48—MLME-ASSOCIATE.indication parameters *(continued)*

Name	Type	Valid range	Description
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.3.2.2 When generated

The MLME-ASSOCIATE.indication primitive is generated by the MLME of the coordinator and issued to its next higher layer to indicate the reception of an association request command (see 7.3.1).

7.1.3.2.3 Appropriate usage

When the next higher layer of a coordinator receives the MLME-ASSOCIATE.indication primitive, the coordinator determines whether to accept or reject the unassociated device using an algorithm outside the scope of this standard. The next higher layer of the coordinator then issues the MLME-ASSOCIATE.response primitive to its MLME.

The association decision and the response should become available at the coordinator within a time of *macResponseWaitTime* (see 7.5.3.1). After this time, the device requesting association attempts to extract the association response command frame from the coordinator, using the method described in 7.5.6.3, in order to determine whether the association was successful.

7.1.3.3 MLME-ASSOCIATE.response

The MLME-ASSOCIATE.response primitive is used to initiate a response to an MLME-ASSOCIATE.indication primitive.

7.1.3.3.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.response primitive are as follows:

```
MLME-ASSOCIATE.response     (
                            DeviceAddress,
                            AssocShortAddress,
                            status,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 49 specifies the parameters for the MLME-ASSOCIATE.response primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 49—MLME-ASSOCIATE.response parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64 bit IEEE address	The address of the device requesting association.
AssocShortAddress	Integer	0x0000–0xffff	The 16-bit short device address allocated by the coordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful.
status	Enumeration	See 7.3.2.3	The status of the association attempt.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.3.3.2 Appropriate usage

The MLME-ASSOCIATE.response primitive is generated by the next higher layer of a coordinator and issued to its MLME in order to respond to the MLME-ASSOCIATE.indication primitive.

7.1.3.3.3 Effect on receipt

When the MLME of a coordinator receives the MLME-ASSOCIATE.response primitive, it generates an association response command (see 7.3.2). The command frame is sent to the device requesting association using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based the DeviceAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with the error status returned by outgoing frame processing.

Upon receipt of the MLME-ASSOCIATE.response primitive, the coordinator attempts to add the information contained in the primitive to its list of pending transactions. If there is no capacity to store the transaction, the MAC sublayer will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5.

If the frame was successfully transmitted and an acknowledgment was received, if requested, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of SUCCESS.

If any parameter in the MLME-ASSOCIATE.response primitive is not supported or is out of range, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of INVALID_PARAMETER.

7.1.3.4 MLME-ASSOCIATE.confirm

The MLME-ASSOCIATE.confirm primitive is used to inform the next higher layer of the initiating device whether its request to associate was successful or unsuccessful.

7.1.3.4.1 Semantics of the service primitive

The semantics of the MLME-ASSOCIATE.confirm primitive are as follows:

```
MLME-ASSOCIATE.confirm      (
                            AssocShortAddress,
                            status,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 50 specifies the parameters for the MLME-ASSOCIATE.confirm primitive.

Table 50—MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
AssocShortAddress	Integer	0x0000–0xffff	The short device address allocated by the coordinator on successful association. This parameter will be equal to 0xffff if the association attempt was unsuccessful.
status	Enumeration	The value of the Status field of the association response command (see 7.3.2.3), SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY INVALID_PARAMETER	The status of the association attempt.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 50—MLME-ASSOCIATE.confirm parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	If the primitive was generated following failed outgoing processing of an association request command: The security level to be used (see Table 95 in 7.6.2.2.1). If the primitive was generated following receipt of an association response command: The security level purportedly used by the received frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was generated following failed outgoing processing of an association request command: The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00. If the primitive was generated following receipt of an association response command: The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated following failed outgoing processing of an association request command: The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated following receipt of an association response command: The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 50—MLME-ASSOCIATE.confirm parameters *(continued)*

Name	Type	Valid range	Description
KeyIndex	Integer	0x01–0xff	If the primitive was generated following failed outgoing processing of an association request command: The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated following receipt of an association response command: The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.3.4.2 When generated

The MLME-ASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-ASSOCIATE.request primitive. If the request was successful, the status parameter will indicate a successful association, as contained in the Status field of the association response command. Otherwise, the status parameter indicates either an error code from the received association response command or the appropriate error code from Table 50. The status values are fully described in 7.1.3.1.3 and subclauses referenced by 7.1.3.1.3.

7.1.3.4.3 Appropriate usage

On receipt of the MLME-ASSOCIATE.confirm primitive, the next higher layer of the initiating device is notified of the result of its request to associate with a coordinator. If the association attempt was successful, the status parameter will indicate a successful association, as contained in the Status field of the association response command, and the device will be provided with a 16-bit short address (see Table 87 in 7.5.3.1). If the association attempt was unsuccessful, the address will be equal to 0xffff, and the status parameter will indicate the error.

7.1.3.5 Association message sequence charts

Figure 31 illustrates a sequence of messages that may be used by a device that is not tracking the beacon of the coordinator (see 7.5.6.3) to successfully associate with a PAN. Figure 80 and Figure 81 (see 7.7) illustrate this same scenario, including steps taken by the PHY, for a device associating with a coordinator and for a coordinator allowing association by a device, respectively.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

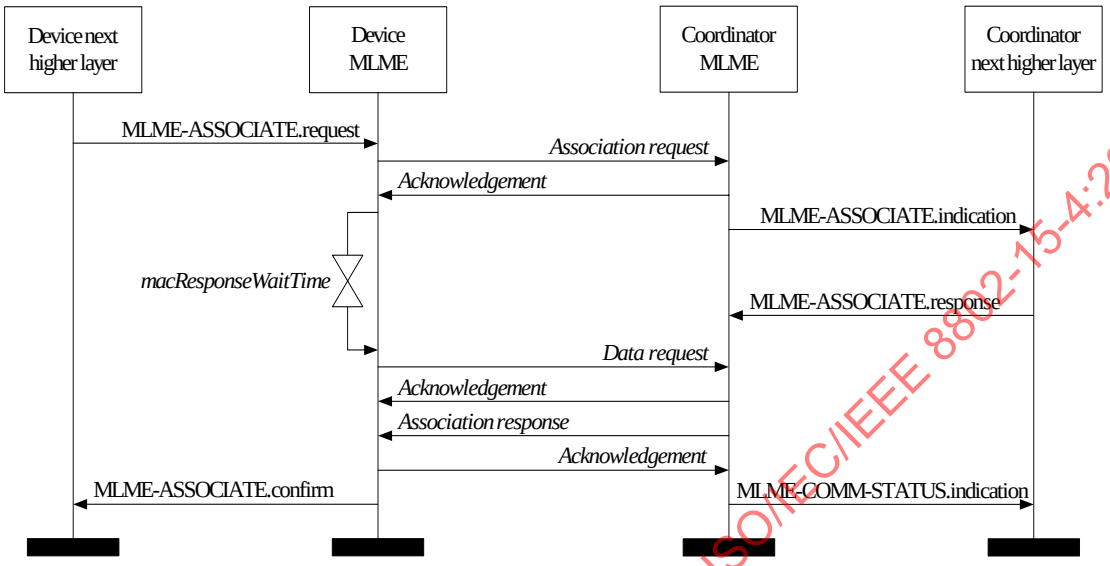


Figure 31—Message sequence chart for association

7.1.4 Disassociation primitives

The MLME-SAP disassociation primitives define how a device can disassociate from a PAN.

All devices shall provide an interface for these disassociation primitives.

7.1.4.1 MLME-DISASSOCIATE.request

The MLME-DISASSOCIATE.request primitive is used by an associated device to notify the coordinator of its intent to leave the PAN. It is also used by the coordinator to instruct an associated device to leave the PAN.

7.1.4.1.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.request primitive are as follows:

```
MLME-DISASSOCIATE.request       (
                                DeviceAddrMode,
                                DevicePANId,
                                DeviceAddress,
                                DisassociateReason,
                                TxIndirect,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex
                                )
```

Table 51 specifies the parameters for the MLME-DISASSOCIATE.request primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 51—MLME-DISASSOCIATE.request parameters

Name	Type	Valid range	Description
DeviceAddrMode	Integer	0x02–0x03	The addressing mode of the device to which to send the disassociation notification command.
DevicePANId	Integer	0x0000–0xffff	The PAN identifier of the device to which to send the disassociation notification command.
DeviceAddress	Device address	As specified by the DeviceAddrMode parameter.	The address of the device to which to send the disassociation notification command.
DisassociateReason	Integer	0x00–0xff	The reason for the disassociation (see 7.3.3.2).
TxIndirect	Boolean	TRUE or FALSE	TRUE if the disassociation notification command is to be sent indirectly.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.4.1.2 Appropriate usage

The MLME-DISASSOCIATE.request primitive is generated by the next higher layer of an associated device and issued to its MLME to request disassociation from the PAN. It is also generated by the next higher layer of the coordinator and issued to its MLME to instruct an associated device to leave the PAN.

7.1.4.1.3 Effect on receipt

On receipt of the MLME-DISASSOCIATE.request primitive, the MLME compares the DevicePANId parameter with *macPANId*. If the DevicePANId parameter is not equal to *macPANId*, the MLME issues the MLME-DISASSOCIATE.confirm primitive with a status of INVALID_PARAMETER. If the DevicePANId parameter is equal to *macPANId*, the MLME evaluates the primitive address fields.

If the DeviceAddrMode parameter is equal to 0x02 and the DeviceAddress parameter is equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to 0x03 and the DeviceAddress parameter is equal to *macCoordExtendedAddress*, the TxIndirect parameter is ignored, and the MLME sends a disassociation notification command (see 7.3.3) to its coordinator in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the DeviceAddrMode parameter is equal to 0x02 and the DeviceAddress parameter is not equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to 0x03 and the DeviceAddress parameter is not equal to *macCoordExtendedAddress*, and if this primitive was received by the MLME of a coordinator with the TxIndirect parameter set to TRUE, the disassociation notification command will be sent using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3.

If the DeviceAddrMode parameter is equal to 0x02 and the DeviceAddress parameter is not equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to 0x03 and the DeviceAddress parameter is not equal to *macCoordExtendedAddress*, and if this primitive was received by the MLME of a coordinator with the TxIndirect parameter set to FALSE, the MLME sends a disassociation notification command to the device in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN.

Otherwise, the MLME issues the MLME-DISASSOCIATE.confirm primitive with a status of INVALID_PARAMETER and does not generate a disassociation notification command.

If the disassociation notification command is to be sent using indirect transmission and there is no capacity to store the transaction, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransaction-PersistenceTime*, the transaction information will be discarded, and the MLME will issue the MLME-DISASSOCIATE.confirm with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 7.5.5.

If the disassociation notification command cannot be sent due to a CSMA-CA algorithm failure and this primitive was received either by the MLME of a coordinator with the TxIndirect parameter set to FALSE or by the MLME of a device, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on the DeviceAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with the error status returned by outgoing frame processing.

If the MLME successfully transmits a disassociation notification command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received and this primitive was received either by the MLME of a coordinator with the TxIndirect parameter set to FALSE or by the MLME of a device, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of NO_ACK (see 7.5.6.4).

If the MLME successfully transmits a disassociation notification command and receives an acknowledgment in return, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of SUCCESS.

On receipt of the disassociation notification command, the MLME of the recipient issues the MLME-DISASSOCIATE.indication primitive.

If any parameter in the MLME-DISASSOCIATE.request primitive is not supported or is out of range, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of INVALID_PARAMETER.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.4.2 MLME-DISASSOCIATE.indication

The MLME-DISASSOCIATE.indication primitive is used to indicate the reception of a disassociation notification command.

7.1.4.2.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.indication primitive are as follows:

```
MLME-DISASSOCIATE.indication    (
                                DeviceAddress,
                                DisassociateReason,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex
                                )
```

Table 52 specifies the parameters for the MLME-DISASSOCIATE.indication primitive.

Table 52—MLME-DISASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting disassociation.
DisassociateReason	Integer	0x00–0xff	The reason for the disassociation (see 7.3.3.2).
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received MAC command frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.4.2.2 When generated

The MLME-DISASSOCIATE.indication primitive is generated by the MLME and issued to its next higher layer on receipt of a disassociation notification command.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.4.2.3 Appropriate usage

The next higher layer is notified of the reason for the disassociation.

7.1.4.3 MLME-DISASSOCIATE.confirm

The MLME-DISASSOCIATE.confirm primitive reports the results of an MLME-DISASSOCIATE.request primitive.

7.1.4.3.1 Semantics of the service primitive

The semantics of the MLME-DISASSOCIATE.confirm primitive are as follows:

```
MLME-DISASSOCIATE.confirm       (
                                status,
                                DeviceAddrMode,
                                DevicePANId,
                                DeviceAddress
                                )
```

Table 53 specifies the parameters for the MLME-DISASSOCIATE.confirm primitive.

Table 53—MLME-DISASSOCIATE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the disassociation attempt.
DeviceAddrMode	Integer	0x02–0x03	The addressing mode of the device that has either requested disassociation or been instructed to disassociate by its coordinator.
DevicePANId	Integer	0x0000–0xffff	The PAN identifier of the device that has either requested disassociation or been instructed to disassociate by its coordinator.
DeviceAddress	Device address	As specified by the DeviceAddrMode parameter.	The address of the device that has either requested disassociation or been instructed to disassociate by its coordinator.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.4.3.2 When generated

The MLME-DISASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-DISASSOCIATE.request primitive. This primitive returns a status of either SUCCESS, indicating that the disassociation request was successful, or the appropriate error code. The status values are fully described in 7.1.4.1.3 and subclauses referenced by 7.1.4.1.3.

7.1.4.3.3 Appropriate usage

On receipt of the MLME-DISASSOCIATE.confirm primitive, the next higher layer of the initiating device is notified of the result of the disassociation attempt. If the disassociation attempt was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.4.4 Disassociation message sequence charts

The request to disassociate may originate either from a device or from the coordinator through which the device has associated. Figure 32 illustrates the sequence of messages necessary for a device to successfully disassociate itself from the PAN.

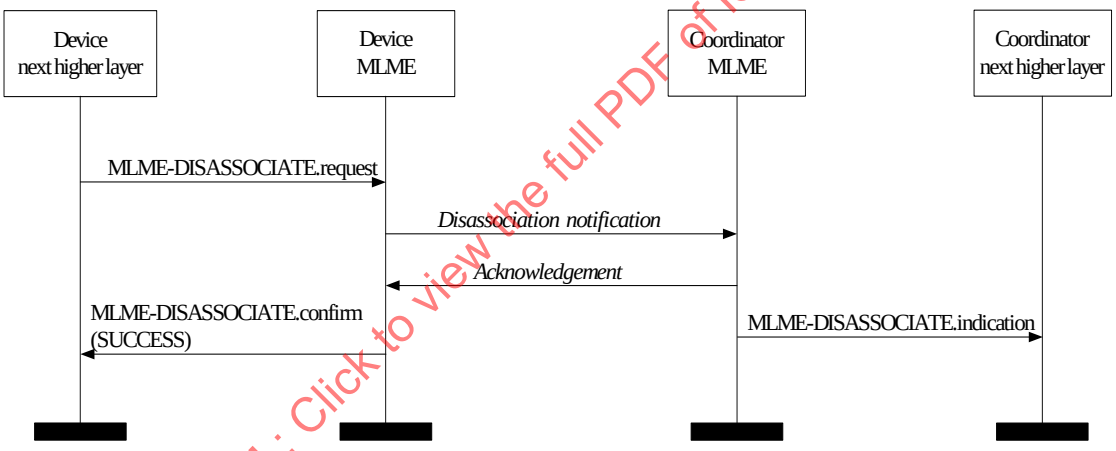


Figure 32—Message sequence chart for disassociation initiated by a device

Figure 33 illustrates the sequence necessary for a coordinator in a beacon-enabled PAN to successfully disassociate a device from its PAN using indirect transmission.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

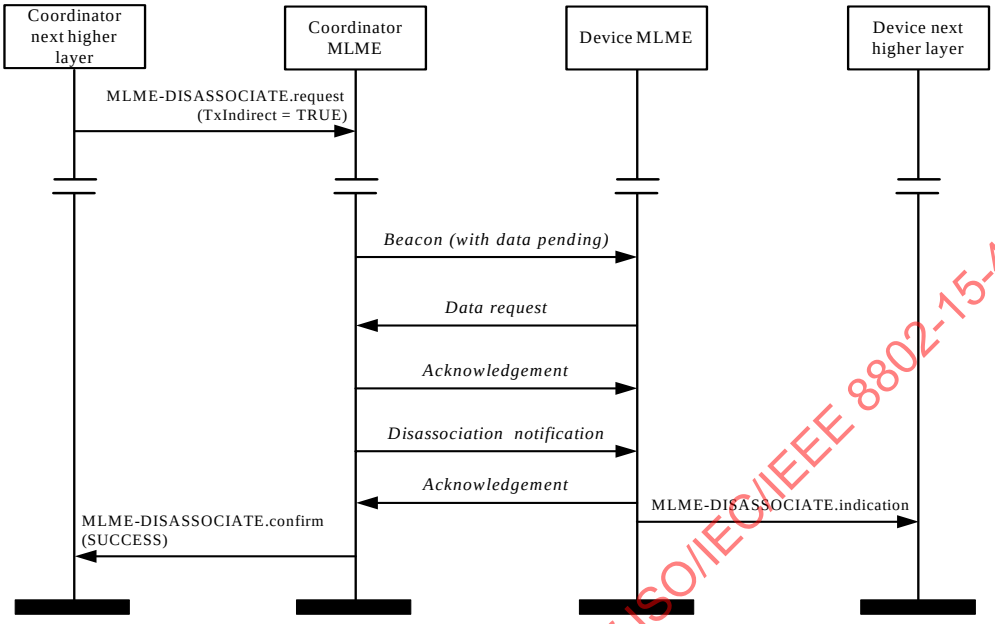


Figure 33—Message sequence chart for disassociation initiated by a coordinator, using indirect transmission, in a beacon-enabled PAN

7.1.5 Beacon notification primitive

The MLME-SAP beacon notification primitive defines how a device may be notified when a beacon is received during normal operating conditions.

All devices shall provide an interface for the beacon notification primitive.

7.1.5.1 MLME-BEACON-NOTIFY.indication

The MLME-BEACON-NOTIFY.indication primitive is used to send parameters contained within a beacon frame received by the MAC sublayer to the next higher layer. The primitive also sends a measure of the LQI and the time the beacon frame was received.

7.1.5.1.1 Semantics of the service primitive

The semantics of the MLME-BEACON-NOTIFY.indication primitive are as follows:

```
MLME-BEACON-NOTIFY.indication   (
                                BSN,
                                PANDescriptor,
                                PendAddrSpec,
                                AddrList,
                                sduLength,
                                sdu
                                )
```

Table 54 specifies the parameters for the MLME-BEACON-NOTIFY.indication primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 54—MLME-BEACON-NOTIFY.indication parameters

Name	Type	Valid range	Description
BSN	Integer	0x00–0xff	The beacon sequence number.
PANDescriptor	PANDescriptor value	See Table 55	The PANDescriptor for the received beacon.
PendAddrSpec	Bitmap	See 7.2.2.1.6	The beacon pending address specification.
AddrList	List of device addresses	—	The list of addresses of the devices for which the beacon source has data.
sduLength	Integer	0 – *aMaxBeaconPayloadLength*	The number of octets contained in the beacon payload of the beacon frame received by the MAC sublayer.
sdu	Set of octets	—	The set of octets comprising the beacon payload to be transferred from the MAC sublayer entity to the next higher layer.

Table 55 describes the elements of the PANDescriptor type.

Table 55—Elements of PANDescriptor

Name	Type	Valid range	Description
CoordAddrMode	Integer	0x02–0x03	The coordinator addressing mode corresponding to the received beacon frame. This value can take one of the following values: 2 = 16-bit short address. 3 = 64-bit extended address.
CoordPANId	Integer	0x0000–0xffff	The PAN identifier of the coordinator as specified in the received beacon frame.
CoordAddress	Device address	As specified by the CoordAddrMode parameter	The address of the coordinator as specified in the received beacon frame.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see 6.1.2)	The current logical channel occupied by the network.
ChannelPage	Integer	Selected from the available channel pages supported by the PHY (see 6.1.2)	The current channel page occupied by the network.
SuperframeSpec	Bitmap	See 7.2.2.1.2	The superframe specification as specified in the received beacon frame.
GTSPermit	Boolean	TRUE or FALSE	TRUE if the beacon is from the PAN coordinator that is accepting GTS requests.
LinkQuality	Integer	0x00–0xff	The LQI at which the network beacon was received. Lower values represent lower LQI (see 6.9.8).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 55—Elements of PANDescriptor *(continued)*

Name	Type	Valid range	Description
TimeStamp	Integer	0x000000–0xffffff	The time at which the beacon frame was received, in symbols. This value is equal to the timestamp taken when the beacon frame was received, as described in 7.5.4.1. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
SecurityFailure	Enumer-ation	SUCCESS, COUNTER_ERROR, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY	SUCCESS if there was no error in the security processing of the frame. One of the other status codes indicating an error in the security processing otherwise (see 7.5.8.2.3).
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received beacon frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.5.1.2 When generated

The MLME-BEACON-NOTIFY.indication primitive is generated by the MLME and issued to its next higher layer upon receipt of a beacon frame either when *macAutoRequest* is set to FALSE or when the beacon frame contains one or more octets of payload.

7.1.5.1.3 Appropriate usage

On receipt of the MLME-BEACON-NOTIFY.indication primitive, the next higher layer is notified of the arrival of a beacon frame at the MAC sublayer.

7.1.6 Primitives for reading PIB attributes

The MLME-SAP get primitives define how to read values from the PIB.

All devices shall provide an interface for these get primitives.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.6.1 MLME-GET.request

The MLME-GET.request primitive requests information about a given PIB attribute.

7.1.6.1.1 Semantics of the service primitive

The semantics of the MLME-GET.request primitive are as follows:

```
MLME-GET.request        (
                        PIBAttribute,
                        PIBAttributeIndex
                        )
```

Table 56 specifies the parameters for the MLME-GET.request primitive.

Table 56—MLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	See Table 86 and Table 88	The identifier of the PIB attribute to read.
PIBAttributeIndex	Integer	Attribute specific; see Table 88	The index within the table of the specified PIB attribute to read. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.

7.1.6.1.2 Appropriate usage

The MLME-GET.request primitive is generated by the next higher layer and issued to its MLME to obtain information from the PIB.

7.1.6.1.3 Effect on receipt

On receipt of the MLME-GET.request primitive, the MLME checks to see if the PIB attribute is a MAC PIB attribute or PHY PIB attribute. If the requested attribute is a MAC attribute, the MLME attempts to retrieve the requested MAC PIB attribute from its database. If the identifier of the PIB attribute is not found in the database, the MLME will issue the MLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttributeIndex parameter specifies an index for a table that is out of range, the MLME will issue the MLME-GET.confirm primitive with a status of INVALID_INDEX. If the requested MAC PIB attribute is successfully retrieved, the MLME will issue the MLME-GET.confirm primitive with a status of SUCCESS.

If the requested attribute is a PHY PIB attribute, the request is passed to the PHY by issuing the PLME-GET.request primitive. Once the MLME receives the PLME-GET.confirm primitive, it will translate the received status value because the status values used by the PHY are not the same as those used by the MLME (e.g., the status values for SUCCESS are 0x00 and 0x07 in the MAC and PHY enumeration tables, respectively). Following the translation, the MLME will issue the MLME-GET.confirm primitive to the next higher layer with the status parameter resulting from the translation and the PIBAttribute and PIBAttributeValue parameters equal to those returned by the PLME primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.6.2 MLME-GET.confirm

The MLME-GET.confirm primitive reports the results of an information request from the PIB.

7.1.6.2.1 Semantics of the service primitive

The semantics of the MLME-GET.confirm primitive are as follows:

```
MLME-GET.confirm        (
                        status,
                        PIBAttribute,
                        PIBAttributeIndex,
                        PIBAttributeValue
                        )
```

Table 57 specifies the parameters for the MLME-GET.confirm primitive.

Table 57—MLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE or INVALID_INDEX	The result of the request for PIB attribute information.
PIBAttribute	Integer	See Table 86 and Table 88	The identifier of the PIB attribute that was read.
PIBAttributeIndex	Integer	Attribute specific; see Table 88	The index within the table or array of the specified PIB attribute to read. This parameter is valid only for MAC PIB attributes that are tables or arrays; it is ignored when accessing PHY PIB attributes.
PIBAttributeValue	Various	Attribute specific; see Table 86 and Table 88	The value of the indicated PIB attribute that was read. This parameter has zero length when the status parameter is set to UNSUPPORTED_ATTRIBUTE.

7.1.6.2.2 When generated

The MLME-GET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-GET.request primitive. This primitive returns a status of either SUCCESS, indicating that the request to read a PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. When an error code of UNSUPPORTED_ATTRIBUTE is returned, the PIBAttribute value parameter will be set to length zero. The status values are fully described in 7.1.6.1.3.

7.1.6.2.3 Appropriate usage

On receipt of the MLME-GET.confirm primitive, the next higher layer is notified of the results of its request to read a PIB attribute. If the request to read a PIB attribute was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.7 GTS management primitives

The MLME-SAP GTS management primitives define how GTSs are requested and maintained. A device wishing to use these primitives and GTSs in general will already be tracking the beacons of its PAN coordinator.

These GTS management primitives are optional.

7.1.7.1 MLME-GTS.request

The MLME-GTS.request primitive allows a device to send a request to the PAN coordinator to allocate a new GTS or to deallocate an existing GTS. This primitive is also used by the PAN coordinator to initiate a GTS deallocation.

7.1.7.1.1 Semantics of the service primitive

The semantics of the MLME-GTS.request primitive are as follows:

```
MLME-GTS.request        (
                        GTSCharacteristics,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 58 specifies the parameters for the MLME-GTS.request primitive.

Table 58—MLME-GTS.request parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	See 7.3.9.2	The characteristics of the GTS request, including whether the request is for the allocation of a new GTS or the deallocation of an existing GTS.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.7.1.2 Appropriate usage

The MLME-GTS.request primitive is generated by the next higher layer of a device and issued to its MLME to request the allocation of a new GTS or to request the deallocation of an existing GTS. It is also generated by the next higher layer of the PAN coordinator and issued to its MLME to request the deallocation of an existing GTS.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.7.1.3 Effect on receipt

On receipt of the MLME-GTS.request primitive by a device, the MLME of a device attempts to generate a GTS request command (see 7.3.9) with the information contained in this primitive and, if successful, sends it to the PAN coordinator.

If *macShortAddress* is equal to 0xfffe or 0xffff, the device is not permitted to request a GTS. In this case, the MLME issues the MLME-GTS.confirm primitive containing a status of NO_SHORT_ADDRESS.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on *macCoordExtendedAddress* and the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-GTS.confirm primitive with the error status returned by outgoing frame processing.

If the GTS request command cannot be sent due to a CSMA-CA algorithm failure, the MLME will issue the MLME-GTS.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a GTS request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_ACK (see 7.5.6.4).

If a GTS is being allocated (see 7.5.7.2) and the request has been acknowledged, the device will wait for a confirmation via a GTS descriptor specified in a beacon frame from its PAN coordinator. If the MLME of the PAN coordinator can allocate the requested GTS, it will issue the MLME-GTS.indication primitive with the characteristics of the allocated GTS and generate a GTS descriptor with the characteristics of the allocated GTS and the 16-bit short address of the requesting device. If the MLME of the PAN coordinator cannot allocate the requested GTS, it will generate a GTS descriptor with a start slot of zero and the short address of the requesting device.

If the device receives a beacon frame from its PAN coordinator with a GTS descriptor containing a 16-bit short address that matches *macShortAddress*, the device will process the descriptor. If no descriptor for that device is received, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_DATA.

If a descriptor is received that matches the characteristics requested (indicating that the PAN coordinator has approved the GTS allocation request), the MLME of the device will issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to one, indicating a GTS allocation.

If the descriptor is received with a start slot of zero (indicating that the PAN coordinator has denied the GTS allocation request), the device requesting the GTS issues the MLME-GTS.confirm primitive with a status of DENIED, indicating that the GTSCharacteristics parameter is to be ignored.

If a GTS is being deallocated (see 7.5.7.4) at the request of a device and the request has been acknowledged by the PAN coordinator, the device will issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to zero, indicating a GTS deallocation. On receipt of a GTS request command with a request type indicating a GTS deallocation, the PAN coordinator will acknowledge the frame and deallocates the GTS. The MLME of the PAN coordinator will then issue the MLME-GTS.indication primitive with the appropriate GTS characteristics. If the PAN coordinator does not receive the deallocation request, countermeasures can be applied by the PAN coordinator to ensure consistency is maintained (see 7.5.7.6).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the MLME of the PAN coordinator receives an MLME-GTS.request primitive indicating deallocation, the PAN coordinator will deallocate the GTS and issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to zero.

If the device receives a beacon frame from its PAN coordinator with a GTS descriptor containing a short address that matches *macShortAddress* and a start slot equal to zero, the device immediately stops using the GTS. The MLME of the device then notifies the next higher layer of the deallocation by issuing the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS.

If any parameter in the MLME-GTS.request primitive is not supported or is out of range, the MLME will issue the MLME-GTS.confirm primitive with a status of INVALID_PARAMETER.

7.1.7.2 MLME-GTS.confirm

The MLME-GTS.confirm primitive reports the results of a request to allocate a new GTS or deallocate an existing GTS.

7.1.7.2.1 Semantics of the service primitive

The semantics of the MLME-GTS.confirm primitive are as follows:

```
MLME-GTS.confirm        (
                        GTSCharacteristics,
                        status
                        )
```

Table 59 specifies the parameters for the MLME-GTS.confirm primitive.

Table 59—MLME-GTS.confirm parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	See 7.3.9.2	The characteristics of the GTS.
status	Enumeration	SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER.	The status of the GTS request.

7.1.7.2.2 When generated

The MLME-GTS.confirm primitive is generated by the MLME and issued to its next higher layer in response to a previously issued MLME-GTS.request primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the request to allocate or deallocate a GTS was successful, this primitive will return a status of SUCCESS and the Characteristics Type field of the GTSCharacteristics parameter will have the value of one or zero, respectively. Otherwise, the status parameter will indicate the appropriate error code. The reasons for these status values are fully described in 7.1.7.1.3 and subclauses referenced by 7.1.7.1.3.

7.1.7.2.3 Appropriate usage

On receipt of the MLME-GTS.confirm primitive the next higher layer is notified of the result of its request to allocate or deallocate a GTS. If the request was successful, the status parameter will indicate a successful GTS operation. Otherwise, the status parameter will indicate the error.

7.1.7.3 MLME-GTS.indication

The MLME-GTS.indication primitive indicates that a GTS has been allocated or that a previously allocated GTS has been deallocated.

7.1.7.3.1 Semantics of the service primitive

The semantics of the MLME-GTS.indication primitive are as follows:

```
MLME-GTS.indication     (
                        DeviceAddress,
                        GTSCharacteristics,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 60 specifies the parameters for the MLME-GTS.indication primitive.

Table 60—MLME-GTS.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	0x0000–0xfffd	The 16-bit short address of the device that has been allocated or deallocated a GTS.
GTSCharacteristics	GTS characteristics	See 7.3.9.2	The characteristics of the GTS.
SecurityLevel	Integer	0x00–0x07	If the primitive was generated when a GTS deallocation is initiated by the PAN coordinator itself, the security level to be used is set to 0x00. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The security level purportedly used by the received MAC command frame (see Table 95 in 7.6.2.2.1).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 60—MLME-GTS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeyIdMode	Integer	0x00–0x03	If the primitive was generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	If the primitive was generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.1.7.3.2 When generated

The MLME-GTS.indication primitive is generated by the MLME of the PAN coordinator to its next higher layer whenever a GTS is allocated or deallocated following the reception of a GTS request command (see 7.3.9) by the MLME. The MLME of the PAN coordinator also generates this primitive when a GTS deallocation is initiated by the PAN coordinator itself. The Characteristics Type field in the GTSCharacteristics parameter will be equal to one if a GTS has been allocated or zero if a GTS has been deallocated.

This primitive is generated by the MLME of a device and issued to its next higher layer when the PAN coordinator has deallocated one of its GTSs. In this case, the Characteristics Type field of the GTSCharacteristics parameter is equal to zero.

© IEEE 2010 – All rights reserved

7.1.7.3.3 Appropriate usage

On receipt of the MLME-GTS.indication primitive the next higher layer is notified of the allocation or de-allocation of a GTS.

7.1.7.4 GTS management message sequence charts

Figure 34 and Figure 35 illustrate the sequence of messages necessary for successful GTS management. The first depicts the message flow for the case in which the device initiates the GTS allocation. The second depicts the message flow for the two cases for which a GTS deallocation occurs, first, by a device (a) and, second, by the PAN coordinator (b).

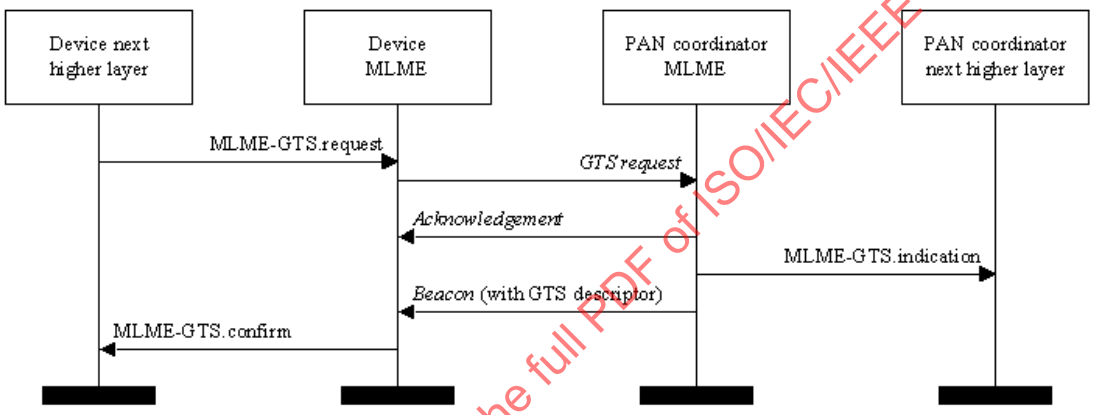


Figure 34—Message sequence chart for GTS allocation initiated by a device

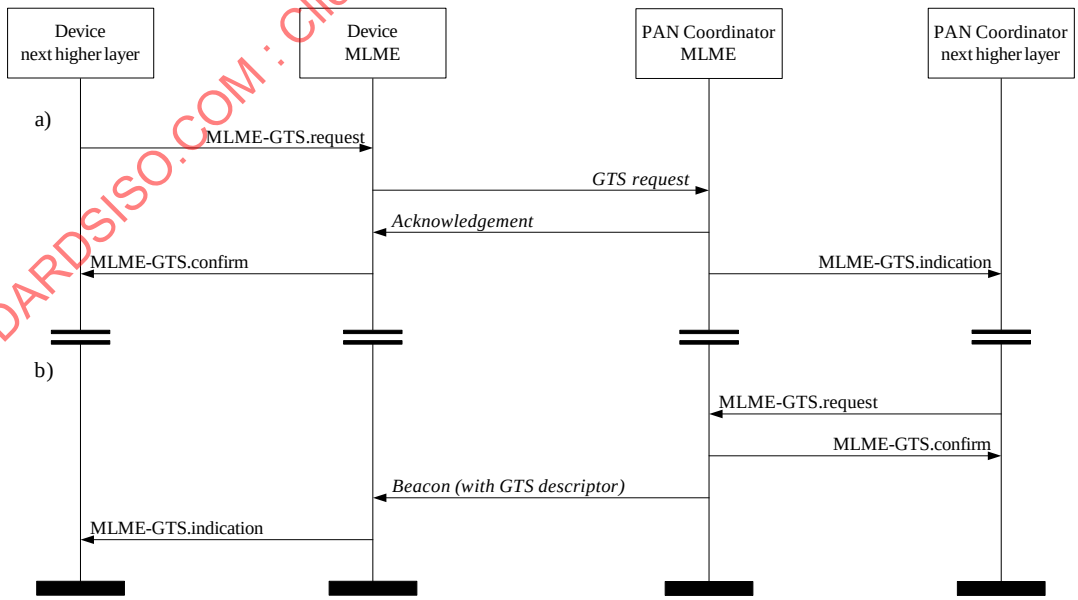


Figure 35—Message sequence chart for GTS deallocation initiated by a device (a) and the PAN coordinator (b)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.8 Primitives for orphan notification

MLME-SAP orphan notification primitives define how a coordinator can issue a notification of an orphaned device.

These orphan notification primitives are optional for an RFD.

7.1.8.1 MLME-ORPHAN.indication

The MLME-ORPHAN.indication primitive allows the MLME of a coordinator to notify the next higher layer of the presence of an orphaned device.

7.1.8.1.1 Semantics of the service primitive

The semantics of the MLME-ORPHAN.indication primitive are as follows:

```
MLME-ORPHAN.indication      (
                            OrphanAddress,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 61 specifies the parameters for the MLME-ORPHAN.indication primitive.

Table 61—MLME-ORPHAN.indication parameters

Name	Type	Valid range	Description
OrphanAddress	Device address	Extended 64-bit IEEE address	The address of the orphaned device.
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received MAC command frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.8.1.2 When generated

The MLME-ORPHAN.indication primitive is generated by the MLME of a coordinator and issued to its next higher layer on receipt of an orphan notification command (see 7.3.6).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.8.1.3 Appropriate usage

The effect on receipt of the MLME-ORPHAN.indication primitive is that the next higher layer is notified of the orphaned device. The next higher layer then determines whether the device was previously associated and issues the MLME-ORPHAN.response primitive to the MLME with its decision (see 7.5.2.1.4).

If the device was previously associated with the coordinator, it will send the MLME-ORPHAN.response primitive with the AssociatedMember parameter set to TRUE and the ShortAddress parameter set to the corresponding 16-bit short address allocated to the orphaned device. If the device was not previously associated with the coordinator, it will send the MLME-ORPHAN.response primitive with the AssociatedMember parameter set to FALSE.

7.1.8.2 MLME-ORPHAN.response

The MLME-ORPHAN.response primitive allows the next higher layer of a coordinator to respond to the MLME-ORPHAN.indication primitive.

7.1.8.2.1 Semantics of the service primitive

The semantics of the MLME-ORPHAN.response primitive are as follows:

```
MLME-ORPHAN.response        (
                            OrphanAddress,
                            ShortAddress,
                            AssociatedMember,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 62 specifies the parameters for the MLME-ORPHAN.response primitive.

Table 62—MLME-ORPHAN.response parameters

Name	Type	Valid range	Description
OrphanAddress	Device address	Extended 64-bit IEEE address	The address of the orphaned device.
ShortAddress	Integer	0x0000–0xffff	The 16-bit short address allocated to the orphaned device if it is associated with this coordinator. The special short address 0xfffe indicates that no short address was allocated, and the device will use its 64-bit extended address in all communications. If the device was not associated with this coordinator, this field will contain the value 0xffff and be ignored on receipt.
AssociatedMember	Boolean	TRUE or FALSE	TRUE if the orphaned device is associated with this coordinator or FALSE otherwise.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 62—MLME-ORPHAN.response parameters *(continued)*

Name	Type	Valid range	Description
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.8.2.2 Appropriate usage

The MLME-ORPHAN.response primitive is generated by the next higher layer and issued to its MLME when it reaches a decision about whether the orphaned device indicated in the MLME-ORPHAN.indication primitive is associated.

7.1.8.2.3 Effect on receipt

If the AssociatedMember parameter is set to TRUE, the orphaned device is associated with the coordinator. In this case, the MLME generates and sends the coordinator realignment command (see 7.3.8) to the orphaned device containing the value of the ShortAddress field. This command is sent in the CAP if the coordinator is on a beacon-enabled PAN or immediately otherwise. If the AssociatedMember parameter is set to FALSE, the orphaned device is not associated with the coordinator and this primitive will be ignored. If the orphaned device does not receive the coordinator realignment command following its orphan notification within *macResponseWaitTime* symbols, it will assume it is not associated to any coordinator in range.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on the OrphanAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with the error status returned by outgoing frame processing.

If the CSMA-CA algorithm failed due to adverse conditions on the channel, the MAC sublayer will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE.

The MAC sublayer enables its receiver immediately following the transmission of the frame and waits for an acknowledgment from the recipient (see 7.5.6.4). If the MAC sublayer does not receive an acknowledgment from the recipient, it will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of NO_ACK.

If the frame was successfully transmitted and an acknowledgment was received, if requested, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of SUCCESS.

If any parameter in the MLME-ORPHAN.response primitive is not supported or is out of range, the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of INVALID_PARAMETER.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.8.3 Orphan notification message sequence chart

Figure 36 illustrates the sequence of messages necessary for a coordinator to issue a notification of an orphaned device.

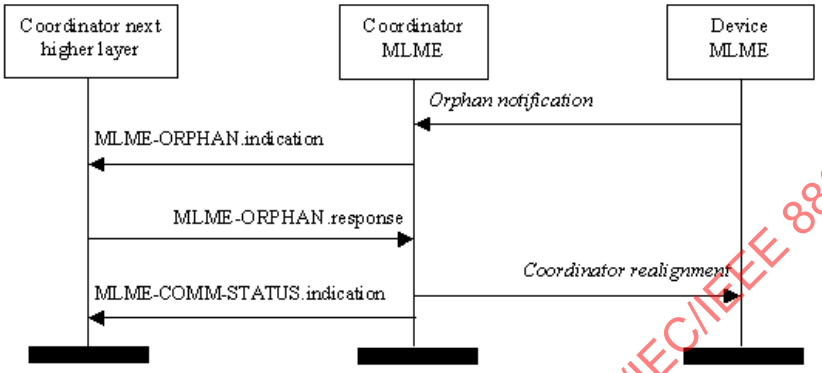


Figure 36—Message sequence chart for orphan notification

7.1.9 Primitives for resetting the MAC sublayer

MLME-SAP reset primitives specify how to reset the MAC sublayer to its default values.

All devices shall provide an interface for these reset primitives.

7.1.9.1 MLME-RESET.request

The MLME-RESET.request primitive allows the next higher layer to request that the MLME performs a reset operation.

7.1.9.1.1 Semantics of the service primitive

The semantics of the MLME-RESET.request primitive are as follows:

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

Table 63 specifies the parameter for the MLME-RESET.request primitive.

Table 63—MLME-RESET.request parameter

Name	Type	Valid range	Description
SetDefaultPIB	Boolean	TRUE or FALSE	If TRUE, the MAC sublayer is reset, and all MAC PIB attributes are set to their default values. If FALSE, the MAC sublayer is reset, but all MAC PIB attributes retain their values prior to the generation of the MLME-RESET.request primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.9.1.2 Appropriate usage

The MLME-RESET.request primitive is generated by the next higher layer and issued to the MLME to request a reset of the MAC sublayer to its initial conditions. The MLME-RESET.request primitive is issued prior to the use of the MLME-START.request or the MLME-ASSOCIATE.request primitives.

7.1.9.1.3 Effect on receipt

On receipt of the MLME-RESET.request primitive, the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of FORCE_TRX_OFF. On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MAC sublayer is then set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultPIB parameter is set to TRUE, the MAC PIB attributes are set to their default values.

The MLME-RESET.confirm primitive with a status of SUCCESS is issued on completion.

7.1.9.2 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

7.1.9.2.1 Semantics of the service primitive

The semantics of the MLME-RESET.confirm primitive are as follows:

```
MLME-RESET.confirm      (
                        status
                        )
```

Table 64 specifies the parameter for the MLME-RESET.confirm primitive.

Table 64—MLME-RESET.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS	The result of the reset operation.

7.1.9.2.2 When generated

The MLME-RESET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RESET.request primitive and following the receipt of the PLME-SET-TRX-STATE.confirm primitive.

7.1.9.2.3 Appropriate usage

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of its request to reset the MAC sublayer. This primitive returns a status of SUCCESS indicating that the request to reset the MAC sublayer was successful.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.10 Primitives for specifying the receiver enable time

MLME-SAP receiver state primitives define how a device can enable or disable the receiver at a given time.

These receiver state primitives are optional.

7.1.10.1 MLME-RX-ENABLE.request

The MLME-RX-ENABLE.request primitive allows the next higher layer to request that the receiver is either enabled for a finite period of time or disabled.

7.1.10.1.1 Semantics of the service primitive

The semantics of the MLME-RX-ENABLE.request primitive are as follows:

```
MLME-RX-ENABLE.request      (
                            DeferPermit,
                            RxOnTime,
                            RxOnDuration
                            )
```

Table 65 specifies the parameters for the MLME-RX-ENABLE.request primitive.

Table 65—MLME-RX-ENABLE.request parameters

Name	Type	Valid range	Description
DeferPermit	Boolean	TRUE or FALSE	TRUE if the requested operation can be deferred until the next superframe if the requested time has already passed. FALSE if the requested operation is only to be attempted in the current superframe. This parameter is ignored for nonbeacon-enabled PANs. If the issuing device is the PAN coordinator, the term *superframe* refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated.
RxOnTime	Integer	0x000000–0xffffff	The number of symbols measured from the start of the superframe before the receiver is to be enabled or disabled. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant. This parameter is ignored for nonbeacon-enabled PANs. If the issuing device is the PAN coordinator, the term *superframe* refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated.
RxOnDuration	Integer	0x000000–0xffffff	The number of symbols for which the receiver is to be enabled. If this parameter is equal to 0x000000, the receiver is to be disabled.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.10.1.2 Appropriate usage

The MLME-RX-ENABLE.request primitive is generated by the next higher layer and issued to the MLME to enable the receiver for a fixed duration, at a time relative to the start of the current or next superframe on a beacon-enabled PAN or immediately on a nonbeacon-enabled PAN. This primitive may also be generated to cancel a previously generated request to enable the receiver. The receiver is enabled or disabled exactly once per primitive request.

7.1.10.1.3 Effect on receipt

The MLME will treat the request to enable or disable the receiver as secondary to other responsibilities of the device (e.g., GTSs, coordinator beacon tracking, beacon transmissions). When the primitive is issued to enable the receiver, the device will enable its receiver until either the device has a conflicting responsibility or the time specified by RxOnDuration has expired. In the case of a conflicting responsibility, the device will interrupt the receive operation. After the completion of the interrupting operation, the RxOnDuration will be checked to determine whether the time has expired. If so, the operation is complete. If not, the receiver is re-enabled until either the device has another conflicting responsibility or the time specified by RxOnDuration has expired. When the primitive is issued to disable the receiver, the device will disable its receiver unless the device has a conflicting responsibility.

On a nonbeacon-enabled PAN, the MLME ignores the DeferPermit and RxOnTime parameters and requests that the PHY enable or disable the receiver immediately. If the request is to enable the receiver, the receiver will remain enabled until RxOnDuration symbols have elapsed.

Before attempting to enable the receiver on a beacon-enabled PAN, the MLME first determines whether (RxOnTime + RxOnDuration) is less than the beacon interval, as defined by *macBeaconOrder*. If (RxOnTime + RxOnDuration) is not less than the beacon interval, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of ON_TIME_TOO_LONG.

The MLME then determines whether the receiver can be enabled in the current superframe. The PAN coordinator issuing this primitive makes the determination based on its own superframe. A device that is not the PAN coordinator makes the determination based on the superframe of the coordinator through which it is associated. If the current number of symbols measured from the start of the superframe is less than (RxOnTime – *aTurnaroundTime*), the MLME attempts to enable the receiver in the current superframe. If the current number of symbols measured from the start of the superframe is greater than or equal to (RxOnTime – *aTurnaroundTime*) and DeferPermit is equal to TRUE, the MLME defers until the next superframe and attempts to enable the receiver in that superframe. Otherwise, if the MLME cannot enable the receiver in the current superframe and is not permitted to defer the receive operation until the next superframe, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of PAST_TIME.

If the RxOnDuration parameter is equal to zero, the MLME requests that the PHY disable its receiver.

If any parameter in the MLME-RX-ENABLE.request primitive is not supported or is out of range, the MAC sublayer will issue the MLME-RX-ENABLE.confirm primitive with a status of INVALID_PARAMETER.

If the request to enable or disable the receiver was successful, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of SUCCESS.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.10.2 MLME-RX-ENABLE.confirm

The MLME-RX-ENABLE.confirm primitive reports the results of the attempt to enable or disable the receiver.

7.1.10.2.1 Semantics of the service primitive

The semantics of the MLME-RX-ENABLE.confirm primitive are as follows:

```
MLME-RX-ENABLE.confirm      (
                            status
                            )
```

Table 66 specifies the parameter for the MLME-RX-ENABLE.confirm primitive.

Table 66—MLME-RX-ENABLE.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, PAST_TIME, ON_TIME_TOO_LONG or INVALID_PARAMETER	The result of the request to enable or disable the receiver.

7.1.10.2.2 When generated

The MLME-RX-ENABLE.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RX-ENABLE.request primitive.

7.1.10.2.3 Appropriate usage

On receipt of the MLME-RX-ENABLE.confirm primitive, the next higher layer is notified of its request to enable or disable the receiver. This primitive returns a status of either SUCCESS, if the request to enable or disable the receiver was successful, or the appropriate error code. The status values are fully described in 7.1.10.1.3.

7.1.10.3 Message sequence chart for changing the state of the receiver

Figure 37 illustrates the sequence of messages necessary for enabling the receiver for a fixed duration when the device does not have any conflicting responsibilities. Figure 37a illustrates the case for a beacon-enabled PAN where it is assumed both that the MLME-RX-ENABLE.request has been received by the MLME without sufficient time available to enable the receiver in the current superframe and that the DeferPermit parameter is TRUE. Figure 37b illustrates the case for a nonbeacon-enabled PAN where the receiver is enabled immediately.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

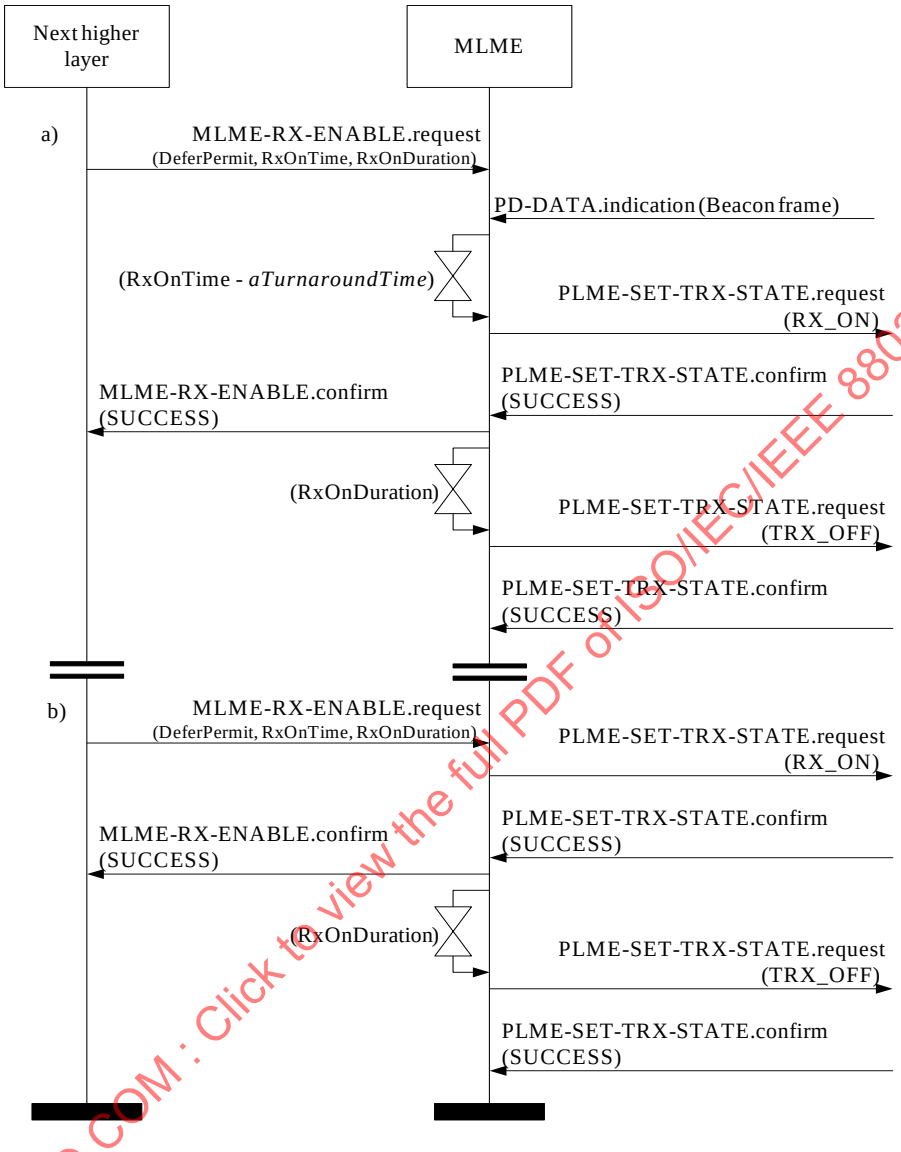


Figure 37—Message sequence chart for changing the state of the receiver

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.1.11 Primitives for channel scanning

MLME-SAP scan primitives define how a device can determine the energy usage or the presence or absence of PANs in a communications channel.

All devices shall provide an interface for these scan primitives.

7.1.11.1 MLME-SCAN.request

The MLME-SCAN.request primitive is used to initiate a channel scan over a given list of channels. A device can use a channel scan to measure the energy on the channel, search for the coordinator with which it associated, or search for all coordinators transmitting beacon frames within the POS of the scanning device.

© IEEE 2010 – All rights reserved

7.1.11.1.1 Semantics of the service primitive

The semantics of the MLME-SCAN.request primitive are as follows:

```
MLME-SCAN.request       (
                        ScanType,
                        ScanChannels,
                        ScanDuration,
                        ChannelPage,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 67 specifies the parameters for the MLME-SCAN.request primitive.

Table 67—MLME-SCAN.request parameters

Name	Type	Valid range	Description
ScanType	Integer	0x00–0x03	Indicates the type of scan performed: 0x00 = ED scan (optional for RFD). 0x01 = active scan (optional for RFD). 0x02 = passive scan. 0x03 = orphan scan.
ScanChannels	Bitmap	27-bit field	The 27 bits (b_0, b_1,... b_{26}) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 channels supported by the ChannelPage parameter.
ScanDuration	Integer	0–14	A value used to calculate the length of time to spend scanning each channel for ED, active, and passive scans. This parameter is ignored for orphan scans. The time spent scanning each channel is [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter.
ChannelPage	Integer	0–31	The channel page on which to perform the scan (see 6.1.2).
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 67—MLME-SCAN.request parameters *(continued)*

Name	Type	Valid range	Description
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

7.1.11.1.2 Appropriate usage

The MLME-SCAN.request primitive is generated by the next higher layer and issued to its MLME to initiate a channel scan to search for activity within the POS of the device. This primitive can be used to perform an ED scan to determine channel usage, an active or passive scan to locate beacon frames containing any PAN identifier, or an orphan scan to locate a PAN to which the device is currently associated. See 7.5.2.1 for a description of each type of scan in detail.

All devices shall be capable of performing passive scans and orphan scans; ED scans and active scans are optional for an RFD. However, an RFD may support active scanning to participate in a nonbeacon-enabled network.

7.1.11.1.3 Effect on receipt

If the MLME receives the MLME-SCAN.request primitive while performing a previously initiated scan operation, it issues the MLME-SCAN.confirm primitive with a status of SCAN_IN_PROGRESS. Otherwise, the MLME initiates a scan in all channels specified in the ScanChannels parameter.

The ED scan is performed on each channel by the MLME repeatedly issuing the PLME-ED.request primitive to the PHY until [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter, have elapsed. The MLME notes the maximum energy measurement and moves on to the next channel in the channel list. See 7.5.2.1.1 for more detailed information on the ED channel scan procedure.

The active scan is performed on each channel by the MLME first sending a beacon request command (see 7.3.7). The MLME then enables the receiver and records the information contained in each received beacon in a PAN descriptor structure (see Table 55 in 7.1.5.1.1). The active scan on a particular channel terminates when the number of PAN descriptors stored equals an implementation-specified maximum or when [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter, have elapsed, whichever comes first. See 7.5.2.1.2 for more detailed information on the active channel scan procedure.

The passive scan is performed on each channel by the MLME enabling its receiver and recording the information contained in each received beacon in a PAN descriptor structure (see Table 55 in 7.1.5.1.1). The passive scan on a particular channel terminates when the number of PAN descriptors stored equals an implementation-specified maximum or when [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter, have elapsed, whichever comes first. See 7.5.2.1.3 for more detailed information on the passive channel scan procedure.

The orphan scan is performed on each channel by the MLME first sending an orphan notification command (see 7.3.6). If the device receives a coordinator realignment command in return, the MLME will disable its

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

receiver. Otherwise, the scan is repeated on the next channel in the channel list. See 7.5.2.1.4 for more detailed information on the orphan channel scan procedure.

The SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters are used only in an orphan scan.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on *macCoordExtendedAddress*, the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-SCAN.confirm primitive with the error status returned by outgoing frame processing.

The results of an ED scan are recorded in a list of ED values and reported to the next higher layer through the MLME-SCAN.confirm primitive with a status of SUCCESS.

The results of an active or passive scan are reported to the next higher layer through the MLME-SCAN.confirm primitive. If the scan is successful and *macAutoRequest* is set to TRUE, the primitive results will include a set of PAN descriptor values. If the scan is successful and *macAutoRequest* is set to FALSE, the primitive results will contain a null set of PAN descriptor values; each PAN descriptor value will be sent individually to the next higher layer using separate MLME-BEACON-NOTIFY (see 7.1.5.1) primitives. In both cases, the MLME-SCAN.confirm primitive will contain a list of unscanned channels and a status of SUCCESS.

If, during an active scan, the MLME is unable to transmit a beacon request command on a channel specified by the ScanChannels parameter due to a channel access failure, the channel will appear in the list of unscanned channels returned by the MLME-SCAN.confirm primitive. If the MLME was able to send a beacon request command on at least one of the channels but no beacons were found, the MLME-SCAN.confirm primitive will contain a null set of PAN descriptor values, regardless of the value of *macAutoRequest*, and a status of NO_BEACON.

The results of an orphan scan are reported to the next higher layer through the MLME-SCAN.confirm primitive. If successful, the MLME-SCAN.confirm primitive will contain a status of SUCCESS. If the MLME is unable to transmit an orphan notification command on a channel specified by the ScanChannels parameter due to a channel access failure, the channel will appear in the list of unscanned channels returned by the MLME-SCAN.confirm primitive. If the MLME was able to send an orphan notification command on at least one of the channels but the device did not receive a coordinator realignment command, the MLME-SCAN.confirm primitive will contain a status of NO_BEACON. In this case, the PANDescriptorList and EnergyDetectList parameters of the MLME-SCAN.confirm primitive are null.

If, during an ED, active, or passive scan, the implementation-specified maximum is reached thus terminating the scan procedure, the MAC sublayer will issue the MLME-SCAN.confirm primitive with a status of LIMIT_REACHED.

If any parameter in the MLME-SCAN.request primitive is not supported or is out of range, the MAC sublayer will issue the MLME-SCAN.confirm primitive with a status of INVALID_PARAMETER.

7.1.11.2 MLME-SCAN.confirm

The MLME-SCAN.confirm primitive reports the result of the channel scan request.

7.1.11.2.1 Semantics of the service primitive

The semantics of the MLME-SCAN.confirm primitive are as follows:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

```
MLME-SCAN.confirm       (
                        status,
                        ScanType,
                        ChannelPage,
                        UnscannedChannels,
                        ResultListSize,
                        EnergyDetectList,
                        PANDescriptorList
                        )
```

Table 68 specifies the parameters for the MLME-SCAN.confirm primitive.

Table 68—MLME-SCAN.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, LIMIT_REACHED, NO_BEACON, SCAN_IN_PROGRESS, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the scan request.
ScanType	Integer	0x00–0x03	Indicates the type of scan performed: 0x00 = ED scan (optional for RFD). 0x01 = active scan (optional for RFD). 0x02 = passive scan. 0x03 = orphan scan.
ChannelPage	Integer	0–31	The channel page on which the scan was performed (see 6.1.2).
UnscannedChannels	Bitmap	27 bit field	Indicates which channels given in the request were not scanned (1 = not scanned, 0 = scanned or not requested). This parameter is not valid for ED scans.
ResultListSize	Integer	Implementation specific	The number of elements returned in the appropriate result lists. This value is zero for the result of an orphan scan.
EnergyDetectList	List of integers	0x00–0xff for each integer	The list of energy measurements, one for each channel searched during an ED scan. This parameter is null for active, passive, and orphan scans.
PANDescriptorList	List of PAN descriptor values	See Table 55	The list of PAN descriptors, one for each beacon found during an active or passive scan if *macAutoRequest* is set to TRUE. This parameter is null for ED and orphan scans or when *macAutoRequest* is set to FALSE during an active or passive scan.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.11.2.2 When generated

The MLME-SCAN.confirm primitive is generated by the MLME and issued to its next higher layer when the channel scan initiated with the MLME-SCAN.request primitive has completed. If the MLME-SCAN.request primitive requested an active, passive, or orphan scan, the EnergyDetectList parameter will be null. If the MLME-SCAN.request primitive requested an ED or orphan scan, the PANDescriptorList parameter will be null; this is also the case if the MLME-SCAN.request primitive requested an active or passive scan with *macAutoRequest* set to FALSE. If the MLME-SCAN.request primitive requested an orphan scan, the ResultListSize parameter will be set to zero.

The MLME-SCAN.confirm primitive returns a status of either SUCCESS, indicating that the requested scan was successful, or the appropriate error code. The status values are fully described in 7.1.11.1.3 and subclauses referenced by 7.1.11.1.3.

7.1.11.2.3 Appropriate usage

On receipt of the MLME-SCAN.confirm primitive, the next higher layer is notified of the results of the scan procedure. If the requested scan was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.11.3 Channel scan message sequence charts

Figure 79, Figure 82, Figure 83, and Figure 86 (see 7.7) illustrate the sequence of messages necessary to perform an ED scan, a passive scan, an active scan, and an orphan scan, respectively. These figures include steps taken by the PHY.

7.1.12 Communication status primitive

The MLME-SAP communication status primitive defines how the MLME communicates to the next higher layer about transmission status, when the transmission was instigated by a response primitive, and about security errors on incoming packets.

All devices shall provide an interface for this communication status primitive.

7.1.12.1 MLME-COMM-STATUS.indication

The MLME-COMM-STATUS.indication primitive allows the MLME to indicate a communications status.

7.1.12.1.1 Semantics of the service primitive

The semantics of the MLME-COMM-STATUS.indication primitive are as follows:

```
MLME-COMM-STATUS.indication     (
                                PANId,
                                SrcAddrMode,
                                SrcAddr,
                                DstAddrMode,
                                DstAddr,
                                status,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex
                                )
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 69 specifies the parameters for the MLME-COMM-STATUS.indication primitive.

Table 69—MLME-COMM-STATUS.indication parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000–0xffff	The 16-bit PAN identifier of the device from which the frame was received or to which the frame was being sent.
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive. This value can take one of the following values: 0 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the frame causing the error originated.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the device for which the frame was intended.
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The communications status.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 69—MLME-COMM-STATUS.indication parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	If the primitive was generated following a transmission instigated through a response primitive: The security level to be used (see Table 95 in 7.6.2.2.1). If the primitive was generated on receipt of a frame that generates an error in its security processing: The security level purportedly used by the received frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was generated following a transmission instigated through a response primitive: The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated following a transmission instigated through a response primitive: The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 69—MLME-COMM-STATUS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeyIndex	Integer	0x01–0xff	If the primitive was generated following a transmission instigated through a response primitive: The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.12.1.2 When generated

The MLME-COMM-STATUS.indication primitive is generated by the MLME and issued to its next higher layer either following a transmission instigated through a response primitive or on receipt of a frame that generates an error in its security processing (see 7.5.8.2.3).

The MLME-COMM-STATUS.indication primitive is generated by the MAC sublayer entity following either the MLME-ASSOCIATE.response primitive or the MLME-ORPHAN.response primitive. This primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, an error code of TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK or INVALID_PARAMETER (these status values are fully described in 7.1.3.3.3 and 7.1.8.2.3), or an error code resulting from failed security processing (these status values are fully described in 7.5.8.2.1 and 7.5.8.2.3).

7.1.12.1.3 Appropriate usage

On receipt of the MLME-COMM-STATUS.indication primitive, the next higher layer is notified of the communication status of a transmission or notified of an error that has occurred during the secure processing of incoming frame.

7.1.13 Primitives for writing PIB attributes

MLME-SAP set primitives define how PIB attributes may be written.

All devices shall provide an interface for these set primitives.

7.1.13.1 MLME-SET.request

The MLME-SET.request primitive attempts to write the given value to the indicated PIB attribute.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.13.1.1 Semantics of the primitive

The semantics of the MLME-SET.request primitive are as follows:

```
MLME-SET.request        (
                        PIBAttribute,
                        PIBAttributeIndex,
                        PIBAttributeValue
                        )
```

Table 70 specifies the parameters for the MLME-SET.request primitive.

Table 70—MLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	See Table 86 and Table 88	The identifier of the PIB attribute to write.
PIBAttributeIndex	Integer	Attribute specific; see Table 88	The index within the table of the specified PIB attribute to write. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.
PIBAttributeValue	Various	Attribute specific; see Table 86 and Table 88	The value to write to the indicated PIB attribute.

7.1.13.1.2 Appropriate usage

The MLME-SET.request primitive is generated by the next higher layer and issued to its MLME to write the indicated PIB attribute.

7.1.13.1.3 Effect on receipt

On receipt of the MLME-SET.request primitive, the MLME checks to see if the PIB attribute is a MAC PIB attribute or PHY PIB attribute. If the requested attribute is a MAC attribute, the MLME attempts to write the given value to the indicated MAC PIB attribute in its database. If the PIBAttribute parameter specifies an attribute that is a read-only attribute (see Table 86 and Table 88), the MLME will issue the MLME-SET.confirm primitive with a status of READ_ONLY. If the PIBAttribute parameter specifies an attribute that is not found in the database, the MLME will issue the MLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttributeIndex parameter specifies an index for a table that is out of range, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_INDEX. If the PIBAttributeValue parameter specifies a value that is out of the valid range for the given attribute, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_PARAMETER. If the requested MAC PIB attribute is successfully written, the MLME will issue the MLME-SET.confirm primitive with a status of SUCCESS.

If the PIBAttribute parameter indicates that *macBeaconPayloadLength* is to be set and the length of the resulting beacon frame exceeds *aMaxPHYPacketSize* (e.g., due to the additional overhead required for security processing), the MAC sublayer shall not update *macBeaconPayloadLength* and will issue the MLME-GET.confirm primitive with a status of INVALID_PARAMETER.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the requested attribute is a PHY PIB attribute, the request is passed to the PHY by issuing the PLME-SET.request primitive. Once the MLME receives the PLME-SET.confirm primitive, it will translate the received status value because the status values used by the PHY are not the same as those used by the MLME (e.g., the status values for SUCCESS are 0x00 and 0x07 in the MAC and PHY enumeration tables, respectively). Following the translation, the MLME will issue the MLME-SET.confirm primitive to the next higher layer with the status parameter resulting from the translation and the PIBAttribute parameter equal to that returned by the PLME primitive.

7.1.13.2 MLME-SET.confirm

The MLME-SET.confirm primitive reports the results of an attempt to write a value to a PIB attribute.

7.1.13.2.1 Semantics of the service primitive

The semantics of the MLME-SET.confirm primitive are as follows:

```
MLME-SET.confirm        (
                        status,
                        PIBAttribute,
                        PIBAttributeIndex
                        )
```

Table 71 specifies the parameters for the MLME-SET.confirm primitive.

Table 71—MLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, READ_ONLY, UNSUPPORTED_ATTRIBUTE, INVALID_INDEX or INVALID_PARAMETER	The result of the request to write the PIB attribute.
PIBAttribute	Integer	See Table 86 and Table 88	The identifier of the PIB attribute that was written.
PIBAttributeIndex	Integer	Attribute specific; see Table 88	The index within the table of the specified PIB attribute to write. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.

7.1.13.2.2 When generated

The MLME-SET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-SET.request primitive. The MLME-SET.confirm primitive returns a status of either SUCCESS, indicating that the requested value was written to the indicated PIB attribute, or the appropriate error code. The status values are fully described in 7.1.13.1.3.

7.1.13.2.3 Appropriate usage

On receipt of the MLME-SET.confirm primitive, the next higher layer is notified of the result of its request to set the value of a PIB attribute. If the requested value was written to the indicated PIB attribute, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.14 Primitives for updating the superframe configuration

MLME-SAP start primitives define how an FFD can request to start using a new superframe configuration in order to initiate a PAN, begin transmitting beacons on an already existing PAN, thus facilitating device discovery, or to stop transmitting beacons.

These start primitives are optional for an RFD.

7.1.14.1 MLME-START.request

The MLME-START.request primitive allows the PAN coordinator to initiate a new PAN or to begin using a new superframe configuration. This primitive may also be used by a device already associated with an existing PAN to begin using a new superframe configuration.

7.1.14.1.1 Semantics of the service primitive

The semantics of the MLME-START.request primitive are as follows:

```
MLME-START.request          (
                            PANId,
                            LogicalChannel,
                            ChannelPage,
                            StartTime,
                            BeaconOrder,
                            SuperframeOrder,
                            PANCoordinator,
                            BatteryLifeExtension,
                            CoordRealignment,
                            CoordRealignSecurityLevel,
                            CoordRealignKeyIdMode,
                            CoordRealignKeySource,
                            CoordRealignKeyIndex,
                            BeaconSecurityLevel,
                            BeaconKeyIdMode,
                            BeaconKeySource,
                            BeaconKeyIndex
                            )
```

Table 72 specifies the parameters for the MLME-START.request primitive.

Table 72—MLME-START.request parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000–0xffff	The PAN identifier to be used by the device.
LogicalChannel	Integer	Selected from the available logical channels specified by the ChannelPage parameter	The logical channel on which to start using the new superframe configuration.
ChannelPage	Integer	Selected from the available channel pages supported by the PHY (see 6.1.2)	The channel page on which to begin using the new superframe configuration.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 72—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
StartTime	Integer	0x000000–0xffffff	The time at which to begin transmitting beacons. If this parameter is equal to 0x000000, beacon transmissions will begin immediately. Otherwise, the specified time is relative to the received beacon of the coordinator with which the device synchronizes. This parameter is ignored if either the BeaconOrder parameter has a value of 15 or the PANCoordinator parameter is TRUE. The time is specified in symbols and is rounded to a backoff slot boundary. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
BeaconOrder	Integer	0–15	How often the beacon is to be transmitted. A value of 15 indicates that the coordinator will not transmit periodic beacons. See 7.5.1.1 for an explanation of the relationship between the beacon order and the beacon interval.
SuperframeOrder	Integer	0–*BO* or 15	The length of the active portion of the superframe, including the beacon frame. If the BeaconOrder parameter (*BO*) has a value of 15, this parameter is ignored. See 7.5.1.1 for an explanation of the relationship between the superframe order and the superframe duration.
PANCoordinator	Boolean	TRUE or FALSE	If this value is TRUE, the device will become the PAN coordinator of a new PAN. If this value is FALSE, the device will begin using a new superframe configuration on the PAN with which it is associated.
BatteryLifeExtension	Boolean	TRUE or FALSE	If this value is TRUE, the receiver of the beaconing device is disabled *macBattLifeExtPeriods* full backoff periods after the interframe spacing (IFS) period following the beacon frame. If this value is FALSE, the receiver of the beaconing device remains enabled for the entire CAP. This parameter is ignored if the BeaconOrder parameter has a value of 15.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 72—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
CoordRealignment	Boolean	TRUE or FALSE	TRUE if a coordinator realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.
CoordRealignSecurity-Level	Integer	0x00–0x07	The security level to be used for coordinator realignment command frames (see Table 95 in 7.6.2.2.1).
CoordRealignKeyId-Mode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the CoordRealignSecurityLevel parameter is set to 0x00.
CoordRealignKey-Source	Set of 0, 4, or 8 octets	As specified by the CoordRealignKeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the CoordRealignKeyId-Mode parameter is ignored or set to 0x00.
CoordRealignKeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the CoordRealignKeyIdMode parameter is ignored or set to 0x00.
BeaconSecurityLevel	Integer	0x00–0x07	The security level to be used for beacon frames (see Table 95 in 7.6.2.2.1).
BeaconKeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.
BeaconKeySource	Set of 0, 4, or 8 octets	As specified by the BeaconKeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.
BeaconKeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.1.14.1.2 Appropriate usage

The MLME-START.request primitive is generated by the next higher layer and issued to its MLME to request that a device start using a new superframe configuration.

7.1.14.1.3 Effect on receipt

If the MLME-START.request primitive is received when *macShortAddress* is set to 0xffff, the MLME will issue the MLME-START.confirm primitive with a status of NO_SHORT_ADDRESS.

When the CoordRealignment parameter is set to TRUE, the coordinator attempts to transmit a coordinator realignment command frame as described in 7.5.2.3.2. If the transmission of the coordinator realignment

© IEEE 2010 – All rights reserved

command fails due to a channel access failure, the MLME will not make any changes to the superframe configuration (i.e., no PIB attributes will be changed) and will issue an MLME-START.confirm with a status of CHANNEL_ACCESS_FAILURE. If the coordinator realignment command is successfully transmitted, the MLME updates the appropriate PIB parameters with the values of the BeaconOrder, SuperframeOrder, PANId, ChannelPage, and LogicalChannel parameters, as described in 7.5.2.3.4, and will issue an MLME-START.confirm with a status of SUCCESS.

When the CoordRealignment parameter is set to FALSE, the MLME updates the appropriate PIB parameters with the values of the BeaconOrder, SuperframeOrder, PANId, ChannelPage, and LogicalChannel parameters, as described in 7.5.2.3.4.

The address used by the coordinator in its beacon frames is determined by the current value of *macShortAddress*, which is set by the next higher layer before issuing this primitive. If the BeaconOrder parameter is less than 15, the MLME sets *macBattLifeExt* to the value of the BatteryLifeExtension parameter. If the BeaconOrder parameter equals 15, the value of the BatteryLifeExtension parameter is ignored.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame, as described in 7.5.8.2.1. If the CoordRealignment parameter is set to TRUE, the CoordRealignSecurityLevel, CoordRealignKeyIdMode, CoordRealignKeySource, and CoordRealignKeyIndex parameters will be used to process the MAC command frame. If the BeaconOrder parameter indicates a beacon-enabled network, the BeaconSecurityLevel, BeaconKeyIdMode, BeaconKeySource, and BeaconKeyIndex parameters will be used to process the beacon frame. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-START.confirm primitive with the error status returned by outgoing frame processing.

If the length of the beacon frame exceeds *aMaxPHYPacketSize* (e.g., due to the additional overhead required for security processing), the MAC sublayer shall discard the beacon frame and issue the MLME-START.confirm primitive with a status of FRAME_TOO_LONG.

The MLME shall ignore the StartTime parameter if the BeaconOrder parameter is equal to 15 because this indicates a nonbeacon-enabled PAN. If the BeaconOrder parameter is less than 15, the MLME examines the StartTime parameter to determine the time to begin transmitting beacons; the time is defined in symbols and is rounded to a backoff slot boundary. If the PAN coordinator parameter is set to TRUE, the MLME ignores the StartTime parameter and begins beacon transmissions immediately. Setting the StartTime parameter to 0x000000 also causes the MLME to begin beacon transmissions immediately. If the PANCoordinator parameter is set to FALSE and the StartTime parameter is nonzero, the MLME calculates the beacon transmission time by adding StartTime symbols to the time, obtained from the local clock, when the MLME receives the beacon of the coordinator through which it is associated. If the time calculated causes the outgoing superframe to overlap the incoming superframe, the MLME shall not begin beacon transmissions. In this case, the MLME issues the MLME-START.confirm primitive with a status of SUPERFRAME_OVERLAP. Otherwise, the MLME then begins beacon transmissions when the current time, obtained from the local clock, equals the number of calculated symbols.

If the StartTime parameter is nonzero and the MLME is not currently tracking the beacon of the coordinator through which it is associated, the MLME will issue the MLME-START.confirm primitive with a status of TRACKING_OFF.

On completion of this procedure, the MLME responds with the MLME-START.confirm primitive. If the attempt to start using a new superframe configuration was successful, the status parameter will be set to SUCCESS. If any parameter is not supported or is out of range, the status parameter will be set to INVALID_PARAMETER.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.14.2 MLME-START.confirm

The MLME-START.confirm primitive reports the results of the attempt to start using a new superframe configuration.

7.1.14.2.1 Semantics of the service primitive

The semantics of the MLME-START.confirm primitive are as follows:

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

Table 73 specifies the parameters for the MLME-START.confirm primitive.

Table 73—MLME-START.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_SHORT_ADDRESS, SUPERFRAME_OVERLAP, TRACKING_OFF, INVALID_PARAMETER, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or CHANNEL_ACCESS_FAILURE	The result of the attempt to start using an updated superframe configuration.

7.1.14.2.2 When generated

The MLME-START.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-START.request primitive. The MLME-START.confirm primitive returns a status of either SUCCESS, indicating that the MAC sublayer has started using the new superframe configuration, or the appropriate error code. The status values are fully described in 7.1.14.1.3 and subclauses referenced by 7.1.14.1.3.

7.1.14.2.3 Appropriate usage

On receipt of the MLME-START.confirm primitive, the next higher layer is notified of the result of its request to start using a new superframe configuration. If the MAC sublayer has been successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

7.1.14.3 Message sequence chart for updating the superframe configuration

Figure 38 illustrates the sequence of messages necessary for initiating beacon transmissions in an FFD. Figure 78 (see 7.7) illustrates the sequence of messages necessary for the PAN coordinator to start beaconing on a new PAN; this figure includes steps taken by the PHY.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

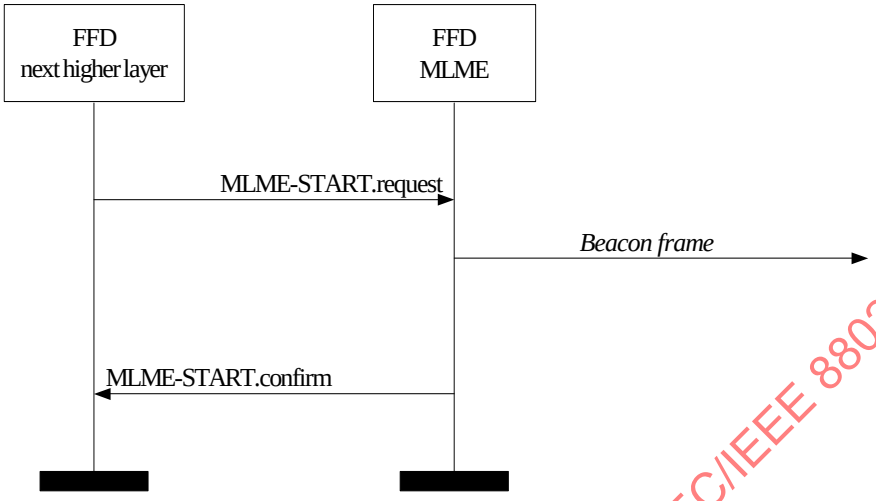


Figure 38—Message sequence chart for updating the superframe configuration

7.1.15 Primitives for synchronizing with a coordinator

MLME-SAP synchronization primitives define how synchronization with a coordinator may be achieved and how a loss of synchronization is communicated to the next higher layer.

All devices shall provide an interface for the indication primitive. The request primitive is optional.

7.1.15.1 MLME-SYNC.request

The MLME-SYNC.request primitive requests to synchronize with the coordinator by acquiring and, if specified, tracking its beacons.

7.1.15.1.1 Semantics of the service primitive

The semantics of the MLME-SYNC.request primitive are as follows:

```
MLME-SYNC.request       (
                        LogicalChannel,
                        ChannelPage,
                        TrackBeacon
                        )
```

Table 74 specifies the parameters for the MLME-SYNC.request primitive.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 74—MLME-SYNC.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The logical channel on which to attempt coordinator synchronization.
ChannelPage	Integer	Selected from the available logical channels supported by the PHY (see 6.1.2)	The channel page on which to attempt coordinator synchronization.
TrackBeacon	Boolean	TRUE or FALSE	TRUE if the MLME is to synchronize with the next beacon and attempt to track all future beacons. FALSE if the MLME is to synchronize with only the next beacon.

7.1.15.1.2 Appropriate usage

The MLME-SYNC.request primitive is generated by the next higher layer of a device on a beacon-enabled PAN and issued to its MLME to synchronize with the coordinator.

7.1.15.1.3 Effect on receipt

If the MLME-SYNC.request primitive is received by the MLME on a beacon-enabled PAN, it will first set *phyCurrentPage* and *phyCurrentChannel* equal to the values of the ChannelPage and LogicalChannel parameters, respectively; both attributes are updated by issuing the PLME-SET.request primitive. If the TrackBeacon parameter is equal to TRUE, the MLME will track the beacon, i.e., enable its receiver just before the expected time of each beacon so that the beacon frame can be processed. If the TrackBeacon parameter is equal to FALSE, the MLME will locate the beacon, but not continue to track it.

If this primitive is received by the MLME while it is currently tracking the beacon, the MLME will not discard the primitive, but rather treat it as a new synchronization request.

If the beacon could not be located either on its initial search or during tracking, the MLME will issue the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

7.1.15.2 MLME-SYNC-LOSS.indication

The MLME-SYNC-LOSS.indication primitive indicates the loss of synchronization with a coordinator.

7.1.15.2.1 Semantics of the service primitive

The semantics of the MLME-SYNC-LOSS.indication primitive are as follows:

```
MLME-SYNC-LOSS.indication       (
                                LossReason,
                                PANId,
                                LogicalChannel,
                                ChannelPage,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex
                                )
```

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 75 specifies the parameters for the MLME-SYNC-LOSS.indication primitive.

Table 75—MLME-SYNC-LOSS.indication parameters

Name	Type	Valid range	Description
LossReason	Enumeration	PAN_ID_CONFLICT, REALIGNMENT, or BEACON_LOST	The reason that synchronization was lost.
PANId	Integer	0x0000–0xffff	The PAN identifier with which the device lost synchronization or to which it was realigned.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see 6.1.2).	The logical channel on which the device lost synchronization or to which it was realigned.
ChannelPage	Integer	Selected from the available channel pages supported by the PHY (see 6.1.2).	The channel page on which the device lost synchronization or to which it was realigned.
SecurityLevel	Integer	0x00-0x07	If the primitive was either generated by the device itself following loss of synchronization or generated by the PAN coordinator upon detection of a PAN ID conflict, the security level is set to 0x00. If the primitive was generated following the reception of either a coordinator realignment command or a PAN ID conflict notification command: The security level purportedly used by the received MAC frame (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was either generated by the device itself following loss of synchronization or generated by the PAN coordinator upon detection of a PAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a PAN ID conflict notification command: The mode used to identify the key purportedly used by the originator of the received frame (see Table 96 in 7.6.2.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 75—MLME-SYNC-LOSS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was either generated by the device itself following loss of synchronization or generated by the PAN coordinator upon detection of a PAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a PAN ID conflict notification command: The originator of the key purportedly used by the originator of the received frame (see 7.6.2.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	If the primitive was either generated by the device itself following loss of synchronization or generated by the PAN coordinator upon detection of a PAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a PAN ID conflict notification command: The index of the key purportedly used by the originator of the received frame (see 7.6.2.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

7.1.15.2.2 When generated

The MLME-SYNC-LOSS.indication primitive is generated by the MLME of a device and issued to its next higher layer in the event of a loss of synchronization with the coordinator. It is also generated by the MLME of the PAN coordinator and issued to its next higher layer in the event of a PAN ID conflict.

If a device that is associated through the PAN coordinator has detected a PAN identifier conflict and communicated it to the PAN coordinator, the MLME will issue this primitive with the LossReason parameter set to PAN_ID_CONFLICT. Similarly, if the PAN coordinator receives a PAN ID conflict notification command (see 7.3.5), the MLME will issue this primitive with the LossReason parameter set to PAN_ID_CONFLICT.

If a device has received the coordinator realignment command (see 7.3.8) from the coordinator through which it is associated and the MLME was not carrying out an orphan scan, the MLME will issue this primitive with the LossReason parameter set to REALIGNMENT and the PANId, LogicalChannel, ChannelPage, and security-related parameters set as described in 7.5.2.3.3.

If a device has not heard the beacon for *aMaxLostBeacons* consecutive superframes following an MLME-SYNC.request primitive, either initially or during tracking, the MLME will issue this primitive with the LossReason parameter set to BEACON_LOST. The PANId, LogicalChannel and ChannelPage parameters

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

shall be set according to the coordinator with which synchronization was lost. The SecurityLevel parameter shall be set to zero and the KeyIdMode, KeySource, and KeyIndex parameters shall be ignored. If the beacon was being tracked, the MLME will not attempt to track the beacon any further.

7.1.15.2.3 Appropriate usage

On receipt of the MLME-SYNC-LOSS.indication primitive, the next higher layer is notified of a loss of synchronization.

7.1.15.3 Message sequence chart for synchronizing with a coordinator

Figure 39 illustrates the sequence of messages necessary for a device to synchronize with a coordinator. In Figure 39a, a single synchronization request is issued. The MLME then searches for a beacon and, if found, determines whether the coordinator has any data pending for the device. If so, the data are requested as described in 7.5.6.3. In Figure 39b, a track synchronization request is issued. The MLME then searches for a beacon and, if found, attempts to keep track of it using a timer that expires just before the expected time of the next beacon.

For both examples Figure 39a and Figure 39b, the received beacon frames do not contain payload, and *macAutoRequest* is set to TRUE. The MLME also checks for any data pending in the coordinator for the device when a beacon frame is received.

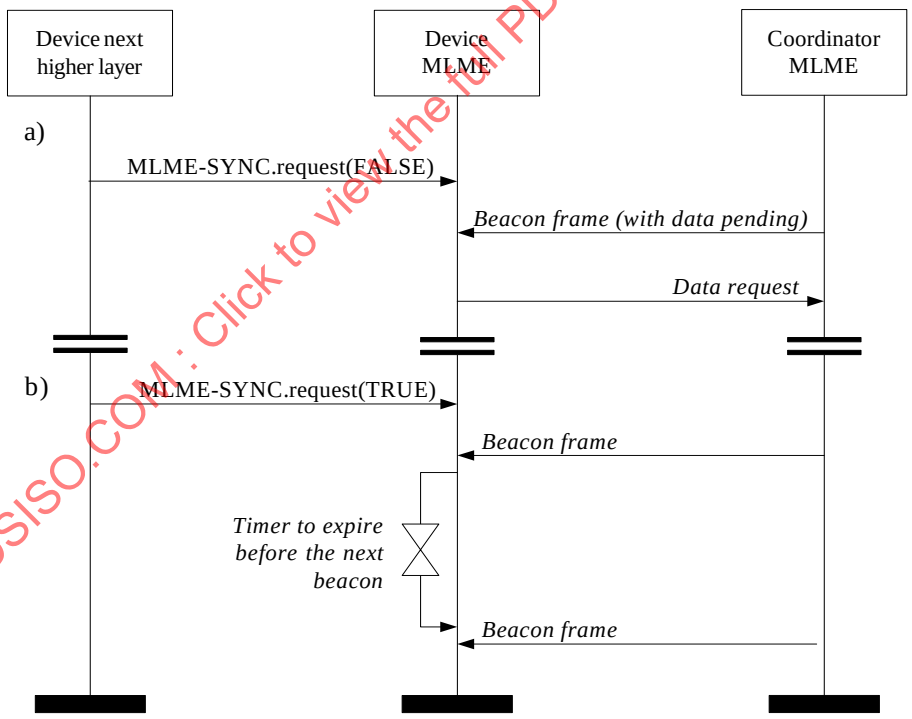


Figure 39—Message sequence chart for synchronizing to a coordinator in a beacon-enabled PAN

7.1.16 Primitives for requesting data from a coordinator

MLME-SAP polling primitives define how to request data from a coordinator.

All devices shall provide an interface for these polling primitives.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.16.1 MLME-POLL.request

The MLME-POLL.request primitive prompts the device to request data from the coordinator.

7.1.16.1.1 Semantics of the service primitive

The semantics of the MLME-POLL.request primitive are as follows:

```
MLME-POLL.request       (
                        CoordAddrMode,
                        CoordPANId,
                        CoordAddress,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 76 specifies the parameter for the MLME-POLL.request primitive.

Table 76—MLME-POLL.request parameters

Name	Type	Valid range	Description
CoordAddrMode	Integer	0x02–0x03	The addressing mode of the coordinator to which the poll is intended. This parameter can take one of the following values: 2 = 16-bit short address, 3 = 64-bit extended address.
CoordPANId	Integer	0x0000–0xfffe	The PAN identifier of the coordinator to which the poll is intended.
CoordAddress	Device-Address	As specified by the CoordAddrMode parameter	The address of the coordinator to which the poll is intended.
SecurityLevel	Integer	0x00–0x07	The security level to be used (see Table 95 in 7.6.2.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (see Table 96 in 7.6.2.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.6.2.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.6.2.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.1.16.1.2 Appropriate usage

The MLME-POLL.request primitive is generated by the next higher layer and issued to its MLME when data are to be requested from a coordinator.

7.1.16.1.3 Effect on receipt

On receipt of the MLME-POLL.request primitive, the MLME generates and sends a data request command (see 7.3.4). If the poll is directed to the PAN coordinator, the data request command may be generated without any destination address information present. Otherwise, the data request command is always generated with the destination address information in the CoordPANId and CoordAddress parameters.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame based on the CoordAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.5.8.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-POLL.confirm primitive with the error status returned by outgoing frame processing.

If the data request command cannot be sent due to a CSMA-CA algorithm failure, the MLME will issue the MLME-POLL.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a data request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_ACK (see 7.5.6.4).

If an acknowledgment is received, the MLME will request that the PHY enable its receiver if the Frame Pending subfield of the acknowledgment frame is set to one. If the Frame Pending subfield of the acknowledgment frame is set to zero, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

If a frame is received from the coordinator with a zero length payload or if the frame is a MAC command frame, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA. If a frame is received from the coordinator with nonzero length payload, the MLME will issue the MLME-POLL.confirm primitive with a status of SUCCESS. In this case, the actual data are indicated to the next higher layer using the MCPS-DATA.indication primitive (see 7.1.1.3).

If a frame is not received within *macMaxFrameTotalWaitTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, even though the acknowledgment to the data request command has its Frame Pending subfield set to one, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

If any parameter in the MLME-POLL.request primitive is not supported or is out of range, the MLME will issue the MLME-POLL.confirm primitive with a status of INVALID_PARAMETER.

7.1.16.2 MLME-POLL.confirm

The MLME-POLL.confirm primitive reports the results of a request to poll the coordinator for data.

7.1.16.2.1 Semantics of the service primitive

The semantics of the MLME-POLL.confirm primitive are as follows:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

```
MLME-POLL.confirm       (
                        status
                        )
```

Table 77 specifies the parameters for the MLME-POLL.confirm primitive.

Table 77—MLME-POLL.confirm parameters

Name	Type	Valid range	Description
status	Integer	SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the data request.

7.1.16.2.2 When generated

The MLME-POLL.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-POLL.request primitive. If the request was successful, the status parameter will be equal to SUCCESS, indicating a successful poll for data. Otherwise, the status parameter indicates the appropriate error code. The status values are fully described in 7.1.16.1.3 and the subclauses referenced by 7.1.16.1.3.

7.1.16.2.3 Appropriate usage

On receipt of the MLME-POLL.confirm primitive, the next higher layer is notified of the status of the procedure to request data from the coordinator.

7.1.16.3 Message sequence chart for requesting data from a coordinator

Figure 40 illustrates the sequence of messages necessary, including the layer behavior of the device and the over-the-air interface, for a device to request data from a coordinator.

In both scenarios Figure 40a and Figure 40b, a poll request is issued to the MLME, which then sends a data request command to the coordinator. In Figure 40a, the corresponding acknowledgment has the Frame Pending (FP) subfield set to zero and the MLME issues the poll request confirmation immediately. In Figure 40b, the corresponding acknowledgment has the Frame Pending subfield set to one and the MLME enables the receiver in anticipation of the data frame from the coordinator. On receipt of this data frame, the MLME issues a poll request confirmation followed by a data indication containing the data of the received frame.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

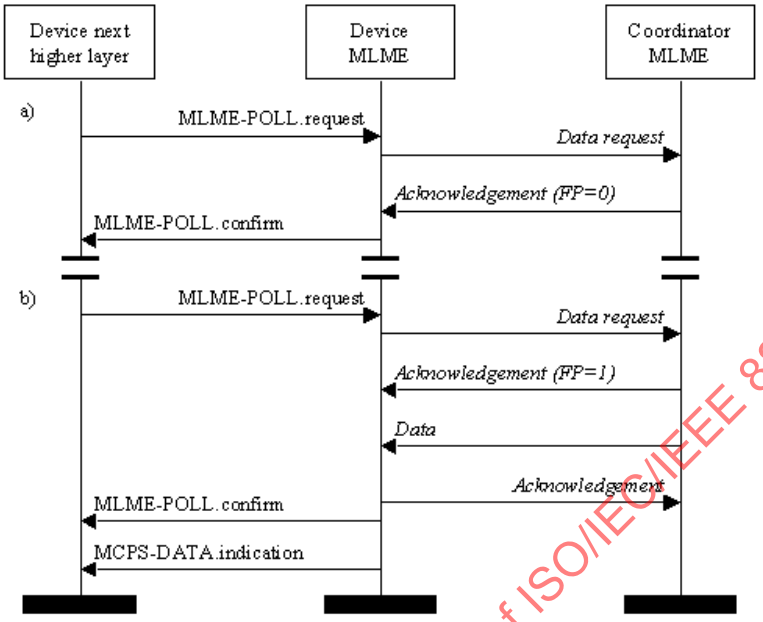


Figure 40—Message sequence chart for requesting data from the coordinator

7.1.17 MAC enumeration description

This subclause explains the meaning of the enumerations used in the primitives defined in the MAC sublayer specification. Table 78 shows a description of the MAC enumeration values.

Table 78—MAC enumerations description

Enumeration	Value	Description
SUCCESS	0x00	The requested operation was completed successfully. For a transmission request, this value indicates a successful transmission.
—	0x01–0xda	Reserved for MAC command status and reason code values.
—	0x80–0xda, 0xfe–0xff	Reserved.
BEACON_LOSS	0xe0	The beacon was lost following a synchronization request.
CHANNEL_ACCESS_FAILURE	0xe1	A transmission could not take place due to activity on the channel, i.e., the CSMA-CA mechanism has failed.
COUNTER_ERROR	0xdb	The frame counter purportedly applied by the originator of the received frame is invalid.
DENIED	0xe2	The GTS request has been denied by the PAN coordinator.
DISABLE_TRX_FAILURE	0xe3	The attempt to disable the transceiver has failed.
FRAME_TOO_LONG	0xe5	Either a frame resulting from processing has a length that is greater than *aMaxPHYPacketSize* or a requested transaction is too large to fit in the CAP or GTS.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 78—MAC enumerations description *(continued)*

Enumeration	Value	Description
IMPROPER_KEY_TYPE	0xdc	The key purportedly applied by the originator of the received frame is not allowed to be used with that frame type according to the key usage policy of the recipient.
IMPROPER_SECURITY_LEVEL	0xdd	The security level purportedly applied by the originator of the received frame does not meet the minimum security level required/expected by the recipient for that frame type.
INVALID_ADDRESS	0xf5	A request to send data was unsuccessful because neither the source address parameters nor the destination address parameters were present.
INVALID_GTS	0xe6	The requested GTS transmission failed because the specified GTS either did not have a transmit GTS direction or was not defined.
INVALID_HANDLE	0xe7	A request to purge an MSDU from the transaction queue was made using an MSDU handle that was not found in the transaction table.
INVALID_INDEX	0xf9	An attempt to write to a MAC PIB attribute that is in a table failed because the specified table index was out of range.
INVALID_PARAMETER	0xe8	A parameter in the primitive is either not supported or is out of the valid range.
LIMIT_REACHED	0xfa	A scan operation terminated prematurely because the number of PAN descriptors stored reached an implementation-specified maximum.
NO_ACK	0xe9	No acknowledgment was received after *macMaxFrameRetries*.
NO_BEACON	0xea	A scan operation failed to find any network beacons.
NO_DATA	0xeb	No response data were available following a request.
NO_SHORT_ADDRESS	0xec	The operation failed because a 16-bit short address was not allocated.
ON_TIME_TOO_LONG	0xf6	A receiver enable request was unsuccessful because it specified a number of symbols that was longer than the beacon interval.
OUT_OF_CAP	0xed	A receiver enable request was unsuccessful because it could not be completed within the CAP. The enumeration description is not used in this standard, and it is included only to meet the backwards compatibility requirements for IEEE Std 802.15.4-2003.
PAN_ID_CONFLICT	0xee	A PAN identifier conflict has been detected and communicated to the PAN coordinator.
PAST_TIME	0xf7	A receiver enable request was unsuccessful because it could not be completed within the current superframe and was not permitted to be deferred until the next superframe.
READ_ONLY	0xfb	A SET/GET request was issued with the identifier of an attribute that is read only.
REALIGNMENT	0xef	A coordinator realignment command has been received.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 78—MAC enumerations description *(continued)*

Enumeration	Value	Description
SCAN_IN_PROGRESS	0xfc	A request to perform a scan operation failed because the MLME was in the process of performing a previously initiated scan operation.
SECURITY_ERROR	0xe4	Cryptographic processing of the received secured frame failed.
SUPERFRAME_OVERLAP	0xfd	The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the instructed start time overlapped the transmission time of the beacon of its coordinator.
TRACKING_OFF	0xf8	The device was instructed to start sending beacons based on the timing of the beacon transmissions of its coordinator, but the device is not currently tracking the beacon of its coordinator.
TRANSACTION_EXPIRED	0xf0	The transaction has expired and its information was discarded.
TRANSACTION_OVERFLOW	0xf1	There is no capacity to store the transaction.
TX_ACTIVE	0xf2	The transceiver was in the transmitter enabled state when the receiver was requested to be enabled. The enumeration description is not used in this standard, and it is included only to meet the backwards compatibility requirements for IEEE Std 802.15.4-2003.
UNAVAILABLE_KEY	0xf3	The key purportedly used by the originator of the received frame is not available or, if available, the originating device is not known or is blacklisted with that particular key.
UNSUPPORTED_ATTRIBUTE	0xf4	A SET/GET request was issued with the identifier of a PIB attribute that is not supported.
UNSUPPORTED_LEGACY	0xde	The received frame was purportedly secured using security based on IEEE Std 802.15.4-2003, and such security is not supported by this standard.
UNSUPPORTED_SECURITY	0xdf	The security purportedly applied by the originator of the received frame is not supported.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.2 MAC frame formats

This subclause specifies the format of the MAC frame (MPDU). Each MAC frame consists of the following basic components:

a) A MHR, which comprises frame control, sequence number, address information, and security-related information.

b) A MAC payload, of variable length, which contains information specific to the frame type. Acknowledgment frames do not contain a payload.

c) A MFR, which contains a FCS.

The frames in the MAC sublayer are described as a sequence of fields in a specific order. All frame formats in this subclause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from 0 (leftmost and least significant) to $k - 1$ (rightmost and most significant), where the length of the field is k bits. Fields that are

© IEEE 2010 – All rights reserved

longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

For every MAC frame, all reserved bits shall be set to zero upon transmission and shall be ignored upon receipt.

7.2.1 General MAC frame format

The MAC frame format is composed of a MHR, a MAC payload, and a MFR. The fields of the MHR appear in a fixed order; however, the addressing fields may not be included in all frames. The general MAC frame shall be formatted as illustrated in Figure 41.

<table>
<tr><th>Octets: 2</th><th>1</th><th>0/2</th><th>0/2/8</th><th>0/2</th><th>0/2/8</th><th>0/5/6/10/14</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td rowspan="2">Frame Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="5">Addressing fields</td></tr>
<tr><td colspan="7">MHR</td><td>MAC Payload</td><td>MFR</td></tr>
</table>

Figure 41—General MAC frame format

7.2.1.1 Frame Control field

The Frame Control field is 2 octets in length and contains information defining the frame type, addressing fields, and other control flags. The Frame Control field shall be formatted as illustrated in Figure 42.

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame Type	Security Enabled	Frame Pending	Ack. Request	PAN ID Compression	Reserved	Dest. Addressing Mode	Frame Version	Source Addressing Mode

Figure 42—Format of the Frame Control field

7.2.1.1.1 Frame Type subfield

The Frame Type subfield is 3 bits in length and shall be set to one of the nonreserved values listed in Table 79.

7.2.1.1.2 Security Enabled subfield

The Security Enabled subfield is 1 bit in length, and it shall be set to one if the frame is protected by the MAC sublayer and shall be set to zero otherwise. The Auxiliary Security Header field of the MHR shall be present only if the Security Enabled subfield is set to one.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 79—Values of the Frame Type subfield

Frame type value $b_2\ b_1\ b_0$	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100–111	Reserved

7.2.1.1.3 Frame Pending subfield

The Frame Pending subfield is 1 bit in length and shall be set to one if the device sending the frame has more data for the recipient. This subfield shall be set to zero otherwise (see 7.5.6.3).

The Frame Pending subfield shall be used only in beacon frames or frames transmitted either during the CAP by devices operating on a beacon-enabled PAN or at any time by devices operating on a nonbeacon-enabled PAN.

At all other times, it shall be set to zero on transmission and ignored on reception.

7.2.1.1.4 Acknowledgment Request subfield

The Acknowledgment Request subfield is 1 bit in length and specifies whether an acknowledgment is required from the recipient device on receipt of a data or MAC command frame. If this subfield is set to one, the recipient device shall send an acknowledgment frame only if, upon reception, the frame passes the third level of filtering (see 7.5.6.2). If this subfield is set to zero, the recipient device shall not send an acknowledgment frame.

7.2.1.1.5 PAN ID Compression subfield

The PAN ID Compression subfield is 1 bit in length and specifies whether the MAC frame is to be sent containing only one of the PAN identifier fields when both source and destination addresses are present. If this subfield is set to one and both the source and destination addresses are present, the frame shall contain only the Destination PAN Identifier field, and the Source PAN Identifier field shall be assumed equal to that of the destination. If this subfield is set to zero and both the source and destination addresses are present, the frame shall contain both the Source PAN Identifier and Destination PAN Identifier fields. If only one of the addresses is present, this subfield shall be set to zero, and the frame shall contain the PAN identifier field corresponding to the address. If neither address is present, this subfield shall be set to zero, and the frame shall not contain either PAN identifier field.

7.2.1.1.6 Destination Addressing Mode subfield

The Destination Addressing Mode subfield is 2 bits in length and shall be set to one of the nonreserved values listed in Table 80.

If this subfield is equal to zero and the Frame Type subfield does not specify that this frame is an acknowledgment or beacon frame, the Source Addressing Mode subfield shall be nonzero, implying that the frame is directed to the PAN coordinator with the PAN identifier as specified in the Source PAN Identifier field.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 80—Possible values of the Destination Addressing Mode and Source Addressing Mode subfields

Addressing mode value b_1 b_0	Description
00	PAN identifier and address fields are not present.
01	Reserved.
10	Address field contains a 16-bit short address.
11	Address field contains a 64-bit extended address.

7.2.1.1.7 Frame Version subfield

The Frame Version subfield is 2 bits in length and specifies the version number corresponding to the frame.

This subfield shall be set to 0x00 to indicate a frame compatible with IEEE Std 802.15.4-2003 and 0x01 to indicate an IEEE 802.15.4 frame. All other subfield values shall be reserved for future use. See 7.2.3 for details on frame compatibility.

7.2.1.1.8 Source Addressing Mode subfield

The Source Addressing Mode subfield is 2 bits in length and shall be set to one of the nonreserved values listed in Table 80.

If this subfield is equal to zero and the Frame Type subfield does not specify that this frame is an acknowledgment frame, the Destination Addressing Mode subfield shall be nonzero, implying that the frame has originated from the PAN coordinator with the PAN identifier as specified in the Destination PAN Identifier field.

7.2.1.2 Sequence Number field

The Sequence Number field is 1 octet in length and specifies the sequence identifier for the frame.

For a beacon frame, the Sequence Number field shall specify a BSN. For a data, acknowledgment, or MAC command frame, the Sequence Number field shall specify a DSN that is used to match an acknowledgment frame to the data or MAC command frame.

7.2.1.3 Destination PAN Identifier field

The Destination PAN Identifier field, when present, is 2 octets in length and specifies the unique PAN identifier of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast PAN identifier, which shall be accepted as a valid PAN identifier by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode subfield of the Frame Control field is nonzero.

7.2.1.4 Destination Address field

The Destination Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Destination Addressing Mode subfield of the Frame Control field (see 7.2.1.1.6), and

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

specifies the address of the intended recipient of the frame. A 16-bit value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid 16-bit short address by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode subfield of the Frame Control field is nonzero.

7.2.1.5 Source PAN Identifier field

The Source PAN Identifier field, when present, is 2 octets in length and specifies the unique PAN identifier of the originator of the frame. This field shall be included in the MAC frame only if the Source Addressing Mode and PAN ID Compression subfields of the Frame Control field are nonzero and equal to zero, respectively.

The PAN identifier of a device is initially determined during association on a PAN, but may change following a PAN identifier conflict resolution (see 7.5.2.2).

7.2.1.6 Source Address field

The Source Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Source Addressing Mode subfield of the Frame Control field (see 7.2.1.1.8), and specifies the address of the originator of the frame. This field shall be included in the MAC frame only if the Source Addressing Mode subfield of the Frame Control field is nonzero.

7.2.1.7 Auxiliary Security Header field

The Auxiliary Security Header field has a variable length and specifies information required for security processing, including how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (see 7.6.1). This field shall be present only if the Security Enabled subfield is set to one. For details on formatting, see 7.6.2.

7.2.1.8 Frame Payload field

The Frame Payload field has a variable length and contains information specific to individual frame types. If the Security Enabled subfield is set to one in the Frame Control field, the frame payload is protected as defined by the security suite selected for that frame.

7.2.1.9 FCS field

The FCS field is 2 octets in length and contains a 16-bit ITU-T CRC. The FCS is calculated over the MHR and MAC payload parts of the frame.

The FCS shall be calculated using the following standard generator polynomial of degree 16:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1 \tag{12}$$

The FCS shall be calculated for transmission using the following algorithm:

— Let $M(x) = b_0 x^{k-1} + b_1 x^{k-2} + \ldots + b_{k-2} x + b_{k-1}$ be the polynomial representing the sequence of bits for which the checksum is to be computed.

— Multiply $M(x)$ by x^{16}, giving the polynomial $x^{16} \times M(x)$.

— Divide $x^{16} \times M(x)$ modulo 2 by the generator polynomial, $G_{16}(x)$, to obtain the remainder polynomial, $R(x) = r_0 x^{15} + r_1 x^{14} + \ldots + r_{14} x + r_{15}$.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

— The FCS field is given by the coefficients of the remainder polynomial, *R(x)*.

Here, binary polynomials are represented as bit strings, in highest polynomial degree first order.

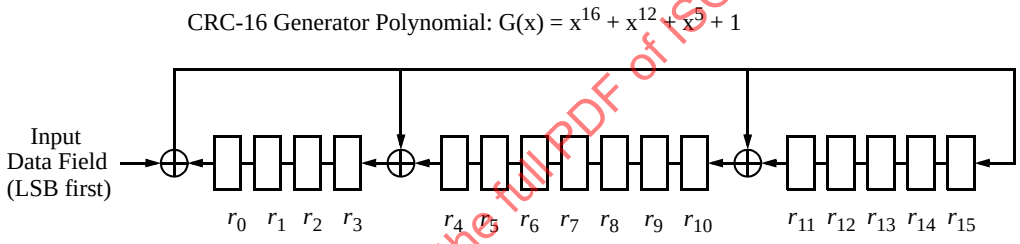
As an example, consider an acknowledgment frame with no payload and the following 3 byte MHR:

0100 0000 0000 0000 0101 0110 [leftmost bit (b_0) transmitted first in time]
b_0.....b_{23}

The FCS for this case would be the following:

0010 0111 1001 1110 [leftmost bit (r_0) transmitted first in time]
r_0.....r_{15}

A typical implementation is depicted in Figure 43.



1. Initialize the remainder register (r_0 through r_{15}) to zero.
2. Shift MHR and payload into the divider in the order of transmission (LSB first).
3. After the last bit of the data field is shifted into the divider, the remainder register contains the FCS.
4. The FCS is appended to the data field so that r_0 is transmitted first.

Figure 43—Typical FCS implementation

7.2.2 Format of individual frame types

Four frame types are defined: beacon, data, acknowledgment, and MAC command. These frame types are discussed in 7.2.2.1 through 7.2.2.4.

7.2.2.1 Beacon frame format

The beacon frame shall be formatted as illustrated in Figure 44.

The GTS fields shall be formatted as illustrated in Figure 45, and the pending address fields shall be formatted as illustrated in Figure 46.

The order of the fields of the beacon frame shall conform to the order of the general MAC frame as illustrated in Figure 41.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Octets: 2	1	4/10	0/5/6/10/14	2	variable	variable	variable	2
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Superframe Specification	GTS fields (Figure 45)	Pending address fields (Figure 46)	Beacon Payload	FCS
MHR				MAC Payload				MFR

Figure 44—Beacon frame format

Octets: 1	0/1	variable
GTS Specification	GTS Directions	GTS List

Figure 45—Format of the GTS information fields

Octets: 1	variable
Pending Address Specification	Address List

Figure 46—Format of the pending address information fields

7.2.2.1.1 Beacon frame MHR fields

The MHR for a beacon frame shall contain the Frame Control field, the Sequence Number field, the Source PAN Identifier field, and the Source Address field.

In the Frame Control field, the Frame Type subfield shall contain the value that indicates a beacon frame, as shown in Table 79, and the Source Addressing Mode subfield shall be set as appropriate for the address of the coordinator transmitting the beacon frame. If protection is used for the beacon, the Security Enabled subfield shall be set to one. The Frame Version subfield shall be set to one only if the Security Enabled subfield is set to one. If a broadcast data or command frame is pending, the Frame Pending subfield shall be set to one. All other subfields shall be set to zero and ignored on reception.

The Sequence Number field shall contain the current value of *macBSN*.

The addressing fields shall comprise only the source address fields. The Source PAN Identifier and Source Address fields shall contain the PAN identifier and address, respectively, of the device transmitting the beacon.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the beacon frame, as specified in 7.2.1.7.

7.2.2.1.2 Superframe Specification field

The Superframe Specification field is 16 bits in length and shall be formatted as illustrated in Figure 47.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Bits: 0–3	4-7	8-11	12	13	14	15
Beacon Order	Superframe Order	Final CAP Slot	Battery Life Extension (BLE)	Reserved	PAN Coordinator	Association Permit

Figure 47—Format of the Superframe Specification field

The Beacon Order subfield is 4 bits in length and shall specify the transmission interval of the beacon. See 7.5.1.1 for an explanation of the relationship between the beacon order and the beacon interval.

The Superframe Order subfield is 4 bits in length and shall specify the length of time during which the superframe is active (i.e., receiver enabled), including the beacon frame transmission time. See 7.5.1.1 for an explanation of the relationship between the superframe order and the superframe duration.

The Final CAP Slot subfield is 4 bits in length and specifies the final superframe slot utilized by the CAP. The duration of the CAP, as implied by this subfield, shall be greater than or equal to the value specified by *aMinCAPLength*. However, an exception is allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3).

The Battery Life Extension (BLE) subfield is 1 bit in length and shall be set to one if frames transmitted to the beaconing device during its CAP are required to start in or before *macBattLifeExtPeriods* full backoff periods after the IFS period following the beacon. Otherwise, the BLE subfield shall be set to zero.

The PAN Coordinator subfield is 1 bit in length and shall be set to one if the beacon frame is being transmitted by the PAN coordinator. Otherwise, the PAN Coordinator subfield shall be set to zero.

The Association Permit subfield is 1 bit in length and shall be set to one if *macAssociationPermit* is set to TRUE (i.e., the coordinator is accepting association to the PAN). The association permit bit shall be set to zero if the coordinator is currently not accepting association requests on its network.

7.2.2.1.3 GTS Specification field

The GTS Specification field is 8 bits in length and shall be formatted as illustrated in Figure 48.

Bits: 0-2	3-6	7
GTS Descriptor Count	Reserved	GTS Permit

Figure 48—Format of the GTS Specification field

The GTS Descriptor Count subfield is 3 bits in length and specifies the number of 3-octet GTS descriptors contained in the GTS List field of the beacon frame. If the value of this subfield is greater than zero, the size of the CAP shall be allowed to dip below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length caused by the inclusion of the subfield. If the value of this subfield is zero, the GTS Directions field and GTS List field of the beacon frame are not present.

The GTS Permit subfield is 1 bit in length and shall be set to one if *macGTSPermit* is equal to TRUE (i.e., the PAN coordinator is accepting GTS requests). Otherwise, the GTS Permit field shall be set to zero.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.2.2.1.4 GTS Directions field

The GTS Directions field is 8 bits in length and shall be formatted as illustrated in Figure 49.

Bits: 0-6	7
GTS Directions Mask	Reserved

Figure 49—Format of the GTS Directions field

The GTS Directions Mask subfield is 7 bits in length and contains a mask identifying the directions of the GTSs in the superframe. The lowest bit in the mask corresponds to the direction of the first GTS contained in the GTS List field of the beacon frame, with the remainder appearing in the order that they appear in the list. Each bit shall be set to one if the GTS is a receive-only GTS or to zero if the GTS is a transmit-only GTS. GTS direction is defined relative to the direction of the data frame transmission by the device.

7.2.2.1.5 GTS List field

The size of the GTS List field is defined by the values specified in the GTS Specification field of the beacon frame and contains the list of GTS descriptors that represents the GTSs that are being maintained. The maximum number of GTS descriptors shall be limited to seven.

Each GTS descriptor is 24 bits in length and shall be formatted as illustrated in Figure 50.

Bits: 0-15	16-19	20-23
Device Short Address	GTS Starting Slot	GTS Length

Figure 50—Format of the GTS descriptor

The Device Short Address subfield is 16 bits in length and shall contain the short address of the device for which the GTS descriptor is intended.

The GTS Starting Slot subfield is 4 bits in length and contains the superframe slot at which the GTS is to begin.

The GTS Length subfield is 4 bits in length and contains the number of contiguous superframe slots over which the GTS is active.

7.2.2.1.6 Pending Address Specification field

The Pending Address Specification field shall be formatted as illustrated in Figure 51.

The Number of Short Addresses Pending subfield is 3 bits in length and indicates the number of 16-bit short addresses contained in the Address List field of the beacon frame.

The Number of Extended Addresses Pending subfield is 3 bits in length and indicates the number of 64-bit extended addresses contained in the Address List field of the beacon frame.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Bits: 0-2	3	4-6	7
Number of Short Addresses Pending	Reserved	Number of Extended Addresses Pending	Reserved

Figure 51—Format of the Pending Address Specification field

7.2.2.1.7 Address List field

The size of the Address List field is determined by the values specified in the Pending Address Specification field of the beacon frame and contains the list of addresses of the devices that currently have messages pending with the coordinator. The address list shall not contain the broadcast short address 0xffff.

The maximum number of addresses pending shall be limited to seven and may comprise both short and extended addresses. All pending short addresses shall appear first in the list followed by any extended addresses. If the coordinator is able to store more than seven transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses.

7.2.2.1.8 Beacon Payload field

The Beacon Payload field is an optional sequence of up to *aMaxBeaconPayloadLength* octets specified to be transmitted in the beacon frame by the next higher layer. The set of octets contained in *macBeaconPayload* shall be copied into this field.

7.2.2.2 Data frame format

The data frame shall be formatted as illustrated in Figure 52.

Octets: 2	1	(see 7.2.2.2.1)	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Data Payload	FCS
MHR				MAC Payload	MFR

Figure 52—Data frame format

The order of the fields of the data frame shall conform to the order of the general MAC frame as illustrated in Figure 41.

7.2.2.2.1 Data frame MHR fields

The MHR for a data frame shall contain the Frame Control field, the Sequence Number field, the destination PAN identifier/address fields, and/or the source PAN identifier/address fields.

In the Frame Control field, the Frame Type subfield shall contain the value that indicates a data frame, as shown in Table 79. If protection is used for the data, the Security Enabled subfield shall be set to one. The Frame Version subfield shall be set to one if either the Security Enabled subfield is set to one or the MAC Payload field is greater than *aMaxMACSafePayloadSize*. All other subfields shall be set appropriately according to the intended use of the data frame. All reserved subfields shall be set to zero and ignored on reception.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The Sequence Number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the Frame Control field.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the data frame, as specified in 7.2.1.7.

7.2.2.2.2 Data Payload field

The payload of a data frame shall contain the sequence of octets that the next higher layer has requested the MAC sublayer to transmit.

7.2.2.3 Acknowledgment frame format

The acknowledgment frame shall be formatted as illustrated in Figure 53.

Octets: 2	**1**	**2**
Frame Control	Sequence Number	FCS
MHR		MFR

Figure 53—Acknowledgment frame format

The order of the fields of the acknowledgment frame shall conform to the order of the general MAC frame as illustrated in Figure 41.

7.2.2.3.1 Acknowledgment frame MHR fields

The MHR for an acknowledgment frame shall contain only the Frame Control field and the Sequence Number field.

In the Frame Control field, the Frame Type subfield shall contain the value that indicates an acknowledgment frame, as shown in Table 79. If the acknowledgment frame is being sent in response to a received data request command, the device sending the acknowledgment frame shall determine whether it has data pending for the recipient. If the device can determine this before sending the acknowledgment frame (see 7.5.6.4.2), it shall set the Frame Pending subfield according to whether there is pending data. Otherwise, the Frame Pending subfield shall be set to one. If the acknowledgment frame is being sent in response to either a data frame or another type of MAC command frame, the device shall set the Frame Pending subfield to zero. All other subfields shall be set to zero and ignored on reception.

The Sequence Number field shall contain the value of the sequence number received in the frame for which the acknowledgment is to be sent.

7.2.2.4 MAC command frame format

The MAC command frame shall be formatted as illustrated in Figure 54.

The order of the fields of the MAC command frame shall conform to the order of the general MAC frame as illustrated in Figure 41.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Octets: 2	1	(see 7.2.2.4.1)	0/5/6/10/14	1	variable	2
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Command Frame Identifier	Command Payload	FCS
MHR				MAC Payload		MFR

Figure 54—MAC command frame format

7.2.2.4.1 MAC command frame MHR fields

The MHR for a MAC command frame shall contain the Frame Control field, the Sequence Number field, the destination PAN identifier/address fields and/or the source PAN identifier/address fields.

In the Frame Control field, the Frame Type subfield shall contain the value that indicates a MAC command frame, as shown in Table 79. If the frame is to be secured, the Security Enabled subfield of the Frame Control field shall be set to one and the frame secured according to the process described in 7.5.8.1.3. Otherwise the Security Enabled subfield of the Frame Control field shall be set to zero. All other subfields shall be set appropriately according to the intended use of the MAC command frame. All reserved subfields shall be set to zero and ignored on reception.

The Sequence Number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the Frame Control field.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the MAC command frame, as specified in 7.2.1.7.

7.2.2.4.2 Command Frame Identifier field

The Command Frame Identifier field identifies the MAC command being used. This field shall be set to one of the nonreserved values listed in Table 82.

7.2.2.4.3 Command Payload field

The Command Payload field contains the MAC command itself. The formats of the individual commands are described in 7.3.

7.2.3 Frame compatibility

All unsecured frames specified in this standard are compatible with unsecured frames compliant with IEEE Std 802.15.4-2003, with two exceptions: a coordinator realignment command frame with the Channel Page field present (see 7.3.8) and any frame with a MAC Payload field larger than *aMaxMACSafePayloadSize* octets.

Compatibility for secured frames is shown in Table 81, which identifies the security operating modes for IEEE Std 802.15.4-2003 and this standard.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 81—Frame compatibility between IEEE Std 802.15.4-2003 and this standard

Frame Control field bit assignments		Functionality
Security enabled b_3	Frame version b_{13} b_{12}	
0	00	No security. Frames are compatible between IEEE Std 802.15.4-2003 and this standard.
0	01	No security. Frames are not compatible between IEEE Std 802.15.4-2003 and this standard.
1	00	Secured frame formatted according to IEEE Std 802.15.4-2003. This type of frame is not supported in this standard.
1	01	Secured frame formatted according to this standard.

7.3 MAC command frames

The command frames defined by the MAC sublayer are listed in Table 82. An FFD shall be capable of transmitting and receiving all command frame types, with the exception of the GTS request command, while the requirements for an RFD are indicated by an "X" in the table. MAC commands shall only be transmitted in the CAP for beacon-enabled PANs or at any time for nonbeacon-enabled PANs.

How the MLME shall construct the individual commands for transmission is detailed in 7.3.1 through 7.3.9. MAC command reception shall abide by the procedure described in 7.5.6.2.

Table 82—MAC command frames

Command frame identifier	Command name	RFD		Subclause
		Tx	Rx	
0x01	Association request	X		7.3.1
0x02	Association response		X	7.3.2
0x03	Disassociation notification	X	X	7.3.3
0x04	Data request	X		7.3.4
0x05	PAN ID conflict notification	X		7.3.5
0x06	Orphan notification	X		7.3.6
0x07	Beacon request			7.3.7
0x08	Coordinator realignment		X	7.3.8
0x09	GTS request			7.3.9
0x0a–0xff	Reserved			—

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.3.1 Association request command

The association request command allows a device to request association with a PAN through the PAN coordinator or a coordinator.

This command shall only be sent by an unassociated device that wishes to associate with a PAN. A device shall only associate with a PAN through the PAN coordinator or a coordinator allowing association, as determined through the scan procedure.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The association request command shall be formatted as illustrated in Figure 55.

octets: (see 7.2.2.4)	**1**	**1**
MHR fields	Command Frame Identifier (see Table 82)	Capability Information

Figure 55—Association request command format

7.3.1.1 MHR fields

The Source Addressing Mode subfield of the Frame Control field shall be set to three (64-bit extended addressing). The Destination Addressing Mode subfield shall be set to the same mode as indicated in the beacon frame to which the association request command refers.

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The Destination PAN Identifier field shall contain the identifier of the PAN to which to associate. The Destination Address field shall contain the address from the beacon frame that was transmitted by the coordinator to which the association request command is being sent. The Source PAN Identifier field shall contain the broadcast PAN identifier (i.e., 0xffff). The Source Address field shall contain the value of *aExtendedAddress*.

7.3.1.2 Capability Information field

The Capability Information field shall be formatted as illustrated in Figure 56.

bits: 0	**1**	**2**	**3**	**4-5**	**6**	**7**
Alternate PAN Coordinator	Device Type	Power Source	Receiver On When Idle	Reserved	Security Capability	Allocate Address

Figure 56—Capability Information field format

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The Alternate PAN Coordinator subfield is 1 bit in length and shall be set to one if the device is capable of becoming the PAN coordinator. Otherwise, the Alternate PAN Coordinator subfield shall be set to zero.

The Device Type subfield is 1 bit in length and shall be set to one if the device is an FFD. Otherwise, the Device Type subfield shall be set to zero to indicate an RFD.

The Power Source subfield is 1 bit in length and shall be set to one if the device is receiving power from the alternating current mains. Otherwise, the Power Source subfield shall be set to zero.

The Receiver On When Idle subfield is 1 bit in length and shall be set to one if the device does not disable its receiver to conserve power during idle periods. Otherwise, the Receiver On When Idle subfield shall be set to zero.

The Security Capability subfield is 1 bit in length and shall be set to one if the device is capable of sending and receiving cryptographically protected MAC frames as specified in 7.5.8.2; it shall be set to zero otherwise.

The Allocate Address subfield is 1 bit in length and shall be set to one if the device wishes the coordinator to allocate a 16-bit short address as a result of the association procedure. Otherwise, it shall be set to zero.

7.3.2 Association response command

The association response command allows the PAN coordinator or a coordinator to communicate the results of an association attempt back to the device requesting association.

This command shall only be sent by the PAN coordinator or coordinator to a device that is currently trying to associate.

All devices shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The association response command shall be formatted as illustrated in Figure 57.

octets: (see 7.2.2.4)	**1**	**2**	**1**
MHR fields	Command Frame Identifier (see Table 82)	Short Address	Association Status

Figure 57—Association response command format

7.3.2.1 MHR fields

The Destination Addressing Mode and Source Addressing Mode subfields of the Frame Control field shall each be set to three (i.e., 64-bit extended addressing).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The PAN ID Compression subfield of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression subfield, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. The Destination Address field shall

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

contain the extended address of the device requesting association. The Source Address field shall contain the value of *aExtendedAddress*.

7.3.2.2 Short Address field

If the coordinator was not able to associate this device to its PAN, the Short Address field shall be set to 0xffff, and the Association Status field shall contain the reason for the failure. If the coordinator was able to associate the device to its PAN, this field shall contain the short address that the device may use in its communications on the PAN until it is disassociated.

A Short Address field value equal to 0xfffe shall indicate that the device has been successfully associated with a PAN, but has not been allocated a short address. In this case, the device shall communicate on the PAN using only its 64-bit extended address.

7.3.2.3 Association Status field

The Association Status field shall contain one of the nonreserved values listed in Table 83.

Table 83—Valid values of the Association Status field

Association status	Description
0x00	Association successful.
0x01	PAN at capacity.
0x02	PAN access denied.
0x03–0x7f	Reserved.
0x80–0xff	Reserved for MAC primitive enumeration values.

7.3.3 Disassociation notification command

The PAN coordinator, a coordinator, or an associated device may send the disassociate notification command.

All devices shall implement this command.

The disassociation notification command shall be formatted as illustrated in Figure 58.

octets: (see 7.2.2.4)	1	1
MHR fields	Command Frame Identifier (see Table 82)	Disassociation Reason

Figure 58—Disassociation notification command format

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.3.3.1 MHR fields

The Destination Addressing Mode subfield of the Frame Control field shall be set according to the addressing mode specified by the corresponding primitive. The Source Addressing Mode subfield shall be set to three (i.e., 64-bit extended addressing).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The PAN ID Compression subfield of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression subfield, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. If the coordinator wants an associated device to leave the PAN, then the Destination Address field shall contain the address of the device being removed from the PAN. If an associated device wants to leave the PAN, then the Destination Address field shall contain the value of either *macCoordShortAddress*, if the Destination Addressing Mode subfield is equal to two, or *macCoordExtendedAddress*, if the Destination Addressing Mode subfield is equal to three. The Source Address field shall contain the value of *aExtendedAddress*.

7.3.3.2 Disassociation Reason field

The Disassociation Reason field shall contain one of the nonreserved values listed in Table 84.

Table 84—Valid disassociation reason codes

Disassociate reason	Description
0x00	Reserved.
0x01	The coordinator wishes the device to leave the PAN.
0x02	The device wishes to leave the PAN.
0x03–0x7f	Reserved.
0x80–0xff	Reserved for MAC primitive enumeration values.

7.3.4 Data request command

The data request command is sent by a device to request data from the PAN coordinator or a coordinator.

There are three cases for which this command is sent. On a beacon-enabled PAN, this command shall be sent by a device when *macAutoRequest* is equal to TRUE and a beacon frame indicating that data are pending for that device is received from its coordinator. The coordinator indicates pending data in its beacon frame by adding the address of the recipient of the data to the Address List field. This command shall also be sent when instructed to do so by the next higher layer on reception of the MLME-POLL.request primitive. In addition, a device may send this command to the coordinator *macResponseWaitTime* symbols after the acknowledgment to an association request command.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The data request command shall be formatted as illustrated in Figure 59.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

octets: (see 7.2.2.4)	1
MHR fields	Command Frame Identifier (see Table 82)

Figure 59—Data request command format

If the data request command is being sent in response to the receipt of a beacon frame indicating that data are pending for that device, the Destination Addressing Mode subfield of the Frame Control field may be set to zero (i.e., destination addressing information not present) if the beacon frame indicated in its Superframe Specification field (see 7.2.2.1.2) that it originated from the PAN coordinator (see 7.2.1.1.6) or set otherwise according to the coordinator to which the data request command is directed. If the destination addressing information is to be included, the Destination Addressing Mode subfield shall be set according to the value of *macCoordShortAddress*. If *macCoordShortAddress* is equal to 0xfffe, extended addressing shall be used: the Destination Addressing Mode subfield shall be set to three, and the Destination Address field shall contain the value of *macCoordExtendedAddress*. Otherwise, short addressing shall be used: the Destination Addressing Mode subfield shall be set to two, and the Destination Address field shall contain the value of *macCoordShortAddress*.

If the data request command is being sent in response to the receipt of a beacon frame indicating that data are pending for that device, the Source Addressing Mode subfield shall be set according to the addressing mode used for the pending address. If the Source Addressing Mode subfield is set to two, short addressing shall be used: the Source Address field shall contain the value of *macShortAddress*. Otherwise, extended addressing shall be used: the Source Addressing Mode subfield shall be set to three, and the Source Address field shall contain the value of *aExtendedAddress*.

If the data request command is triggered by the reception of an MLME-POLL.request primitive from the next higher layer, then the destination addressing information shall be the same as that contained in the primitive. The Source Addressing Mode subfield shall be set according to the value of *macShortAddress*. If *macShortAddress* is less than 0xfffe, short addressing shall be used. Extended addressing shall be used otherwise.

If the data request command is being sent following the acknowledgment to an association request command frame, the Destination Addressing Mode subfield of the Frame Control field shall be set according to the coordinator to which the data request command is directed. If *macCoordShortAddress* is equal to 0xfffe, extended addressing shall be used. Short addressing shall be used otherwise. The Source Addressing Mode subfield shall be set to use extended addressing.

If the Destination Addressing Mode subfield is set to zero (i.e., destination addressing information not present), the PAN ID Compression subfield of the Frame Control field shall be set to zero and the source PAN identifier shall contain the value of *macPANId*. Otherwise, the PAN ID Compression subfield shall be set to one. In this case and in accordance with the PAN ID Compression subfield, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted.

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

7.3.5 PAN ID conflict notification command

The PAN ID conflict notification command is sent by a device to the PAN coordinator when a PAN identifier conflict is detected.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The PAN ID conflict notification command shall be formatted as illustrated in Figure 60.

octets: (see 7.2.2.4)	**1**
MHR fields	Command Frame Identifier (see Table 82)

Figure 60—PAN ID conflict notification command format

The Destination Addressing Mode and Source Addressing Mode subfields of the Frame Control field shall both be set to three (i.e., 64-bit extended addressing).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The PAN ID Compression subfield of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression subfield, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. The Destination Address field shall contain the value of *macCoordExtendedAddress*. The Source Address field shall contain the value of *aExtendedAddress*.

7.3.6 Orphan notification command

The orphan notification command is used by an associated device that has lost synchronization with its coordinator.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The orphan notification command shall be formatted as illustrated in Figure 61.

octets: 15	**1**
MHR fields	Command Frame Identifier (see Table 82)

Figure 61—Orphan notification command format

The Source Addressing Mode subfield of the Frame Control field shall be set to three (i.e., 64-bit extended addressing). The Destination Addressing Mode subfield shall be set to two (i.e., 16-bit short addressing).

The Frame Pending subfield and Acknowledgment Request subfield of the Frame Control field shall be set to zero and ignored upon reception.

The PAN ID Compression subfield of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression subfield, the Destination PAN Identifier field shall contain the value of the broadcast PAN identifier (i.e., 0xffff), while the Source PAN Identifier field shall be omitted. The

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Destination Address field shall contain the broadcast short address (i.e., 0xffff). The Source Address field shall contain the value of *aExtendedAddress*.

7.3.7 Beacon request command

The beacon request command is used by a device to locate all coordinators within its POS during an active scan.

This command is optional for an RFD.

The beacon request command shall be formatted as illustrated in Figure 62.

octets: 7	1
MHR fields	Command Frame Identifier (see Table 82)

Figure 62—Beacon request command format

The Destination Addressing Mode subfield of the Frame Control field shall be set to two (i.e., 16-bit short addressing), and the Source Addressing Mode subfield shall be set to zero (i.e., source addressing information not present).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception. The Acknowledgment Request subfield and Security Enabled subfield shall also be set to zero.

The Destination PAN Identifier field shall contain the broadcast PAN identifier (i.e., 0xffff). The Destination Address field shall contain the broadcast short address (i.e., 0xffff).

7.3.8 Coordinator realignment command

The coordinator realignment command is sent by the PAN coordinator or a coordinator either following the reception of an orphan notification command from a device that is recognized to be on its PAN or when any of its PAN configuration attributes change due to the receipt of an MLME-START.request primitive.

If this command is sent following the reception of an orphan notification command, it is sent directly to the orphaned device. If this command is sent when any PAN configuration attributes (i.e., PAN identifier, short address, logical channel, or channel page) change, it is broadcast to the PAN.

All devices shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The coordinator realignment command shall be formatted as illustrated in Figure 63.

octets: 17/18/23/24	1	2	2	1	2	0/1
MHR fields	Command Frame Identifier (see Table 82)	PAN Identifier	Coordinator Short Address	Logical Channel	Short Address	Channel page

Figure 63—Coordinator realignment command format

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.3.8.1 MHR fields

The Destination Addressing Mode subfield of the Frame Control field shall be set to three (e.g., 64-bit extended addressing) if the command is directed to an orphaned device or set to two (e.g., 16-bit short addressing) if it is to be broadcast to the PAN. The Source Addressing Mode subfield of the Frame Control field shall be set to three (e.g., 64-bit extended addressing).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception.

The Acknowledgment Request subfield of the Frame Control field shall be set to one if the command is directed to an orphaned device or set to zero if the command is to be broadcast to the PAN.

The Frame Version subfield shall be set to 0x01 if the Channel Page field is present. Otherwise it shall be set as specified in 7.2.3.

The Destination PAN Identifier field shall contain the broadcast PAN identifier (e.g., 0xffff). The Destination Address field shall contain the extended address of the orphaned device if the command is directed to an orphaned device. Otherwise, the Destination Address field shall contain the broadcast short address (e.g., 0xffff). The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *aExtendedAddress*.

7.3.8.2 PAN Identifier field

The PAN Identifier field shall contain the PAN identifier that the coordinator intends to use for all future communications.

7.3.8.3 Coordinator Short Address field

The Coordinator Short Address field shall contain the value of *macShortAddress*.

7.3.8.4 Logical Channel field

The Logical Channel field shall contain the logical channel that the coordinator intends to use for all future communications.

7.3.8.5 Short Address field

If the coordinator realignment command is broadcast to the PAN, the Short Address field shall be set to 0xffff and ignored on reception.

If the coordinator realignment command is sent directly to an orphaned device, this field shall contain the short address that the orphaned device shall use to operate on the PAN. If the orphaned device does not have a short address, because it always uses its 64-bit extended address, this field shall contain the value 0xfffe.

7.3.8.6 Channel Page field

The Channel Page field, if present, shall contain the channel page that the coordinator intends to use for all future communications. This field may be omitted if the new channel page is the same as the previous channel page.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.3.9 GTS request command

The GTS request command is used by an associated device that is requesting the allocation of a new GTS or the deallocation of an existing GTS from the PAN coordinator. Only devices that have a 16-bit short address less than 0xfffe shall send this command.

This command is optional.

The GTS request command shall be formatted as illustrated in Figure 64.

octets: 7	1	1
MHR fields	Command Frame Identifier (see Table 82)	GTS Characteristics

Figure 64—GTS request command format

7.3.9.1 MHR fields

The Destination Addressing Mode subfield of the Frame Control field shall be set to zero (e.g., destination addressing information not present), and the Source Addressing Mode subfield shall be set to two (e.g., 16-bit short addressing).

The Frame Pending subfield of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

7.3.9.2 GTS Characteristics field

The GTS Characteristics field shall be formatted as illustrated in Figure 65.

bits: 0–3	4	5	6–7
GTS Length	GTS Direction	Characteristics Type	Reserved

Figure 65—GTS Characteristics field format

The GTS Length subfield shall contain the number of superframe slots being requested for the GTS.

The GTS Direction subfield shall be set to one if the GTS is to be a receive-only GTS. Conversely, this subfield shall be set to zero if the GTS is to be a transmit-only GTS. GTS direction is defined relative to the direction of data frame transmissions by the device.

The Characteristics Type subfield shall be set to one if the characteristics refers to a GTS allocation or zero if the characteristics refers to a GTS deallocation.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.4 MAC constants and PIB attributes

This subclause specifies the constants and attributes required by the MAC sublayer.

7.4.1 MAC constants

The constants that define the characteristics of the MAC sublayer are presented in Table 85.

Table 85—MAC sublayer constants

Constant	Description	Value
aBaseSlotDuration	The number of symbols forming a superframe slot when the superframe order is equal to 0 (see 7.5.1.1).	60
aBaseSuperframeDuration	The number of symbols forming a superframe when the superframe order is equal to 0.	*aBaseSlotDuration* * *aNumSuperframeSlots*
aExtendedAddress	The 64-bit (IEEE) address assigned to the device.	Device specific
aGTSDescPersistenceTime	The number of superframes in which a GTS descriptor exists in the beacon frame of the PAN coordinator.	4
aMaxBeaconOverhead	The maximum number of octets added by the MAC sublayer to the MAC payload of a beacon frame.	75
aMaxBeaconPayloadLength	The maximum size, in octets, of a beacon payload.	*aMaxPHYPacketSize* – *aMaxBeaconOverhead*
aMaxLostBeacons	The number of consecutive lost beacons that will cause the MAC sublayer of a receiving device to declare a loss of synchronization.	4
aMaxMACSafePayloadSize	The maximum number of octets that can be transmitted in the MAC Payload field of an unsecured MAC frame that will be guaranteed not to exceed *aMaxPHYPacketSize*.	*aMaxPHYPacketSize* – *aMaxMPDUUnsecuredOverhead*
aMaxMACPayloadSize	The maximum number of octets that can be transmitted in the MAC Payload field.	*aMaxPHYPacketSize* – *aMinMPDUOverhead*
aMaxMPDUUnsecuredOverhead	The maximum number of octets added by the MAC sublayer to the PSDU without security.	25
aMaxSIFSFrameSize	The maximum size of an MPDU, in octets, that can be followed by a SIFS period.	18
aMinCAPLength	The minimum number of symbols forming the CAP. This ensures that MAC commands can still be transferred to devices when GTSs are being used. An exception to this minimum shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3).	440
aMinMPDUOverhead	The minimum number of octets added by the MAC sublayer to the PSDU.	9
aNumSuperframeSlots	The number of slots contained in any superframe.	16
aUnitBackoffPeriod	The number of symbols forming the basic time period used by the CSMA-CA algorithm.	20

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.4.2 MAC PIB attributes

The MAC PIB comprises the attributes required to manage the MAC sublayer of a device. The attributes contained in the MAC PIB are presented in Table 86. Attributes marked with a dagger (†) are read-only attributes (i.e., attribute can only be set by the MAC sublayer), which can be read by the next higher layer using the MLME-GET.request primitive. All other attributes can be read or written by the next higher layer using the MLME-GET.request or MLME-SET.request primitives, respectively. Attributes marked with a diamond (◆) are optional for an RFD; attributes marked with an asterisk (*) are optional for both device types (i.e., RFD and FFD).

The read-only attribute *macAckWaitDuration* is dependent on a combination of constants and PHY PIB attributes. The formula for relating the constants and attributes is shown in Equation (13).

$$macAckWaitDuration = aUnitBackoffPeriod + aTurnaroundTime + phySHRDuration + \lceil 6 \cdot phySymbolsPerOctet \rceil \quad (13)$$

where

6 represents the number of PHY header octets plus the number of PSDU octets in an acknowledgment frame.

The attribute *macMaxFrameTotalWaitTime* may be set by the next higher layer and is dependent upon a combination of PHY and MAC PIB attributes and constants. The formula relating the attributes and constants is shown in Equation (14):

$$macMaxFrameTotalWaitTime = \left[\left(\sum_{k=0}^{m-1} 2^{macMinBE+k}\right) + (2^{macMaxBE} - 1) \cdot (macMaxCSMABackoffs - m)\right] \bullet aUnitBackoffPeriod + phyMaxFrameDuratio \quad (14)$$

where

m is min(*macMaxBE-macMinBE, macMaxCSMABackoffs).*

Table 86—MAC PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macAckWaitDuration†	0x40	Integer	See Equation (13)	The maximum number of symbols to wait for an acknowledgment frame to arrive following a transmitted data frame. This value is dependent on the supported PHY, which determines both the selected logical channel and channel page. The calculated value is the time to commence transmitting the ACK plus the length of the ACK frame. The commencement time is described in 7.5.6.4.2.	Dependent on currently selected PHY, indicated by *phyCurrentPage*

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macAssociatedPAN-Coord	0x56	Boolean	TRUE or FALSE	Indication of whether the device is associated to the PAN through the PAN coordinator. A value of TRUE indicates the device has associated through the PAN coordinator. Otherwise, the value is set to FALSE.	FALSE
macAssociation-Permit◆	0x41	Boolean	TRUE or FALSE	Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.	FALSE
macAutoRequest	0x42	Boolean	TRUE or FALSE	Indication of whether a device automatically sends a data request command if its address is listed in the beacon frame. A value of TRUE indicates that the data request command is automatically sent. This attribute also affects the generation of the MLME-BEACON-NOTIFY.indication primitive (see 7.1.5.1.2).	TRUE
macBattLifeExt	0x43	Boolean	TRUE or FALSE	Indication of whether BLE, through the reduction of coordinator receiver operation time during the CAP, is enabled. A value of TRUE indicates that it is enabled. Also, see 7.5.1.4 for an explanation of how this attribute affects the backoff exponent in the CSMA-CA algorithm.	FALSE

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macBattLifeExtPeriods	0x44	Integer	6-41	In BLE mode, the number of backoff periods during which the receiver is enabled after the IFS following a beacon. This value is dependent on the supported PHY and is the sum of three terms: Term 1: The value $2^x - 1$, where x is the maximum value of *macMinBE* in BLE mode (equal to two). This term is thus equal to 3 backoff periods. Term 2: The duration of the initial contention window length (see 7.5.1.4). This term is thus equal to 2 backoff periods. Term 3: The Preamble field length and the SFD field length of the supported PHY (see Table 19 and Table 20 in Clause 6), summed together and rounded up (if necessary) to an integer number of back-off periods.	Dependent on currently selected PHY, indicated by *phyCurrentPage*
macBeaconPayload◆	0x45	Set of octets	—	The contents of the beacon payload.	NULL
macBeaconPayloadLength◆	0x46	Integer	0 – *aMaxBeaconPayloadLength*	The length, in octets, of the beacon payload.	0
macBeaconOrder◆	0x47	Integer	0–15	Specification of how often the coordinator transmits its beacon. If *BO* = 15, the coordinator will not transmit a periodic beacon. See 7.5.1.1 for an explanation of the relationship between the beacon order and the beacon interval.	15

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macBeaconTxTime†◆	0x48	Integer	0x000000–0xffffff	The time that the device transmitted its last beacon frame, in symbol periods. The measurement shall be taken at the same symbol boundary within every transmitted beacon frame, the location of which is implementation specific. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest four bits being the least significant.	0x000000
macBSN◆	0x49	Integer	0x00–0xff	The sequence number added to the transmitted beacon frame.	Random value from within the range
macCoordExtended-Address	0x4a	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the coordinator through which the device is associated.	—
macCoordShort-Address	0x4b	Integer	0x0000–0xffff	The 16-bit short address assigned to the coordinator through which the device is associated. A value of 0xfffe indicates that the coordinator is only using its 64-bit extended address. A value of 0xffff indicates that this value is unknown.	0xffff
macDSN	0x4c	Integer	0x00–0xff	The sequence number added to the transmitted data or MAC command frame.	Random value from within the range
*macGTSPermit**	0x4d	Boolean	TRUE or FALSE	TRUE if the PAN coordinator is to accept GTS requests. FALSE otherwise.	TRUE
macMaxBE	0x57	Integer	3–8	The maximum value of the backoff exponent, BE, in the CSMA-CA algorithm. See 7.5.1.4 for a detailed explanation of the backoff exponent.	5
macMaxCSMABackoffs	0x4e	Integer	0–5	The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure.	4

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macMaxFrameTotal-WaitTime	0x58	Integer	See Equation (14)	The maximum number of CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, to wait either for a frame intended as a response to a data request frame or for a broadcast frame following a beacon with the Frame Pending subfield set to one. This attribute, which shall only be set by the next higher layer, is dependent upon *macMinBE*, *macMaxBE*, *macMaxCSMABackoffs* and the number of symbols per octet. See 7.4.2 for the formula relating the attributes.	Dependent on currently selected PHY, indicated by *phy-Current-Page*
macMaxFrameRetries	0x59	Integer	0–7	The maximum number of retries allowed after a transmission failure.	3
macMinBE	0x4f	Integer	0–*macMaxBE*	The minimum value of the backoff exponent (BE) in the CSMA-CA algorithm. See 7.5.1.4 for a detailed explanation of the backoff exponent.	3
macMinLIFSPeriod†		Integer	See Table 3 in Clause 6	The minimum number of symbols forming a LIFS period.	Dependent on currently selected PHY, indicated by *phy-Current-Page*
macMinSIFSPeriod†		Integer	See Table 3 in Clause 6	The minimum number of symbols forming a SIFS period.	Dependent on currently selected PHY, indicated by *phy-Current-Page*
macPANId	0x50	Integer	0x0000–0xffff	The 16-bit identifier of the PAN on which the device is operating. If this value is 0xffff, the device is not associated.	0xffff

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macPromiscuous-Mode◆	0x51	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer is in a promiscuous (receive all) mode. A value of TRUE indicates that the MAC sublayer accepts all frames received from the PHY.	FALSE
macResponseWaitTime	0x5a	Integer	2–64	The maximum time, in multiples of *aBaseSuperframeDuration*, a device shall wait for a response command frame to be available following a request command frame.	32
macRxOnWhenIdle	0x52	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer is to enable its receiver during idle periods. For a beacon-enabled PAN, this attribute is relevant only during the CAP of the incoming superframe. For a nonbeacon-enabled PAN, this attribute is relevant at all times.	FALSE
macSecurityEnabled	0x5d	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer has security enabled. A value of TRUE indicates that security is enabled, while a value of FALSE indicates that security is disabled.	FALSE
macShortAddress	0x53	Integer	0x0000–0xffff	The 16-bit address that the device uses to communicate in the PAN. If the device is the PAN coordinator, this value shall be chosen before a PAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfffe indicates that the device has associated but has not been allocated an address. A value of 0xffff indicates that the device does not have a short address.	0xffff

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 86—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macSuperframeOrder†◆	0x54	Integer	0–15	The length of the active portion of the outgoing superframe, including the beacon frame. If superframe order, *SO*, = 15, the superframe will not be active following the beacon. See 7.5.1.1 for an explanation of the relationship between the superframe order and the superframe duration.	15
macSyncSymbolOffset†	0x5b	Integer	0x000–0x100 for the 2.4 GHz PHY 0x000–0x400 for the 868/915 MHz PHY	The offset, measured in symbols, between the symbol boundary at which the MLME captures the timestamp of each transmitted or received frame, and the onset of the first symbol past the SFD, namely, the first symbol of the Length field.	Implementation specific
macTimestampSupported†	0x5c	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer supports the optional timestamping feature for incoming and outgoing data frames.	Implementation specific
macTransactionPersistenceTime◆	0x55	Integer	0x0000–0xffff	The maximum time (in unit periods) that a transaction is stored by a coordinator and indicated in its beacon. The unit period is governed by *macBeaconOrder*, *BO*, as follows: For $0 \le BO \le 14$, the unit period will be *aBaseSuperframeDuration* * 2^{BO}. For *BO* = *15*, the unit period will be *aBaseSuperframeDuration*.	0x01f4

7.5 MAC functional description

This subclause provides a detailed description of the MAC functionality. Subclause 7.5.1 describes the following two mechanisms for channel access: contention based and contention free. Contention-based access allows devices to access the channel in a distributed fashion using a CSMA-CA backoff algorithm. Contention-free access is controlled entirely by the PAN coordinator through the use of GTSs.

The mechanisms used for starting and maintaining a PAN are described in 7.5.2. Channel scanning is used by a device to assess the current state of a channel (or channels), locate all beacons within its POS, or locate a particular beacon with which it has lost synchronization. Before starting a new PAN, the results of a channel scan can be used to select an appropriate logical channel and channel page, as well as a PAN identifier that is not being used by any other PAN in the area. Because it is still possible for the POS of two PANs with the same PAN identifier to overlap, a procedure exists to detect and resolve this situation. Following a channel scan and suitable PAN identifier selection, an FFD can begin operating as the PAN

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

coordinator. Also described in the subclause is a method to allow a beaconing FFD to discover other such devices during normal operations, i.e., when not scanning.

The mechanisms to allow devices to join or leave a PAN are defined in 7.5.3. The association procedure describes the conditions under which a device may join a PAN and the conditions necessary for a coordinator to permit devices to join. Also described is the disassociation procedure, which can be initiated by the associated device or its coordinator.

The mechanisms to allow devices to acquire and maintain synchronization with a coordinator are described in 7.5.4. Synchronization on a beacon-enabled PAN is described after first explaining how a coordinator generates beacon frames. Following this explanation, synchronization on a nonbeacon-enabled PAN is described. Also described is a procedure to reestablish communication between a device and its coordinator, as it is possible that a device may lose synchronization in the case of either a beacon-enabled or a nonbeacon-enabled PAN.

This standard has been designed so that application data transfers can be controlled by the devices on a PAN rather than by the coordinator. The procedures the coordinator uses to handle multiple transactions while preserving this requirement are described in 7.5.5.

The mechanisms for transmitting, receiving, and acknowledging frames, including frames sent using indirect transmission, are described in 7.5.6. In addition, methods for retransmitting frames are also described.

The mechanisms for allocating and deallocating a GTS are described in 7.5.7. The deallocation process may result in the fragmentation of the GTS space, i.e., an unused slot or slots. The subclause describes a mechanism to resolve fragmentation.

The MAC sublayer uses the mechanisms described in 7.5.8 for all incoming and outgoing frames.

Throughout this subclause, the receipt of a frame is defined as the successful receipt of the frame by the PHY and the successful verification of the FCS by the MAC sublayer, as described in 7.2.1.9.

7.5.1 Channel access

This subclause describes the mechanisms for accessing the physical radio channel.

7.5.1.1 Superframe structure

A coordinator on a PAN can optionally bound its channel time using a superframe structure. A superframe is bounded by the transmission of a beacon frame and can have an active portion and an inactive portion. The coordinator may enter a low-power (sleep) mode during the inactive portion.

The structure of this superframe is described by the values of *macBeaconOrder* and *macSuperframeOrder*. The MAC PIB attribute *macBeaconOrder*, describes the interval at which the coordinator shall transmit its beacon frames. The value of *macBeaconOrder*, *BO*, and the beacon interval, *BI*, are related as follows: for $0 \leq BO \leq 14$, $BI = aBaseSuperframeDuration * 2^{BO}$ symbols. If $BO = 15$, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command. The value of *macSuperframeOrder* shall be ignored if $BO = 15$.

The MAC PIB attribute *macSuperframeOrder* describes the length of the active portion of the superframe, which includes the beacon frame. The value of *macSuperframeOrder*, *SO*, and the superframe duration, *SD*, are related as follows: for $0 \leq SO \leq BO \leq 14$, $SD = aBaseSuperframeDuration * 2^{SO}$ symbols. If $SO = 15$, the superframe shall not remain active after the beacon. If $BO = 15$, the superframe shall not exist (the value of

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

macSuperframeOrder shall be ignored), and *macRxOnWhenIdle* shall define whether the receiver is enabled during periods of transceiver inactivity.

The active portion of each superframe shall be divided into *aNumSuperframeSlots* equally spaced slots of duration 2^{SO} * *aBaseSlotDuration* and is composed of three parts: a beacon, a CAP and a CFP. The beacon shall be transmitted, without the use of CSMA, at the start of slot 0, and the CAP shall commence immediately following the beacon. The start of slot 0 is defined as the point at which the first symbol of the beacon PPDU is transmitted. The CFP, if present, follows immediately after the CAP and extends to the end of the active portion of the superframe. Any allocated GTSs shall be located within the CFP.

The MAC sublayer shall ensure that the integrity of the superframe timing is maintained, e.g., compensating for clock drift error.

PANs that wish to use the superframe structure (referred to as a beacon-enabled PAN) shall set *macBeaconOrder* to a value between 0 and 14, both inclusive, and *macSuperframeOrder* to a value between 0 and the value of *macBeaconOrder*, both inclusive.

PANs that do not wish to use the superframe structure (referred to as a nonbeacon-enabled PAN) shall set both *macBeaconOrder* and *macSuperframeOrder* to 15. In this case, a coordinator shall not transmit beacons, except upon receipt of a beacon request command; all transmissions, with the exception of acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command (see 7.5.6.3), shall use an unslotted CSMA-CA mechanism to access the channel. In addition, GTSs shall not be permitted.

An example of a superframe structure is shown in Figure 66. In this case, the beacon interval, *BI*, is twice as long as the active superframe duration, *SD*, and the CFP contains two GTSs.

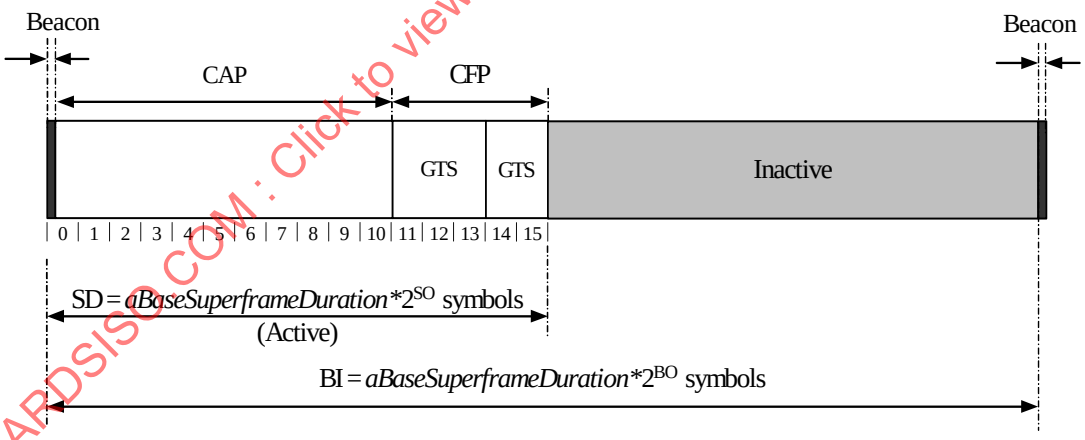


Figure 66—An example of the superframe structure

7.5.1.1.1 Contention access period (CAP)

The CAP shall start immediately following the beacon and complete before the beginning of the CFP on a superframe slot boundary. If the CFP is zero length, the CAP shall complete at the end of the active portion of the superframe. The CAP shall be at least *aMinCAPLength* symbols, unless additional space is needed to temporarily accommodate the increase in the beacon frame length needed to perform GTS maintenance (see 7.2.2.1.3), and shall shrink or grow dynamically to accommodate the size of the CFP.

All frames, except acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command (see 7.5.6.3), transmitted in the CAP shall use a slotted CSMA-CA mechanism to

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

access the channel. A device transmitting within the CAP shall ensure that its transaction is complete (i.e., including the reception of any acknowledgment) one IFS period (see 7.5.1.3) before the end of the CAP. If this is not possible, the device shall defer its transmission until the CAP of the following superframe.

MAC command frames shall always be transmitted in the CAP.

7.5.1.1.2 Contention-free period (CFP)

The CFP shall start on a slot boundary immediately following the CAP and it shall complete before the end of the active portion of the superframe. If any GTSs have been allocated by the PAN coordinator, they shall be located within the CFP and occupy contiguous slots. The CFP shall therefore grow or shrink depending on the total length of all of the combined GTSs.

No transmissions within the CFP shall use a CSMA-CA mechanism to access the channel. A device transmitting in the CFP shall ensure that its transmissions are complete one IFS period (see 7.5.1.3) before the end of its GTS.

7.5.1.2 Incoming and outgoing superframe timing

On a beacon-enabled PAN, a coordinator that is not the PAN coordinator shall maintain the timing of both the superframe in which its coordinator transmits a beacon (the incoming superframe) and the superframe in which it transmits its own beacon (the outgoing superframe). The relative timing of these superframes is defined by the StartTime parameter of the MLME-START.request primitive (see 7.1.14.1 and 7.5.2.4). The relationship between incoming and outgoing superframes is illustrated in Figure 67.

The beacon order and superframe order shall be equal for all superframes on a PAN. All devices shall interact with the PAN only during the active portion of a superframe.

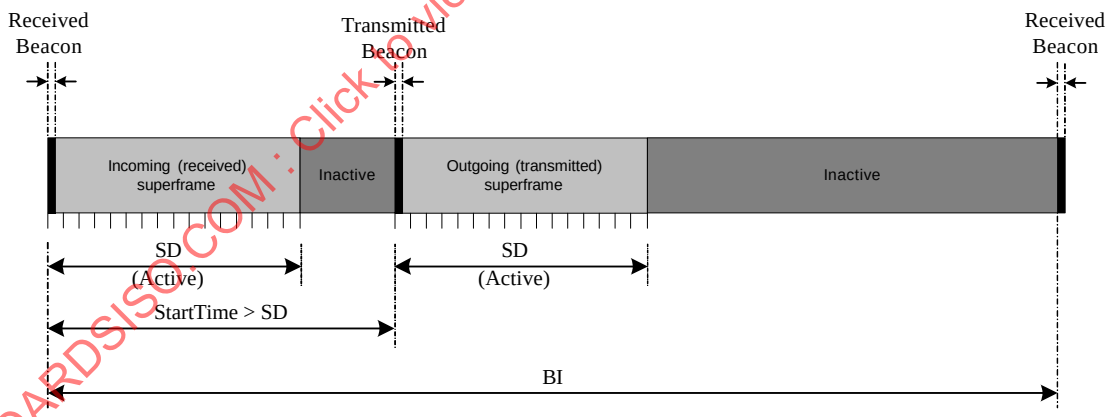


Figure 67—The relationship between incoming and outgoing beacons

7.5.1.3 Interframe spacing (IFS)

The MAC sublayer needs a finite amount of time to process data received by the PHY. To allow for this, two successive frames transmitted from a device shall be separated by at least an IFS period; if the first transmission requires an acknowledgment, the separation between the acknowledgment frame and the second transmission shall be at least an IFS period. The length of the IFS period is dependent on the size of the frame that has just been transmitted. Frames (i.e., MPDUs) of up to *aMaxSIFSFrameSize* octets in length shall be followed by a SIFS period of a duration of at least *macMinSIFSPeriod* symbols. Frames (i.e., MPDUs) with lengths greater than *aMaxSIFSFrameSize* octets shall be followed by a LIFS period of a duration of at least *macMinLIFSPeriod* symbols. These concepts are illustrated in Figure 68.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The CSMA-CA algorithm shall take this requirement into account for transmissions in the CAP.

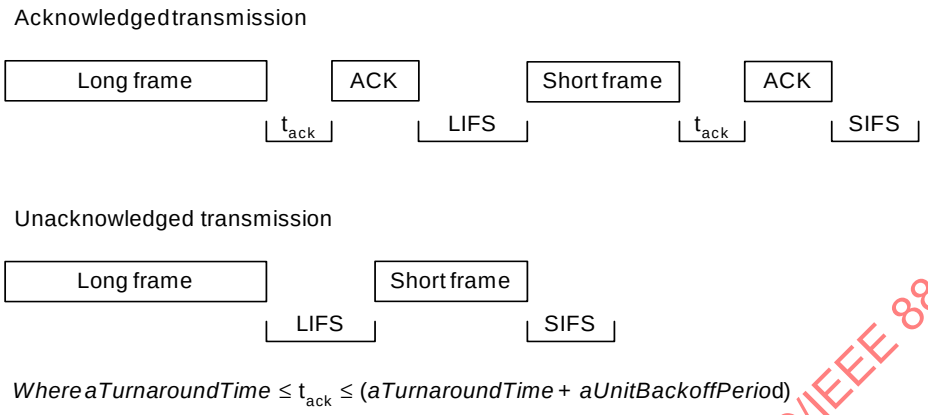


Figure 68—IFS

7.5.1.4 CSMA-CA algorithm

The CSMA-CA algorithm shall be used before the transmission of data or MAC command frames transmitted within the CAP, unless the frame can be quickly transmitted following the acknowledgment of a data request command (see 7.5.6.3 for timing requirements). The CSMA-CA algorithm shall not be used for the transmission of beacon frames in a beacon-enabled PAN, acknowledgment frames, or data frames transmitted in the CFP.

If periodic beacons are being used in the PAN, the MAC sublayer shall employ the slotted version of the CSMA-CA algorithm for transmissions in the CAP of the superframe. Conversely, if periodic beacons are not being used in the PAN or if a beacon could not be located in a beacon-enabled PAN, the MAC sublayer shall transmit using the unslotted version of the CSMA-CA algorithm. In both cases, the algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to *aUnitBackoffPeriod* symbols.

In slotted CSMA-CA, the backoff period boundaries of every device in the PAN shall be aligned with the superframe slot boundaries of the PAN coordinator, i.e., the start of the first backoff period of each device is aligned with the start of the beacon transmission. In slotted CSMA-CA, the MAC sublayer shall ensure that the PHY commences all of its transmissions on the boundary of a backoff period. In unslotted CSMA-CA, the backoff periods of one device are not related in time to the backoff periods of any other device in the PAN.

Each device shall maintain three variables for each transmission attempt: *NB*, *CW* and *BE*. *NB* is the number of times the CSMA-CA algorithm was required to backoff while attempting the current transmission; this value shall be initialized to zero before each new transmission attempt. *CW* is the contention window length, defining the number of backoff periods that need to be clear of channel activity before the transmission can commence; this value shall be initialized to two before each transmission attempt and reset to two each time the channel is assessed to be busy. The *CW* variable is only used for slotted CSMA-CA. *BE* is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to assess a channel. In unslotted systems, or slotted systems with the received BLE subfield (see Figure 47) set to zero, *BE* shall be initialized to the value of *macMinBE*. In slotted systems with the received BLE subfield set to one, this value shall be initialized to the lesser of two and the value of *macMinBE*. Note that if *macMinBE* is set to zero, collision avoidance will be disabled during the first iteration of this algorithm.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Although the receiver of the device is enabled during the CCA analysis portion of this algorithm, the device may discard any frames received during this time.

Figure 69 illustrates the steps of the CSMA-CA algorithm. When using slotted CSMA-CA, the MAC sublayer shall first initialize *NB*, *CW*, and *BE* and then locate the boundary of the next backoff period [step (1)]. For unslotted CSMA-CA, the MAC sublayer shall initialize *NB* and *BE* and then proceed directly to step (2).

The MAC sublayer shall delay for a random number of complete backoff periods in the range 0 to $2^{BE} - 1$ [step (2)] and then request that the PHY perform a CCA [step (3)]. In a slotted CSMA-CA system, the CCA shall start on a backoff period boundary. In an unslotted CSMA-CA system, the CCA shall start immediately.

In a slotted CSMA-CA system with the BLE subfield set to zero, the MAC sublayer shall ensure that, after the random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. Note that any bit padding used by the supported PHY (see 6.7.2.2) must be considered in making this determination. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer shall pause the backoff countdown at the end of the CAP and resume it at the start of the CAP in the next superframe. If the number of backoff periods is less than or equal to the remaining number of backoff periods in the CAP, the MAC sublayer shall apply its backoff delay and then evaluate whether it can proceed. The MAC sublayer shall proceed if the remaining CSMA-CA algorithm steps (i.e., two CCA analyses), the frame transmission, and any acknowledgment can be completed before the end of the CAP. If the MAC sublayer can proceed, it shall request that the PHY perform the CCA in the current superframe. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and apply a further random backoff delay [step (2)] before evaluating whether it can proceed again.

In a slotted CSMA-CA system with the BLE subfield set to one, the MAC sublayer shall ensure that, after the random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. The backoff countdown shall only occur during the first *macBattLifeExtPeriods* full backoff periods after the end of the IFS period following the beacon. The MAC sublayer shall proceed if the remaining CSMA-CA algorithm steps (two CCA analyses), the frame transmission, and any acknowledgment can be completed before the end of the CAP, and the frame transmission will start in one of the first *macBattLifeExtPeriods* full backoff periods after the IFS period following the beacon. If the MAC sublayer can proceed, it shall request that the PHY perform the CCA in the current superframe. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and apply a further random backoff delay [step (2)] before evaluating whether it can proceed again.

If the channel is assessed to be busy [step (4)], the MAC sublayer shall increment both *NB* and *BE* by one, ensuring that *BE* shall be no more than *macMaxBE*. The MAC sublayer in a slotted CSMA-CA system shall also reset *CW* to two. If the value of *NB* is less than or equal to *macMaxCSMABackoffs*, the CSMA-CA algorithm shall return to step (2). If the value of *NB* is greater than *macMaxCSMABackoffs*, the CSMA-CA algorithm shall terminate with a channel access failure status.

If the channel is assessed to be idle [step (5)], the MAC sublayer in a slotted CSMA-CA system shall ensure that the contention window has expired before commencing transmission. To do this, the MAC sublayer shall first decrement *CW* by one and then determine whether it is equal to zero. If it is not equal to zero, the CSMA-CA algorithm shall return to step (3). If it is equal to zero, the MAC sublayer shall begin transmission of the frame on the boundary of the next backoff period. If the channel is assessed to be idle in an unslotted CSMA-CA system, the MAC sublayer shall begin transmission of the frame immediately.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

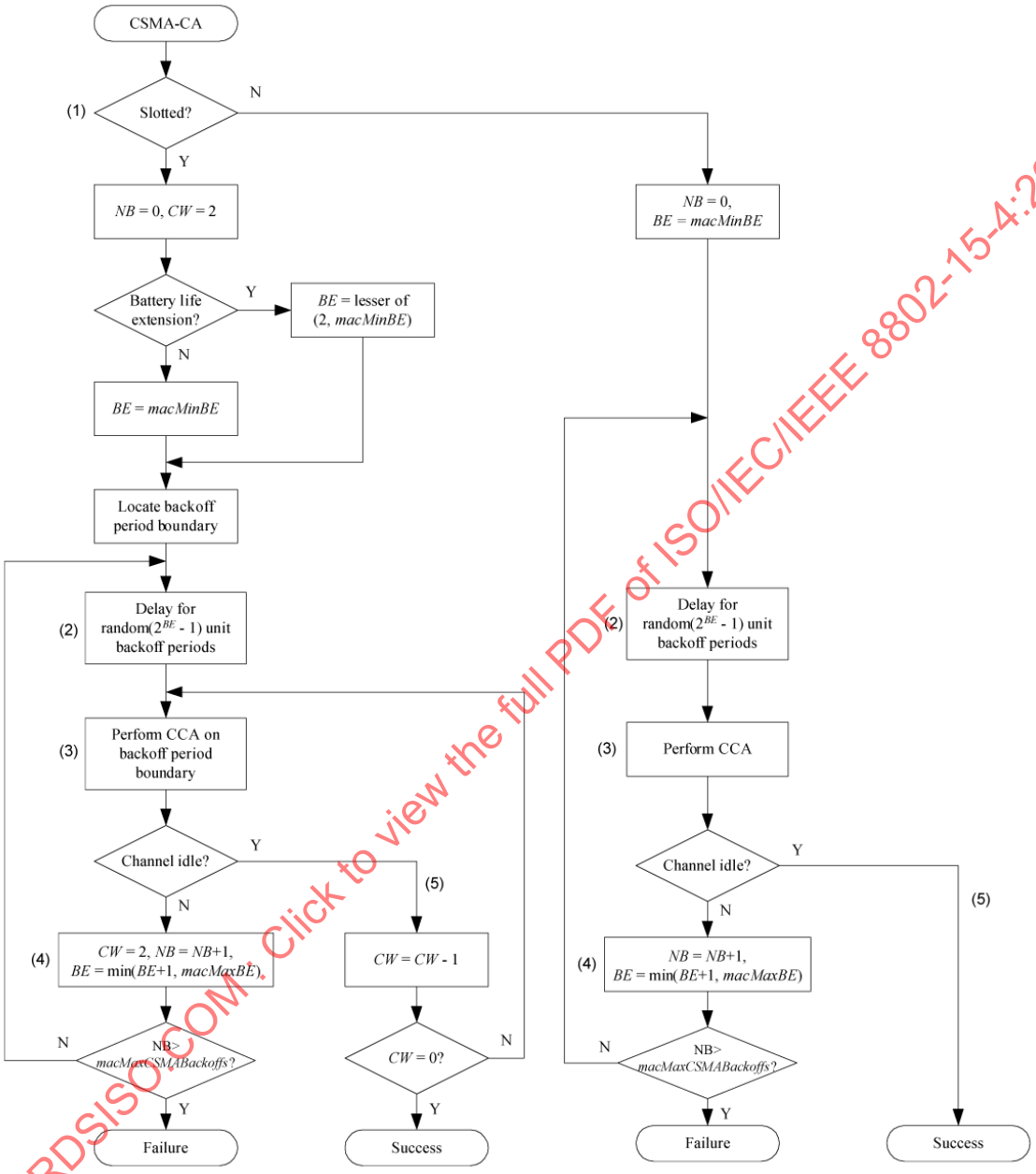


Figure 69—CSMA-CA algorithm

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.5.2 Starting and maintaining PANs

This subclause specifies the procedures for scanning through channels, identifying PAN identifier conflicts, and starting PANs.

7.5.2.1 Scanning through channels

All devices shall be capable of performing passive and orphan scans across a specified list of channels. In addition, an FFD shall be able to perform ED and active scans. The next higher layer should submit a scan

© IEEE 2010 – All rights reserved

request for a particular channel page containing a list of channels chosen only from the channels specified by *phyChannelsSupported* for that particular channel page.

A device is instructed to begin a channel scan through the MLME-SCAN.request primitive. Channels are scanned in order from the lowest channel number to the highest. For the duration of the scan, the device shall suspend beacon transmissions, if applicable, and shall only accept frames received over the PHY data service that are relevant to the scan being performed. Upon the conclusion of the scan, the device shall recommence beacon transmissions. The results of the scan shall be returned via the MLME-SCAN.confirm primitive.

7.5.2.1.1 ED channel scan

An ED scan allows a device to obtain a measure of the peak energy in each requested channel. This could be used by a prospective PAN coordinator to select a channel on which to operate prior to starting a new PAN. During an ED scan, the MAC sublayer shall discard all frames received over the PHY data service.

An ED scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an ED scan. For each logical channel, the MLME shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and then repeatedly perform an ED measurement for [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter in the MLME-SCAN.request primitive. An ED measurement is performed by the MLME issuing the PLME-ED.request (see 6.2.2.3), which is guaranteed to return a value. The maximum ED measurement obtained during this period shall be noted before moving on to the next channel in the channel list. A device shall be able to store between one and an implementation-specified maximum number of channel ED measurements.

The ED scan shall terminate when either the number of channel ED measurements stored equals the implementation-specified maximum or energy has been measured on each of the specified logical channels.

7.5.2.1.2 Active channel scan

An active scan allows a device to locate any coordinator transmitting beacon frames within its POS. This could be used by a prospective PAN coordinator to select a PAN identifier prior to starting a new PAN, or it could be used by a device prior to association.

During an active scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames. If a beacon frame is received that contains the address of the scanning device in its list of pending addresses, the scanning device shall not attempt to extract the pending data.

Before commencing an active scan, the MAC sublayer shall store the value of *macPANId* and then set it to 0xffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current PAN (see 7.5.6.2). On completion of the scan, the MAC sublayer shall restore the value of *macPANId* to the value stored before the scan began.

An active scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an active scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and send a beacon request command (see 7.3.7). Upon successful transmission of the beacon request command, the device shall enable its receiver for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the ScanDuration parameter. During this time, the device shall reject all nonbeacon frames and record the information contained in all unique beacons in a PAN descriptor structure (see Table 55 in 7.1.5.1.1). If a beacon frame is received when *macAutoRequest* is set to TRUE, the list of PAN descriptor structures shall be stored by the MAC sublayer until the scan is complete; at this time, the list shall be sent to the next higher layer in the PANDescriptorList parameter of the MLME-SCAN.confirm primitive. A device

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

shall be able to store between one and an implementation-specified maximum number of PAN descriptors. A beacon frame shall be assumed to be unique if it contains both a PAN identifier and a source address that has not been seen before during the scan of the current channel. If a beacon frame is received when *macAutoRequest* is set to FALSE, each recorded PAN descriptor is sent to the next higher layer in a separate MLME-BEACON-NOTIFY.indication primitive. A received beacon frame containing one or more octets of payload shall also cause the PAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY.indication primitive.

If a protected beacon frame is received, i.e., the Security Enabled subfield in the Frame Control field is set to one, the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.5.8.2.3.

The security-related elements of the PAN descriptor corresponding to the beacon (see Table 55) shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the PAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

The information from the unsecured frame shall be recorded in the PAN descriptor even if the status from the unsecuring process indicated an error.

If a coordinator of a beacon-enabled PAN receives the beacon request command, it shall ignore the command and continue transmitting its periodic beacons as usual. If a coordinator of a nonbeacon-enabled PAN receives this command, it shall transmit a single beacon frame using unslotted CSMA-CA.

If *macAutoRequest* is set to TRUE, the active scan on a particular channel shall terminate when the number of beacons found equals the implementation-specified limit or the channel has been scanned for the full time, as specified in 7.5.2.1.2. If *macAutoRequest* is set to FALSE, the active scan on a particular channel shall terminate when the channel has been scanned for the full time. If a channel was not scanned for the full time, it shall be considered to be unscanned.

If *macAutoRequest* is set to TRUE, the entire scan procedure shall terminate when the number of PAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned. If *macAutoRequest* is set to FALSE, the entire scan procedure shall only terminate when every channel in the set of available channels has been scanned.

7.5.2.1.3 Passive channel scan

A passive scan, like an active scan, allows a device to locate any coordinator transmitting beacon frames within its POS. The beacon request command, however, is not transmitted. This type of scan could be used by a device prior to association.

During a passive scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames. If a beacon frame is received that contains the address of the scanning device in its list of pending addresses, the scanning device shall not attempt to extract the pending data.

Before commencing a passive scan, the MAC sublayer shall store the value of *macPANId* and then set it to 0xffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current PAN (see 7.5.6.2). On completion of the scan, the MAC sublayer shall restore the value of *macPANId* to the value stored before the scan began.

A passive scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate a passive scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and then enable its receiver for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of the

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

ScanDuration parameter. During this time, the device shall reject all nonbeacon frames and record the information contained in all unique beacons in a PAN descriptor structure (see Table 55 in 7.1.5.1.1). If a beacon frame is received when *macAutoRequest* is set to TRUE, the list of PAN descriptor structures shall be stored by the MAC sublayer until the scan is complete; at this time, the list shall be sent to the next higher layer in the PANDescriptorList parameter of the MLME-SCAN.confirm primitive. A device shall be able to store between one and an implementation- specified maximum number of PAN descriptors. A beacon frame shall be assumed to be unique if it contains both a PAN identifier and a source address that has not been seen before during the scan of the current channel. If a beacon frame is received when *macAutoRequest* is set to FALSE, each recorded PAN descriptor is sent to the next higher layer in a separate MLME-BEACON-NOTIFY.indication primitive. Once the scan is complete, the MLME-SCAN.confirm shall be issued to the next higher layer with a null PANDescriptorList. A received beacon frame containing one or more octets of payload shall also cause the PAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY.indication primitive.

If a protected beacon frame is received (i.e., the Security Enabled subfield in the Frame Control field is set to one), the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.5.8.2.3.

The security-related elements of the PAN descriptor corresponding to the beacon (see Table 55) shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the PAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

The information from the unsecured frame shall be recorded in the PAN descriptor even if the status from the unsecuring process indicated an error.

If *macAutoRequest* is set to TRUE, the passive scan on a particular channel shall terminate when the number of beacons found equals the implementation specified limit or the channel has been scanned for the full time, as specified in 7.5.2.1.3. If *macAutoRequest* is set to FALSE, the passive scan on a particular channel shall terminate when the channel has been scanned for the full time. If a channel was not scanned for the full time, it shall be considered to be unscanned.

If *macAutoRequest* is set to TRUE, the entire scan procedure shall terminate when the number of PAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned. If *macAutoRequest* is set to FALSE, the entire scan procedure shall terminate only when every channel in the set of available channels has been scanned.

7.5.2.1.4 Orphan channel scan

An orphan scan allows a device to attempt to relocate its coordinator following a loss of synchronization. During an orphan scan, the MAC sublayer shall discard all frames received over the PHY data service that are not coordinator realignment command frames.

An orphan scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an orphan scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and then send an orphan notification command (see 7.3.6). Upon successful transmission of the orphan notification command, the device shall enable its receiver for at most *macResponseWaitTime* symbols. If the device successfully receives a coordinator realignment command (see 7.3.8) within this time, the device shall terminate the scan.

If a coordinator receives the orphan notification command, the MLME shall send the MLME-ORPHAN.indication primitive to the next higher layer. The next higher layer should then search its device list for the device indicated by the primitive. If the next higher layer finds a record of the device, it should

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

send a coordinator realignment command to the orphaned device using the MLME-ORPHAN.response primitive. The process of searching for the device and sending the coordinator realignment command shall occur within *macResponseWaitTime* symbols. The coordinator realignment command shall contain its current PAN identifier, *macPANId*, its current logical channel and channel page, and the 16-bit short address of the orphaned device. If the next higher layer of the coordinator finds no record of the device, it should ignore the MLME-ORPHAN.indication primitive and not send the MLME-ORPHAN.response primitive; therefore, the MLME shall not send a coordinator realignment command.

The orphan scan shall terminate when the device receives a coordinator realignment command or the specified set of logical channels has been scanned.

7.5.2.2 PAN identifier conflict resolution

In some instances a situation could occur in which two PANs exist in the same POS with the same PAN identifier. If this conflict happens, the PAN coordinator and its devices shall perform the PAN identifier conflict resolution procedure.

This procedure is optional for an RFD.

7.5.2.2.1 Detection

The PAN coordinator shall conclude that a PAN identifier conflict is present if either of the following apply:

— A beacon frame is received by the PAN coordinator with the PAN Coordinator subfield (see 7.2.2.1.2) set to one and the PAN identifier equal to *macPANId*.

— A PAN ID conflict notification command (see 7.3.5) is received by the PAN coordinator from a device associated with it on its PAN.

A device that is associated through the PAN coordinator (i.e., *macAssociatedPANCoord* is set to TRUE) shall conclude that a PAN identifier conflict is present if the following applies:

— A beacon frame is received by the device with the PAN Coordinator subfield set to one, the PAN identifier equal to *macPANId*, and an address that is equal to neither *macCoordShortAddress* nor *macCoordExtendedAddress*.

A device that is associated through a coordinator that is not the PAN coordinator shall not be capable of detecting a PAN identifier conflict.

7.5.2.2.2 Resolution

On the detection of a PAN identifier conflict by a device, it shall generate the PAN ID conflict notification command (see 7.3.5) and send it to its PAN coordinator. Since the PAN ID conflict notification command contains an acknowledgment request (see 7.3.3.1), the PAN coordinator shall confirm its receipt by sending an acknowledgment frame. Once the device has received the acknowledgment frame from the PAN coordinator, the MLME shall issue an MLME-SYNC-LOSS.indication primitive with the LossReason parameter set to PAN_ID_CONFLICT. If the device does not receive an acknowledgment frame, the MLME shall not inform the next higher layer of the PAN identifier conflict.

On the detection of a PAN identifier conflict by the PAN coordinator, the MLME shall issue an MLME-SYNC-LOSS.indication to the next higher layer with the LossReason parameter set to PAN_ID_CONFLICT. The next higher layer of the PAN coordinator may then perform an active scan and, using the information from the scan, select a new PAN identifier. The algorithm for selecting a suitable PAN identifier is out of the scope of this standard. If the next higher layer does select a new PAN identifier, it may then issue an MLME-START.request with the CoordRealignment parameter set to TRUE in order to realign the PAN, as described in 7.5.2.3.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.2.3 Starting and realigning a PAN

This subclause specifies procedures for the PAN coordinator starting a PAN, coordinators realigning a PAN, and devices being realigned on a PAN.

7.5.2.3.1 Starting a PAN

A PAN should be started by an FFD only after having first performed a MAC sublayer reset, by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE, an active channel scan, and a suitable PAN identifier selection. The algorithm for selecting a suitable PAN identifier from the list of PAN descriptors returned from the active channel scan procedure is out of the scope of this standard. In addition, an FFD should set *macShortAddress* to a value less than 0xffff.

An FFD is instructed to begin operating a PAN through the use of the MLME-START.request primitive (see 7.1.14.1) with the PANCoordinator parameter set to TRUE and the CoordRealignment parameter set to FALSE. On receipt of this primitive, the MAC sublayer shall update the superframe configuration and channel parameters as specified in 7.5.2.3.4. After completing this, the MAC sublayer shall issue the MLME-START.confirm primitive with a status of SUCCESS and begin operating as the PAN coordinator.

7.5.2.3.2 Realigning a PAN

If a coordinator receives the MLME-START.request primitive (see 7.1.14.1) with the CoordRealignment parameter set to TRUE, the coordinator shall attempt to transmit a coordinator realignment command containing the new parameters for PANId, LogicalChannel, and, if present, ChannelPage.

When the coordinator is already transmitting beacons and the CoordRealignment parameter is set to TRUE, the next scheduled beacon shall be transmitted on the current channel using the current superframe configuration, with the Frame Pending subfield of the Frame Control field set to one. Immediately following the transmission of the beacon, the coordinator realignment command shall also be transmitted on the current channel using CSMA-CA.

When the coordinator is not already transmitting beacons and the CoordRealignment parameter is set to TRUE, the coordinator realignment command shall be transmitted immediately on the current channel using CSMA-CA.

If the transmission of the coordinator realignment command fails due to a channel access failure, the MLME shall notify the next higher layer by issuing the MLME-START.confirm primitive with a status of CHANNEL_ACCESS_FAILURE. The next higher layer may then choose to issue the MLME-START.request primitive again.

Upon successful transmission of the coordinator realignment command, the new superframe configuration and channel parameters shall be put into operation as described in 7.5.2.3.4 at the subsequent scheduled beacon, or immediately if the coordinator is not already transmitting beacons, and the MAC sublayer shall issue the MLME-START.confirm primitive with a status of SUCCESS.

7.5.2.3.3 Realignment in a PAN

If a device has received the coordinator realignment command (see 7.3.8) from the coordinator through which it is associated and the MLME was not carrying out an orphan scan, the MLME shall issue the MLME-SYNC-LOSS.indication primitive with the LossReason parameter set to REALIGNMENT and the PANId, LogicalChannel, ChannelPage, and the security-related parameters set to the respective fields in the coordinator realignment command. The next higher layer of a coordinator may then issue an MLME-START.request primitive with the CoordRealignment parameter set to TRUE. The next higher layer of a

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

device that is not a coordinator may instead change the superframe configuration or channel parameters through use of the MLME-SET.request primitive.

7.5.2.3.4 Updating superframe configuration and channel PIB attributes

To update the superframe configuration and channel attributes, the MLME shall assign values from the MLME-START.request primitive parameters to the appropriate PIB attributes. The MLME shall set *macBeaconOrder* to the value of the BeaconOrder parameter. If *macBeaconOrder* is equal to 15, the MLME will also set *macSuperframeOrder* to 15. In this case, this primitive configures a nonbeacon-enabled PAN. If *macBeaconOrder* is less than 15, the MAC sublayer will set *macSuperframeOrder* to the value of the SuperframeOrder parameter. The MAC sublayer shall also update *macPANID* with the value of the PANId parameter and update *phyCurrentPage* and *phyCurrentChannel* with the values of the ChannelPage and LogicalChannel parameters, respectively, by twice issuing the PLME-SET.request primitive.

7.5.2.4 Beacon generation

A device shall be permitted to transmit beacon frames only if *macShortAddress* is not equal to 0xffff.

An FFD shall use the MLME-START.request primitive to begin transmitting beacons only if the BeaconOrder parameter is less than 15. The FFD may begin beacon transmission either as the PAN coordinator of a new PAN or as a device on a previously established PAN, depending upon the setting of the PANCoordinator parameter (see 7.1.14.1). The FFD shall begin beacon transmission on a previously established PAN only once it has successfully associated with that PAN.

If the FFD is the PAN coordinator (i.e., the PANCoordinator parameter is set to TRUE), the MAC sublayer shall ignore the StartTime parameter and begin beacon transmissions immediately. Setting the StartTime parameter to zero shall also cause the MAC sublayer to begin beacon transmissions immediately. If the FFD is not the PAN coordinator and the StartTime parameter is nonzero, the time to begin beacon transmissions shall be calculated using the following method. The StartTime parameter, which is rounded to a backoff slot boundary, shall be added to the time, obtained from the local clock, when the MAC sublayer receives the beacon of the coordinator through which it is associated. The MAC sublayer shall then begin beacon transmissions when the current time, obtained from the local clock, equals the number of calculated symbols. In order for the beacon transmission time to be calculated by the MAC sublayer, the MAC sublayer shall first track the beacon of the coordinator through which it is associated. If the MLME-START.request primitive is issued with a nonzero StartTime parameter and the MAC sublayer is not currently tracking the beacon of its coordinator, the MLME shall not begin beacon transmissions but shall instead issue the MLME-START.confirm primitive with a status of TRACKING_OFF.

If a device misses between one and (*aMaxLostBeacons*–1) consecutive beacon frames from its coordinator, the device shall continue to transmit its own beacons based on both *macBeaconOrder* (see 7.5.2.3.4) and its local clock. If the device then receives a beacon frame from its coordinator and, therefore, does not lose synchronization, the device shall resume transmitting its own beacons based on the StartTime parameter and the incoming beacon. If a device does lose synchronization with its coordinator, the MLME of the device shall issue the MLME-SYNC-LOSS.indication primitive to the next higher layer and immediately stop transmitting its own beacons. The next higher layer may, at any time following the reception of the MLME-SYNC-LOSS.indication primitive, resume beacon transmissions by issuing a new MLME-START.request primitive.

On receipt of the MLME-START.request primitive, the MAC sublayer shall set the PAN identifier in *macPANId* and use this value in the Source PAN Identifier field of the beacon frame. The address used in the Source Address field of the beacon frame shall contain the value of *aExtendedAddress* if *macShortAddress* is equal to 0xfffe or *macShortAddress* otherwise.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The time of transmission of the most recent beacon shall be recorded in *macBeaconTxTime* and shall be computed so that its value is taken at the same symbol boundary in each beacon frame, the location of which is implementation specific. The symbol boundary, which is specified by the *macSyncSymbolOffset* attribute, is the same as that used in the timestamp of the incoming beacon frame, as described in 7.5.4.1.

All beacon frames shall be transmitted at the beginning of each superframe at an interval equal to *aBaseSuperframeDuration* * 2^n symbols, where *n* is the value of *macBeaconOrder* (the construction of the beacon frame is specified in 7.2.2.1).

Beacon transmissions shall be given priority over all other transmit and receive operations.

7.5.2.5 Device discovery

The PAN coordinator or a coordinator indicates its presence on a PAN to other devices by transmitting beacon frames. This allows the other devices to perform device discovery.

A coordinator that is not the PAN coordinator shall begin transmitting beacon frames only when it has successfully associated with a PAN. The transmission of beacon frames by the device is initiated through the use of the MLME-START.request primitive with the PANCoordinator parameter set to FALSE. On receipt of this primitive, the MLME shall begin transmitting beacons based on the StartTime parameter (see 7.5.2.4) using the identifier of the PAN with which the device has associated, *macPANId*, and its extended address, *aExtendedAddress*, if *macShortAddress* is equal to 0xfffe, or its short address, *macShortAddress*, otherwise. A beacon frame shall be transmitted at a rate of one beacon frame every *aBaseSuperframeDuration* * 2^n symbols, where *n* is the value of *macBeaconOrder*.

7.5.3 Association and disassociation

This subclause specifies the procedures for association and disassociation.

7.5.3.1 Association

A device shall attempt to associate only after having first performed a MAC sublayer reset, by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE, and then having completed either an active channel scan (see 7.5.2.1.2) or a passive channel scan (see 7.5.2.1.3). The results of the channel scan would have then been used to choose a suitable PAN. The algorithm for selecting a suitable PAN with which to associate from the list of PAN descriptors returned from the channel scan procedure is out of the scope of this standard.

Following the selection of a PAN with which to associate, the next higher layers shall request through the MLME-ASSOCIATE.request primitive that the MLME configures the following PHY and MAC PIB attributes to the values necessary for association:

- *phyCurrentChannel* shall be set equal to the LogicalChannel parameter of the MLME-ASSOCIATE.request primitive.
- *phyCurrentPage* shall be set equal to the ChannelPage parameter of the MLME-ASSOCIATE.request primitive.
- *macPANId* shall be set equal to the CoordPANId parameter of the MLME-ASSOCIATE.request primitive.
- *macCoordExtendedAddress* or *macCoordShortAddress*, depending on which is known from the beacon frame from the coordinator through which it wishes to associate, shall be set equal to the CoordAddress parameter of the MLME-ASSOCIATE.request primitive.

A coordinator shall allow association only if *macAssociationPermit* is set to TRUE. Similarly, a device should attempt to associate only with a PAN through a coordinator that is currently allowing association, as

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

indicated in the results of the scanning procedure. If a coordinator with *macAssociationPermit* set to FALSE receives an association request command from a device, the command shall be ignored.

In order to optimize the association procedure on a beacon-enabled PAN, a device may begin tracking the beacon of the coordinator through which it wishes to associate a priori. This is achieved by the next higher layer issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE.

A device that is instructed to associate with a PAN, through the MLME-ASSOCIATE.request primitive, shall try to associate only with an existing PAN and shall not attempt to start its own PAN.

The MAC sublayer of an unassociated device shall initiate the association procedure by sending an association request command (see 7.3.1) to the coordinator of an existing PAN; if the association request command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer. Because the association request command contains an acknowledgment request (see 7.3.1.1), the coordinator shall confirm its receipt by sending an acknowledgment frame.

The acknowledgment to an association request command does not mean that the device has associated. The next higher layer of the coordinator needs time to determine whether the current resources available on the PAN are sufficient to allow another device to associate. The next higher layer should make this decision within *macResponseWaitTime* symbols. If the next higher layer of the coordinator finds that the device was previously associated on its PAN, all previously obtained device-specific information should be removed. If sufficient resources are available, the next higher layer should allocate a 16-bit short address to the device, and the MAC sublayer shall generate an association response command (see 7.3.2) containing the new address and a status indicating a successful association. If sufficient resources are not available, the next higher layer of the coordinator should inform the MAC sublayer, and the MLME shall generate an association response command containing a status indicating a failure (see Table 83). The association response command shall be sent to the device requesting association using indirect transmission, i.e., the association response command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3.

If the Allocate Address subfield of the Capability Information field (see 7.3.1.2) of the association request command is set to one, the next higher layer of the coordinator shall allocate a 16-bit address with a range depending on the addressing mode supported by the coordinator, as described in Table 87. If the Allocate Address subfield of the association request command is set to zero, the 16-bit short address shall be equal to 0xfffe. A short address of 0xfffe is a special case that indicates that the device has associated, but has not been allocated a short address by the coordinator. In this case, the device shall use only its 64-bit extended address to operate on the network.

On receipt of the acknowledgment to the association request command, the device shall wait for at most *macResponseWaitTime* symbols for the coordinator to make its association decision; the PIB attribute *macResponseWaitTime* is a network-topology-dependent parameter and may be set to match the specific requirements of the network that a device is trying to join. If the device is tracking the beacon, it shall attempt to extract the association response command from the coordinator whenever it is indicated in the beacon frame. If the device is not tracking the beacon, it shall attempt to extract the association response command from the coordinator after *macResponseWaitTime* symbols. If the device does not extract an association response command frame from the coordinator within *macResponseWaitTime* symbols, the MLME shall issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA, and the association attempt shall be deemed a failure. In this case, the next higher layer shall terminate any tracking of the beacon. This is achieved by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to FALSE.

Because the association response command contains an acknowledgment request (see 7.3.2.1), the device requesting association shall confirm its receipt by sending an acknowledgment frame. If the Association Status field of the command indicates that the association was successful, the device shall store the address

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

contained in the 16-bit Short Address field of the command in *macShortAddress*; communication on the PAN using this short address shall depend on its range, as described in Table 87. If the original beacon selected for association following a scan contained the short address of the coordinator, the extended address of the coordinator, contained in the MHR of the association response command frame, shall be stored in *macCoordExtendedAddress.*

Table 87—Usage of the 16-bit short address

Value of *macShortAddress*	Description
0x0000–0xfffd	If a source address is included, the device shall use short source addressing mode for beacon and data frames and the appropriate source addressing mode specified in 7.3 for MAC command frames.
0xfffe	If a source address is included, the device shall use extended source addressing mode for beacon and data frames and the appropriate source addressing mode specified in 7.3 for MAC command frames.
0xffff	The device is not associated and, therefore, shall not perform any data frame communication. The device shall use the appropriate source addressing mode specified in 7.3 for MAC command frames.

If the Association Status field of the command indicates that the association was unsuccessful, the device shall set *macPANId* to the default value (0xffff).

7.5.3.2 Disassociation

The disassociation procedure is initiated by the next higher layer by issuing the MLME-DISASSOCIATE.request primitive to the MLME.

When a coordinator wants one of its associated devices to leave the PAN, the MLME of the coordinator shall send the disassociation notification command in the manner specified by the TxIndirect parameter of the MLME-DISASSOCIATE.request primitive previously sent by the next higher layer. If TxIndirect is TRUE, the MLME of the coordinator shall send the disassociation notification command to the device using indirect transmission, i.e., the disassociation notification command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 7.5.6.3. If the command frame is not successfully extracted by the device, the coordinator should consider the device disassociated. Otherwise, the MLME shall send the disassociation notification command to the device directly. In this case, if the disassociation notification command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer.

Because the disassociation command contains an acknowledgment request (see 7.3.3.1), the receiving device shall confirm its receipt by sending an acknowledgment frame. If the direct or indirect transmission fails, the coordinator should consider the device disassociated.

If an associated device wants to leave the PAN, the MLME of the device shall send a disassociation notification command to its coordinator. If the disassociation notification command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer. Because the disassociation command contains an acknowledgment request (see 7.3.3.1), the coordinator shall confirm its receipt by sending an acknowledgment frame. However, even if the acknowledgment is not received, the device should consider itself disassociated.

If the source address contained in the disassociation notification command is equal to *macCoord-ExtendedAddress*, the device should consider itself disassociated. If the command is received by a coordinator and the source is not equal to *macCoordExtendedAddress*, it shall verify that the source address

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

corresponds to one of its associated devices; if so, the coordinator should consider the device disassociated. If none of the above conditions is satisfied, the command shall be ignored.

An associated device shall disassociate itself by removing all references to the PAN; the MLME shall set *macPANId*, *macShortAddress*, *macAssociatedPANCoord*, *macCoordShortAddress* and *macCoordExtended-Address* to the default values. The next higher layer of a coordinator should disassociate a device by removing all references to that device.

The next higher layer of the requesting device shall be notified of the result of the disassociation procedure through the MLME-DISASSOCIATE.confirm primitive.

7.5.4 Synchronization

This subclause specifies the procedures for coordinators to generate beacon frames and for devices to synchronize with a coordinator. For PANs supporting beacons, synchronization is performed by receiving and decoding the beacon frames. For PANs not supporting beacons, synchronization is performed by polling the coordinator for data.

7.5.4.1 Synchronization with beacons

All devices operating on a beacon-enabled PAN (i.e., *macBeaconOrder* < 15) shall be able to acquire beacon sychronization in order to detect any pending messages or to track the beacon. Devices shall be permitted to acquire beacon synchronization only with beacons containing the PAN identifier specified in *macPANId*. If *macPANId* specifies the broadcast PAN identifier (0xffff), a device shall not attempt to acquire beacon synchronization.

A device is instructed to attempt to acquire the beacon through the MLME-SYNC.request primitive. If tracking is specified in the MLME-SYNC.request primitive, the device shall attempt to acquire the beacon and keep track of it by regular and timely activation of its receiver. If tracking is not specified, the device shall either attempt to acquire the beacon only once or terminate the tracking after the next beacon if tracking was enabled through a previous request.

To acquire beacon synchronization, a device shall enable its receiver and search for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of *macBeaconOrder*. If a beacon frame containing the current PAN identifier of the device is not received, the MLME shall repeat this search. Once the number of missed beacons reaches *aMaxLostBeacons*, the MLME shall notify the next higher layer by issuing the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOSS.

The MLME shall timestamp each received beacon frame at the same symbol boundary within each frame, the location of which is described by the *macSyncSymbolOffset* attribute. The symbol boundary shall be the same as that used in the timestamp of the outgoing beacon frame, stored in *macBeaconTxTime*. The timestamp value shall be that of the local clock of the device at the time of the symbol boundary. The timestamp is intended to be a relative time measurement that may or may not be made absolute, at the discretion of the implementer.

If a protected beacon frame is received (i.e., the Security Enabled subfield in the Frame Control field is set to one), the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.5.8.2.3.

If the status from the unsecuring process is not SUCCESS, the MLME shall issue an MLME-COMM-STATUS.indication primitive with the status parameter set to the status from the unsecuring process, indicating the error.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The security-related elements of the PAN descriptor corresponding to the beacon (see Table 55) shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the PAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

If a beacon frame is received, the MLME shall discard the beacon frame if the Source Address and the Source PAN Identifier fields of the MHR of the beacon frame do not match the coordinator source address (*macCoordShortAddress* or *macCoordExtendedAddress*, depending on the addressing mode) and the PAN identifier of the device (*macPANId*).

If a valid beacon frame is received and *macAutoRequest* is set to FALSE, the MLME shall indicate the beacon parameters to the next higher layer by issuing the MLME-BEACON-NOTIFY.indication primitive. If a beacon frame is received and *macAutoRequest* is set to TRUE, the MLME shall first issue the MLME-BEACON-NOTIFY.indication primitive if the beacon contains any payload. The MLME shall then compare its address with those addresses in the Address List field of the beacon frame. If the Address List field contains the 16-bit short or 64-bit extended address of the device and the source PAN identifier matches *macPANId*, the MLME shall follow the procedure for extracting pending data from the coordinator (see 7.5.6.3).

If beacon tracking is activated, the MLME shall enable its receiver at a time prior to the next expected beacon frame transmission, i.e., just before the known start of the next superframe. If the number of consecutive beacons missed by the MLME reaches *aMaxLostBeacons*, the MLME shall respond with the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

7.5.4.2 Synchronization without beacons

All devices operating on a nonbeacon-enabled PAN (*macBeaconOrder* = 15) shall be able to poll the coordinator for data at the discretion of the next higher layer.

A device is instructed to poll the coordinator when the MLME receives the MLME-POLL.request primitive. On receipt of this primitive, the MLME shall follow the procedure for extracting pending data from the coordinator (see 7.5.6.3).

7.5.4.3 Orphaned device realignment

If the next higher layer receives repeated communications failures following its requests to transmit data, it may conclude that it has been orphaned. A single communications failure occurs when a device transaction fails to reach the coordinator, i.e., an acknowledgment is not received after *macMaxFrameRetries* attempts at sending the data. If the next higher layer concludes that it has been orphaned, it may instruct the MLME to either perform the orphaned device realignment procedure, or to reset the MAC sublayer and then perform the association procedure.

If the decision has been made by the next higher layer to perform the orphaned device realignment procedure, it will have issued an MLME-SCAN.request with the ScanType parameter set to orphan scan and the ScanChannel parameter containing the list of channels to be scanned. Upon receiving this primitive, the MAC sublayer shall begin an orphan scan, as described in 7.5.2.1.4.

If the orphan scan is successful (i.e., its PAN has been located), the device shall update its MAC PIB with the PAN information contained in the coordinator realignment command (see 7.3.8).

7.5.5 Transaction handling

Because this standard favors very low cost devices that, in general, will be battery powered, transactions can be instigated from the devices themselves rather than from the coordinator. In other words, either the

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

coordinator needs to indicate in its beacon when messages are pending for devices or the devices themselves need to poll the coordinator to determine whether they have any messages pending. Such transfers are called indirect transmissions.

The coordinator shall begin handling a transaction on receipt of an indirect transmission request either via the MCPS-DATA.request primitive or via a request from the MLME to send a MAC command instigated by a primitive from the next higher layer, such as the MLME-ASSOCIATE.response primitive (see 7.1.3.3). On completion of the transaction, the MAC sublayer shall indicate a status value to the next higher layer. If a request primitive instigated the indirect transmission, the corresponding confirm primitive shall be used to convey the appropriate status value. Conversely, if a response primitive instigated the indirect transmission, the MLME-COMM-STATUS.indication primitive shall be used to convey the appropriate status value. The MLME-COMM-STATUS.indication primitive can be related to its corresponding response primitive by examining the Destination Address field.

The information contained in the indirect transmission request forms a transaction, and the coordinator shall be capable of storing at least one transaction. On receipt of an indirect transmission request, if there is no capacity to store another transaction, the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_OVERFLOW in the appropriate corresponding primitive.

If the coordinator is capable of storing more than one transaction, it shall ensure that all the transactions for the same device are sent in the order in which they arrived at the MAC sublayer. For each transaction sent, if another exists for the same device, the MAC sublayer shall set its Frame Pending subfield to one, indicating the additional pending data.

Each transaction shall persist in the coordinator for at most *macTransactionPersistenceTime*. If the transaction is not successfully extracted by the appropriate device within this time, the transaction information shall be discarded and the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_EXPIRED in the appropriate corresponding primitive. In order to be successfully extracted, an acknowledgment shall be received if one was requested.

If the transaction was successful, the transaction information shall be discarded, and the MAC sublayer shall indicate to the next higher layer a status of SUCCESS in the appropriate corresponding primitive.

If the coordinator transmits beacons, it shall list the addresses of the devices to which each transaction is associated in the Address List field and indicate the number of addresses in the Pending Address Specification field of the beacon frame. If the coordinator is able to store more than seven pending transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses. For transactions requiring a GTS, the PAN coordinator shall not add the address of the recipient to its list of pending addresses in the beacon frame. Instead it shall transmit the transaction in the GTS allocated for the device (see 7.5.7.3).

On a beacon-enabled PAN, if there is a transaction pending for the broadcast address, the Frame Pending subfield of the Frame Control field in the beacon frame shall be set to one, and the pending message shall be transmitted immediately following the beacon using the CSMA-CA algorithm. If there is a second message pending for the broadcast address, its transmission shall be delayed until the following superframe. Only one broadcast message shall be allowed to be sent indirectly per superframe.

On a beacon-enabled PAN, a device that receives a beacon containing its address in the list of pending addresses shall attempt to extract the data from the coordinator. On a nonbeacon-enabled PAN, a device shall attempt to extract the data from the coordinator on receipt of the MLME-POLL.request primitive. The procedure for extracting pending data from the coordinator is described in 7.5.6.3. If a device receives a beacon with the Frame Pending subfield set to one, it shall leave its receiver enabled for up to *macMaxFrameTotalWaitTime* symbols to receive the broadcast data frame from the coordinator.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.6 Transmission, reception, and acknowledgment

This subclause describes the fundamental procedures for transmission, reception, and acknowledgment.

7.5.6.1 Transmission

Each device shall store its current DSN value in the MAC PIB attribute *macDSN* and initialize it to a random value; the algorithm for choosing a random number is out of the scope of this standard. Each time a data or a MAC command frame is generated, the MAC sublayer shall copy the value of *macDSN* into the Sequence Number field of the MHR of the outgoing frame and then increment it by one. Each device shall generate exactly one DSN regardless of the number of unique devices with which it wishes to communicate. The value of *macDSN* shall be permitted to roll over.

Each coordinator shall store its current BSN value in the MAC PIB attribute *macBSN* and initialize it to a random value; the algorithm for choosing a random number is out of the scope of this standard. Each time a beacon frame is generated, the MAC sublayer shall copy the value of *macBSN* into the Sequence Number field of the MHR of the outgoing frame and then increment it by one. The value of *macBSN* shall be permitted to roll over.

It should be noted that both the DSN and BSN are 8-bit values and, therefore, have limited use to the next higher layer (e.g., in the case of the DSN, in detecting retransmitted frames).

The Source Address field, if present, shall contain the address of the device sending the frame. When a device has associated and has been allocated a 16-bit short address (i.e., *macShortAddress* is not equal to 0xfffe or 0xffff), it shall use that address in preference to its 64-bit extended address (i.e., *aExtendedAddress*) wherever possible. When a device has not yet associated to a PAN or *macShortAddress* is equal to 0xffff, it shall use its 64-bit extended address in all communications requiring the Source Address field. If the Source Address field is not present, the originator of the frame shall be assumed to be the PAN coordinator, and the Destination Address field shall contain the address of the recipient.

The Destination Address field, if present, shall contain the address of the intended recipient of the frame, which may be either a 16-bit short address or a 64-bit extended address. If the Destination Address field is not present, the recipient of the frame shall be assumed to be the PAN coordinator, and the Source Address field shall contain the address of the originator.

If both destination and source addressing information is present, the MAC sublayer shall compare the destination and source PAN identifiers. If the PAN identifiers are identical, the PAN ID Compression subfield of the Frame Control field shall be set to one, and the source PAN identifier shall be omitted from the transmitted frame. If the PAN identifiers are different, the PAN ID Compression subfield of the Frame Control field shall be set to zero, and both Destination PAN Identifier and Source PAN Identifier fields shall be included in the transmitted frame. If only either the destination or the source addressing information is present, the PAN ID Compression subfield of the Frame Control field shall be set to zero, and the PAN identifier field of the single address shall be included in the transmitted frame.

If the frame is to be transmitted on a beacon-enabled PAN, the transmitting device shall attempt to find the beacon before transmitting. If the beacon is not being tracked (see 7.5.4.1) and hence the device does not know where the beacon will appear, it shall enable its receiver and search for at most [*aBaseSuperframeDuration* * (2^n + 1)] symbols, where *n* is the value of *macBeaconOrder*, in order to find the beacon. If the beacon is not found after this time, the device shall transmit the frame following the successful application of the unslotted version of the CSMA-CA algorithm (see 7.5.1.4). Once the beacon has been found, either after a search or due to its being tracked, the frame shall be transmitted in the appropriate portion of the superframe. Transmissions in the CAP shall follow a successful application of the slotted version of the CSMA-CA algorithm (see 7.5.1.4), and transmissions in a GTS shall not use CSMA-CA.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the frame is to be transmitted on a nonbeacon-enabled PAN, the frame shall be transmitted following the successful application of the unslotted version of the CSMA-CA algorithm (see 7.5.1.4).

For either a beacon-enabled PAN or a nonbeacon-enabled PAN, if the transmission is direct and originates due to a primitive issued by the next higher layer and the CSMA-CA algorithm fails, the next higher layer shall be notified. If the transmission is indirect and the CSMA-CA algorithm fails, the frame shall remain in the transaction queue until it is requested again and successfully transmitted or until the transaction expires.

The device shall process the frame using the outgoing frame security procedure described in 7.5.8.2.1.

If the status from the outgoing frame security procedure is not SUCCESS, the MLME shall issue the corresponding confirm or MLME-COMM-STATUS.indication primitive with the status parameter set to the status from the outgoing frame security procedure, indicating the error.

To transmit the frame, the MAC sublayer shall first enable the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the constructed frame shall then be transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MAC sublayer shall disable the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON or TRX_OFF to the PHY, depending on whether the receiver is to be enabled following the transmission. In the case where the Acknowledgment Request subfield of the Frame Control field is set to one, the MAC sublayer shall enable the receiver immediately following the transmission of the frame by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY.

7.5.6.2 Reception and rejection

Each device may choose whether the MAC sublayer is to enable its receiver during idle periods. During these idle periods, the MAC sublayer shall still service transceiver task requests from the next higher layer. A transceiver task shall be defined as a transmission request with acknowledgment reception, if required, or a reception request. On completion of each transceiver task, the MAC sublayer shall request that the PHY enables or disables its receiver, depending on the values of *macBeaconOrder* and *macRxOnWhenIdle*. If *macBeaconOrder* is less than 15, the value of *macRxOnWhenIdle* shall be considered relevant only during idle periods of the CAP of the incoming superframe. If *macBeaconOrder* is equal to 15, the value of *macRxOnWhenIdle* shall be considered relevant at all times.

Due to the nature of radio communications, a device with its receiver enabled will be able to receive and decode transmissions from all devices complying with this standard that are currently operating on the same channel and are in its POS, along with interference from other sources. The MAC sublayer shall, therefore, be able to filter incoming frames and present only the frames that are of interest to the upper layers.

For the first level of filtering, the MAC sublayer shall discard all received frames that do not contain a correct value in their FCS field in the MFR (see 7.2.1.9). The FCS field shall be verified on reception by recalculating the purported FCS over the MHR and MAC payload of the received frame and by subsequently comparing this value with the received FCS field. The FCS field of the received frame shall be considered to be correct if these values are the same and incorrect otherwise.

The second level of filtering shall be dependent on whether the MAC sublayer is currently operating in promiscuous mode. In promiscuous mode, the MAC sublayer shall pass all frames received after the first filter directly to the upper layers without applying any more filtering or processing. The MAC sublayer shall be in promiscuous mode if *macPromiscuousMode* is set to TRUE.

If the MAC sublayer is not in promiscuous mode (i.e., *macPromiscuousMode* is set to FALSE), it shall accept only frames that satisfy all of the following third-level filtering requirements:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

— The Frame Type subfield shall not contain a reserved frame type.
— The Frame Version subfield shall not contain a reserved value.
— If a destination PAN identifier is included in the frame, it shall match *macPANId* or shall be the broadcast PAN identifier (0xffff).
— If a short destination address is included in the frame, it shall match either *macShortAddress* or the broadcast address (0xffff). Otherwise, if an extended destination address is included in the frame, it shall match *aExtendedAddress*.
— If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match *macPANId* unless *macPANId* is equal to 0xffff, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
— If only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is the PAN coordinator and the source PAN identifier matches *macPANId*.

If any of the third-level filtering requirements are not satisfied, the MAC sublayer shall discard the incoming frame without processing it further. If all of the third-level filtering requirements are satisfied, the frame shall be considered valid and processed further. For valid frames that are not broadcast, if the Frame Type subfield indicates a data or MAC command frame and the Acknowledgment Request subfield of the Frame Control field is set to one, the MAC sublayer shall send an acknowledgment frame. Prior to the transmission of the acknowledgment frame, the sequence number included in the received data or MAC command frame shall be copied into the Sequence Number field of the acknowledgment frame. This step will allow the transaction originator to know that it has received the appropriate acknowledgment frame.

If the PAN ID Compression subfield of the Frame Control field is set to one and both destination and source addressing information is included in the frame, the MAC sublayer shall assume that the omitted Source PAN Identifier field is identical to the Destination PAN Identifier field.

The device shall process the frame using the incoming frame security procedure described in 7.5.8.2.3.

If the status from the incoming frame security procedure is not SUCCESS, the MLME shall issue the corresponding confirm or MLME-COMM-STATUS.indication primitive with the status parameter set to the status from the incoming frame security procedure, indicating the error, and with the security-related parameters set to the corresponding parameters returned by the unsecuring process.

If the valid frame is a data frame, the MAC sublayer shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive containing the frame information. The security-related parameters of the MCPS-DATA.indication primitive shall be set to the corresponding parameters returned by the unsecuring process.

If the valid frame is a MAC command or beacon frame, it shall be processed by the MAC sublayer accordingly, and a corresponding confirm or indication primitive may be sent to the next higher layer. The security-related parameters of the corresponding confirm or indication primitive shall be set to the corresponding parameters returned by the unsecuring process.

7.5.6.3 Extracting pending data from a coordinator

A device on a beacon-enabled PAN can determine whether any frames are pending for it by examining the contents of the received beacon frame, as described in 7.5.4.1. If the address of the device is contained in the Address List field of the beacon frame and *macAutoRequest* is TRUE, the MLME of the device shall send a data request command (see 7.3.4) to the coordinator during the CAP with the Acknowledgment Request subfield of the Frame Control field set to one; the only exception to this is if the beacon frame is received while performing an active or passive scan (see 7.5.2.1). There are two other cases for which the MLME

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

shall send a data request command to the coordinator. The first case is when the MLME receives the MLME-POLL.request primitive. In the second case, a device may send a data request command *macResponseWaitTime* symbols after the acknowledgment to a request command frame, such as during the association procedure. If the data request is intended for the PAN coordinator, the destination address information may be omitted.

If the data request command originated from an MLME-POLL.request primitive, the MLME shall perform the security process on the data request command based on the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters of the MLME-POLL.request primitive, according to 7.5.8.2.1. Otherwise, the MLME shall perform the security process on the data request command based on the *macAutoRequestSecurityLevel*, *macAutoRequestKeyIdMode*, *macAutoRequestKeySource*, and *macAutoRequestKeyIndex* PIB attributes, according to 7.5.8.2.1.

On successfully receiving a data request command, the coordinator shall send an acknowledgment frame, thus confirming its receipt. If the coordinator has enough time to determine whether the device has a frame pending before sending the acknowledgment frame (see 7.5.6.4.2), it shall set the Frame Pending subfield of the Frame Control field of the acknowledgment frame accordingly to indicate whether a frame is actually pending for the device. If this is not possible, the coordinator shall set the Frame Pending subfield of the acknowledgment frame to one.

On receipt of the acknowledgment frame with the Frame Pending subfield set to zero, the device shall conclude that there are no data pending at the coordinator.

On receipt of the acknowledgment frame with the Frame Pending subfield set to one, a device shall enable its receiver for at most *macMaxFrameTotalWaitTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, to receive the corresponding data frame from the coordinator. If there is an actual data frame pending within the coordinator for the requesting device, the coordinator shall send the frame to the device using one of the mechanisms described in this subclause. If there is no data frame pending for the requesting device, the coordinator shall send a data frame without requesting acknowledgment to the device containing a zero length payload, indicating that no data are present, using one of the mechanisms described in this subclause.

The data frame following the acknowledgment of the data request command shall be transmitted using one of the following mechanisms:

— Without using CSMA-CA, if the MAC sublayer can commence transmission of the data frame between *aTurnaroundTime* and (*aTurnaroundTime* + *aUnitBackoffPeriod*) symbols, on a backoff slot boundary, and there is time remaining in the CAP for the message, appropriate IFS, and acknowledgment (see 6.4.1 for an explanation of *aTurnAroundTime*). If a requested acknowledgment frame is not received following this data frame, the process shall begin anew following the receipt of a new data request command.

— Using CSMA-CA, otherwise.

If the requesting device does not receive a data frame from the coordinator within *macMaxFrameTotalWaitTime* CAP symbols in a beacon-enabled PAN, or symbols in a nonbeacon-enabled PAN, or if the requesting device receives a data frame from the coordinator with a zero length payload, it shall conclude that there are no data pending at the coordinator. If the requesting device does receive a data frame from the coordinator, it shall send an acknowledgment frame, if requested, thus confirming receipt.

If the Frame Pending subfield of the Frame Control field of the data frame received from the coordinator is set to one, the device still has more data pending with the coordinator. In this case it may extract the data by sending a new data request command to the coordinator.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.6.4 Use of acknowledgments and retransmissions

A data or MAC command frame shall be sent with the Acknowledgment Request subfield of its Frame Control field set appropriately for the frame. A beacon or acknowledgment frame shall always be sent with the Acknowledgment Request subfield set to zero. Similarly, any frame that is broadcast shall be sent with its Acknowledgment Request subfield set to zero.

7.5.6.4.1 No acknowledgment

A frame transmitted with its Acknowledgment Request subfield set to zero shall not be acknowledged by its intended recipient. The originating device shall assume that the transmission of the frame was successful.

The message sequence chart in Figure 70 shows the scenario for transmitting a single frame of data from an originator to a recipient without requiring an acknowledgment. In this case, the originator transmits the data frame with the Acknowledgment Request (AR) subfield of the Frame Control field equal to zero.

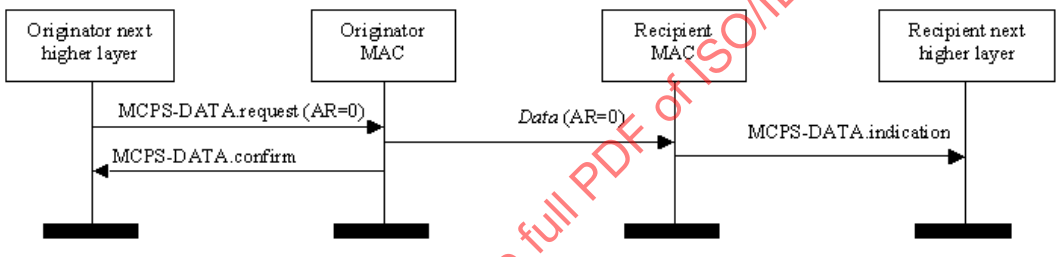


Figure 70—Successful data transmission without an acknowledgment

7.5.6.4.2 Acknowledgment

A frame transmitted with the Acknowledgment Request subfield of its Frame Control field set to one shall be acknowledged by the recipient. If the intended recipient correctly receives the frame, it shall generate and send an acknowledgment frame containing the same DSN from the data or MAC command frame that is being acknowledged.

The transmission of an acknowledgment frame in a nonbeacon-enabled PAN or in the CFP shall commence *aTurnaroundTime* symbols after the reception of the last symbol of the data or MAC command frame. The transmission of an acknowledgment frame in the CAP shall commence either *aTurnaroundTime* symbols after the reception of the last symbol of the data or MAC command frame or at a backoff slot boundary. In the latter case, the transmission of an acknowledgment frame shall commence between *aTurnaroundTime* and (*aTurnaroundTime* + *aUnitBackoffPeriod*) symbols after the reception of the last symbol of the data or MAC command frame. The constant *aTurnaroundTime* is defined in Table 22 (see 6.4.1).

The message sequence chart in Figure 71 shows the scenario for transmitting a single frame of data from an originator to a recipient with an acknowledgment. In this case, the originator indicates to the recipient that it requires an acknowledgment by transmitting the data frame with the Acknowledgment Request (AR) subfield of the Frame Control field set to one.

7.5.6.4.3 Retransmissions

A device that sends a frame with the Acknowledgment Request subfield of its Frame Control field set to zero shall assume that the transmission was successfully received and shall hence not perform the retransmission procedure.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

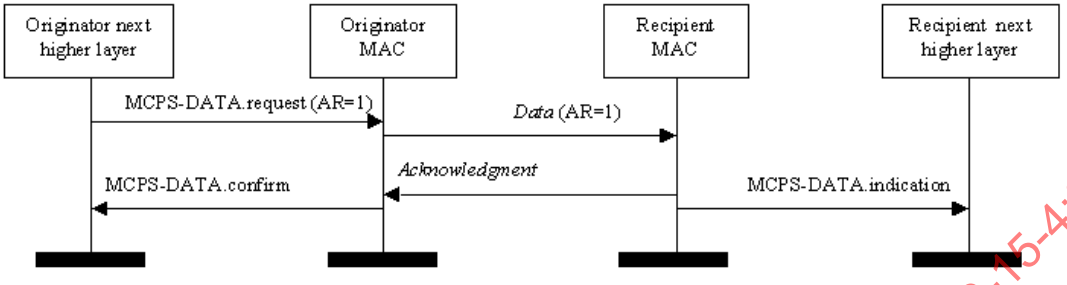


Figure 71—Successful data transmission with an acknowledgment

A device that sends a data or MAC command frame with its Acknowledgment Request subfield set to one shall wait for at most *macAckWaitDuration* symbols for the corresponding acknowledgment frame to be received. If an acknowledgment frame is received within *macAckWaitDuration* symbols and contains the same DSN as the original transmission, the transmission is considered successful, and no further action regarding retransmission shall be taken by the device. If an acknowledgment is not received within *macAckWaitDuration* symbols or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

If a single transmission attempt has failed and the transmission was indirect, the coordinator shall not retransmit the data or MAC command frame. Instead, the frame shall remain in the transaction queue of the coordinator and can only be extracted following the reception of a new data request command. If a new data request command is received, the originating device shall transmit the frame using the same DSN as was used in the original transmission.

If a single transmission attempt has failed and the transmission was direct, the device shall repeat the process of transmitting the data or MAC command frame and waiting for the acknowledgment, up to a maximum of *macMaxFrameRetries* times. The retransmitted frame shall contain the same DSN as was used in the original transmission. Each retransmission shall only be attempted if it can be completed within the same portion of the superframe, i.e., the CAP or a GTS in which the original transmission was attempted. If this timing is not possible, the retransmission shall be deferred until the same portion in the next superframe. If an acknowledgment is still not received after *macMaxFrameRetries* retransmissions, the MAC sublayer shall assume the transmission has failed and notify the next higher layer of the failure.

7.5.6.5 Promiscuous mode

A device may activate promiscuous mode by setting *macPromiscuousMode*. If the MLME is requested to set *macPromiscuousMode* to TRUE, the MLME shall then request that the PHY enable its receiver. This request is achieved when the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of RX_ON.

When in promiscuous mode, the MAC sublayer shall process received frames according to 7.5.6.2 and pass all frames correctly received to the next higher layer using the MCPS-DATA.indication primitive. The source and destination addressing mode parameters shall each be set to 0x00, the MSDU parameter shall contain the MHR concatenated with the MAC payload (see Figure 41), and the msduLength parameter shall contain the total number of octets in the MHR concatenated with the MAC payload. The mpduLinkQuality shall be valid.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If the MLME is requested to set *macPromiscuousMode* to FALSE, the MLME shall request that the PHY set its receiver to the state specified by *macRxOnWhenIdle*. This is achieved by the MLME issuing the PLME-SET-TRX-STATE.request primitive (see 6.2.2.7) with the state set accordingly.

7.5.6.6 Transmission scenarios

Due to the imperfect nature of the radio medium, a transmitted frame does not always reach its intended destination. Figure 72 illustrates three different data transmission scenarios:

— *Successful data transmission.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the data frame to the next higher layer. The originator MAC sublayer receives the acknowledgment from the recipient before its timer expires and then disables and resets the timer. The data transfer is now complete, and the originator MAC sublayer issues a success confirmation to the next higher layer.

— *Lost data frame.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer does not receive the data frame and so does not respond with an acknowledgment. The timer of the originator MAC sublayer expires before an acknowledgment is received; therefore, the data transfer has failed. If the transmission was direct, the originator retransmits the data, and this entire sequence may be repeated up to a maximum of *macMaxFrameRetries* times; if a data transfer attempt fails a total of (1 + *macMaxFrameRetries*) times, the originator MAC sublayer will issue a failure confirmation to the next higher layer. If the transmission was indirect, the data frame will remain in the transaction queue until either another request for the data is received and correctly acknowledged or until *macTransactionPersistenceTime* is reached. If *macTransactionPersistenceTime* is reached, the transaction information will be discarded, and the MAC sublayer will issue a failure confirmation to the next higher layer.

— *Lost acknowledgment frame.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* symbols. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the data frame to the next higher layer. The originator MAC sublayer does not receive the acknowledgment frame, and its timer expires. Therefore, the data transfer has failed. If the transmission was direct, the originator retransmits the data, and this entire sequence may be repeated up to a maximum of *macMaxFrameRetries* times. If a data transfer attempt fails a total of (1 + *macMaxFrameRetries*) times, the originator MAC sublayer will issue a failure confirmation to the next higher layer. If the transmission was indirect, the data frame will remain in the transaction queue either until another request for the data is received and correctly acknowledged or until *macTransactionPersistenceTime* is reached. If *macTransactionPersistenceTime* is reached, the transaction information will be discarded, and the MAC sublayer will issue a failure confirmation to the next higher layer.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

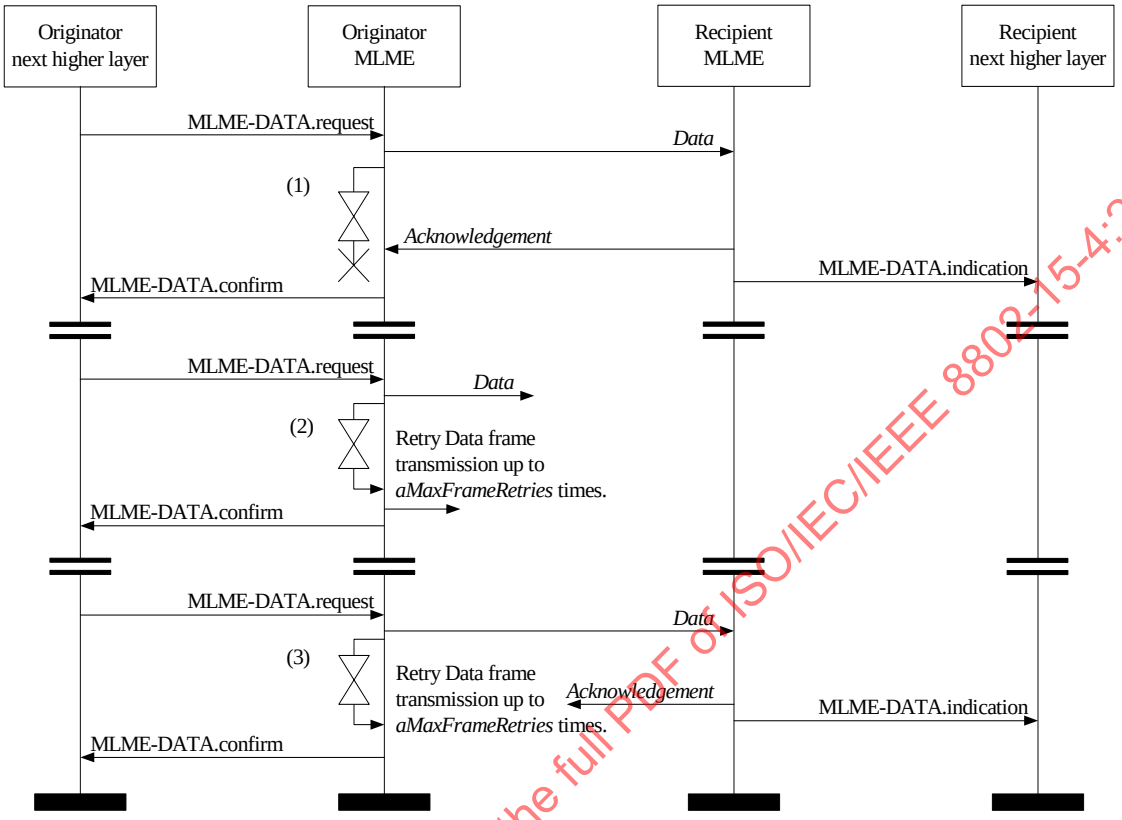


Figure 72—Transmission scenarios, using direct transmission, for frame reliability

7.5.7 GTS allocation and management

A GTS allows a device to operate on the channel within a portion of the superframe that is dedicated (on the PAN) exclusively to that device. A GTS shall be allocated only by the PAN coordinator, and it shall be used only for communications between the PAN coordinator and a device associated with the PAN through the PAN coordinator. A single GTS may extend over one or more superframe slots. The PAN coordinator may allocate up to seven GTSs at the same time, provided there is sufficient capacity in the superframe.

A GTS shall be allocated before use, with the PAN coordinator deciding whether to allocate a GTS based on the requirements of the GTS request and the current available capacity in the superframe. GTSs shall be allocated on a first-come-first-served basis, and all GTSs shall be placed contiguously at the end of the superframe and after the CAP. Each GTS shall be deallocated when the GTS is no longer required, and a GTS can be deallocated at any time at the discretion of the PAN coordinator or by the device that originally requested the GTS. A device that has been allocated a GTS may also operate in the CAP.

A data frame transmitted in an allocated GTS shall use only short addressing.

The management of GTSs shall be undertaken by the PAN coordinator only. To facilitate GTS management, the PAN coordinator shall be able to store all the information necessary to manage seven GTSs. For each GTS, the PAN coordinator shall be able to store its starting slot, length, direction, and associated device address.

The GTS direction, which is relative to the data flow from the device that owns the GTS, is specified as either transmit or receive. The device address and direction shall, therefore, uniquely identify each GTS.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Each device may request one transmit GTS and/or one receive GTS. For each allocated GTS, the device shall be able to store its starting slot, length, and direction. If a device has been allocated a receive GTS, it shall enable its receiver for the entirety of the GTS. In the same way, the PAN coordinator shall enable its receiver for the entirety of the GTS if a device has been allocated a transmit GTS. If a data frame is received during a receive GTS and an acknowledgment is requested, the device shall transmit the acknowledgment frame as usual. Similarly, a device shall be able to receive an acknowledgment frame during a transmit GTS.

A device shall attempt to allocate and use a GTS only if it is currently tracking the beacons. The MLME is instructed to track beacons by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE. If a device loses synchronization with the PAN coordinator, all its GTS allocations shall be lost.

The use of GTSs is optional.

7.5.7.1 CAP maintenance

The PAN coordinator shall preserve the minimum CAP length of *aMinCAPLength* and take preventative action if the minimum CAP is not satisfied. However, an exception shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance. If preventative action becomes necessary, the action chosen is left up to the implementation, but may include one or more of the following:

— Limiting the number of pending addresses included in the beacon.
— Not including a payload field in the beacon frame.
— Deallocating one or more of the GTSs

7.5.7.2 GTS allocation

A device is instructed to request the allocation of a new GTS through the MLME-GTS.request primitive, with GTS characteristics set according to the requirements of the intended application.

To request the allocation of a new GTS, the MLME shall send the GTS request command (see 7.3.9) to the PAN coordinator. The Characteristics Type subfield of the GTS Characteristics field of the request shall be set to one (GTS allocation), and the length and direction subfields shall be set according to the desired characteristics of the required GTS. Because the GTS request command contains an acknowledgment request (see 7.3.3.1), the PAN coordinator shall confirm its receipt by sending an acknowledgment frame.

On receipt of a GTS request command indicating a GTS allocation request, the PAN coordinator shall first check if there is available capacity in the current superframe, based on the remaining length of the CAP and the desired length of the requested GTS. The superframe shall have available capacity if the maximum number of GTSs has not been reached and allocating a GTS of the desired length would not reduce the length of the CAP to less than *aMinCAPLength*. GTSs shall be allocated on a first-come-first-served basis by the PAN coordinator provided there is sufficient bandwidth available. The PAN coordinator shall make this decision within *aGTSDescPersistenceTime* superframes.

On receipt of the acknowledgment to the GTS request command, the device shall continue to track beacons and wait for at most *aGTSDescPersistenceTime* superframes. If no GTS descriptor for the device appears in the beacon within this time, the MLME of the device shall notify the next higher layer of the failure. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive (see 7.1.7.2) with a status of NO_DATA.

When the PAN coordinator determines whether capacity is available for the requested GTS, it shall generate a GTS descriptor with the requested specifications and the 16-bit short address of the requesting device. If the GTS was allocated successfully, the PAN coordinator shall set the start slot in the GTS descriptor to the

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

superframe slot at which the GTS begins and the length in the GTS descriptor to the length of the GTS. In addition, the PAN coordinator shall notify the next higher layer of the new GTS. This notification is achieved when the MLME of the PAN coordinator issues the MLME-GTS.indication primitive (see 7.1.7.3) with the characteristics of the allocated GTS. If there was not sufficient capacity to allocate the requested GTS, the start slot shall be set to zero and the length to the largest GTS length that can currently be supported. The PAN coordinator shall then include this GTS descriptor in its beacon and update the GTS Specification field of the beacon frame accordingly. The PAN coordinator shall also update the Final CAP Slot subfield of the Superframe Specification field of the beacon frame, indicating the final superframe slot utilized by the decreased CAP. The GTS descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes, after which it shall be removed automatically. The PAN coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress*, the device shall process the descriptor. The MLME of the device shall then notify the next higher layer of whether the GTS allocation request was successful. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive with a status of SUCCESS (if the start slot in the GTS descriptor was greater than zero) or DENIED (if the start slot was equal to zero or if the length did not match the requested length).

7.5.7.3 GTS usage

When the MAC sublayer of a device that is not the PAN coordinator receives an MCPS-DATA.request primitive (see 7.1.1.1) with the TxOptions parameter indicating a GTS transmission, it shall determine whether it has a valid transmit GTS. If a valid GTS is found, the MAC sublayer shall transmit the data during the GTS, i.e., between its starting slot and its starting slot plus its length. At this time, the MAC sublayer shall transmit the MPDU immediately without using CSMA-CA, provided the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer shall defer the transmission until the specified GTS in the next superframe. Note that the MAC must allow for the PHY overhead (see 6.7.2.2) in making this determination.

If the device has any receive GTSs, the MAC sublayer of the device shall ensure that the receiver is enabled at a time prior to the start of the GTS and for the duration of the GTS, as indicated by its starting slot and its length.

When the MAC sublayer of the PAN coordinator receives an MCPS-DATA.request primitive with the TxOptions parameter indicating a GTS transmission, it shall determine whether it has a valid receive GTS corresponding to the device with the requested destination address. If a valid GTS is found, the PAN coordinator shall defer the transmission until the start of the receive GTS. In this case, the address of the device with the message requiring a GTS transmission shall not be added to the list of pending addresses in the beacon frame (see 7.5.5). At the start of the receive GTS, the MAC sublayer shall transmit the data without using CSMA-CA, provided the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer shall defer the transmission until the specified GTS in the next superframe.

For all allocated transmit GTSs (relative to the device), the MAC sublayer of the PAN coordinator shall ensure that its receiver is enabled at a time prior to the start and for the duration of each GTS.

Before commencing transmission in a GTS, each device shall ensure that the data transmission, the acknowledgment, if requested, and the IFS, suitable to the size of the data frame, can be completed before the end of the GTS.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

If a device misses the beacon at the beginning of a superframe, it shall not use its GTSs until it receives a subsequent beacon correctly. If a loss of synchronization occurs due to the loss of the beacon, the device shall consider all of its GTSs deallocated.

7.5.7.4 GTS deallocation

A device is instructed to request the deallocation of an existing GTS through the MLME-GTS.request primitive (see 7.1.7.1), using the characteristics of the GTS it wishes to deallocate. From this point onward, the GTS to be deallocated shall not be used by the device, and its stored characteristics shall be reset.

To request the deallocation of an existing GTS, the MLME shall send the GTS request command (see 7.3.9) to the PAN coordinator. The Characteristics Type subfield of the GTS Characteristics field of the request shall be set to zero (i.e., GTS deallocation), and the length and direction subfields shall be set according to the characteristics of the GTS to deallocate. Because the GTS request command contains an acknowledgment request (see 7.3.3.1), the PAN coordinator shall confirm its receipt by sending an acknowledgment frame. On receipt of the acknowledgment to the GTS request command, the MLME shall notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive (see 7.1.7.2) with a status of SUCCESS and a GTSCharacteristics parameter with its Characteristics Type subfield set to zero. If the GTS request command is not received correctly by the PAN coordinator, it shall determine that the device has stopped using its GTS by the procedure described in 7.5.7.6.

On receipt of a GTS request command with the Characteristics Type subfield of the GTS Characteristics field set to zero (GTS deallocation), the PAN coordinator shall attempt to deallocate the GTS. If the GTS characteristics contained in the GTS request command do not match the characteristics of a known GTS, the PAN coordinator shall ignore the request. If the GTS characteristics contained in the GTS request command match the characteristics of a known GTS, the MLME of the PAN coordinator shall deallocate the specified GTS and notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive (see 7.1.7.3) with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero. The PAN coordinator shall also update the Final CAP Slot subfield of the Superframe Specification field of the beacon frame, indicating the final superframe slot utilized by the increased CAP. It shall not add a descriptor to the beacon frame to describe the deallocation.

GTS deallocation may be initiated by the PAN coordinator due to a deallocation request from the next higher layer, the expiration of the GTS (see 7.5.7.6), or maintenance required to maintain the minimum CAP length, *aMinCAPLength* (see 7.5.7.1).

When a GTS deallocation is initiated by the next higher layer of the PAN coordinator, the MLME shall receive the MLME-GTS.request primitive with the GTS Characteristics field of the request set to zero (i.e., GTS deallocation) and the length and direction subfields set according to the characteristics of the GTS to deallocate.

When a GTS deallocation is initiated by the PAN coordinator either due to the GTS expiring or due to CAP maintenance, the MLME shall notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero.

In the case of any deallocation initiated by PAN coordinator, the PAN coordinator shall deallocate the GTS and add a GTS descriptor into its beacon frame corresponding to the deallocated GTS, but with its starting slot set to zero. The descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes. The PAN coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a start slot equal to zero, the device shall immediately stop using the GTS. The MLME of the device shall then notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero.

7.5.7.5 GTS reallocation

The deallocation of a GTS may result in the superframe becoming fragmented. For example, Figure 73 shows three stages of a superframe with allocated GTSs. In stage 1, three GTSs are allocated starting at slots 14, 10, and 8, respectively. If GTS 2 is now deallocated (stage 2), there will be a gap in the superframe during which nothing can happen. To solve this, GTS 3 will have to be shifted to fill the gap, thus increasing the size of the CAP (stage 3).

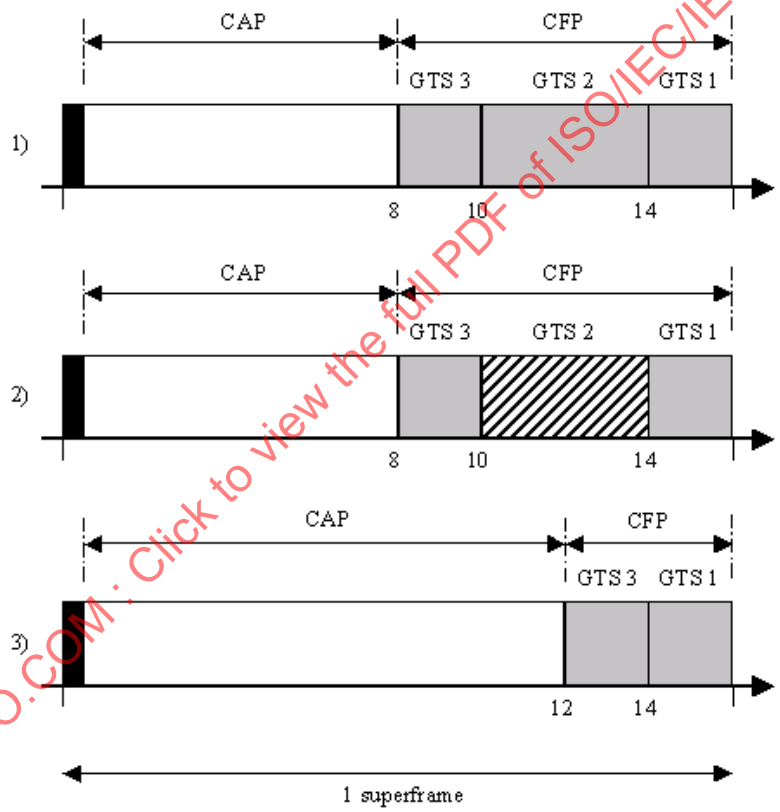


Figure 73—CFP defragmentation on GTS deallocations

The PAN coordinator shall ensure that any gaps occurring in the CFP, appearing due to the deallocation of a GTS, are removed to maximize the length of the CAP.

When a GTS is deallocated by the PAN coordinator, it shall add a GTS descriptor into its beacon frame indicating that the GTS has been deallocated. If the deallocation is initiated by a device, the PAN coordinator shall not add a GTS descriptor into its beacon frame to indicate the deallocation. For each device with an allocated GTS having a starting slot lower than the GTS being deallocated, the PAN coordinator shall update the GTS with the new starting slot and add a GTS descriptor to its beacon corresponding to this adjusted GTS. The new starting slot is computed so that no space is left between this GTS and either the end of the CFP, if the GTS appears at the end of the CFP, or the start of the next GTS in the CFP.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

In situations where multiple reallocations occur at the same time, the PAN coordinator may choose to perform the reallocation in stages. The PAN coordinator shall keep each GTS descriptor in its beacon for *aGTSDescPersistenceTime* superframes.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a direction and length corresponding to one of its GTSs, the device shall adjust the starting slot of the GTS corresponding to the GTS descriptor and start using it immediately.

In cases where it is necessary for the PAN coordinator to include a GTS descriptor in its beacon, it shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length. After *aGTSDescPersistenceTime* superframes, the PAN coordinator shall remove the GTS descriptor from the beacon.

7.5.7.6 GTS expiration

The MLME of the PAN coordinator shall attempt to detect when a device has stopped using a GTS using the following rules:

— For a transmit GTS, the MLME of the PAN coordinator shall assume that a device is no longer using its GTS if a data frame is not received from the device in the GTS at least every 2**n* superframes, where *n* is defined below.

— For receive GTSs, the MLME of the PAN coordinator shall assume that a device is no longer using its GTS if an acknowledgment frame is not received from the device at least every 2**n* superframes, where *n* is defined below. If the data frames sent in the GTS do not require acknowledgment frames, the MLME of the PAN coordinator will not be able to detect whether a device is using its receive GTS. However, the PAN coordinator is capable of deallocating the GTS at any time.

The value of *n* is defined as follows:

$n = 2^{(8-macBeaconOrder)}$ $\quad 0 \le macBeaconOrder \le 8$

$n = 1$ $\quad 9 \le macBeaconOrder \le 14$

7.5.8 Frame security

The MAC sublayer is responsible for providing security services on specified incoming and outgoing frames when requested to do so by the higher layers. This standard supports the following security services (see 5.5.6 for definitions):

— Data confidentiality

— Data authenticity

— Replay protection

The information determining how to provide the security is found in the security-related PIB (see Table 88 in 7.6.1).

7.5.8.1 Security-related MAC PIB attributes

The security-related MAC PIB attributes contain:

— Key table (*macKeyTable*, *macKeyTableEntries*)

— Device table (*macDeviceTable*, *macDeviceTableEntries*)

— Minimum security level table (*macSecurityLevelTable*, *macSecurityLevelTableEntries*)

— Frame counter (*macFrameCounter*)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

— Automatic request attributes (*macAutoRequestSecurityLevel*, *macAutoRequestKeyIdMode*, *macAutoRequestKeySource*, *macAutoRequestKeyIndex*)
— Default key source (*macDefaultKeySource*)
— PAN coordinator address (*macPANCoordExtendedAddress*, *macPANCoordShortAddress*)

7.5.8.1.1 Key table

The key table holds key descriptors (keys with related key-specific information) that are required for security processing of outgoing and incoming frames. Key-specific information in the key table is identified based on information explicitly contained in the requesting primitive or in the received frame, as described in the outgoing frame key retrieval procedure (see 7.5.8.2.2) and the incoming frame security material retrieval procedure (see 7.5.8.2.4), as well as in the KeyDescriptor lookup procedure (see 7.5.8.2.5).

7.5.8.1.2 Device table

The device table holds device descriptors (device-specific addressing information and security-related information) that, when combined with key-specific information from the key table, provide all the keying material needed to secure outgoing (see 7.5.8.2.1) and unsecure incoming frames (see 7.5.8.2.3). Device-specific information in the device table is identified based on the originator of the frame, as described in the DeviceDescriptor lookup procedure (see 7.5.8.2.7), and on key-specific information, as described in the blacklist checking procedure (see 7.5.8.2.6).

7.5.8.1.3 Minimum security level table

The minimum security level table holds information regarding the minimum security level the device expects to have been applied by the originator of a frame, depending on frame type and, if it concerns a MAC command frame, the command frame identifier. Security processing of an incoming frame will fail if the frame is not adequately protected, as described in the incoming frame security procedure (see 7.5.8.2.3) and in the incoming security level checking procedure (see 7.5.8.2.8).

7.5.8.1.4 Frame counter

The 4-octet frame counter is used to provide replay protection and semantic security of the cryptographic building block used for securing outgoing frames. The frame counter is included in each secured frame and is one of the elements required for the unsecuring operation at the recipient(s). The frame counter is incremented each time an outgoing frame is secured, as described in the outgoing frame security procedure (see 7.5.8.2.1). When the frame counter reaches its maximum value of 0xffffffff, the associated keying material can no longer be used, thus requiring all keys associated with the device to be updated. This provides a mechanism for ensuring that the keying material for every frame is unique and, thereby, provides for sequential freshness.

7.5.8.1.5 Automatic request attributes

Automatic request attributes hold all the information needed to secure outgoing frames generated automatically and not as a result of a higher layer primitive, as is the case with automatic data requests.

7.5.8.1.6 Default key source

The default key source is information commonly shared between originator and receipient(s) of a secured frame, which, when combined with additional information explicitly contained in the requesting primitive or in the received frame, allows an originator or a recipient to determine the key required for securing or unsecuring this frame, respectively. This provides a mechanism for significantly reducing the overhead of security information contained in secured frames in particular use cases (see 7.5.8.2.2 and 7.5.8.2.4).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.8.1.7 PAN coordinator address

The address of the PAN coordinator is information commonly shared between all devices in a PAN, which, when combined with additional information explicitly contained in the requesting primitive or in the received frame, allows an originator of a frame directed to the PAN coordinator or a recipient of a frame originating from the PAN coordinator to determine the key and security-related information required for securing or unsecuring, respectively, this frame (see 7.5.8.2.2 and 7.5.8.2.4).

7.5.8.2 Functional description

A device may optionally implement security. A device that does not implement security shall not provide a mechanism for the MAC sublayer to perform any cryptographic transformation on incoming and outgoing frames nor require any PIB attributes associated with security. A device that implements security shall provide a mechanism for the MAC sublayer to provide cryptographic transformations on incoming and outgoing frames using information in the PIB attributes associated with security when the *macSecurityEnabled* attribute is set to TRUE.

If the MAC sublayer is required to transmit a frame or receives an incoming frame, the MAC sublayer shall process the frame as specified in 7.5.8.2.1 and 7.5.8.2.3, respectively.

7.5.8.2.1 Outgoing frame security procedure

The inputs to this procedure are the frame to be secured and the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters from the originating primitive or automatic request PIB attributes. The outputs from this procedure are the status of the procedure and, if this status is SUCCESS, the secured frame.

The outgoing frame security procedure involves the following steps as applicable:

a) If the Security Enabled subfield of the Frame Control field of the frame to be secured is set to zero, the procedure shall set the security level to zero.

b) If the Security Enabled subfield of the Frame Control field of the frame to be secured is set to one, the procedure shall set the security level to the SecurityLevel parameter. If the resulting security level is zero, the procedure shall return with a status of UNSUPPORTED_SECURITY.

c) If the *macSecurityEnabled* attribute is set to FALSE and the security level is not equal to zero, the procedure shall return with a status of UNSUPPORTED_SECURITY.

d) The procedure shall determine whether the frame to be secured satisfies the constraint on the maximum length of MAC frames, as follows:

   1) The procedure shall set the length *M*, in octets, of the Authentication field to zero if the security level is equal to zero and shall determine this value from the security level and Table 95 otherwise.

   2) The procedure shall determine the length AuxLen, in octets, of the auxiliary security header (see 7.6.2) using KeyIdMode and the security level.

   3) The procedure shall determine the data expansion as AuxLen + *M*.

   4) The procedure shall check whether the length of the frame to be secured, including data expansion and FCS, is less than or equal to *aMaxPHYPacketSize*. If this check fails, the procedure shall return with a status of FRAME_TOO_LONG.

e) If the security level is zero, the procedure shall set the secured frame to be the frame to be secured and return with the secured frame and a status of SUCCESS.

f) The procedure shall set the frame counter to the *macFrameCounter* attribute. If the frame counter has the value 0xffffffff, the procedure shall return with a status of COUNTER_ERROR.

g) The procedure shall obtain the key using the outgoing frame key retrieval procedure as described in 7.5.8.2.2. If that procedure fails, the procedure shall return with a status of UNAVAILABLE_KEY.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

h) The procedure shall insert the auxiliary security header into the frame, with fields set as follows:

1) The Security Level subfield of the Security Control field shall be set to the security level.

2) The Key Identifier Mode subfield of the Security Control field shall be set to the KeyIdMode parameter.

3) The Frame Counter field shall be set to the frame counter.

4) If the KeyIdMode parameter is set to a value not equal to zero, the Key Source and Key Index subfields of the Key Identifier field shall be set to the KeySource and KeyIndex parameters, respectively.

i) The procedure shall then use *aExtendedAddress*, the frame counter, the security level, and the key to produce the secured frame according to the transformation process known as CCM* [or the extension of CCM, which is the combined counter with CBC-MAC (i.e., cipher block chaining message authentication code) mode of operation] that is described in the security operations (see 7.6.3.4).

1) If the SecurityLevel parameter specifies the use of encryption (see Table 95 in 7.6.2.2.1), the encryption operation shall be applied only to the actual payload field within the MAC payload, i.e., the Beacon Payload field (see 7.2.2.1.8), Command Payload field (see 7.2.2.4.3), or Data Payload field (see 7.2.2.2.2), depending on the frame type. The corresponding payload field is passed to the CCM* transformation process described in 7.6.3.4 as the unsecured payload (see Table 97 in 7.6.3.4.2). The resulting encrypted payload shall substitute the original payload.

2) The remaining fields in the MAC payload part of the frame shall be passed to the CCM* transformation process described in 7.6.3.4 as the nonpayload fields (see Table 97).

3) The ordering and exact manner of performing the encryption and integrity operations and the placement of the resulting encrypted data or integrity code within the MAC Payload field shall be as defined in 7.6.3.4.

j) The procedure shall increment the frame counter by one and set the *macFrameCounter* attribute to the resulting value.

k) The procedure shall return with the secured frame and a status of SUCCESS.

7.5.8.2.2 Outgoing frame key retrieval procedure

The inputs to this procedure are the frame to be secured and the KeyIdMode, KeySource, and KeyIndex parameters from the originating primitive. The outputs from this procedure are a passed or failed status and, if passed, a key.

The outgoing frame key retrieval procedure involves the following steps as applicable:

a) If the KeyIdMode parameter is set to 0x00 (implicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:

1) If the Destination Addressing Mode subfield of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the short address is used), the key lookup data shall be set to the 2-octet Source PAN Identifier field of the frame right-concatenated (see B.2.1) with the 2-octet *macPANCoordShortAddress* attribute right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

2) If the Destination Addressing Mode subfield of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the key lookup data shall be set to the 8-octet *macPANCoordExtendedAddress* attribute right-concatenated (see B.2.1) with the single octet 0x00. The key lookup size shall be set to nine.

3) If the Destination Addressing Mode subfield of the Frame Control field of the frame is set to 0x02, the key lookup data shall be set to the 2-octet Destination PAN Identifier field of the

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

frame right-concatenated (see B.2.1) with the 2-octet Destination Address field of the frame right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

4) If the Destination Addressing Mode subfield of the Frame Control field of the frame is set to 0x03, the key lookup data shall be set to the 8-octet Destination Address field of the frame right-concatenated (see B.2.1) with the single octet 0x00. The key lookup size shall be set to nine.

b) If the KeyIdMode parameter is set to a value not equal to 0x00 (explicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:

1) If the KeyIdMode parameter is set to 0x01, the key lookup data shall be set to the 8-octet *macDefaultKeySource* attribute right-concatenated (see B.2.1) with the single octet KeyIndex parameter. The key lookup size shall be set to nine.

2) If the KeyIdMode parameter is set to 0x02, the key lookup data shall be set to the 4-octet KeySource parameter right-concatenated (see B.2.1) with the single octet KeyIndex parameter. The key lookup size shall be set to five.

3) If the KeyIdMode parameter is set to 0x03, the key lookup data shall be set to the 8-octet KeySource parameter right-concatenated (see B.2.1) with the single octet KeyIndex parameter. The key lookup size shall be set to nine.

c) The procedure shall obtain the KeyDescriptor by passing the key lookup data and the key lookup size to the KeyDescriptor lookup procedure as described in 7.5.8.2.5. If that procedure returns with a failed status, this procedure shall also return with a failed status.

d) The MAC sublayer shall set the key to the Key element of the KeyDescriptor.

e) The procedure shall return with a passed status, having obtained the key identifier and the key.

NOTE—For broadcast frames, the outgoing frame key retrieval procedure will result in a failed status if implicit key identification is used. Hence, explicit key identification should be used for broadcast frames.[7]

7.5.8.2.3 Incoming frame security procedure

The input to this procedure is the frame to be unsecured. The outputs from this procedure are the unsecured frame, the security level, the key identifier mode, the key source, the key index, and the status of the procedure. All outputs of this procedure are assumed to be invalid unless and until explicitly set in this procedure. It is assumed that the PIB attributes associating KeyDescriptors in *macKeyTable* with a single, unique device or a number of devices will have been established by the next higher layer.

The incoming frame security procedure involves the following steps as applicable:

a) If the Security Enabled subfield of the Frame Control field of the frame to be unsecured is set to zero, the procedure shall set the security level to zero.

b) If the Security Enabled subfield of the Frame Control field of the frame to be unsecured is set to one and the Frame Version subfield of the Frame Control field of the frame to be unsecured is set to zero, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of UNSUPPORTED_LEGACY.

c) If the Security Enabled subfield of the Frame Control field of the frame to be unsecured is set to one, the procedure shall set the security level and the key identifier mode to the corresponding subfields of the Security Control field of the auxiliary security header of the frame to be unsecured and shall set the key source and key index to the corresponding subfields of the Key Identifier field of the auxiliary security header of the frame to be unsecured, if present. If the resulting security level is zero, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of UNSUPPORTED_SECURITY.

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

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

d) If the *macSecurityEnabled* attribute is set to FALSE, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of SUCCESS if the security level is equal to zero and with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of UNSUPPORTED_SECURITY otherwise.

e) The procedure shall determine whether the frame to be unsecured meets the minimum security level by passing the security level, the frame type, and, depending on whether the frame is a MAC command frame, the first octet of the MAC payload (i.e., command frame identifier for a MAC command frame) to the incoming security level checking procedure as described in 7.5.8.2.8. If that procedure fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of IMPROPER_SECURITY_LEVEL.

f) If the security level is set to zero, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of SUCCESS.

g) The procedure shall obtain the KeyDescriptor, DeviceDescriptor, and KeyDeviceDescriptor using the incoming frame security material retrieval procedure described in 7.5.8.2.4. If that procedure fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of UNAVAILABLE_KEY.

h) The procedure shall determine whether the frame to be unsecured conforms to the key usage policy by passing the KeyDescriptor, the frame type, and, depending on whether the frame is a MAC command frame, the first octet of the MAC payload (i.e., command frame identifier for a MAC command frame) to the incoming key usage policy checking procedure as described in 7.5.8.2.9. If that procedure fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of IMPROPER_KEY_TYPE.

i) If the Exempt element of the DeviceDescriptor is set to FALSE and if the incoming security level checking procedure of Step e above had as output the “conditionally passed” status, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of IMPROPER_SECURITY_LEVEL.

j) The procedure shall set the frame counter to the Frame Counter field of the auxiliary security header of the frame to be unsecured. If the frame counter has the value 0xffffffff, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of COUNTER_ERROR.

k) The procedure shall determine whether the frame counter is greater than or equal to the FrameCounter element of the DeviceDescriptor. If this check fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of COUNTER_ERROR.

l) The procedure shall then use the ExtAddress element of the DeviceDescriptor, the frame counter, the security level, and the Key element of the KeyDescriptor to produce the unsecured frame according to the CCM* inverse transformation process described in the security operations (see 7.6.3.5).

   1) If the security level specifies the use of encryption (see Table 95 in 7.6.2.2.1), the decryption operation shall be applied only to the actual payload field within the MAC payload, i.e., the Beacon Payload field (see 7.2.2.1.8), Command Payload field (see 7.2.2.4.3), or Data Payload field (see 7.2.2.2.2), depending on the frame type. The corresponding payload field shall be passed to the CCM* inverse transformation process described in 7.6.3.5 as the secure payload.

   2) The remaining fields in the MAC payload part of the frame shall be passed to the CCM* inverse transformation process described in 7.6.3.5 as the nonpayload fields.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

m) If the CCM* inverse transformation process fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of SECURITY_ERROR.

n) The procedure shall increment the frame counter by one and set the FrameCounter element of the DeviceDescriptor to the resulting value.

o) If the FrameCounter element is equal to 0xffffffff, the procedure shall set the Blacklisted element of the KeyDeviceDescriptor.

p) The procedure shall return with the unsecured frame, the security level, the key identifier mode, the key source, the key index, and a status of SUCCESS.

7.5.8.2.4 Incoming frame security material retrieval procedure

The input to this procedure is the frame to be unsecured. The outputs from this procedure are a passed or failed status and, if passed, a KeyDescriptor, a DeviceDescriptor, and a KeyDeviceDescriptor.

The incoming frame security material retrieval procedure involves the following steps as applicable:

a) If the Key Identifier Mode subfield of the Security Control field of the auxiliary security header of the frame is set to 0x00 (implicit key identification), the procedure shall determine the key lookup data and the key lookup size as follows:

   1) If the source address mode of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the short address is used), the key lookup data shall be set to the 2-octet Destination PAN Identifier field of the frame right-concatenated (see B.2.1) with the 2-octet *macPANCoordShortAddress* attribute right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

   2) If the source address mode of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the key lookup data shall be set to the 8-octet *macPANCoordExtendedAddress* attribute right-concatenated (see B.2.1) with the single octet 0x00. The key lookup size shall be set to nine.

   3) If the source address mode of the Frame Control field of the frame is set to 0x02, the key lookup data shall be set to the 2-octet Source PAN Identifier field of the frame, or to the 2-octet Destination PAN Identifier field of the frame if the PAN ID Compression subfield of the Frame Control field of the frame is set to one, right-concatenated (see B.2.1) with the 2-octet Source Address field of the frame right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

   4) If the source address mode of the Frame Control field of the frame is set to 0x03, the key lookup data shall be set to the 8-octet Source Address field of the frame right-concatenated (see B.2.1) with the single octet 0x00. The key lookup size shall be set to nine.

b) If the Key Identifier Mode subfield of the Security Control field of the auxiliary security header of the frame is set to a value not equal to 0x00 (explicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:

   1) If the key identifier mode is set to 0x01, the key lookup data shall be set to the 8-octet *macDefaultKeySource* attribute right-concatenated (see B.2.1) with the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to nine.

   2) If the key identifier mode is set to 0x02, the key lookup data shall be set to the right-concatenation (see B.2.1) of the 4-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to five.

   3) If the key identifier mode is set to 0x03, the key lookup data shall be set to the right-concatenation (see B.2.1) of the 8-octet Key Source subfield and the 1-octet Key Index subfield

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to nine.

c) The procedure shall obtain the KeyDescriptor by passing the key lookup data and the key lookup size to the KeyDescriptor lookup procedure as described in 7.5.8.2.5. If that procedure returns with a failed status, the procedure shall also return with a failed status.

d) The procedure shall determine the device lookup data and the device lookup size as follows:

1) If the source address mode of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the short address is used), the device lookup data shall be set to the 2-octet Destination PAN Identifier field of the frame right-concatenated (see B.2.1) with the 2-octet *macPANCoordShortAddress* attribute. The device lookup size shall be set to four.

2) If the source address mode of the Frame Control field of the frame is set to 0x00 and the *macPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the device lookup data shall be set to the 8-octet *macPANCoordExtendedAddress* attribute. The device lookup size shall be set to eight.

3) If the source address mode of the Frame Control field of the frame is set to 0x02, the device lookup data shall be set to the 2-octet Source PAN Identifier field of the frame, or to the 2-octet Destination PAN Identifier field of the frame if the PANID Compression subfield of the Frame Control field of the frame is set to one, right-concatenated (see B.2.1) with the 2-octet Source Address field of the frame. The device lookup size shall be set to four.

4) If the source address mode of the Frame Control field of the frame is set to 0x03, the device lookup data shall be set to the 8-octet Source Address field of the frame. The device lookup size shall be set to eight.

e) The procedure shall obtain the DeviceDescriptor and the KeyDeviceDescriptor by passing the KeyDescriptor, the device lookup data, and the device lookup size to the blacklist checking procedure as described in 7.5.8.2.6. If that procedure returns with a failed status, the procedure shall also return with a failed status.

f) The procedure shall return with a passed status having obtained the KeyDescriptor, the DeviceDescriptor, and the KeyDeviceDescriptor.

7.5.8.2.5 KeyDescriptor lookup procedure

The inputs to this procedure are the key lookup data and the key lookup size. The outputs from this procedure are a passed or failed status and, if passed, a KeyDescriptor.

The KeyDescriptor lookup procedure involves the following steps as applicable:

a) For each KeyDescriptor in the *macKeyTable* attribute and for each KeyIdLookupDescriptor in the KeyIdLookupList of the KeyDescriptor, the procedure shall check whether the LookupDataSize element of the KeyIdLookupDescriptor indicates the same integer value (see Figure 94) as the key lookup size and whether the LookupData element of the KeyIdLookupDescriptor is equal to the key lookup data. If both checks pass (i.e., there is a match), the procedure shall return with this (matching) KeyDescriptor and a passed status.

b) The procedure shall return with a failed status.

7.5.8.2.6 Blacklist checking procedure

The inputs to this procedure are the KeyDescriptor, the device lookup data, and the device lookup size. The outputs from this procedure are a passed or failed status and, if passed, a DeviceDescriptor and a KeyDeviceDescriptor.

The blacklist checking procedure involves the following steps as applicable:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

a) For each KeyDeviceDescriptor in the KeyDeviceList of the KeyDescriptor:
   1) The procedure shall obtain the DeviceDescriptor using the DeviceDescriptorHandle element of the KeyDeviceDescriptor.
   2) If the UniqueDevice element of the KeyDeviceDescriptor is set to TRUE, the procedure shall return with the DeviceDescriptor, the KeyDeviceDescriptor, and a passed status if the BlackListed element of the KeyDeviceDescriptor is set to FALSE, or the procedure shall return with a failed status if this Blacklisted element is set to TRUE.
   3) If the UniqueDevice element of the KeyDeviceDescriptor is set to FALSE, the procedure shall execute the DeviceDescriptor lookup procedure as described in 7.5.8.2.7, with the device lookup data and the device lookup size as inputs. If the corresponding output of that procedure is a passed status, the procedure shall return with the DeviceDescriptor, the KeyDeviceDescriptor, and a passed status if the Blacklisted element of the KeyDeviceDescriptor is set to FALSE, or the procedure shall return with a failed status if this Blacklisted element is set to TRUE.
b) The procedure shall return with a failed status.

7.5.8.2.7 DeviceDescriptor lookup procedure

The inputs to this procedure are the DeviceDescriptor, the device lookup data, and the device lookup size. The output from this procedure is a passed or failed status.

The DeviceDescriptor lookup procedure involves the following steps as applicable:

a) If the device lookup size is four and the device lookup data is equal to the PAN ID element of the DeviceDescriptor right-concatenated (see B.2.1) with the ShortAddress element of the DeviceDescriptor, this procedure shall return with a passed status.
b) If the device lookup size is eight and the device lookup data is equal to the ExtAddress element of the DeviceDescriptor, this procedure shall return with a passed status.
c) The procedure shall return with a failed status.

7.5.8.2.8 Incoming security level checking procedure

The inputs to this procedure are the incoming security level, the frame type and the command frame identifier. The output from this procedure is a passed, failed, or “conditionally passed” status.

The incoming security level checking procedure involves the following steps as applicable:

a) For each SecurityLevelDescriptor in the *macSecurityLevelTable* attribute:
   1) If the frame type is not equal to 0x03 and the frame type is equal to the FrameType element of the SecurityLevelDescriptor, the procedure shall compare the incoming security level (as SEC1) with the SecurityMinimum element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 7.6.2.2.1. If this comparison fails (i.e., evaluates to FALSE), the procedure shall return with a “conditionally passed” status if the DeviceOverrideSecurityMinimum element of the SecurityLevelDescriptor is set to TRUE and the security level is set to zero and with a failed status otherwise.
   2) If the frame type is equal to 0x03, the frame type is equal to the FrameType element of the SecurityLevelDescriptor, and the command frame identifier is equal to the CommandFrameIdentifier element of the SecurityLevelDescriptor, the procedure shall compare the incoming security level (as SEC1) with the SecurityMinimum element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 7.6.2.2.1. If this comparison fails (i.e., evaluates to FALSE), the procedure shall return with a “conditionally passed” status if the DeviceOverrideSecurityMinimum element of the SecurityLevelDescriptor is set to TRUE and the security level is set to zero and with a failed status otherwise.
b) The procedure shall return with a passed status.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.5.8.2.9 Incoming key usage policy checking procedure

The inputs to this procedure are the KeyDescriptor, the frame type, and the command frame identifier. The output from this procedure is a passed or failed status.

The incoming key usage policy checking procedure involves the following steps as applicable:

a) For each KeyUsageDescriptor in the KeyUsageList of the KeyDescriptor:

   1) If the frame type is not equal to 0x03 and the frame type is equal to the FrameType element of the KeyUsageDescriptor, the procedure shall return with a passed status.

   2) If the frame type is equal to 0x03, the frame type is equal to the FrameType element of the KeyUsageDescriptor, and the command frame identifier is equal to the CommandFrameIdentifier element of the KeyUsageDescriptor, the procedure shall return with a passed status.

b) The procedure shall return with a failed status.

7.6 Security suite specifications

7.6.1 PIB security material

The PIB security-related attributes are presented in Table 88, Table 89, Table 90, Table 91, Table 92, Table 93, and Table 94.

Table 88—Security-related MAC PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macKeyTable	0x71	List of KeyDescriptor entries (see Table 89)	—	A table of KeyDescriptor entries, each containing keys and related information required for secured communications.	(empty)
macKeyTableEntries	0x72	Integer	Implementation specific	The number of entries in *macKeyTable*.	0
macDeviceTable	0x73	List of DeviceDescriptor entries (see Table 93)	—	A table of DeviceDescriptor entries, each indicating a remote device with which this device securely communicates.	(empty)
macDeviceTableEntries	0x74	Integer	Implementation specific	The number of entries in *macDeviceTable*.	0
macSecurityLevelTable	0x75	Table of SecurityLevel Descriptor entries (see Table 92)	—	A table of SecurityLevelDescriptor entries, each with information about the minimum security level expected depending on incoming frame type and subtype.	(empty)
macSecurityLevelTableEntries	0x76	Integer	Implementation specific	The number of entries in *macSecurityLevelTable*.	0
macFrameCounter	0x77	Integer	0x00000000–0xffffffff	The outgoing frame counter for this device.	0x00000000

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 88—Security-related MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macAutoRequest-SecurityLevel	0x78	Integer	0x00–0x07	The security level used for automatic data requests.	0x06
macAutoRequest-KeyIdMode	0x79	Integer	0x00–0x03	The key identifier mode used for automatic data requests. This attribute is invalid if the *macAuto-RequestSecurityLevel* attribute is set to 0x00.	0x00
macAutoRequest-KeySource	0x7a	As specified by the *mac-AutoRequest-KeyIdMode* parameter	—	The originator of the key used for automatic data requests. This attribute is invalid if the *macAuto-RequestKeyIdMode* element is invalid or set to 0x00.	All octets 0xff
macAutoRequest-KeyIndex	0x7b	Integer	0x01–0xff	The index of the key used for automatic data requests. This attribute is invalid if the *macAuto-RequestKeyIdMode* attribute is invalid or set to 0x00.	All octets 0xff
macDefaultKey-Source	0x7c	Set of 8 octets	—	The originator of the default key used for key identifier mode 0x01.	All octets 0xff
macPANCoord-ExtendedAddress	0x7d	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the PAN coordinator.	—
macPANCoordShort-Address	0x7e	Integer	0x0000–0xffff	The 16-bit short address assigned to the PAN coordinator. A value of 0xfffe indicates that the PAN coordinator is only using its 64-bit extended address. A value of 0xffff indicates that this value is unknown.	0x0000

Table 89—Elements of KeyDescriptor

Name	Type	Range	Description
KeyIdLookupList	List of KeyId-LookupDescriptor entries (see Table 94)	—	A list of KeyIdLookupDescriptor entries used to identify this KeyDescriptor.
KeyIdLookupListEntries	Integer	Implementation specific	The number of entries in KeyIdLookupList.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 89—Elements of KeyDescriptor *(continued)*

Name	Type	Range	Description
KeyDeviceList	List of KeyDevice-Descriptor entries (see Table 91)	—	A list of KeyDeviceDescriptor entries indicating which devices are currently using this key, including their blacklist status.
KeyDeviceListEntries	Integer	Implementation specific	The number of entries in KeyDeviceList.
KeyUsageList	List of KeyUsage-Descriptor entries (see Table 90)	—	A list of KeyUsageDescriptor entries indicating which frame types this key may be used with.
KeyUsageListEntries	Integer		The number of entries in KeyUsageList.
Key	Set of 16 octets	—	The actual value of the key.

Table 90—Elements of KeyUsageDescriptor

Name	Type	Range	Description
FrameType	Integer	0x00–0x03	See 7.2.1.1.1.
CommandFrameIdentifier	Integer	0x00–0x09	See Table 82.

Table 91—Elements of KeyDeviceDescriptor

Name	Type	Range	Description
DeviceDescriptorHandle	Integer	Implementa-tion specific	Handle to the DeviceDescriptor corresponding to the device (see Table 93).
UniqueDevice	Boolean	TRUE or FALSE	Indication of whether the device indicated by DeviceDescriptorHandle is uniquely associated with the KeyDescriptor, i.e., it is a link key as opposed to a group key.
Blacklisted	Boolean	TRUE or FALSE	Indication of whether the device indicated by DeviceDescriptorHandle previously communicated with this key prior to the exhaustion of the frame counter. If TRUE, this indicates that the device shall not use this key further because it exhausted its use of the frame counter used with this key.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 92—Elements of SecurityLevelDescriptor

Name	Type	Range	Description
FrameType	Integer	0x00–0x03	See 7.2.1.1.1.
CommandFrameIdentifier	Integer	0x00–0x09	See Table 82.
SecurityMinimum	Integer	0x00–0x07	The minimal required/expected security level for incoming MAC frames with the indicated frame type and, if present, command frame type (see Table 95 in 7.6.2.2.1).
DeviceOverrideSecurity-Minimum	Boolean	TRUE or FALSE	Indication of whether originating devices for which the Exempt flag is set may override the minimum security level indicated by the SecurityMinimum element. If TRUE, this indicates that for originating devices with Exempt status, the incoming security level zero is acceptable, in addition to the incoming security levels meeting the minimum expected security level indicated by the SecurityMinimum element.

Table 93—Elements of DeviceDescriptor

Name	Type	Range	Description
PANId	Device PAN ID	0x0000–0xffff	The 16-bit PAN identifier of the device in this DeviceDescriptor.
ShortAddress	Device short address	0x0000–0xffff	The 16-bit short address of the device in this DeviceDescriptor. A value of 0xfffe indicates that this device is using only its extended address. A value of 0xffff indicates that this value is unknown.
ExtAddress	IEEE address	Any valid 64-bit device address	The 64-bit IEEE extended address of the device in this DeviceDescriptor. This element is also used in unsecuring operations on incoming frames.
FrameCounter	Integer	0x00000000–0xffffffff	The incoming frame counter of the device in this DeviceDescriptor. This value is used to ensure sequential freshness of frames.
Exempt	Boolean	TRUE or FALSE	Indication of whether the device may override the minimum security level settings defined in Table 92.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 94—Elements of KeyIdLookupDescriptor

Name	Type	Range	Description
LookupData	Set of 5 or 9 octets	—	Data used to identify the key.
LookupDataSize	Integer	0x00–0x01	A value of 0x00 indicates a set of 5 octets; a value of 0x01 indicates a set of 9 octets.

7.6.2 Auxiliary security header

The Auxiliary Security Header field has a variable length and contains information required for security processing, including a Security Control field, a Frame Counter field, and a Key Identifier field. The Auxiliary Security Header field shall be present only if the Security Enabled subfield of the Frame Control field is set to one. The Auxiliary Security Header field shall be formatted as illustrated in Figure 74.

Octets: 1	4	0/1/5/9
Security Control	Frame Counter	Key Identifier

Figure 74—Format of the auxiliary security header

7.6.2.1 Integer and octet representation

The auxiliary security header is a MAC frame field (see 7.2.1.7) and, therefore, uses the representation conventions specified in 7.2.

7.6.2.2 Security Control field

The Security Control field is 1 octet in length and is used to provide information about what protection is applied to the frame. The Security Control field shall be formatted as shown in Figure 75.

Bit: 0–2	3–4	5–7
Security Level	Key Identifier Mode	Reserved

Figure 75—Security Control field format

7.6.2.2.1 Security Level subfield

The Security Level subfield is 3 bits in length and indicates the actual frame protection that is provided. This value can be adapted on a frame-by-frame basis and allows for varying levels of data authenticity (to allow minimization of security overhead in transmitted frames where required) and for optional data confidentiality. The cryptographic protection offered by the various security levels is shown in Table 95. When nontrivial protection is required, replay protection is always provided.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 95—Security levels available to the MAC sublayer

Security level identifier	Security Control field (Figure 75) b_2 b_1 b_0	Security attributes	Data confidentiality	Data authenticity (including length *M* of authentication tag, in octets)
0x00	'000'	None	OFF	NO (*M* = 0)
0x01	'001'	MIC-32	OFF	YES (*M* = 4)
0x02	'010'	MIC-64	OFF	YES (*M* = 8)
0x03	'011'	MIC-128	OFF	YES (*M* = 16)
0x04	'100'	ENC	ON	NO (*M* = 0)
0x05	'101'	ENC-MIC-32	ON	YES (*M* = 4)
0x06	'110'	ENC-MIC-64	ON	YES (*M* = 8)
0x07	'111'	ENC-MIC-128	ON	YES (*M* = 16)

Security levels can be ordered according to the corresponding cryptographic protection offered. Here, a first security level SEC1 is greater than or equal to a second security level SEC2 if and only if SEC1 offers at least the protection offered by SEC2, both with respect to data confidentiality and with respect to data authenticity. The statement "SEC1 is greater than or equal to SEC2" shall be evaluated as TRUE if both of the following conditions apply:

a) Bit position b_2 in SEC1 is greater than or equal to bit position b_2 in SEC2 (where Encryption OFF < Encryption ON).

b) The integer value of bit positions b_1 b_0 in SEC1 is greater than or equal to the integer value of bit positions b_1 b_0 in SEC2 (where increasing integer values indicate increasing levels of data authenticity provided, i.e., message integrity code (MIC)-0 < MIC-32 < MIC-64 < MIC-128).

Otherwise, the statement shall be evaluated as FALSE.

For example, ENC-MIC-64 ≥ MIC-64 is TRUE because ENC-MIC-64 offers the same data authenticity protection as MIC-64, plus confidentiality. On the other hand, MIC-128 ≥ ENC-MIC-64 is FALSE because even though MIC-128 offers stronger data authenticity than ENC-MIC-64, it offers no confidentiality.

7.6.2.2.2 Key Identifier Mode subfield

The Key Identifier Mode subfield is 2 bits in length and indicates whether the key that is used to protect the frame can be derived implicitly or explicitly; furthermore, it is used to indicate the particular representations of the Key Identifier field (see 7.6.2.4) if derived explicitly. The Key Identifier Mode subfield shall be set to one of the values listed in Table 96. The Key Identifier field of the auxiliary security header (see 7.6.2.4) shall be present only if this subfield has a value that is not equal to 0x00.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 96—Values of the key identifier mode

Key identifier mode	Key Identifier Mode subfield b_1 b_0	Description	Key Identifier field length (octets)
0x00	'00'	Key is determined implicitly from the originator and receipient(s) of the frame, as indicated in the frame header.	0
0x01	'01'	Key is determined from the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header in conjunction with *macDefaultKeySource*.	1
0x02	'10'	Key is determined explicitly from the 4-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header.	5
0x03	'11'	Key is determined explicitly from the 8-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header.	9

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

7.6.2.3 Frame Counter field

The Frame Counter field is 4 octets in length and represents the *macFrameCounter* attribute of the originator of a protected frame. It is used to provide semantic security of the cryptographic mechanism used to protect a frame and to offer replay protection.

7.6.2.4 Key Identifier field

The Key Identifier field has a variable length and identifies the key that is used for cryptographic protection of outgoing frames, either explicitly or in conjunction with implicitly defined side information. The Key Identifier field shall be present only if the Key Identifier Mode subfield of the Security Control field of the auxiliary security header (see 7.6.2.2.2) is set to a value different from 0x00. The Key Identifier field shall be formatted as illustrated in Figure 76.

Octets: 0/4/8	1
Key Source	Key Index

Figure 76—Format for the Key Identifier field, if present

7.6.2.4.1 Key Source subfield

The Key Source subfield, when present, is either 4 octets or 8 octets in length, according to the value specified by the Key Identifier Mode subfield of the Security Control field (see 7.6.2.2.2), and indicates the originator of a group key.

© IEEE 2010 – All rights reserved

7.6.2.4.2 Key Index subfield

The Key Index subfield is 1 octet in length and allows unique identification of different keys with the same originator.

It is the responsibility of each key originator to make sure that actively used keys that it issues have distinct key indices and that the key indices are all different from 0x00.

7.6.3 Security operations

This subclause describes the parameters for the CCM* security operations, as specified in B.3.2.

7.6.3.1 Integer and octet representation

The integer and octet representation conventions specified in B.2 are used throughout 7.6.3.

7.6.3.2 CCM* Nonce

The CCM* nonce is a 13-octet string and is used for the advanced encryption standard (AES)-CCM* mode of operation (see B.2.2). The nonce shall be formatted as shown in Figure 77, with the leftmost field in the figure defining the first (and leftmost) octets and the rightmost field defining the last (and rightmost) octet of the nonce.

Octets: 8	**4**	**1**
Source address	Frame counter	Security level

Figure 77—CCM* nonce

The source address shall be set to the extended address *aExtendedAddress* of the device originating the frame, the frame counter to the value of the respective field in the auxiliary security header (see 7.6.2), and the security level to the security level identifier corresponding to the Security Level subfield of the Security Control field of the auxiliary security header as defined in Table 95.

The source address, frame counter, and security level shall be represented as specified in 7.6.3.1.

7.6.3.3 CCM* prerequisites

Securing a frame involves the use of the CCM* mode encryption and authentication transformation, as described in B.4.1. Unsecuring a frame involves the use of the CCM* decryption and authentication checking process, as described in B.4.2. The prerequisites for the CCM* forward and inverse transformations are as follows:

— The underlying block cipher shall be the AES encryption algorithm as specified in B.3.1.

— The bit ordering shall be as defined in 7.6.3.1.

— The length in octets of the Length field *L* shall be 2 octets.

— The length of the Authentication field *M* shall be 0 octets, 4 octets, 8 octets, or 16 octets, as required.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

7.6.3.3.1 Authentication field length

The length of the Authentication field *M* for the CCM* forward transformation and the CCM* inverse transformation is determined from Table 95, using the Security Level subfield of the Security Control field of the auxiliary security header of the frame.

7.6.3.4 CCM* transformation data representation

This subclause describes how the inputs and output of the CCM* forward transformation, as described in B.4.1, are formed:

The inputs are

— Key
— Nonce
— *a* data
— *m* data

The output is *c* data.

7.6.3.4.1 Key and nonce data inputs

The Key data for the CCM* forward transformation is passed by the outgoing frame security procedure described in 7.5.8.2.1. The Nonce data for the CCM* transformation is constructed as described in 7.6.3.2.

7.6.3.4.2 *a* data and *m* data

In the CCM* transformation process, the data fields shall be applied as in Table 97.

Table 97—*a* data and *m* data for all security levels

Security level identifier	*a* data	*m* data
0x00	None	None
0x01	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x02	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x03	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x04	None	Unsecured payload fields
0x05	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields
0x06	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields
0x07	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields

7.6.3.4.3 *c* data output

In the CCM* transformation process, the data fields that are applied, or right-concatenated and applied, represent octet strings.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

The secured payload fields right-concatenated with the authentication tag shall substitute the unsecured payload field in the original unsecured frame to form the secured frame (see Table 98).

Table 98—*c* data for all security levels

Security level identifier	*c* data
0x00	None
0x01	MIC-32
0x02	MIC-64
0x03	MIC-128
0x04	Secured payload fields
0x05	Secured payload fields || MIC-32
0x06	Secured payload fields || MIC-64
0x07	Secured payload fields || MIC-128

7.6.3.5 CCM* inverse transformation data representation

This subclause describes how the inputs and output of the CCM* inverse transformation, as described in B.4.2, are formed.

The inputs are

— Key
— Nonce
— *c* data
— *a* data

The output is *m* data.

7.6.3.5.1 Key and nonce data inputs

The Key data for the CCM* inverse transformation is passed by the incoming frame security procedure described in 7.5.8.2.3. The Nonce data for the CCM* transformation is constructed as described in 7.6.3.2.

7.6.3.5.2 *c* data and *a* data

In the CCM* inverse transformation process, the data fields shall be applied as in Table 99.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table 99—*c* data and *a* data for all security levels

Security level identifier	*c* data	*a* data
0x00	None	None
0x01	MIC-32	MHR || Auxiliary security header || Nonpayload fields ||Secured payload fields
0x02	MIC-64	MHR || Auxiliary security header || Nonpayload fields || Secured payload fields
0x03	MIC-128	MHR || Auxiliary security header || Nonpayload fields || Secured payload fields
0x04	Secured payload fields	MHR || Auxiliary security header || Nonpayload fields
0x05	Secured payload fields || MIC-32	MHR || Auxiliary security header || Nonpayload fields
0x06	Secured payload fields || MIC-64	MHR || Auxiliary security header || Nonpayload fields
0x07	Secured payload fields || MIC-128	MHR || Auxiliary security header || Nonpayload fields

7.6.3.5.3 *m* data output

The *m* data shall then substitute secured payload fields and authentication tag in the original secured frame to form the unsecured frame.

7.7 Message sequence charts illustrating MAC-PHY interaction

This subclause illustrates the main tasks specified in this standard. Each task is described by use of a message sequence chart to illustrate the chronological order, rather than the exact timing, of the primitives required for each task.

The primitives necessary for the PAN coordinator to start a new PAN are shown in Figure 78. The first action the next higher layer takes after resetting the MAC sublayer is to initiate a scan to search for other PANs in the area. An active scan is required, and an ED scan may optionally be performed. The steps for performing an active scan and an ED scan are shown in Figure 83 and Figure 79, respectively.

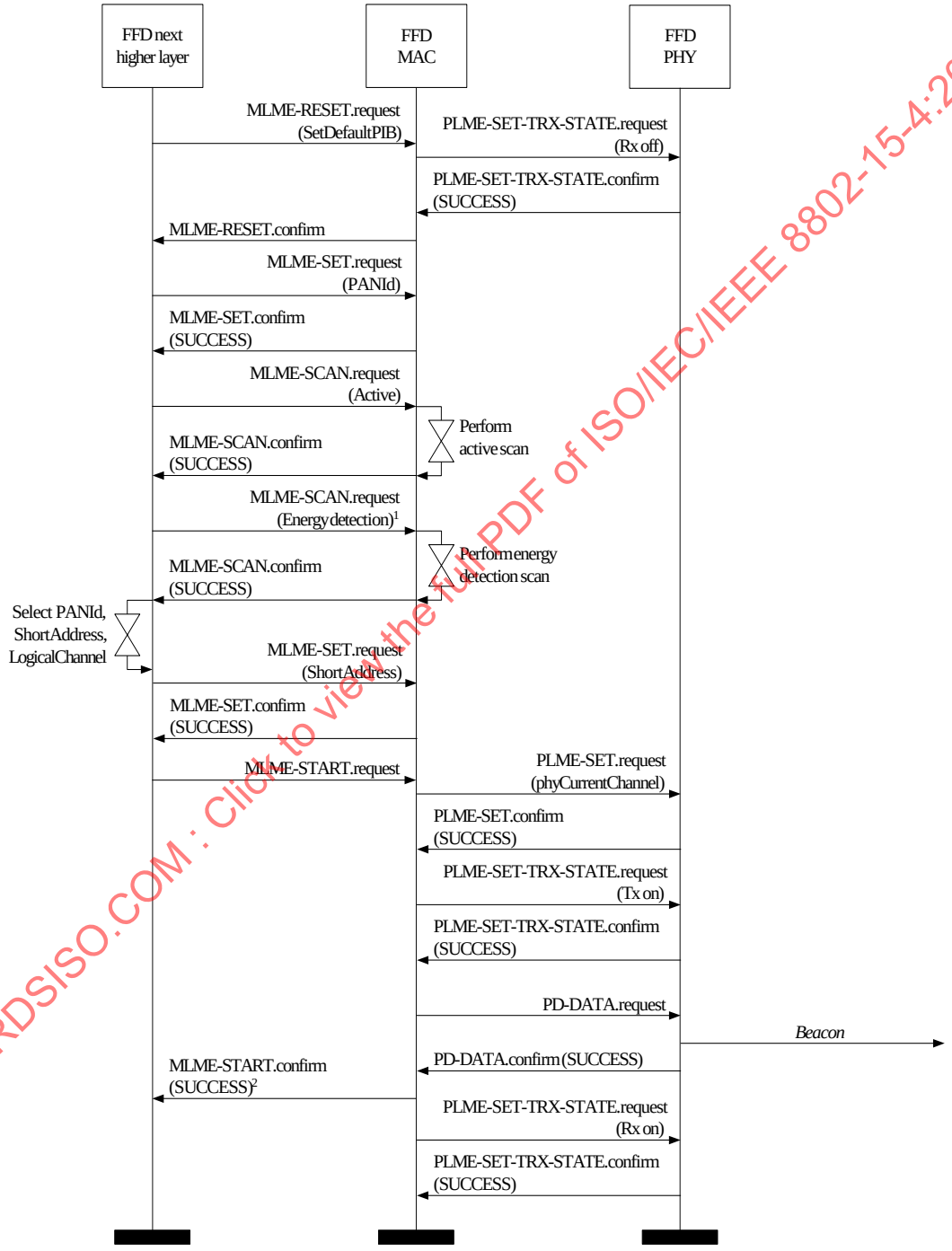
Once a new PAN is established, the PAN coordinator is ready to accept requests from other devices to join the PAN. Figure 80 shows the primitives issued by a device requesting association, while Figure 81 illustrates the steps taken by a coordinator allowing association. In the process of joining a PAN, the device requesting association will perform either a passive or an active scan to determine which PANs in the area are allowing association; Figure 82 and Figure 83 detail the primitives necessary to complete a passive scan and an active scan, respectively.

The primitives necessary for transmitting and receiving a single data packet are shown next. The actions taken by the originator of the packet are shown in Figure 84, while the actions taken by the recipient are shown in Figure 85.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

When a device becomes unable to communicate to its coordinator any longer, the device can use an orphan scan to rediscover its coordinator. The primitives necessary for the realignment of an orphaned device are shown in Figure 86.

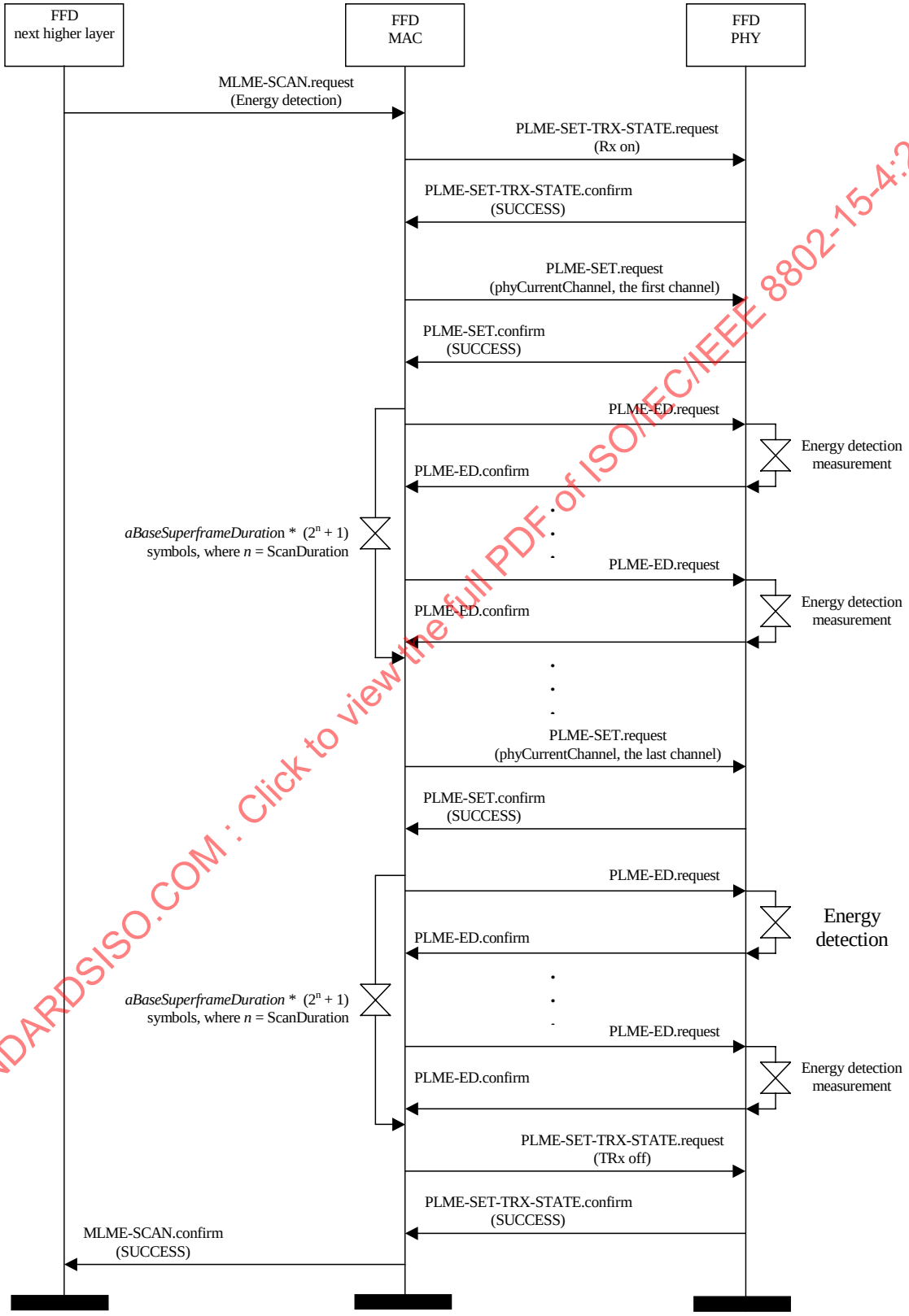


[1] The energy detection scan is optional.
[2] The MLME-START.confirm and PLME-SET-TRX-STATE.request primitives may not need to be in the order shown.

Figure 78—PAN start message sequence chart—PAN coordinator

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 79—ED scan message sequence chart

© IEEE 2010 – All rights reserved

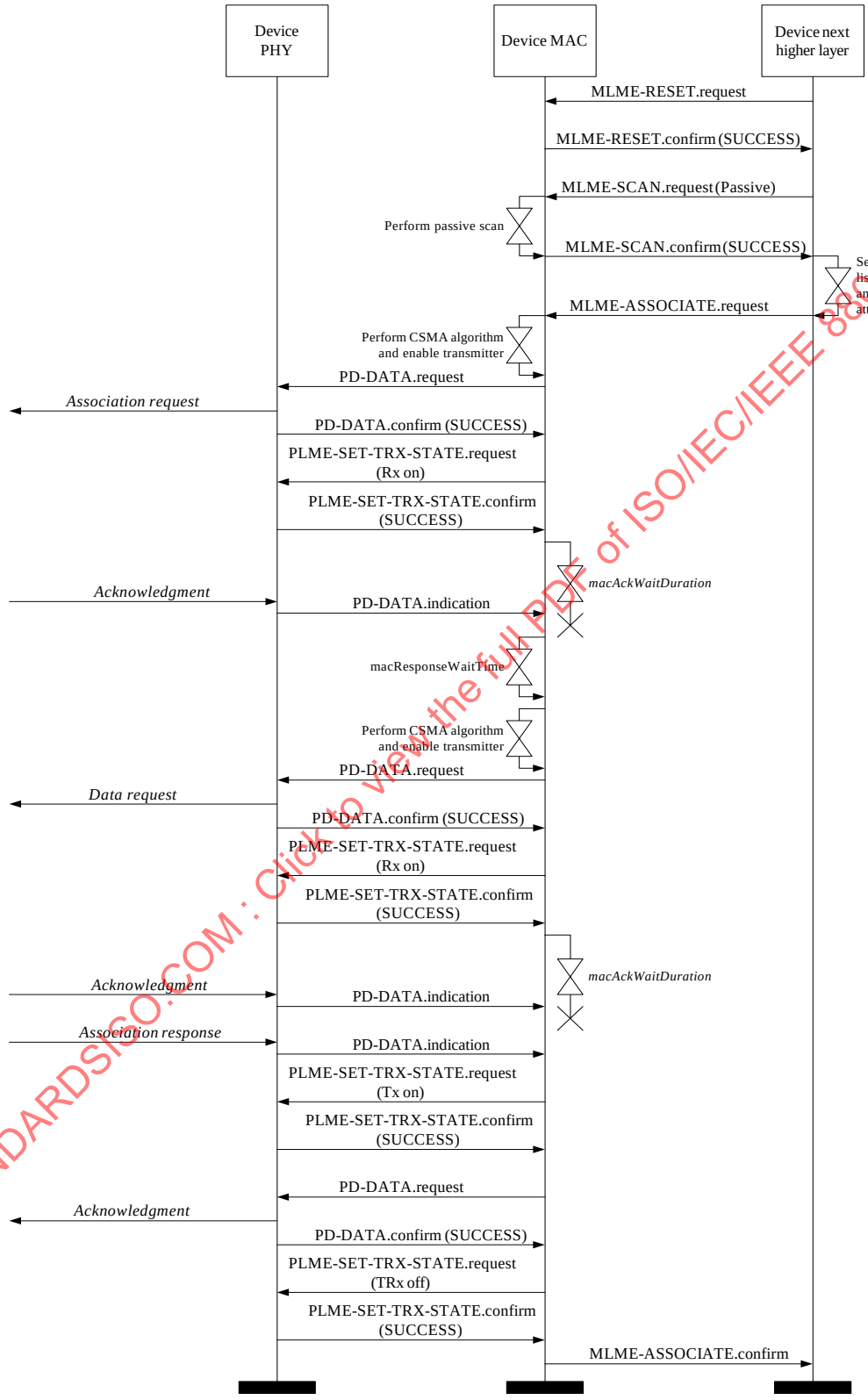
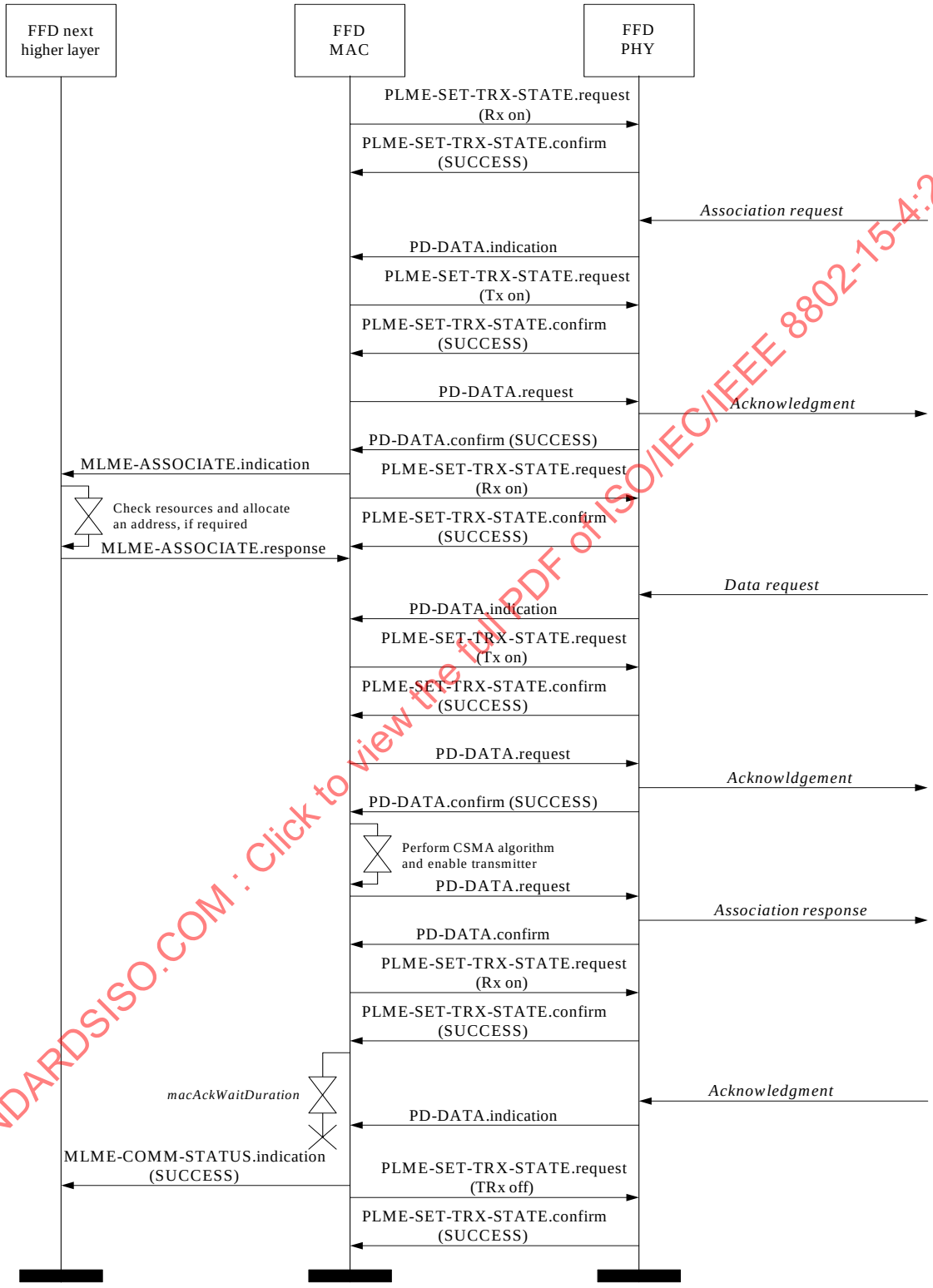


Figure 80—Association message sequence chart—device

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

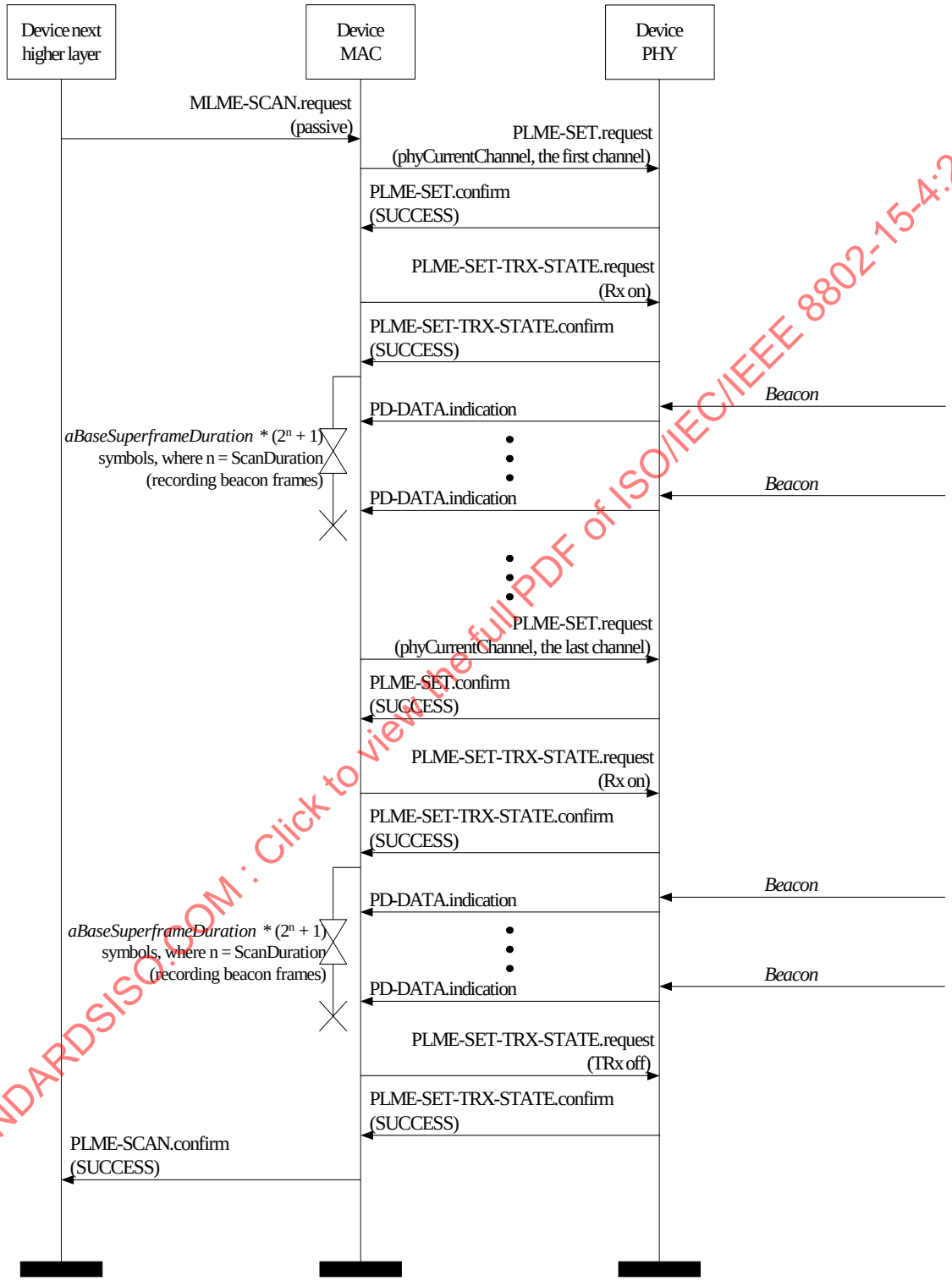
© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 81—Association message sequence chart—coordinator

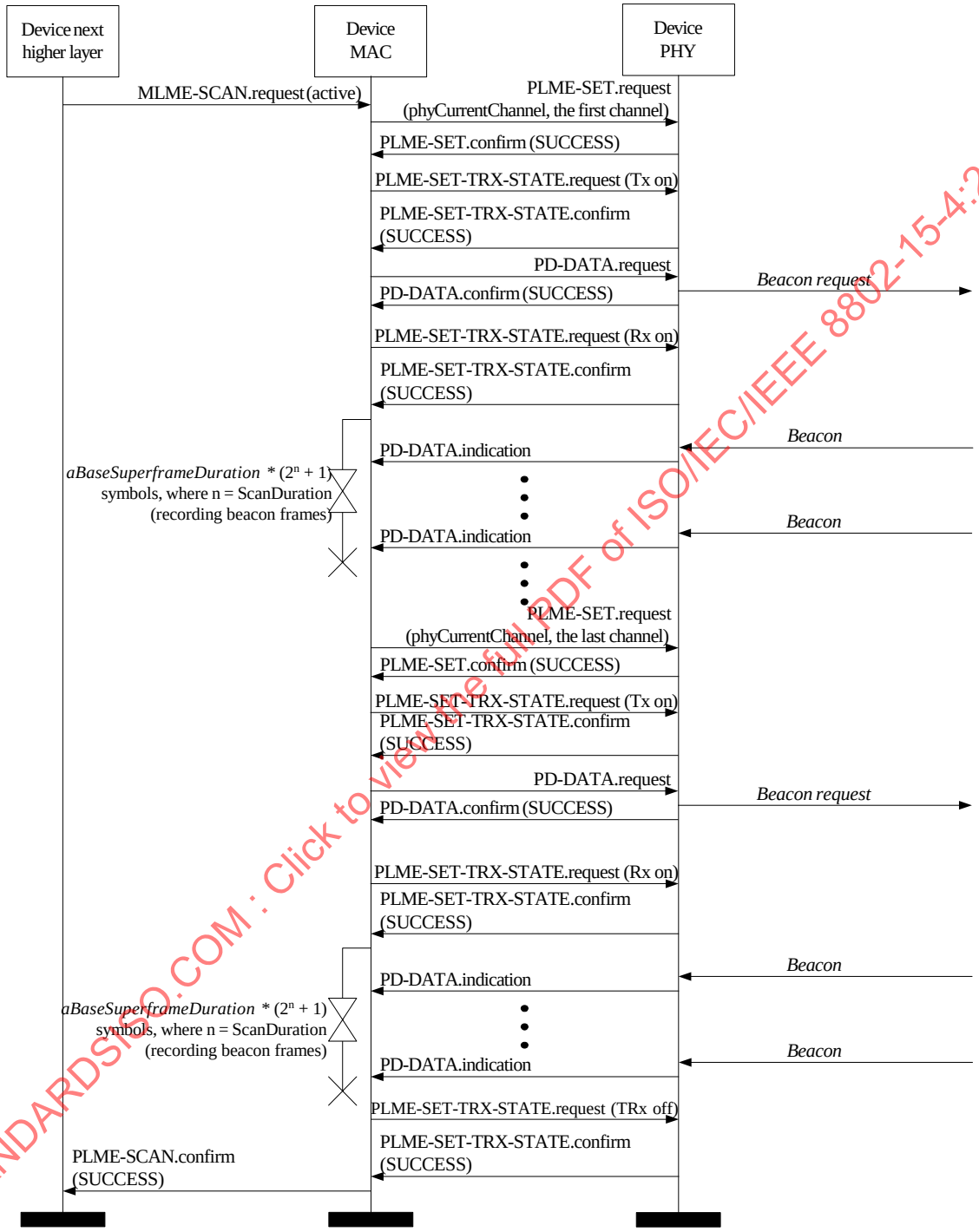
© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 82—Passive scan message sequence chart

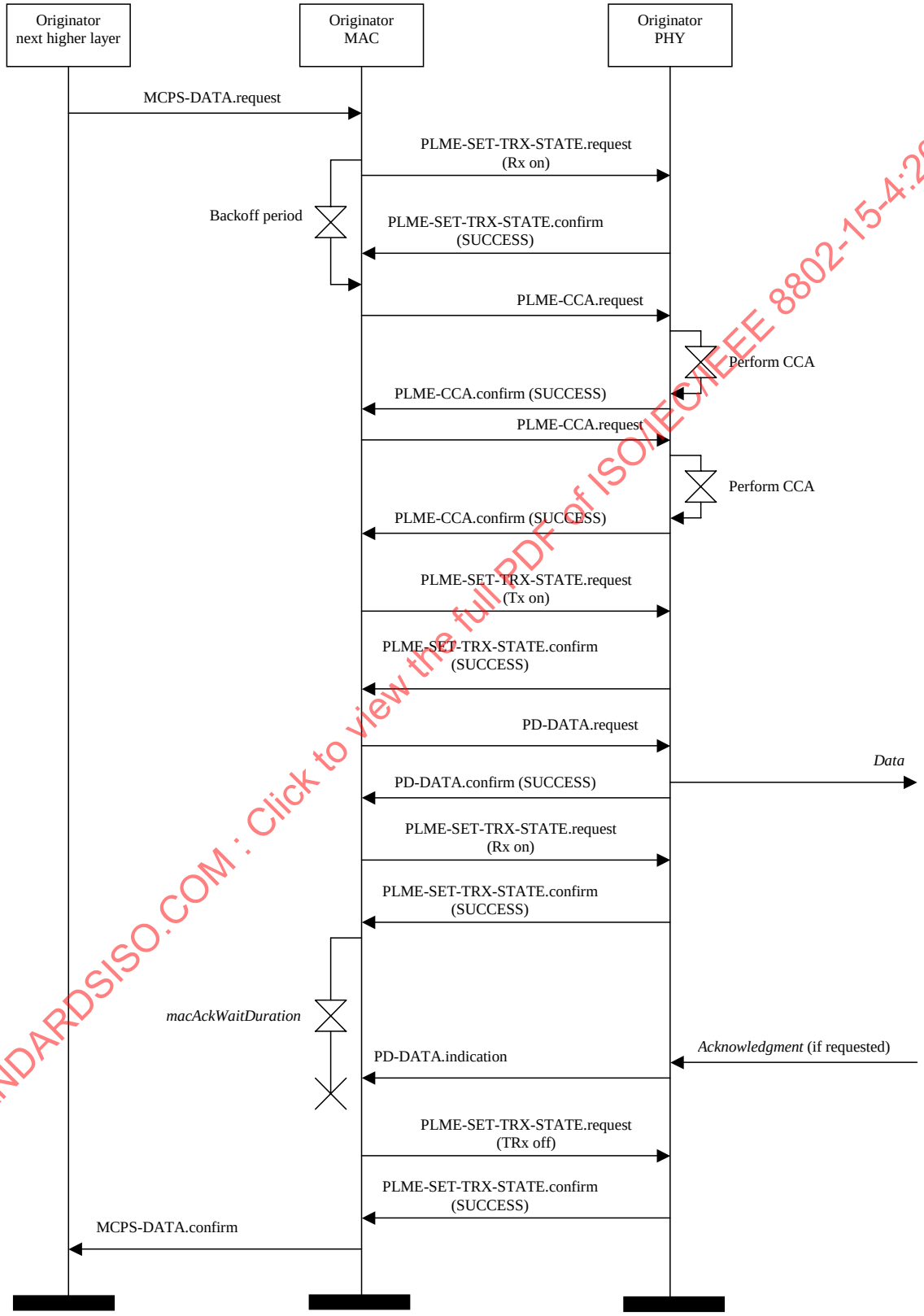
© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 83—Active scan message sequence chart

© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 84—Data transmission message sequence chart—originator

© IEEE 2010 – All rights reserved

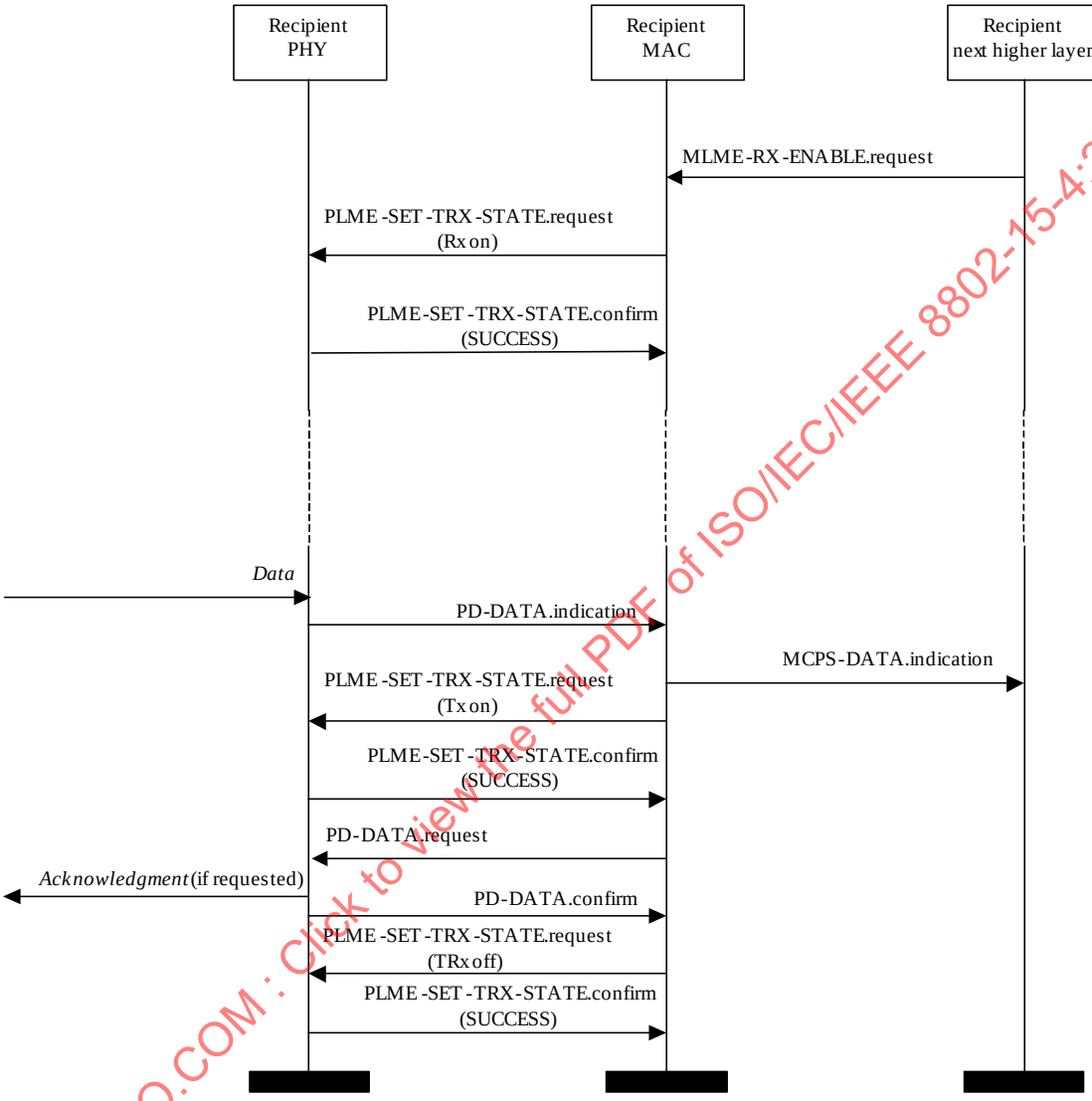
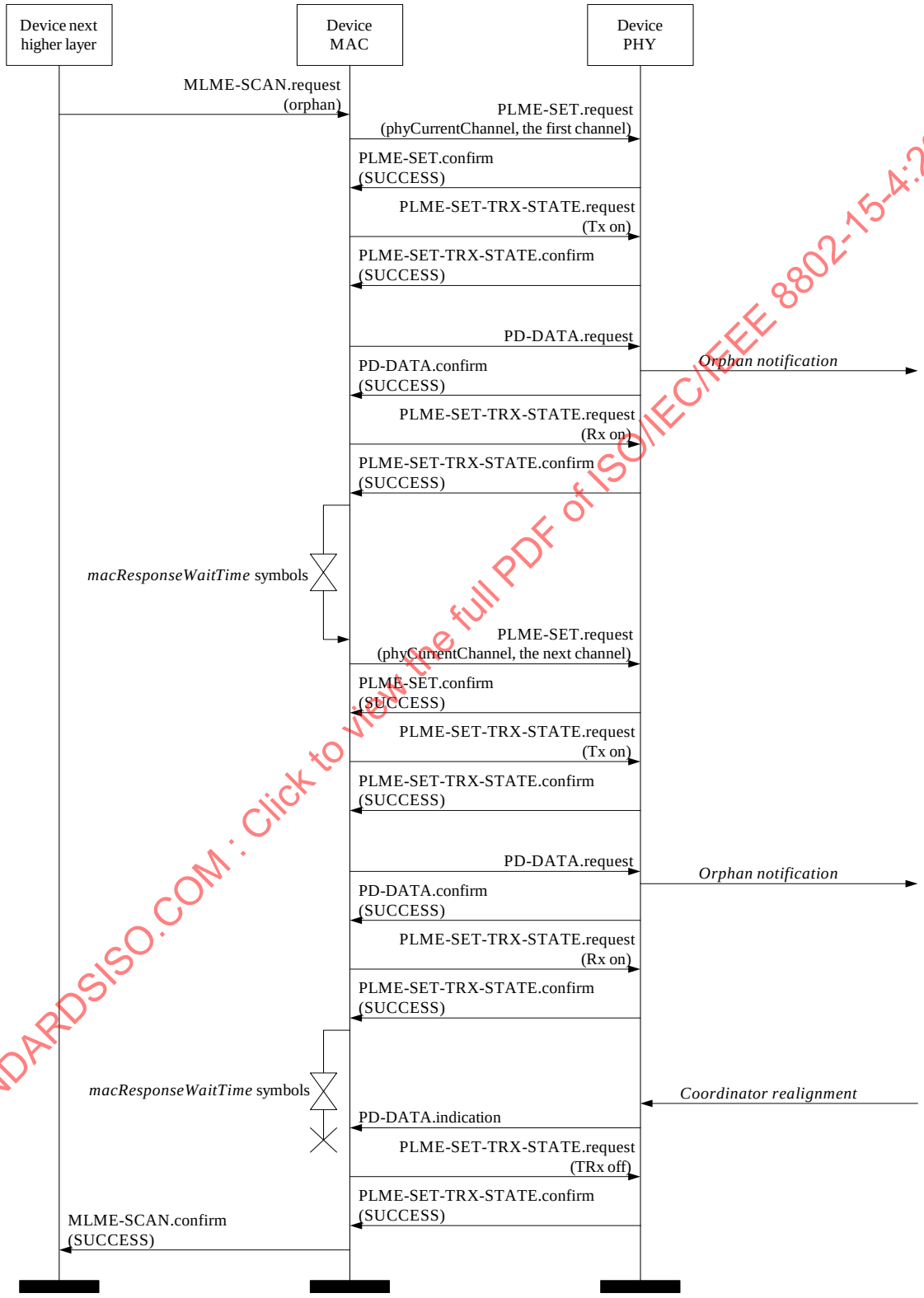


Figure 85—Data transmission message sequence chart—recipient

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved



STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

Figure 86—Orphaned device realignment message sequence chart

© IEEE 2010 – All rights reserved

(blank page)

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Annex A

(normative)

Service-specific convergence sublayer (SSCS)

A.1 IEEE 802.2 convergence sublayer

The IEEE 802.2 convergence sublayer exists above the IEEE 802.15.4 MCPS. This sublayer provides an interface between an instance of an IEEE 802.2 LLC sublayer and the IEEE 802.15.4 MCPS.

A.1.1 MA-UNITDATA.request

The MA-UNITDATA.request primitive requests the transfer of a LLC protocol data unit (LPDU) (i.e., MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or multiple peer IEEE 802.2 Type 1 LLC sublayer entities in the case of a group address.

A.1.1.1 Semantics of the service primitive

The semantics of the MA-UNITDATA.request primitive is as follows:

```
MA-UNITDATA.request     (
                        SrcAddr,
                        DstAddr,
                        RoutingInformation,
                        data,
                        priority,
                        ServiceClass
                        )
```

Table A.1 specifies the parameters for the MA-UNITDATA.request primitive.

Table A.1—MA-UNITDATA.request parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU is being transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
data	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

A.1.1.2 Appropriate usage

The MA-UNITDATA.request primitive is generated by a local IEEE 802.2 Type 1 LLC sublayer entity when an LPDU (MSDU) is to be transferred to a peer IEEE 802.2 Type 1 LLC sublayer entity or entities.

A.1.1.3 Effect on receipt

On receipt of the MA-UNITDATA.request primitive, the MAC sublayer entity shall begin the transmission of the supplied MSDU.

The MAC sublayer first builds an MPDU to transmit from the supplied arguments. The MPDU shall be transmitted using the CSMA-CA algorithm in the contention period of the frame and without requesting a handshake.

If the CSMA-CA algorithm indicates a busy channel, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE. If the MPDU was successfully transmitted, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of SUCCESS.

A.1.2 MA-UNITDATA.indication

The MA-UNITDATA.indication primitive indicates the transfer of an LPDU (i.e., MSDU) from the MAC sublayer to the local IEEE 802.2 Type 1 LLC sublayer entity.

A.1.2.1 Semantics of the service primitive

The semantics of the MA-UNITDATA.indication primitive is as follows:

```
MA-UNITDATA.indication      (
                            SrcAddr,
                            DstAddr,
                            RoutingInformation,
                            data,
                            ReceptionStatus,
                            priority,
                            ServiceClass
                            )
```

Table A.2 specifies the parameters for the MA-UNITDATA.indication primitive.

A.1.2.2 When generated

On receipt of a data packet at the local MAC sublayer entity, the FCS field is checked. If it is valid, the MAC sublayer shall issue the MA-UNITDATA.indication primitive to the IEEE 802.2 Type 1 LLC sublayer entity, indicating the arrival of a MSDU. If the FCS is not valid, the packet shall be discarded, and the IEEE 802.2 Type 1 LLC sublayer entity shall not be informed.

A.1.2.3 Appropriate usage

The appropriate usage of the MA-UNITDATA.indication primitive by the IEEE 802.2 Type 1 LLC sublayer entity is not specified in this standard.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table A.2—MA-UNITDATA.indication parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been received
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
data	Set of octets	—	The set of octets forming the MSDU received by the MAC sublayer entity.
ReceptionStatus	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

A.1.3 MA-UNITDATA-STATUS.indication

The MA-UNITDATA-STATUS.indication primitive reports the results of a request to transfer a LPDU (MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or to multiple peer IEEE 802.2 Type 1 LLC sublayer entities.

A.1.3.1 Semantics of the service primitive

The semantics of the MA-UNITDATA-STATUS.indication primitive is as follows:

```
MA-UNITDATA-STATUS.indication   (
                                SrcAddr,
                                DstAddr,
                                status,
                                ProvPriority,
                                ProvServiceClass
                                )
```

Table A.3 specifies the parameters for the MA-UNITDATA-STATUS.indication primitive.

A.1.3.2 When generated

The MA-UNITDATA-STATUS.indication primitive is generated by the MAC sublayer entity in response to an MA-UNITDATA.request primitive issued by the IEEE 802.2 Type 1 LLC sublayer.

A.1.3.3 Appropriate usage

The receipt of the MA-UNITDATA-STATUS.indication primitive by the IEEE 802.2 Type 1 LLC sublayer entity signals the completion of the current data transmission.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table A.3—MA-UNITDATA-STATUS.indication parameters

Name	Type	Valid Range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU has been transferred.
status	Enumeration	SUCCESS, TRANSMISSION_PENDING, NO_BEACON, or CHANNEL_ACCESS_FAILURE	The status of the last MSDU transmission.
ProvPriority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ProvServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Annex B

(normative)

CCM* mode of operation

B.1 Introduction

CCM* is a generic combined encryption and authentication block cipher mode. The CCM* mode coincides with the original specification for the combined counter with CBC-MAC (cipher block chaining message authentication code) mode of operation (known as CCM) (ANSI X9.63-2001 [B1], Appendix A of NIST Pub 800-38C [B16]) for messages that require authentication and, possibly, encryption, but also offers support for messages that require only encryption. Moreover, it can be used in implementation environments for which the use of variable-length authentication tags, rather than fixed-length authentication tags only, is beneficial.

B.2 Notation and representation

B.2.1 Strings and string operations

A string is a sequence of symbols over a specific set (e.g., the binary alphabet {0,1} or the set of all octets). The length of a string is the number of symbols it contains (over the same alphabet). The empty string is the string of length 0. The right-concatenation of two strings x and y (over the same alphabet) of length m and n, respectively (notation: $x \| y$), is the string z of length $m + n$ that coincides with x on its leftmost (most significant) m symbols and with y on its rightmost (least significant) n symbols. An octet is a symbol string of length 8. In the context of this standard, all octets are strings over the binary alphabet.

B.2.2 Integers, octets, and their representation

Throughout this standard, the representation of integers as octet strings and of octet strings as binary strings shall be fixed. All integers shall be represented as octet strings in most-significant-octet-first order. All octets shall be represented as bit strings of length eight in most-significant-bit-first order.

For example, the 32-bit integer 0x12345678 is represented as an octet string of {0x12, 0x34, 0x56, 0x78}, and the first octet of that octet string is represented as a bit string of {0,0,0,1,0,0,1,0}.

B.3 Symmetric-key cryptographic building blocks

The symmetric-key cryptographic primitives and mechanisms are defined for use with all security-processing operations specified in this standard.

B.3.1 Block cipher

The block cipher used in this standard shall be the advanced encryption standard (AES)-128, as specified in FIPS Pub 197. This block cipher shall be used with symmetric keys with the same size as that of the block cipher: 128 bits. The generation of these keys is outside the scope of this standard.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

B.3.2 Mode of operation

The block cipher mode of operation used in this specification shall be the generic CCM* mode of operation, as specified in B.4, with the following instantiations:

a) Each entity shall use the block cipher *E* as specified in B.3.1.

b) All integers shall be represented as octet strings as specified in B.2.2.

c) All octets shall be represented as binary strings as specified in B.2.2.

d) The parameter *L* shall have the integer value 2.

e) The parameter *M* shall have one of the following integer values: 0, 4, 8, or 16.

B.4 Specification of generic CCM* mode of operation

Prerequisites:

The following are the prerequisites for the operation of the generic CCM* mode:

a) A block cipher encryption function *E* shall have been chosen, with a 128-bit block size. The length in bits of the keys used by the chosen encryption function is denoted by keylen.

b) A fixed representation of integers as octet strings shall have been chosen (e.g., most-significant-octet-first order or least-significant-octet-first order).

c) A fixed representation of octets as binary strings shall have been chosen (e.g., most-significant-bit-first order or least-significant-bit-first order).

d) The length *L* of the message Length field, in octets, shall have been chosen. Valid values for *L* are the integers 2, 3, …, 8 (the value $L = 1$ is reserved).

e) The length *M* of the Authentication field, in octets, shall have been chosen. Valid values for *M* are the integers 0, 4, 6, 8, 10, 12, 14, and 16 (the value $M = 0$ corresponds to disabling authenticity because then the Authentication field is the empty string).

B.4.1 CCM* mode encryption and authentication transformation

Inputs:

The CCM* mode forward transformation takes the following as inputs:

a) A bit string Key of length keylen bits to be used as the key. Each entity shall have evidence that access to this key is restricted to the entity itself and its intended key sharing group member(s).

b) A nonce *N* of $15 - L$ octets. Within the scope of any encryption key Key, the nonce value shall be unique.

c) An octet string *m* of length $l(m)$ octets, where $0 \le l(m) < 2^{8L}$.

d) An octet string *a* of length $l(a)$ octets, where $0 \le l(a) < 2^{64}$.

The nonce *N* shall encode the potential values for *M* so that the actual value of *M* can be uniquely determined from *N*. The exact format of the nonce *N* is outside the scope of this specification and shall be determined and fixed by the actual implementation environment of the CCM* mode.

NOTE—The exact format of the nonce *N* is left to the application, to allow simplified hardware and software implementations in particular settings. Actual implementations of the CCM* mode may restrict the values of *M* that are allowed throughout the life cycle of the encryption key Key to a strict subset of those allowed in the generic CCM* mode. If so, the format of the nonce *N* shall be such that one can uniquely determine from *N* the actually used value of *M* in that particular subset. In particular, if *M* is fixed and the value $M = 0$ is not allowed, then there are no restrictions on *N*, in which case the CCM* mode reduces to the CCM mode.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Actions:

The CCM* mode forward transformation involves the execution, in order, of an input transformation (B.4.1.1), an authentication transformation (B.4.1.2), and an encryption transformation (B.4.1.3).

B.4.1.1 Input transformation

This step involves the transformation of the input strings *a* and *m* to the strings AuthData and PlainTextData, to be used by the authentication transformation and the encryption transformation, respectively.

This step involves the following steps, in order:

a) Form the octet string representation $L(a)$ of the length $l(a)$ of the octet string *a*, as follows:
   1) If $l(a) = 0$, then $L(a)$ is the empty string.
   2) If $0 < l(a) < 2^{16} - 2^8$, then $L(a)$ is the 2-octets encoding of $l(a)$.
   3) If $2^{16} - 2^8 \leq l(a) < 2^{32}$, then $L(a)$ is the right-concatenation of the octet 0xff, the octet 0xfe, and the 4-octet encoding of $l(a)$.
   4) If $2^{32} \leq l(a) < 2^{64}$, then $L(a)$ is the right-concatenation of the octet 0xff, the octet 0xff, and the 8-octet encoding of $l(a)$.

b) Right-concatenate the octet string $L(a)$ with the octet string *a* itself. Note that the resulting string contains $l(a)$ and *a* encoded in a reversible manner.

c) Form the padded message AddAuthData by right-concatenating the resulting string with the smallest non-negative number of all-zero octets so that the octet string AddAuthData has length divisible by 16.

d) Form the padded message PlaintextData by right-concatenating the octet string *m* with the smallest non-negative number of all-zero octets so that the octet string PlaintextData has length divisible by 16.

e) Form the message AuthData consisting of the octet strings AddAuthData and PlaintextData: AuthData = AddAuthData || PlaintextData.

B.4.1.2 Authentication transformation

The data AuthData that was established in B.4.1.1 shall be tagged using the tagging transformation as follows:

a) Form the 1-octet Flags field consisting of the 1-bit Reserved field, the 1-bit Adata field, and particular 3-bit representations of the integers *M* and *L*, as follows:

   Flags = Reserved || Adata || *M* || *L*.

   Here, the 1-bit Reserved field is reserved for future expansions and shall be set to '0'. The 1-bit Adata field is set to '0' if $l(a) = 0$ and set to '1' if $l(a) > 0$. The M field is the 3-bit representation of the integer $(M - 2)/2$ if $M > 0$ and of the integer 0 if $M = 0$, in most-significant-bit-first order. The L field is the 3-bit representation of the integer $L - 1$, in most-significant-bit-first order.

b) Form the 16-octet B_0 field consisting of the 1-octet Flags field defined in step a) in this subclause, the $(15 - L)$-octet Nonce field *N*, and the *L*-octet representation of the Length field $l(m)$, as follows:

   B_0 = Flags || Nonce *N* || $l(m)$.

c) Parse the message AuthData as $B_1 \| B_2 \| \ldots \| B_t$, where each message block B_i is a 16-octet string.

d) The CBC-MAC value X_{t+1} is defined by

   $X_0 := 0^{128}$; $X_{i+1} := E(\text{Key}, X_i \oplus B_i)$ for i = 0, …, t.

   Here, $E(K, x)$ is the ciphertext that results from encryption of the plaintext x, using the established block cipher encryption function *E* with key K; the string 0^{128} is the 16-octet all-zero bit string.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

e) The authentication tag T is the result of omitting all but the leftmost M octets of the CBC-MAC value X_{t+1} thus computed.

B.4.1.3 Encryption transformation

The data PlaintextData that was established in B.4.1.1 (step d) and the authentication tag T that was established in B.4.1.2 (step e) shall be encrypted using the encryption transformation as follows:

a) Form the 1-octet Flags field consisting of two 1-bit Reserved fields, and particular 3-bit representations of the integers 0 and L, as follows:

Flags = Reserved || Reserved || 0 || L.

Here, the two 1-bit Reserved fields are reserved for future expansions and shall be set to '0'. The '0' field is the 3-bit representation of the integer 0, in most-significant-bit-first order. The L field is the 3-bit representation of the integer $L - 1$, in most-significant-bit-first order.

b) Define the 16-octet A_i field consisting of the 1-octet Flags field defined in step a) in this subclause, the $(15 - L)$-octet Nonce field N, and the L-octet representation of the integer i, as follows:

A_i = Flags || Nonce N || Counter i, for i = 0, 1, 2, …

Note that this definition ensures that all the A_i fields are distinct from the B_0 fields that are actually used, as those have a Flags field with a nonzero encoding of M in the positions where all A_i fields have an all-zero encoding of the integer 0 (see B.4.1.2, step b).

c) Parse the message PlaintextData as M1 || … ||M_t, where each message block M_i is a 16-octet string.

d) The ciphertext blocks C_1, …, C_t are defined by

$C_i := E(\text{Key}, A_i) \oplus M_i$ for i = 1, 2, …, t.

e) The string Ciphertext is the result of omitting all but the leftmost $l(m)$ octets of the string C_1 || … || C_t.

f) Define the 16-octet encryption block S_0 by

$S_0 := E(\text{Key}, A_0)$.

g) The encrypted authentication tag U is the result of XOR-ing the string consisting of the leftmost M octets of S_0 and the authentication tag T.

Output:

If any of the above operations has failed, then output “invalid.” Otherwise, output the right-concatenation c of the encrypted message Ciphertext and the encrypted authentication tag U.

B.4.2 CCM* mode decryption and authentication checking transformation

Inputs:

The CCM* mode inverse transformation takes the following as inputs:

a) A bit string Key of length keylen bits to be used as the key. Each entity shall have evidence that access to this key is restricted to the entity itself and its intended key-sharing group member(s).

b) A nonce N of $15 - L$ octets. Within the scope of any encryption key Key, the nonce value shall be unique.

c) An octet string c of length $l(c)$ octets, where $0 \le l(c) - M < 2^{8L}$.

d) An octet string a of length $l(a)$ octets, where $0 \le l(a) < 2^{64}$.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Actions:

The CCM* mode inverse transformation involves the execution, in order, of a decryption transformation (B.4.2.1) and an authentication checking transformation (B.4.2.2).

B.4.2.1 Decryption transformation

The decryption transformation involves the following steps, in order:

a) Parse the message c as $C \| U$, where the rightmost string U is an M-octet string. If this operation fails, output "invalid" and stop. U is the purported encrypted authentication tag. Note that the leftmost string C has length $l(c) - M$ octets.

b) Form the padded message CiphertextData by right-concatenating the string C with the smallest non-negative number of all-zero octets so that the octet string CiphertextData has length divisible by 16.

c) Use the encryption transformation in B.4.1.3, with as inputs the data CiphertextData and the tag U.

d) Parse the output string resulting from applying this transformation as $m \| T$, where the rightmost string T is an M-octet string. T is the purported authentication tag. Note that the leftmost string m has length $l(c) - M$ octets.

B.4.2.2 Authentication checking transformation

The authentication checking transformation involves the following steps, in order:

a) Form the message AuthData using the input transformation in B.4.1.1, with as inputs the string a and the octet string m that was established in B.4.2.1 (step d).

b) Use the authentication transformation in B.4.1.2, with as input the message AuthData.

c) Compare the output tag MACTag resulting from this transformation with the tag T that was established in B.4.2.1 (step d). If MACTag = T, output "valid"; otherwise, output "invalid" and stop.

Output:

If any of the above verifications has failed, then output "invalid," and reject the octet strings a and m. Otherwise, accept the octet strings a and m, and accept one of the key sharing group member(s) as the source of a and m.

B.4.3 Restrictions

All implementations shall limit the total amount of data that is encrypted with a single key. The CCM* encryption and authentication transformation shall invoke not more than 2^{61} block cipher encryption function invocations with the same key in total.

The CCM* decryption and authentication checking transformation shall not expose any information if any verification check fails. The only information that may be exposed in this case is that the authenticity verification transformation failed; all other information, such as the purported plaintext, shall be destroyed.

NOTE 1—With regard to security of the CCM* mode of operation, the CCM* mode coincides with the original CCM mode specification (ANSI X9.63-2001 [B1]) for messages that require authentication and, possibly, encryption, but also offers support for messages that require only encryption. Moreover, it can be used in implementation environments for which the use of variable-length authentication tags, rather than fixed-length authentication tags only, is beneficial. As with the CCM mode, the CCM* mode requires only one key. The CCM* specification differs from the CCM specification, as follows:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

— The CCM* mode allows the length of the Authentication field *M* to be zero as well (the value *M* = 0 corresponding to disabling authenticity because then the Authentication field is the empty string).
— The CCM* mode imposes a further restriction on the nonce *N*: it shall encode the potential values for *M* so that one can uniquely determine from *N* the actually used value of *M*.

As a result, if *M* is fixed and the value *M* = 0 is not allowed, then there are no additional restrictions on *N*, in which case the CCM* mode reduces to the CCM mode. In particular, the proof of the CCM mode applies (Jonsson [B13] and [B14]).

For fixed-length authentication tags, the CCM* mode is equally secure as the original CCM mode. For variable-length authentication tags, the CCM* mode completely avoids, by design, the vulnerabilities that do apply to the original CCM mode.

For fixed-length authentication tags, the security proof of the original CCM mode carries over to that of the CCM* mode (also for *M* = 0), by observing that the proof of the original CCM mode relies on the following properties, which slightly relax those stated in Jonsson [B13] and [B14] (relaxed property indicated in italics):

— The B_0 field uniquely determines the value of the nonce *N*.
— The authentication transformation operates on input strings $B_0 \| B_1 \| B_2 \| \ldots \| B_i$ from which one can uniquely determine the input strings *a* and *m* (as well as the nonce *N*). In fact, for any two input strings corresponding to distinct triples (*N*, *m*, *a*), neither one is a prefix string of the other.
— All the A_i fields are distinct from the B_0 fields *that are actually used* (over the lifetime of the key), as those have a Flags field with a nonzero encoding of *M* in the positions where all A_i fields have an all-zero encoding of the integer 0.

Hence, if *M* is fixed, then the CCM* mode offers the same security properties as the original CCM mode: confidentiality over the input string *m* and data authenticity over the input strings *a* and *m*, relative to the length of the authentication tag. Obviously, if *M* = 0, then no data authenticity is provided by the CCM* mode itself (but may be provided by an external mechanism).

For variable-length authentication tags, the original CCM mode is known to be vulnerable to specific attacks (see, e.g., Section 3.4 of Rogaway and Wagner [B17]). These attacks may arise with the original CCM mode because the decryption transformation does not depend on the length of the authentication tag itself. The CCM* mode avoids these attacks altogether, by requiring that one shall be able to uniquely determine the length of the applicable authentication tag from the A_i fields (i.e., from the counters blocks).

NOTE 2—With regard to the interoperability between CCM mode and CCM* mode of operation, the CCM* mode reduces to the CCM mode in all implementation environments where the length of the authentication tag is fixed and where the value *M* = 0 (encryption-only) is not allowed. In particular, the CCM* mode is compatible with the CCM mode, as specified in IEEE Std 802.11™-2004 (for WLANs) [B7], IEEE Std 802.15.3™-2003 (for WPANs) [B10], and IEEE Std 802.15.4-2003 (for older WPANs).

NOTE 3—For test vectors for cryptographic building blocks, see Annex C.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Annex C

(informative)

Test vectors for cryptographic building blocks

With regard to the CCM* mode of operation (see Annex B), this annex provides sample test vectors for the IEEE 802.15.4 community, aimed at assisting in building interoperable security implementations.

C.1 AES block cipher

See FIPS Pub 197 for sample test vectors for the block cipher specified in B.3.1.

C.2 Mode of operation

This subclause provides sample test vectors for the mode of operation as specified in B.3.2, illustrated in the context of different MAC frame types.

C.2.1 MAC beacon frame

C.2.1.1 Description

The example below illustrates security processing of a beacon frame that is transmitted by the coordinator using its extended source address. In this example, the Superframe Specification field is set to 0xCF55 (e.g., beacon order and superframe order have integer value 5, while the final CAP slot has integer value 15), and there are no pending addresses. This example uses source address 0xacde480000000001, PAN identifier 0x4321, and beacon payload 0x51 0x52 0x53 0x54; the frame counter has integer value 5. The security level is set to 0x02 (MIC-64, or 64-bit data authenticity).

For simplicity, all frames in this example are shown without the FCS field (because security processing is independent of it).

Secured beacon frame:

08 D0 84 21 43 01 00 00 00 00 48 DE AC || 02 05 00 00 00 || 55 CF 00 00 51 52 53 54 22 3B C1 EC 84 1A B5 53.

Corresponding unsecured beacon frame:

00 C0 84 21 43 01 00 00 00 00 48 DE AC || 55 CF 00 00 51 52 53 54.

Prerequisite:

The following prerequisite is established for this mode of operation: the parameter *M* shall have the integer value 8.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

C.2.1.2 CCM* mode encryption and authentication transformation

Inputs:

The inputs to the CCM* mode forward transformation are as follows:

a) The key Key of size keylen = 128 bits to be used:

Key = C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF.

b) The nonce N of $15 – L = 13$ octets to be used:

Nonce = AC DE 48 00 00 00 00 01 || 00 00 00 05 || 02.

c) The octet string m of length $l(m) = 0$ octets to be used:

m = (empty string).

d) The octet string a of length $l(a) = 26$ octets to be used:

a = 08 D0 84 21 43 01 00 00 00 00 48 DE AC || 02 || 05 00 00 00 || 55 CF 00 00 51 52 53 54.

Actions:

The CCM* mode forward transformation involves the execution, in order, of an input transformation (C.2.1.2.1), an authentication transformation (C.2.1.2.2), and an encryption transformation (C.2.1.2.3).

C.2.1.2.1 Input transformation

This step involves the transformation of the input strings a and m to the strings AuthData and PlainTextData, to be used by the authentication transformation and the encryption transformation, respectively.

a) Form the octet string representation $L(a)$ of the length $l(a)$ of the octet string a:

$L(a)$ = 00 1A.

b) Right-concatenate the octet string $L(a)$ with the octet string a itself:

$L(a) \| a$ = 00 1A || 08 D0 84 21 43 01 00 00 00 00 48 DE AC 02 05 00 00 00 55 CF 00 00 51 52 53 54.

c) Form the padded message AddAuthData by right-concatenating the resulting string with the smallest non-negative number of all-zero octets so that the octet string AddAuthData has length divisible by 16:

AddAuthData = 00 1A 08 D0 84 21 43 01 00 00 00 00 48 DE AC 02 05 00 00 00 55 CF 00 00 51 52 53 54 00 00 00 00.

d) Form the padded message PlaintextData by right-concatenating the octet string m with the smallest non-negative number of all-zero octets so that the octet string PlaintextData has length divisible by 16:

PlaintextData = (empty string).

e) Form the message AuthData consisting of the octet strings AddAuthData and PlaintextData:

AuthData = 00 1A 08 D0 84 21 43 01 00 00 00 00 48 DE AC 02 05 00 00 00 55 CF 00 00 51 52 53 54 00 00 00 00.

C.2.1.2.2 Authentication transformation

The data AuthData that was established in C.2.1.2.1 shall be tagged using the tagging transformation as follows:

a) Form the 1-octet Flags field:

Flags = 59.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

b) Form the 16-octet B_0 field:

B_0 = 59 || AC DE 48 00 00 00 00 01 00 00 00 05 02 || 00 00.

c) Parse the message AuthData as B_1 || B_2, where each message block B_i is a 16-octet string.

d) The CBC-MAC value X_3 is computed as shown in Table C.1.

Table C.1—Computation of CBC-MAC value

i	B_i	X_i
0	59 AC DE 48 00 00 00 00 01 00 00 00 05 02 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1	00 1A 08 D0 84 21 43 01 00 00 00 00 48 DE AC 02	C4 A4 D0 BD 70 73 7E 32 11 2E 51 9A CA A2 01 F1
2	05 00 00 00 55 CF 00 00 51 52 53 54 00 00 00 00	A9 70 2C 6E E1 7E DE E0 C7 32 88 0A 40 41 7F 9C
3	—	AB 6B 19 E7 5B 75 2D 9A 6E F0 CC 13 09 98 EB D0

e) The authentication tag *T* is the result of omitting all but the leftmost *M* = 8 octets of the CBC-MAC value X_3:

T = AB 6B 19 E7 5B 75 2D 9A.

C.2.1.2.3 Encryption transformation

The data PlaintextData that was established in C.2.1.2.1 (step d), and the authentication tag *T* that was established in C.2.1.2.2 (step e) shall be encrypted using the encryption transformation as follows:

a) Form the 1-octet Flags field:

Flags = 01.

b) Define the 16-octet A_i field as shown in Table C.2.

Table C.2—A_i fields

i	A_i
0	01 AC DE 48 00 00 00 00 01 00 00 00 05 02 00 00

c) Define the 16-octet encryption block S_0:

S_0:= *E*(Key, A_0) = 89 50 D8 0B DF 6F 98 C9 63 F2 D5 A1 08 A1 55 C7.

d) The encrypted authentication tag *U* is the result of XOR-ing the string consisting of the leftmost *M* = 8 octets of S_0 and the authentication tag *T*:

U = 22 3B C1 EC 84 1A B5 53

Output:

The octet string *c* = 22 3B C1 EC 84 1A B5 53.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

C.2.1.3 CCM* mode decryption and authentication checking transformation

Inputs:

The inputs to the CCM* mode inverse transformation are as follows:

a) The key Key of size keylen = 128 bits to be used:

Key = C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF.

b) The nonce N of $15 - L = 13$ octets to be used:

Nonce = AC DE 48 00 00 00 00 01 || 00 00 00 05 || 02.

c) The octet string c of length $l(c) = 8$ octets to be used:

c = 22 3B C1 EC 84 1A B5 53.

d) The octet string a of length $l(a) = 26$ octets to be used:

a = 08 D0 84 21 43 01 00 00 00 00 48 DE AC || 02 || 05 00 00 00 || 55 CF 00 00 51 52 53 54.

Actions:

The CCM* mode inverse transformation involves the execution, in order, of a decryption transformation (C.2.1.3.1) and an authentication checking transformation (C.2.1.3.2).

C.2.1.3.1 Decryption transformation

The decryption transformation involves the following steps, in order:

a) Parse the message c as $C \| U$, where the rightmost string U is an 8-octet string:

C = (empty string);
U = 22 3B C1 EC 84 1A B5 53.

b) Form the 1-octet Flags field:

Flags = 01.

c) Define the 16-octet A_i field as shown in Table C.3.

Table C.3—A_i fields

i	A_i
0	01 AC DE 48 00 00 00 00 01 00 00 00 05 02 00 00

d) Define the 16-octet encryption block S_0:

$S_0 = E(\text{Key}, A_0)$ = 89 50 D8 0B DF 6F 98 C9 63 F2 D5 A1 08 A1 55 C7.

e) The purported authentication tag T is the result of XOR-ing the string consisting of the leftmost $M = 8$ octets of S_0 and the octet string U:

T = AB 6B 19 E7 5B 75 2D 9A.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

C.2.1.3.2 Authentication checking transformation

The authentication checking transformation involves the following steps, in order:

a) Form the message AuthData using the input transformation in C.2.1.2.1, with as inputs the string *a* and the octet string *m* = (empty string):

AuthData = 00 1A 08 D0 84 21 43 01 00 00 00 00 48 DE AC 02 05 00 00 00 55 CF
00 00 51 52 53 54 00 00 00 00.

b) Use the authentication transformation in C.2.1.2.2, with as input the message AuthData to compute the authentication tag MACTag:

MACTag = AB 6B 19 E7 5B 75 2D 9A.

c) Compare the output tag MACTag resulting from this transformation with the tag *T* that was established in C.2.1.3.1 (step e):

T = AB 6B 19 E7 5B 75 2D 9A = MACTag.

Output:

Because MACTag = *T*, output "valid," accept the octet strings *a* and *m*, and accept one of the key sharing group member(s) as the source of *a* and *m*.

C.2.2 MAC data frame

C.2.2.1 Description

The example below illustrates security processing of a data frame that is transmitted using extended addresses, with PAN identifier compression and acknowledgment enabled. This example uses source address 0xacde480000000001, destination address 0xacde480000000002, PAN identifier 0x4321, and data payload 0x61 0x62 0x63 0x64; the frame counter has integer value 5. The security level is set to 0x04 (ENC, or data confidentiality without data authenticity).

For simplicity, all frames in this example are shown without the FCS field (because security processing is independent of it).

Secured data frame:

69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 05 00 00 00 || D4 3E 02 2B.

Corresponding unsecured data frame:

61 CC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 61 62 63 64.

Prerequisite:

The following prerequisite is established for this mode of operation: the parameter *M* shall have the integer value 0.

C.2.2.2 CCM* mode encryption and authentication transformation

Inputs:

The inputs to the CCM* mode forward transformation are as follows:

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

a) The key Key of size keylen = 128 bits to be used:

Key = C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF.

b) The nonce N of $15 – L = 13$ octets to be used:

Nonce = AC DE 48 00 00 00 00 01 || 00 00 00 05 || 04.

c) The octet string m of length $l(m) = 4$ octets to be used:

m = 61 62 63 64.

d) The octet string a of length $l(a) = 26$ octets to be used:

a = 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 || 05 00 00 00.

Actions:

The CCM* mode forward transformation involves the execution, in order, of an input transformation (C.2.2.2.1), an authentication transformation (C.2.2.2.2), and an encryption transformation (C.2.2.2.3).

C.2.2.2.1 Input transformation

This step involves the transformation of the input strings a and m to the strings AuthData and PlainTextData, to be used by the authentication transformation and the encryption transformation, respectively.

a) Form the octet string representation $L(a)$ of the length $l(a)$ of the octet string a:

$L(a)$ = 00 1A.

b) Right-concatenate the octet string $L(a)$ and the octet string a itself:

$L(a) \| a$ = 00 1A || 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC 04 05 00 00 00.

c) Form the padded message AddAuthData by right-concatenating the resulting string with the smallest non-negative number of all-zero octets so that the octet string AddAuthData has length divisible by 16:

AddAuthData = 00 1A 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC 04 05 00 00 00 00 00 00 00.

d) Form the padded message PlaintextData by right-concatenating the octet string m with the smallest non-negative number of all-zero octets so that the octet string PlaintextData has length divisible by 16:

PlaintextData = 61 62 63 64 00 00 00 00 00 00 00 00 00 00 00 00.

e) Form the message AuthData consisting of the octet strings AddAuthData and PlaintextData:

AuthData = 00 1A 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC 04 05 00 00 00 00 00 00 00 61 62 63 64 00 00 00 00 00 00 00 00 00 00 00 00.

C.2.2.2.2 Authentication transformation

T = (empty string).

C.2.2.2.3 Encryption transformation

The data PlaintextData shall be encrypted using the encryption transformation as follows:

a) Form the 1-octet Flags field:

Flags = 01.

b) Define the 16-octet A_i field as shown in Table C.4.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

Table C.4—A_i fields

i	A_i
1	01 AC DE 48 00 00 00 00 01 00 00 00 05 04 00 01

c) Parse the message PlaintextData as M_1, where each message block M_i is a 16-octet string.

d) The ciphertext block C_1 is computed as shown in Table C.5.

Table C.5—Computation of ciphertext

i	AES(Key, A_i)	C_i = AES(Key, A_i) ⊕ M_i
1	B5 5C 61 4F A6 8B 7E E0 CB 77 37 EB A8 1D 33 41	D4 3E 02 2B A6 8B 7E E0 CB 77 37 EB A8 1D 33 41

e) The string Ciphertext is the result of omitting all but the leftmost $l(m) = 4$ octets of the string C_1:

CipherText = D4 3E 02 2B.

Output:

c = D4 3E 02 2B.

C.2.2.3 CCM* mode decryption and authentication checking transformation

Inputs:

The inputs to the CCM* mode inverse transformation are as follows:

a) The key Key of size keylen = 128 bits to be used:

Key = C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF.

b) The nonce N of $15 - L = 13$ octets to be used:

Nonce = AC DE 48 00 00 00 00 01 || 00 00 00 05 || 04.

c) The octet string c of length $l(c) = 4$ octets to be used:

c = D4 3E 02 2B.

d) The octet string a of length $l(a) = 26$ octets to be used:

a = 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC || 04 || 05 00 00 00.

Actions:

The CCM* mode inverse transformation involves the execution, in order, of a decryption transformation (C.2.2.3.1) and an authentication checking transformation (C.2.2.3.2).

C.2.2.3.1 Decryption transformation

The decryption transformation involves the following steps, in order:

a) Parse the message c as $C \| U$, where the rightmost string U is a 0-octet string:

C = D4 3E 02 2B.
U = (empty string).

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved

b) Form the padded message CiphertextData by right-concatenating the string *C* with the smallest non-negative number of all-zero octets so that the octet string CiphertextData has length divisible by 16:

CipherTextData = D4 3E 02 2B 00 00 00 00 00 00 00 00 00 00 00 00.

c) Form the 1-octet Flags field:

Flags = 01.

d) Define the 16-octet A_i field as shown in Table C.6.

Table C.6—A_i fields

i	A_i
1	01 AC DE 48 00 00 00 00 01 00 00 00 05 04 00 01

e) Parse the message CiphertextData as C_1, where each message block C_i is a 16-octet string.

f) The plaintext block P_1 is computed as shown in Table C.7.

Table C.7—Computation of plaintext

i	AES(Key, A_i)	P_i = AES(Key, A_i) ⊕ C_i
1	B5 5C 61 4F A6 8B 7E E0 CB 77 37 EB A8 1D 33 41	61 62 63 64 00 00 00 00 00 00 00 00 00 00 00 00

g) The octet string *m* is the result of omitting all but the leftmost *l*(*m*) = 4 octets of the string P_1:

m = 61 62 63 64.

h) The purported authentication tag *T* is the empty string:

T = (empty string).

C.2.2.3.2 Authentication checking transformation

The authentication checking transformation involves the following steps, in order:

a) Form the message AuthData using the input transformation in C.2.2.2.1, with as inputs the string *a* and the octet string *m* that was established in C.2.2.3.1 (step g):

AuthData = 00 1A 69 DC 84 21 43 02 00 00 00 00 48 DE AC 01 00 00 00 00 48 DE AC
04 05 00 00 00 00 00 00 00 61 62 63 64 00 00 00 00 00 00 00 00 00 00 00 00.

b) Use the authentication transformation in C.2.2.2.2, with as input the message AuthData to compute the authentication tag MACTag:

MACTag = (empty string).

c) Compare the output tag MACTag resulting from this transformation with the tag *T* that was established in C.2.2.3.1 (step h):

T = (empty string) = MACTag.

Output:

Because MACTag = *T*, output "valid," accept the octet strings *a* and *m*, and accept one of the key sharing group member(s) as the source of *a* and *m*.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2010

© IEEE 2010 – All rights reserved