

---

---

**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)*



IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018



**COPYRIGHT PROTECTED DOCUMENT**

© IEEE 2016

All rights reserved. Unless otherwise specified, or required in the context of its implementation, 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 ISO, IEC or IEEE at the respective address below.

ISO copyright office  
Ch. de Blandonnet 8 • CP 401  
CH-1214 Vernier, Geneva, Switzerland  
Tel. +41 22 749 01 11  
Fax +41 22 749 09 47  
copyright@iso.org  
www.iso.org

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

stds.ipr@ieee.org  
www.ieee.org

## 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 of the IEEE Computer Society (as IEEE 802.15.4-2015). 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.

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 8802-15-4:2010), which has been technically revised.

A list of all parts in the ISO/IEC/IEEE 8802 series can be found on the ISO website.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018

**IEEE Std 802.15.4™-2015**

(Revision of  
IEEE Std 802.15.4-2011)

# IEEE Standard for Low-Rate Wireless Networks

Sponsor

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

Approved 5 December 2015

**IEEE-SA Standards Board**

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018

**Abstract:** 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) are defined in this standard. A variety of physical layers (PHYs) have been defined that cover a wide variety of frequency bands.

**Keywords:** ad hoc network, IEEE 802.15.4™, 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 © 2016 by The Institute of Electrical and Electronics Engineers, Inc.  
All rights reserved. Published 22 April 2016. Printed in the United States of America.

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

Print: ISBN 978-1-5044-0854-5 STD20893  
PDF: ISBN 978-1-5044-0846-2 STDPD20893

*IEEE prohibits discrimination, harassment and bullying.*

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

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.

## Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

### Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

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

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. 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.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

## Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

## Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. 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 of IEEE.

## Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

## Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

## Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.



## Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, 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.

## Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, 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 order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at <http://ieeexplore.ieee.org/Xplore/home.jsp> or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

## Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

## 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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, 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 the IEEE Standards Association.

## Participants

At the time this standard was sent to sponsor ballot, the IEEE P802.15 Working Group had the following voting members:

**Robert F. Heile, Chair**  
**Rick Alfvén, Co-Vice Chair**  
**Patrick W. Kinney, Co-Vice Chair**  
**James P. K. Gilb, Working Group Technical Editor**  
**Patrick W. Kinney, Secretary**

**Patrick W. Kinney, 802.15 Maintenance Committee Chair**  
**Benjamin A. Rolfe, 802.15 Maintenance Committee Vice Chair**  
**James P. K. Gilb, Technical Editor**

|                           |                    |                          |
|---------------------------|--------------------|--------------------------|
| Mounir Achir              | Shinsuke Hara      | Robert Moskowitz         |
| Keiji Akiyama             | Timothy Harrington | Jinesh P. Nair           |
| Arthur Astrin             | James Hartman      | Chiu Ngo                 |
| Philip Beecher            | Marco Hernandez    | Paul Nikolich            |
| Frederik Beer             | Ken Hiraga         | John Notor               |
| Chandrashekhar P. S. Bhat | Iwao Hosako        | Hiroyo Ogawa             |
| Kiran Bynam               | Yeong Min Jang     | Taejoon Park             |
| Edgar Callaway            | Seong-Soon Joo     | Glenn Parsons            |
| Chris Calvert             | Akifumi Kasamatsu  | Charles Perkins          |
| Radhakrishna Canchi       | Shuzo Kato         | Albert Petrick           |
| Kapseok Chang             | Toyoyuki Kato      | Clinton C. Powell        |
| Soo-Young Chang           | Jeritt Kent        | Verotiana Rabarijaona    |
| Clint Chaplin             | Jaehwan Kim        | Ivan Reede               |
| Stephen Chasko            | Youngsoo Kim       | Richard Roberts          |
| Paul Chilton              | Shoichi Kitazawa   | Ruben E. Salazar Cardozo |
| Sangsung Choi             | Tero Kivinen       | Noriyuki Sato            |
| Hendricus de Ruijter      | Ryuji Kohno        | Norihiko Sekine          |
| Guido Dolmans             | Fumihide Kojima    | Kunal Shah               |
| Igor Dotlic               | Thomas Kuerner     | Stephen Shellhammer      |
| Stefan Drude              | Byung-Jae Kwak     | Shusaku Shimada          |
| Dietmar Eggert            | Jae Seung Lee      | Gary Stuebing            |
| Shahriar Emami            | Myung Lee          | Don Sturek               |
| Andrew Estrada            | Sangjae Lee        | Mineo Takai              |
| David Evans               | Huan-Bang Li       | Kou Togashi              |
| George Flammer            | Liang Li           | Billy Verso              |
| Kiyoshi Fukui             | Qing Li            | Gabriel Villardi         |
| Matthew Gillmore          | Michael Lynch      | Brian Weis               |
| Tim Godfrey               | Itaru Maekawa      | Makoto Yaita             |
| Elad Gottlieb             | Hiroyuki Matsumura | Peter Yee                |
| Jussi Haapola             | Michael McLaughlin | Yu Zeng                  |
| Rainer Hach               | Michael McInnis    | Chunhui (Allan) Zhu      |
|                           | Kenichi Mori       |                          |

Major contributions in the revision process were received from the following individuals:

|                    |                       |                          |
|--------------------|-----------------------|--------------------------|
| Philip Beecher     | Jeritt Kent           | Benjamin A. Rolfe        |
| Monique Brown      | Patrick W. Kinney     | Ruben E. Salazar Cardozo |
| Edgar Callaway     | Tero Kivinen          | Cristina Seibert         |
| Soo-Young Chang    | Amarjeet Kumar        | Kunal Shal               |
| James P. K. Gilb   | Michael McInnis       | Shusakh Shimada          |
| Tim Godfrey        | Robert Moskowitz      | Chin-Sean Sum            |
| Jussi Haapola      | Clinton C. Powell     | Larry Taylor             |
| Timothy Harrington | Verotiana Rabarijaona | Billy Verso              |

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

|                      |                      |                          |
|----------------------|----------------------|--------------------------|
| Rick Alfvén          | Noriyuki Ikeuchi     | Yoshihiro Ohba           |
| Nobumitsu Amachi     | Akio Iso             | Okundu Omeni             |
| Butch Anton          | Atsushi Ito          | Satoshi Oyama            |
| Stefan Aust          | Raj Jain             | Stephen Palm             |
| Michael Bahr         | Oyvind Janbu         | Arumugam Paventhan       |
| Madhusudan Banavara  | Michael Johas Teener | Dalibor Pokrajac         |
| Philip E. Beecher    | Adri Jovin           | Clinton C. Powell        |
| Harry Bims           | Naveen Kakani        | Venkatesha Prasad        |
| Gennaro Boggia       | Shinkyo Kaku         | Verotiana Rabarijaona    |
| Riccardo Brama       | Piotr Karocki        | Demir Rakanovic          |
| Nancy Bravin         | Ruediger Kays        | R. K. Rannow             |
| Vern Brethour        | Jeritt Kent          | Maximilian Riegel        |
| Monique Brown        | Stuart Kerry         | Robert Robinson          |
| William Byrd         | Yongbum Kim          | Osman Sakr               |
| Brent Cain           | Patrick W. Kinney    | Ruben E. Salazar Cardozo |
| Edgar Callaway       | Tero Kivinen         | John Santhoff            |
| Radhakrishna Chanchi | Fumihide Kojima      | Naotaka Sato             |
| Juan Carreon         | Bruce Kraemer        | Bartien Sayogo           |
| Suresh Channarasappa | Yasushi Kudoh        | Kunal Shah               |
| Keith Chow           | Amarjeet Kumar       | Shusaku Shimada          |
| Charles Cook         | Jeremy Landt         | Suresh Shrivastava       |
| Steven Crowley       | Hyeong Ho Lee        | Daniel Smolinski         |
| Sourav Dutta         | James Lepp           | Thomas Starai            |
| Richard Edgar        | Arthur H. Light      | Rene Struik              |
| David Evans          | Vincent Lipsio       | Walter Struppler         |
| Liu Fangfang         | William Lumpkins     | Patricia Thaler          |
| Michael Fischer      | Elvis Maculuba       | David Thompson           |
| Avraham Freedman     | James Marin          | Ha-Nguyen Tran           |
| Devon Gayle          | Roger Marks          | Mark-Rene Uchida         |
| James P. K. Gilb     | Michael McInnis      | Lorenzo Vangelista       |
| Gregory Gillooly     | Michael McLaughlin   | Dmitri Varsanofiev       |
| Joel Goergen         | Apurva Mody          | Prabodh Varshney         |
| Randall Groves       | Jose Morales         | Raj Vaswani              |
| Chris Guy            | Ronald Murias        | George Vlantis           |
| Rainer Hach          | Jinesh Nair          | Khurram Waheed           |
| Timothy Harrington   | Juichi Nakada        | Hung-Yu Wei              |
| Marco Hernandez      | Michael Newman       | Brian Weis               |
| Guido Hiertz         | Chiu Ngo             | Andreas Wolf             |
| Werner Hoelzl        | Nick S. A. Nikjoo    | Oren Yuen                |
| Tetsushi Ikegami     | Paul Nikolich        | Daidi Zhong              |

When the IEEE-SA Standards Board approved this standard on 5 December 2015, it had the following membership:

**John Kulick, Chair**  
**Jon Walter Rosdahl, Vice Chair**  
**Richard H. Hulet, Past Chair**  
**Konstantinos Karachalios, Secretary**

|                      |                       |                        |
|----------------------|-----------------------|------------------------|
| Masayuki Ariyoshi    | Joseph L. Koepfinger* | Stephen J. Shellhammer |
| Ted Burse            | David J. Law          | Adrian P. Stephens     |
| Stephen Dukes        | Hung Ling             | Yatin Trivedi          |
| Jean-Phillippe Faure | Andrew Myles          | Phillip Winston        |
| J. Travis Griffith   | T. W. Olsen           | Don Wright             |
| Gary Hoffman         | Glenn Parsons         | Yu Yuan                |
| Michael Janezic      | Ronald C. Peterson    | Daidi Zhong            |
|                      | Annette D. Reilly     |                        |

\*Member Emeritus

## Historical participants

Many individuals have participated in the IEEE P802.15 Working Group during various stages of the standard's development. Since the initial publication, many amendments have added functionality or updated material in this standard, and now three revisions have been published. Here is a historical list of the working group participants who dedicated their valuable time, energy, and knowledge to the advancement of this standard at the time of its original publication and for its revisions. Many of these members also worked on amendments.

The following members of the IEEE P802.15 Working Group participated in the development of the original standard, IEEE Std 802.15.4-2003:

**Robert F. Heile**, *Chair*  
**James D. Allen**, *Vice Chair*  
**Patrick W. Kinney**, *Secretary*  
**Michael D. McInnis**, *Assistant Secretary and Editor*

**Ian C. Gifford**, *Task Group 1 Chair*  
**Stephen J. Shellhammer**, *Task Group 2 Chair*  
**John R. Barr**, *Task Group 3 Chair*

**Patrick W. Kinney**, *Task Group 4 Chair*  
**Phil Jamieson**, *Task Group 4 Vice Chair*  
**José A. Gutierrez**, *Task Group 4 Editor-in-Chief*  
**Marco Naeve**, *Task Group 4 Secretary*

**Monique Bourgeois**, *MAC Technical Editor*  
**Said Moridi**, *PHY Technical Editor*  
**Phil Jamieson**, *Layer Management Technical Editor*  
**Greg Breen**, *Low-Band PHY Technical Editing*  
**Ed Callaway**, *Networking Technical Editing*  
**Paul Gorday**, *High-Band PHY Technical Editing*  
**Marco Naeve**, *General Description Technical Editing*  
**David Cypher**, *PICs/SDLs Technical Editing*  
**Robert D. Poor**, *Coexistence Technical Editing*  
**Farron Dacus**, *Regulatory Technical Editing*

Roberto Aiello  
Masaaki Akahane  
Richard Alfvén  
Arun Arunachalam  
Naiel Askar  
Venkat J. Bahl  
Daniel Bailey  
Jay Bain  
James Baker  
Jaiganesh Balakrishnan  
Anuj Batra  
Timothy Blaney  
Kenneth Boehike  
Stan Bottoms  
Mark V. Bowles  
Chuck Brabenac  
Soo-Young Chang  
Francois Po-Shin Chin  
Aik Chindapol

Craig Conkling  
Anand Dabak  
Kai Dombrowski  
Mary DuVal  
Michael Dydyk  
Jason L. Ellis  
Mark W. Fidler  
Jeff R. Foerster  
David S. Furuno  
Pierre Gandolfo  
Atul Garg  
James Gilb  
Nada Golmie  
Yasuo Harada  
Allen Heberling  
Barry Herold  
Robert Y. Huang  
Eran Igler  
Katsumi Ishii

Jeyhan Karaoguz  
Masami Katagiri  
Joy H. Kelly  
Stuart J. Kerry  
Yongsuk Kim  
Young Hwan Kim  
Günter Kleindl  
Bruce P. Kraemer  
DoHoon Kwon  
Jim Lansford  
David Leeper  
Liang Li  
Yeong-Chang Maa  
Steven March  
Ralph Mason  
Jim Meyer  
Leonard E. Miller  
Akira Miura  
Andreas Molisch

Antonio Mondragon  
 Tony Morelli  
 Chiu Ngo  
 Kei Obara  
 Knut Odman  
 John B. Pardee  
 Jongun Park  
 Dave Patton  
 Marcus Pendergrass  
 Gregg Rasor  
 Ivan Reede  
 Jim Richards  
 Glyn Roberts  
 Richard Roberts  
 William Roberts  
 Chris Rogers  
 Philippe Rouzet

Chandos Rypinski  
 John H. Santhoff  
 Mark Schrader  
 Tom Schuster  
 Erik Schylander  
 Michael Seals  
 Nick Shepherd  
 Gadi Shor  
 William Shvodian  
 Thomas Siep  
 Kazimierz Siwiak  
 Carl Stevenson  
 Rene Struik  
 Shigeru Sugaya  
 Kazuhisa Takamura  
 Katsumi Takaoka

Teik-Kheong Tan  
 Larry Taylor  
 Stephen E. Taylor  
 Hans vanLeeuwen  
 Ritesh Vishwakarma  
 Thierry Walrant  
 Jing Wang  
 Fijio Watanabe  
 Mathew Welborn  
 Richard Wilson  
 Stephen Wood  
 Edward G. Woodrow  
 Hirohisa Yamaguchi  
 Amos Young  
 Song-Lin Young  
 Nakache Yves-paul  
 Jim Zyren

Major contributions were received from the following individuals:

Tony Adamson  
 David Archer  
 David Avery  
 Venkat Bahl  
 Daniel Bailey  
 Edul Batliwala  
 Pratik Bose  
 Boaz Carmeli  
 Farron Dacus  
 Martin Digon  
 Ian C. Gifford

Ed Hogervorst  
 Stephen Korfhage  
 Charles Luebke  
 Masahiro Maeda  
 Ian Marsden  
 Chris Marshall  
 Paul Marshall  
 Fred Martin  
 Ralph Mason  
 Rod Miller  
 Phil Rudland

Niels Schutten  
 Nick Shepherd  
 Ari Singer  
 Ralph D'Souza  
 Carl Stevenson  
 Mark Tilinghast  
 Hans Van Leeuwen  
 Jacco van Muiswinkel  
 Luis Pereira  
 Richard Wilson  
 Wim Zwart

The following members of the IEEE P802.15 Working Group participated in the development of the first revision, IEEE Std 802.15.4-2006:

**Robert F. Heile, Chair**  
**James D. Allen, Vice Chair**  
**Patrick W. Kinney, Assistant Vice Chair**  
**James P. K. Gilb, Editor-in-Chief**  
**Patrick W. Kinney, Secretary**  
**Michael D. McInnis, Assistant Secretary and Editor**

**John R. Barr, Task Group 3b Chair**  
**Reed Fisher, Task Group 3c Chair**  
**Patrick W. Kinney, Task Group 4a Chair**  
**Myung Lee, Task Group 5 Chair**

**Robert D. Poor, Task Group 4b Chair**  
**Marco Naeve, Task Group 4b Vice Chair**  
**Monique B. Brown, Task Group 4b Editor-in-Chief**  
**Eric T. Gnoske, Task Group 4b Secretary**

**Philip E. Beecher, MAC Contributing Editor**  
**Monique B. Brown, MAC Technical Editor**  
**Edgar H. Callaway, Jr., MAC Contributing Editor**  
**Francois Chin, PHY Contributing Editor**  
**Robert C. Cragie, MAC/Security Contributing Editor**

**Paul Gorday**, *PHY Contributing Editor*  
**James P. K. Gilb**, *Draft D3 Editor-in-Chief*  
**Øyvind Janbu**, *MAC/PHY/Security Contributing Editor*  
**Marco Naeve**, *General Description/PICS Editor, MAC Contributing Editor*  
**Clinton C. Powell**, *PHY Technical Editor*  
**Joseph Reddy**, *Security Contributing Editor*  
**Zachary Smith**, *MAC Contributing Editor*  
**René Struik**, *Security Contributing Editor*  
**Andreas C. Wolf**, *PHY Contributing Editor*

Roberto Aiello  
Richard Alfvén  
Mikio Aoki  
Takashi Arita  
Larry Arnett  
Arthur Astrin  
Yasaman Bahreini  
Jay Bain  
Alan Berkema  
Bruce Bosco  
Mark Bowles  
Charles Brabenac  
David Brenner  
Vern Brethour  
Ronald Brown  
Bill Carney  
Kuor-Hsin Chang  
Jonathon Cheah  
Kwan-Wu Chin  
Sarm-Goo Cho  
Sungsoo Choi  
Yun Choi  
Chun-Ting Chou  
Manoj Choudhary  
Celestino Corral  
Joe Decuir  
Javier Del Prado Pavon  
Kai Dombrowski  
Stefan Drude  
Amal Ekbal  
Jason Ellis  
Shahriar Emami  
Paul Everest  
Mark W. Fidler  
Kris Fleming  
Amir Freund  
Camillo Gentile  
Ian Gifford  
Sung-Wook Goh  
Sorn Goldenberg  
Vivek Gupta  
Rainer Hach  
Robert Hall  
Shinsuke Hara  
Jeff Harris  
Allen Heberling  
Eric Heinze  
Barry Herold  
Keisuke Higuchi  
Jin-Meng Ho  
Patrick Houghton  
Robert Huang

Tian-Wei Huang  
Hideto Ikeda  
Tetsushi Ikegami  
Adrian Jennings  
Ho-In Jeon  
Tzyy Hong Jiang  
David Julian  
Jeyhan Karaoguz  
Michael Kelly  
Stuart Kerry  
Jae-Hyon Kim  
Jaeyoung Kim  
Jinkyong Kim  
Yongsuk Kim  
Kursat Kimyacioglu  
Matthias Kindler  
Guenter Kleindl  
Ryuji Kohno  
Mike Krell  
Yasushi Kudo  
Akiomi Kunisa  
Yuzo Kuramochi  
Jiun-You Lai  
Ismail Lakkis  
John Lampe  
Kyung Kuk Lee  
Wooyong Lee  
David Leeper  
Huan-Bang Li  
Haixiang Liang  
Ian Macnamara  
Akira Maeki  
Patricia Martigne  
Abbie Mathew  
Taisuke Matsumoto  
Gustaf Max  
Michael McLaughlin  
Charlie Mellone  
Klaus Meyer  
Samuel Mo  
Andreas Molisch  
Mark Moore  
Ken Naganuma  
Yves-Paul Nakache  
Hiroyuki Nakase  
Saishankar Nandagopalan  
Chiu Ngo  
Erwin Noble  
John O'Connor  
Knut Odman  
Hiroyo Ogawa  
Yasuyuki Okuma

Philip Orlik  
Laurent Ouvry  
John Pardee  
Nirmalendu Patra  
Dave Patton  
Xiaoming Peng  
Tony Pollock  
Vidyasagar Premkumar  
Yihong Qi  
Raad Raad  
Pekka Ranta  
Dani Raphaeli  
Gregg Rasor  
Charles Razzell  
Ivan Reede  
Yuko Rikuta  
Terry Robar  
Glyn Roberts  
Richard Roberts  
Benjamin A. Rolfé  
Philippe Rouzet  
Chandos Rypinski  
Ali Sadri  
Saeid Safavi  
Zafer Sahinoglu  
Tomoki Saito  
Syed Saleem  
Kamran Sayrafian  
Jean Schwoerer  
Erik Schylander  
Alireza Seyedi  
Sanjeev Sharma  
Siddharth Shetty  
John Shi  
Shusaku Shimada  
Yuichi Shiraki  
Gadi Shor  
William Shvodian  
Thomas Siep  
Michael Sim  
Kazimierz Siwiak  
V. Somayazulu  
Amjad Soomro  
Carl Stevenson  
Kazuaki Takahashi  
Kenichi Takizawa  
Teik-Kheong Tan  
Mike Tanahashi  
Yasushi Tanaka  
James Taylor  
Arnaud Tonnerre  
Ichihiko Toyoda

Jerry Upton  
Bart Van Poucke  
Chris Weber  
Matthew Welborn  
Magnus Wiklund

Gerald Wineinger  
Patrick Worfolk  
Tracy Wright  
Hirohisa Yamaguchi  
Kamya Yekeh Yazdandoost

Su-Khiong Yong  
Zhan Yu  
Serdar Yurdakul  
Mahmoud Zadeh  
Bin Zhen

Major contributions in the revision process were received from the following individuals:

Jon Adams  
Helmut P. Adamski  
Jonathan Avey  
Jon Beniston  
Bernd Grohmann  
José A. Gutierrez  
Jesper Holm  
ZhiJian Hu  
Phil A. Jamieson  
Yuen-Sam Kwok

Colin Lanzl  
Myung Lee  
Zhongding Lei  
Liang Li  
Yong Liu  
Frederick Martin  
Frank Poegel  
Matthias Scheide  
D. C. Seward

Huai-Rong Shao  
Mark Shea  
Stephen J. Shellhammer  
Mark A. Tillinghast  
Johannes Van Leeuwen  
Richard Wilson  
Ping Xiong  
Bing Xu  
Chen Yang Yang  
Chunhui Zhu

The following members of the IEEE P802.15 Working Group participated in the development of the second revision, IEEE Std 802.15.4-2011:

**Robert F. Heile, Chair**

**Rick Alfvin, Co-Vice Chair**

**Patrick W. Kinney, Co-Vice Chair**

**James P. K. Gilb, Working Group Technical Editor**

**Patrick W. Kinney, Secretary**

**James P. K. Gilb, Task Group 4i Chair and Technical Editor**

Emad Afifi  
Gahng-Seop Ahn  
Roberto Aiello  
Arthur Astrin  
Taehan Bae  
Michael Bahr  
John Barr  
Anuj Batra  
Tuncer Baykas  
Philip E. Beecher  
Ashutosh Bhatia  
Ghulam Bhatti  
Gary Birk  
Mathew Boytim  
Peter David Bradley  
Nancy Bravin  
David Britz  
Monique B. Brown  
Sverre Brubk  
Brian Buchanan  
John Buffington  
Kiran Bynam  
Brent Cain  
Edgar H. Callaway  
Chris Calvert

Ruben E. Salazar Cardozo  
Douglas Castor  
Jaesang Cha  
Russell Chandler  
Kuo-Hsin Chang  
Soo-Young Chang  
Clint Chaplin  
Hind Chebbo  
Chang-Soon Choi  
Sangsung Choi  
Ciaran Connell  
David Cypher  
Matthew Dahl  
David Davenport  
Mark Dawkins  
Hendricus de Ruijter  
Upkar Dhaliwal  
Gang Ding  
Paul Dixon  
Guido Dolmans  
Igor Dotlic  
Michael Dow  
Dietmar Eggert  
David Evans  
Charles Farlow

John Farserotu  
Jeffrey Fischbeck  
Mike Fischer  
George Flammer  
Ryosuke Fujiwara  
Noriyasu Fukatsu  
Kiyoshi Fukui  
John Geiger  
Gregory Gillooly  
Tim Godfrey  
Paul Gorday  
Elad Gottlieb  
Robert Hall  
Shinsuke Hara  
Hiroshi Harada  
Timothy Harrington  
Rodney Hemminger  
Marco Hernandez  
Garth Hillman  
Jin-Meng Ho  
Wei Hong  
Srinath Hosur  
David Howard  
Jung-Hwan Hwang  
Taeho Hwang



|                    |                    |                     |
|--------------------|--------------------|---------------------|
| Ichirou Ida        | Robert Moskowitz   | Steve Shearer       |
| Tetsushi Ikegami   | Hamilton Moy       | Stephen Shellhammer |
| Akio Iso           | Peter Murray       | Shusaku Shimada     |
| Yeong Min Jang     | Theodore Myers     | Chang Sub Shin      |
| Adrian Jennings    | Chiu Ngo           | Cheol Ho Shin       |
| Wuncheol Jeong     | Paul Nikolich      | Michael Sim         |
| Steven Jillings    | Hirohito Nishiyama | Jonathan Simon      |
| Noh-Gyoung Kang    | David Olson        | Jaeseung Son        |
| Tae-Gyu Kang       | Okundu Omeni       | Paul Stadnik        |
| Shuzo Kato         | Ryoji Ono          | René Struik         |
| Tatsuya Kato       | Laurent Ouvry      | Chin-Sean Sum       |
| Jeritt Kent        | James Pace         | Hui-Hsia Sung       |
| Prithpal Khakuria  | Hyung-Il Park      | Gu Sungi            |
| Dae Ho Kim         | Jahng Park         | Kenichi Takizawa    |
| Dong-Sun Kim       | Seung-Hoon Park    | Hirokazu Tanaka     |
| Dukhyun Kim        | Taejoon Park       | Larry Taylor        |
| Jaehwan Kim        | Ranjeet Patro      | Mark Thompson       |
| Jeffrey King       | Al Petrick         | James Tomcik        |
| Ryuji Kohno        | Dalibor Pokrajac   | Ichihiko Toyoda     |
| Fumihide Kojima    | Daniel Popa        | David Tracey        |
| Bruce Kraemer      | Stephen Pope       | Khanh Tran          |
| Raymond Krasinski  | Clinton C. Powell  | Jerry Upton         |
| Masahiro Kuroda    | Richard Powell     | Jana van Greunen    |
| John Lampe         | Chang-Woo Pyo      | Hartman van Wyk     |
| Zhou Lan           | Mohammad Rahman    | Michel Veillette    |
| Khanh Le           | Sridhar Rajagopal  | Billy Verso         |
| Cheolhyo Lee       | Jayaram Ramasastry | Bhupender Virk      |
| Hyungsoo Lee       | Marc Reed          | Joachim Walewski    |
| Myung Lee          | Ivan Reede         | Junyi Wang          |
| Daniel Lewis       | Richard Roberts    | Quan Wang           |
| Huan-Bang Li       | Craig Rodine       | Xiang Wang          |
| Liang Li           | June Chul Roh      | Andy Ward           |
| Sang-Kyu Lim       | Benjamin A. Rolfe  | Scott Weikel        |
| Jeremy Link        | Seung-Moon Ryu     | Nicholas West       |
| Mike Lynch         | Didier Sagan       | Mark Wilbur         |
| Robert Mason       | Kentaro Sakamoto   | Ludwig Winkel       |
| Tomokuni Matsumura | Will San Filippo   | Eun Tae Won         |
| Jeff McCullough    | H. Sanderford      | Alan Chi Wai Wong   |
| Michael McGillan   | Kamran Sayrafian   | Tao Xing            |
| Michael D. McInnis | Timothy Schmidl    | Wen-Bin Yang        |
| Michael McLaughlin | Michael Schmidt    | Yang Yang           |
| Charles Millet     | Jean Schwoerer     | Kazuyuki Yasukawa   |
| Siamak Mirnezami   | Cristina Seibert   | Kamya Yazdandoost   |
| Rishi Mohindra     | Neal Seidl         | Kaoru Yokoo         |
| Emmanuel Monnerie  | Kunal Shah         | Mu Zhao             |
| Rajendra Moorti    |                    | Bin Zhen            |

Major contributions in the revision process were received from the following individuals:

|                   |                    |                   |
|-------------------|--------------------|-------------------|
| Philip E. Beecher | Clint Chaplin      | Clinton C. Powell |
| Vern Brethour     | James P. K. Gilb   | Benjamin A. Rolfe |
| Monique B. Brown  | Patrick W. Kinney  | Timothy Schmidl   |
| Edgar H. Callaway | Michael D. McInnis | René Struik       |
| Kuor-Hsin Chang   |                    | Billy Verso       |



## Introduction

This introduction is not part of IEEE Std 802.15.4™-2015, IEEE Standard for Low-Rate Wireless Networks.

This is the third revision of IEEE Std 802.15.4. From the beginning, the goal of the IEEE P802.15 Working Group was to produce a standard that enabled very low-cost, low-power communications. The initial standard, IEEE Std 802.15.4-2003, defined two optional physical layers (PHYs), operating in different frequency bands with a simple and effective medium access control (MAC).

In 2006, the standard was revised and added two more PHY options. The MAC remained backward compatible, but the revision added MAC frames with an increased version number and a variety of MAC enhancements, including the following:

- Support for a shared time base with a data time stamping mechanism
- Support for beacon scheduling
- Synchronization of broadcast messages in beacon-enabled personal area networks (PANs)
- Improved MAC layer security

In 2011, the standard was revised to include the three amendments approved subsequent to the 2006 revision. This effort added four more PHY options along with the MAC capability to support ranging. Additionally, the organization of the standard was changed so that each PHY would have a separate clause, and the MAC clause was split into functional description, interface specification, and security specification.

The current revision of the standard was created to roll in the amendments approved subsequent to the 2011 revision: six PHY amendments and one MAC amendment, with corrigenda and clarifications. The features added by the amendments include the following:

- Enhanced frame formats maintaining backward compatibility
- Information Elements (IEs)
- Channel agility
- Extended superframe options
- Low-energy mechanisms
- An enhanced acknowledgment frame that can carry data and can be secured
- Prioritized channel access
- A variety of new PHY modulation, coding, and band options to support a wide variety of application needs including radio frequency identification (RFID), smart utility networks (SUNs), television white space (TVWS) operation, low-energy critical infrastructure monitoring (LECIM), and rail communications and control (RCC).

Much of the corrigenda and clarifications were collected from requests from individuals after the revision in 2011. Major corrigenda items included changes to the security text to correct errors and clarify the text, removal of the encrypt only mode, addition of security policy checks for the IEs, corrections regarding personal area network identifier (PAN ID) compression behavior to eliminate ambiguous specification, and changes to the IEs subclauses to include more information necessary for users of this standard.

The Project Authorization Request (PAR) for IEEE Std 802.15.4-2015 was first proposed in July 2013 and was approved in October 2013 by IEEE's New Standards Committee (NesCom). After three working group ballots and two sponsor ballots, the final standard was approved in December 2015, just over two years from start to finish.

## Contents

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 1. Overview .....                                                  | 35 |
| 1.1 Scope .....                                                    | 35 |
| 1.2 Purpose .....                                                  | 35 |
| 2. Normative references .....                                      | 36 |
| 3. Definitions, acronyms, and abbreviations .....                  | 37 |
| 3.1 Definitions .....                                              | 37 |
| 3.2 Acronyms and abbreviations .....                               | 38 |
| 4. Format conventions .....                                        | 42 |
| 4.1 General .....                                                  | 42 |
| 4.2 Fields .....                                                   | 42 |
| 4.3 Numbers .....                                                  | 43 |
| 4.4 Strings .....                                                  | 43 |
| 4.5 Reserved fields and values .....                               | 43 |
| 5. General description .....                                       | 44 |
| 5.1 Introduction .....                                             | 44 |
| 5.2 Special application spaces .....                               | 44 |
| 5.2.1 Smart utility network (SUN) .....                            | 44 |
| 5.2.2 Rail communications and control (RCC) .....                  | 44 |
| 5.2.3 Television white space (TVWS) .....                          | 45 |
| 5.2.4 Radio frequency identification (RFID) .....                  | 45 |
| 5.2.5 Low-energy, critical infrastructure monitoring (LECIM) ..... | 45 |
| 5.2.6 Medical body area network (MBAN) services .....              | 45 |
| 5.3 Components of the IEEE 802.15.4 WPAN .....                     | 45 |
| 5.4 Multi-PHY management (MPM) of the SUN WPAN .....               | 45 |
| 5.5 Network topologies .....                                       | 46 |
| 5.5.1 Star network formation .....                                 | 46 |
| 5.5.2 Peer-to-peer network formation .....                         | 47 |
| 5.6 Architecture .....                                             | 48 |
| 5.6.1 PHY .....                                                    | 49 |
| 5.6.2 MAC sublayer .....                                           | 49 |
| 5.7 Functional overview .....                                      | 50 |
| 5.7.1 Superframe structure .....                                   | 50 |
| 5.7.1.1 Beacon superframe .....                                    | 50 |
| 5.7.1.2 DSME multi-superframe structure .....                      | 51 |
| 5.7.1.3 Slotframes .....                                           | 51 |
| 5.7.1.4 TMCTP superframe .....                                     | 51 |
| 5.7.2 Data transfer model .....                                    | 52 |
| 5.7.2.1 Types of data transfer models .....                        | 52 |
| 5.7.2.2 Data transfer to a coordinator .....                       | 52 |
| 5.7.2.3 Data transfer from a coordinator .....                     | 52 |
| 5.7.2.4 Peer-to-peer data transfers .....                          | 53 |
| 5.7.3 Frame structure .....                                        | 53 |
| 5.7.4 Access methods .....                                         | 53 |

|         |                                                                           |    |
|---------|---------------------------------------------------------------------------|----|
| 5.7.4.1 | Frame acknowledgment .....                                                | 53 |
| 5.7.4.2 | Frak .....                                                                | 54 |
| 5.7.4.3 | Data verification .....                                                   | 54 |
| 5.7.5   | Power consumption considerations .....                                    | 54 |
| 5.7.5.1 | Low-energy mechanisms .....                                               | 54 |
| 5.7.6   | Security .....                                                            | 55 |
| 5.8     | Concept of primitives .....                                               | 56 |
| 5.9     | Deprecation of features .....                                             | 56 |
| 6.      | MAC functional description .....                                          | 57 |
| 6.1     | Device types and conventions .....                                        | 57 |
| 6.2     | Channel access .....                                                      | 57 |
| 6.2.1   | Superframe structure .....                                                | 57 |
| 6.2.1.1 | Contention access period (CAP) .....                                      | 59 |
| 6.2.1.2 | Contention-free period (CFP) .....                                        | 59 |
| 6.2.1.3 | BOP .....                                                                 | 59 |
| 6.2.2   | Incoming and outgoing superframe timing .....                             | 59 |
| 6.2.3   | Enhanced Beacon frame timing for MPM procedure .....                      | 60 |
| 6.2.4   | IFS .....                                                                 | 61 |
| 6.2.5   | Random access methods .....                                               | 61 |
| 6.2.5.1 | CSMA-CA algorithm .....                                                   | 61 |
| 6.2.5.2 | TSCH CCA algorithm .....                                                  | 64 |
| 6.2.5.3 | TSCH CSMA-CA retransmission algorithm .....                               | 64 |
| 6.2.5.4 | CSMA-CA with PCA .....                                                    | 66 |
| 6.2.5.5 | LECIM ALOHA PCA .....                                                     | 69 |
| 6.2.6   | TSCH slotframe structure .....                                            | 69 |
| 6.2.6.1 | General .....                                                             | 69 |
| 6.2.6.2 | Absolute slot number (ASN) .....                                          | 70 |
| 6.2.6.3 | Links .....                                                               | 70 |
| 6.2.6.4 | Multiple slotframes .....                                                 | 70 |
| 6.2.7   | LE functional description .....                                           | 71 |
| 6.2.7.1 | LE contention access period (LE CAP) .....                                | 71 |
| 6.2.7.2 | LE superframe structure .....                                             | 71 |
| 6.2.7.3 | LE incoming and outgoing superframe timing .....                          | 71 |
| 6.2.7.4 | LE scan .....                                                             | 71 |
| 6.2.8   | Superframe use for TMCTP operation .....                                  | 72 |
| 6.2.9   | Rail communications and control network (RCCN) superframe structure ..... | 72 |
| 6.2.10  | Channel hopping .....                                                     | 73 |
| 6.3     | Starting and maintaining PANs .....                                       | 75 |
| 6.3.1   | Scanning through channels .....                                           | 75 |
| 6.3.1.1 | ED channel scan .....                                                     | 75 |
| 6.3.1.2 | Active and passive channel scan .....                                     | 75 |
| 6.3.1.3 | Orphan channel scan .....                                                 | 78 |
| 6.3.1.4 | RIT passive channel scan .....                                            | 80 |
| 6.3.2   | PAN ID conflict resolution .....                                          | 81 |
| 6.3.2.1 | Detection .....                                                           | 81 |
| 6.3.2.2 | Resolution .....                                                          | 82 |
| 6.3.3   | Starting and realigning a PAN .....                                       | 82 |
| 6.3.3.1 | Starting a PAN .....                                                      | 82 |
| 6.3.3.2 | Realigning a PAN .....                                                    | 82 |
| 6.3.3.3 | Realignment in a PAN .....                                                | 84 |
| 6.3.3.4 | Updating superframe configuration and channel PIB attributes .....        | 84 |
| 6.3.4   | Beacon generation .....                                                   | 84 |

|          |                                                                   |     |
|----------|-------------------------------------------------------------------|-----|
| 6.3.5    | Device discovery.....                                             | 86  |
| 6.3.6    | TSCH PAN formation.....                                           | 86  |
| 6.4      | Association and disassociation.....                               | 87  |
| 6.4.1    | Association.....                                                  | 87  |
| 6.4.2    | Disassociation.....                                               | 90  |
| 6.4.3    | Fast association.....                                             | 92  |
| 6.5      | Synchronization.....                                              | 93  |
| 6.5.1    | General.....                                                      | 93  |
| 6.5.2    | Synchronization with beacons.....                                 | 93  |
| 6.5.3    | Synchronization without beacons.....                              | 95  |
| 6.5.4    | Synchronization in TSCH PAN.....                                  | 95  |
| 6.5.4.1  | Timeslot communication.....                                       | 96  |
| 6.5.4.2  | Node synchronization.....                                         | 97  |
| 6.5.5    | Orphaned device realignment.....                                  | 98  |
| 6.6      | Transaction handling.....                                         | 98  |
| 6.7      | Transmission, reception, and acknowledgment.....                  | 99  |
| 6.7.1    | Transmission.....                                                 | 99  |
| 6.7.2    | Reception and rejection.....                                      | 101 |
| 6.7.3    | Extracting pending data from a coordinator.....                   | 102 |
| 6.7.4    | Use of acknowledgments and retransmissions.....                   | 104 |
| 6.7.4.1  | No acknowledgment.....                                            | 104 |
| 6.7.4.2  | Acknowledgment.....                                               | 105 |
| 6.7.4.3  | Retransmissions.....                                              | 106 |
| 6.7.5    | Transmission timing restrictions.....                             | 106 |
| 6.7.6    | Guard time.....                                                   | 107 |
| 6.7.7    | Promiscuous mode.....                                             | 109 |
| 6.7.8    | Transmission scenarios.....                                       | 109 |
| 6.7.9    | Device announcement.....                                          | 110 |
| 6.8      | GTS allocation and management.....                                | 111 |
| 6.8.1    | GTS general requirements.....                                     | 111 |
| 6.8.2    | CAP maintenance.....                                              | 112 |
| 6.8.3    | GTS allocation.....                                               | 112 |
| 6.8.4    | GTS usage.....                                                    | 113 |
| 6.8.5    | GTS deallocation.....                                             | 114 |
| 6.8.6    | GTS reallocation.....                                             | 115 |
| 6.8.7    | GTS expiration.....                                               | 117 |
| 6.9      | Ranging.....                                                      | 117 |
| 6.9.1    | Ranging requirements.....                                         | 117 |
| 6.9.2    | Set-up activities before a ranging exchange.....                  | 117 |
| 6.9.3    | Finish-up activities after a ranging exchange.....                | 117 |
| 6.9.4    | Managing DPS.....                                                 | 118 |
| 6.9.5    | The ranging exchange.....                                         | 119 |
| 6.10     | PHY parameter change notification procedure.....                  | 120 |
| 6.10.1   | Signaling using Beacon frames.....                                | 120 |
| 6.10.2   | Signaling using multipurpose frames.....                          | 120 |
| 6.11     | Deterministic and synchronous multi-channel extension (DSME)..... | 121 |
| 6.11.1   | DSME command requirements.....                                    | 121 |
| 6.11.2   | DSME multi-superframe structure.....                              | 121 |
| 6.11.3   | Channel diversity.....                                            | 123 |
| 6.11.3.1 | Channel adaptation.....                                           | 123 |
| 6.11.3.2 | Channel hopping.....                                              | 124 |
| 6.11.4   | CAP reduction.....                                                | 126 |
| 6.11.5   | DSME GTS allocation and management.....                           | 126 |
| 6.11.5.1 | DSME GTS allocation.....                                          | 127 |

|          |                                                                               |     |
|----------|-------------------------------------------------------------------------------|-----|
| 6.11.5.2 | DSME GTS deallocation .....                                                   | 130 |
| 6.11.5.3 | DSME GTS expiration .....                                                     | 131 |
| 6.11.5.4 | DSME GTS retrieve .....                                                       | 131 |
| 6.11.5.5 | DSME GTS change .....                                                         | 132 |
| 6.11.6   | Beacon scheduling .....                                                       | 132 |
| 6.11.7   | Time synchronization .....                                                    | 133 |
| 6.11.8   | Deferred beacon .....                                                         | 134 |
| 6.11.9   | Passive channel scan .....                                                    | 134 |
| 6.12     | LE transmission, reception and acknowledgment .....                           | 134 |
| 6.12.1   | LE transmission, reception, and acknowledgment with positive handshakes ..... | 134 |
| 6.12.2   | Coordinated sampled listening (CSL) .....                                     | 135 |
| 6.12.2.1 | CSL idle listening .....                                                      | 136 |
| 6.12.2.2 | CSL transmission .....                                                        | 136 |
| 6.12.2.3 | Unicast transmission .....                                                    | 137 |
| 6.12.2.4 | Broadcast transmission .....                                                  | 137 |
| 6.12.2.5 | CSL reception .....                                                           | 138 |
| 6.12.2.6 | CSL over multiple channels .....                                              | 138 |
| 6.12.2.7 | Turning off CSL mode to reduce latency .....                                  | 138 |
| 6.12.3   | RIT .....                                                                     | 138 |
| 6.12.3.1 | General .....                                                                 | 138 |
| 6.12.3.2 | Periodic RIT data request transmission and reception .....                    | 139 |
| 6.12.3.3 | RIT transmission .....                                                        | 141 |
| 6.12.4   | Implicit RIT (I-RIT) .....                                                    | 143 |
| 6.13     | Starting and maintaining TMCTPs .....                                         | 143 |
| 6.14     | MPM procedure for inter-PHY coexistence .....                                 | 146 |
| 6.15     | TVWS access procedures .....                                                  | 149 |
| 6.16     | Channel timing management (CTM) .....                                         | 149 |
| 7.       | MAC frame formats .....                                                       | 151 |
| 7.1      | Device extended address .....                                                 | 151 |
| 7.2      | General MAC frame format .....                                                | 151 |
| 7.2.1    | Frame Control field .....                                                     | 151 |
| 7.2.1.1  | Frame Type field .....                                                        | 152 |
| 7.2.1.2  | Security Enabled field .....                                                  | 152 |
| 7.2.1.3  | Frame Pending field .....                                                     | 152 |
| 7.2.1.4  | AR field .....                                                                | 153 |
| 7.2.1.5  | PAN ID Compression field .....                                                | 153 |
| 7.2.1.6  | Sequence Number Suppression .....                                             | 154 |
| 7.2.1.7  | IE Present field .....                                                        | 154 |
| 7.2.1.8  | Destination Addressing Mode field .....                                       | 154 |
| 7.2.1.9  | Frame Version field .....                                                     | 154 |
| 7.2.1.10 | Source Addressing Mode field .....                                            | 155 |
| 7.2.2    | Sequence Number field .....                                                   | 155 |
| 7.2.3    | Destination PAN ID field .....                                                | 155 |
| 7.2.4    | Destination Address field .....                                               | 155 |
| 7.2.5    | Source PAN ID field .....                                                     | 156 |
| 7.2.6    | Source Address field .....                                                    | 156 |
| 7.2.7    | Auxiliary Security Header field .....                                         | 156 |
| 7.2.8    | IE field .....                                                                | 156 |
| 7.2.9    | Frame Payload field .....                                                     | 156 |
| 7.2.10   | FCS field .....                                                               | 156 |
| 7.3      | Format of individual frame types .....                                        | 158 |
| 7.3.1    | Beacon frame format .....                                                     | 158 |

|          |                                                       |     |
|----------|-------------------------------------------------------|-----|
| 7.3.1.1  | Beacon frame MHR field.....                           | 159 |
| 7.3.1.2  | IEs field.....                                        | 160 |
| 7.3.1.3  | Superframe Specification field .....                  | 160 |
| 7.3.1.4  | GTS Info field .....                                  | 161 |
| 7.3.1.5  | Pending Address field .....                           | 162 |
| 7.3.1.6  | Beacon Payload field .....                            | 162 |
| 7.3.2    | Data frame format .....                               | 163 |
| 7.3.2.1  | Data frame MHR field .....                            | 163 |
| 7.3.2.2  | Data Payload field.....                               | 163 |
| 7.3.3    | Ack frame format.....                                 | 163 |
| 7.3.4    | MAC command frame format.....                         | 165 |
| 7.3.4.1  | MHR field.....                                        | 165 |
| 7.3.4.2  | Command ID field .....                                | 165 |
| 7.3.4.3  | Payload field .....                                   | 165 |
| 7.3.5    | Multipurpose frame format .....                       | 165 |
| 7.3.5.1  | Frame Type field.....                                 | 166 |
| 7.3.5.2  | Long Frame Control field .....                        | 166 |
| 7.3.5.3  | Destination Addressing Mode field .....               | 166 |
| 7.3.5.4  | Source Addressing Mode field .....                    | 166 |
| 7.3.5.5  | PAN ID Present field .....                            | 166 |
| 7.3.5.6  | Security Enabled field.....                           | 167 |
| 7.3.5.7  | Sequence Number Suppression field .....               | 167 |
| 7.3.5.8  | Frame Pending field.....                              | 167 |
| 7.3.5.9  | Frame Version field .....                             | 167 |
| 7.3.5.10 | Ack Request field.....                                | 167 |
| 7.3.5.11 | IEs Present field .....                               | 167 |
| 7.3.5.12 | Sequence Number field .....                           | 167 |
| 7.3.5.13 | Destination PAN ID field .....                        | 167 |
| 7.3.5.14 | Destination Address field.....                        | 167 |
| 7.3.5.15 | Source Address field .....                            | 167 |
| 7.3.5.16 | Auxiliary Security Header field.....                  | 167 |
| 7.3.5.17 | IEs field.....                                        | 168 |
| 7.3.5.18 | Payload field .....                                   | 168 |
| 7.3.6    | Extended frame format .....                           | 168 |
| 7.4      | IEs .....                                             | 168 |
| 7.4.1    | IE list termination .....                             | 168 |
| 7.4.2    | Header IEs.....                                       | 169 |
| 7.4.2.1  | Header IE format .....                                | 169 |
| 7.4.2.2  | Vendor Specific Header IE .....                       | 171 |
| 7.4.2.3  | CSL IE .....                                          | 171 |
| 7.4.2.4  | RIT IE .....                                          | 171 |
| 7.4.2.5  | DSME PAN descriptor IE.....                           | 172 |
| 7.4.2.6  | Rendezvous Time IE .....                              | 174 |
| 7.4.2.7  | Time Correction IE .....                              | 174 |
| 7.4.2.8  | Extended DSME PAN descriptor IE .....                 | 175 |
| 7.4.2.9  | Fragment Sequence Context Description (FSCD) IE ..... | 176 |
| 7.4.2.10 | Simplified Superframe Specification IE .....          | 177 |
| 7.4.2.11 | Simplified GTS Specification IE .....                 | 178 |
| 7.4.2.12 | LECIM Capabilities IE .....                           | 178 |
| 7.4.2.13 | RCC Capabilities IE.....                              | 180 |
| 7.4.2.14 | RCCN Descriptor IE .....                              | 182 |
| 7.4.2.15 | Global Time IE .....                                  | 183 |
| 7.4.2.16 | DA IE.....                                            | 183 |
| 7.4.2.17 | Header Termination 1 IE .....                         | 184 |

|          |                                                |     |
|----------|------------------------------------------------|-----|
| 7.4.2.18 | Header Termination 2 IE .....                  | 184 |
| 7.4.3    | Payload IEs .....                              | 184 |
| 7.4.3.1  | Encapsulated Service Data Unit (ESDU) IE ..... | 184 |
| 7.4.3.2  | MLME IE .....                                  | 185 |
| 7.4.3.3  | Payload Termination IE .....                   | 185 |
| 7.4.4    | Nested IE .....                                | 185 |
| 7.4.4.1  | Format of Nested IE .....                      | 185 |
| 7.4.4.2  | TSCH Synchronization IE .....                  | 188 |
| 7.4.4.3  | TSCH Slotframe and Link IE .....               | 188 |
| 7.4.4.4  | TSCH Timeslot IE .....                         | 190 |
| 7.4.4.5  | Hopping timing IE .....                        | 191 |
| 7.4.4.6  | Enhanced Beacon Filter IE .....                | 191 |
| 7.4.4.7  | MAC Metrics IE .....                           | 192 |
| 7.4.4.8  | All MAC Metrics IE .....                       | 192 |
| 7.4.4.9  | Coexistence Specification IE .....             | 193 |
| 7.4.4.10 | SUN Device Capabilities IE .....               | 193 |
| 7.4.4.11 | SUN FSK Generic PHY IE .....                   | 199 |
| 7.4.4.12 | Mode Switch Parameter IE .....                 | 200 |
| 7.4.4.13 | PHY Parameter Change IE .....                  | 200 |
| 7.4.4.14 | O-QPSK PHY Mode IE .....                       | 201 |
| 7.4.4.15 | PCA Allocation IE .....                        | 201 |
| 7.4.4.16 | LECIM DSSS Operating Mode IE .....             | 202 |
| 7.4.4.17 | LECIM FSK Operating Mode IE .....              | 204 |
| 7.4.4.18 | TVWS PHY Operating Mode Description IE .....   | 205 |
| 7.4.4.19 | TVWS Device Capabilities IE .....              | 208 |
| 7.4.4.20 | TVWS Device Category IE .....                  | 213 |
| 7.4.4.21 | TVWS Device Identification IE .....            | 213 |
| 7.4.4.22 | TVWS Device Location IE .....                  | 214 |
| 7.4.4.23 | TVWS Channel Information Query IE .....        | 215 |
| 7.4.4.24 | TVWS Channel Information Source IE .....       | 217 |
| 7.4.4.25 | CTM IE .....                                   | 218 |
| 7.4.4.26 | Timestamp IE .....                             | 219 |
| 7.4.4.27 | Timestamp Difference IE .....                  | 219 |
| 7.4.4.28 | TMCTP Specification IE .....                   | 219 |
| 7.4.4.29 | RCC PHY Operating Mode IE .....                | 220 |
| 7.4.4.30 | Vendor Specific Nested IE .....                | 221 |
| 7.4.4.31 | Channel hopping IE .....                       | 221 |
| 7.5      | MAC commands .....                             | 222 |
| 7.5.1    | Command ID field .....                         | 222 |
| 7.5.2    | Association Request command .....              | 223 |
| 7.5.3    | Association Response command .....             | 224 |
| 7.5.4    | Disassociation Notification command .....      | 225 |
| 7.5.5    | Data Request command .....                     | 226 |
| 7.5.6    | PAN ID Conflict Notification command .....     | 227 |
| 7.5.7    | Orphan Notification command .....              | 227 |
| 7.5.8    | Beacon Request command .....                   | 228 |
| 7.5.9    | Enhanced Beacon Request command .....          | 228 |
| 7.5.10   | Coordinator realignment command .....          | 228 |
| 7.5.11   | GTS request command .....                      | 230 |
| 7.5.12   | DSME Association Request command .....         | 230 |
| 7.5.13   | DSME Association Response command .....        | 232 |
| 7.5.14   | DSME GTS Request command .....                 | 233 |
| 7.5.15   | DSME GTS Response command .....                | 235 |
| 7.5.16   | DSME GTS Notify command .....                  | 237 |

|          |                                                           |     |
|----------|-----------------------------------------------------------|-----|
| 7.5.17   | DSME Information Request command.....                     | 238 |
| 7.5.18   | DSME Information Response command.....                    | 238 |
| 7.5.19   | DSME Beacon Allocation Notification command.....          | 239 |
| 7.5.20   | DSME Beacon Collision Notification command.....           | 240 |
| 7.5.21   | DSME Link Report command.....                             | 240 |
| 7.5.22   | RIT Data Request command.....                             | 241 |
| 7.5.23   | DBS Request command.....                                  | 242 |
| 7.5.24   | DBS Response command.....                                 | 243 |
| 7.5.25   | RIT Data Response command.....                            | 244 |
| 7.5.26   | Vendor Specific command.....                              | 244 |
| 8.       | MAC services .....                                        | 245 |
| 8.1      | Overview.....                                             | 245 |
| 8.2      | MAC management service.....                               | 245 |
| 8.2.1    | Primitives supported by the MLME-SAP interface.....       | 245 |
| 8.2.2    | Common requirements for MLME primitives.....              | 247 |
| 8.2.3    | Association primitives .....                              | 248 |
| 8.2.3.1  | MLME-ASSOCIATE.request.....                               | 248 |
| 8.2.3.2  | MLME-ASSOCIATE.indication.....                            | 249 |
| 8.2.3.3  | MLME-ASSOCIATE.response.....                              | 251 |
| 8.2.3.4  | MLME-ASSOCIATE.confirm.....                               | 252 |
| 8.2.4    | Disassociation primitives.....                            | 254 |
| 8.2.4.1  | MLME-DISASSOCIATE.request.....                            | 254 |
| 8.2.4.2  | MLME-DISASSOCIATE.indication.....                         | 256 |
| 8.2.4.3  | MLME-DISASSOCIATE.confirm.....                            | 256 |
| 8.2.5    | Communications notification primitives.....               | 257 |
| 8.2.5.1  | MLME-BEACON-NOTIFY.indication.....                        | 257 |
| 8.2.5.2  | MLME-COMM-STATUS.indication.....                          | 260 |
| 8.2.5.3  | MLME-IE-NOTIFY.indication.....                            | 262 |
| 8.2.6    | Primitives for reading and writing PIB attributes.....    | 264 |
| 8.2.6.1  | MLME-GET.request.....                                     | 264 |
| 8.2.6.2  | MLME-GET.confirm.....                                     | 264 |
| 8.2.6.3  | MLME-SET.request.....                                     | 265 |
| 8.2.6.4  | MLME-SET.confirm.....                                     | 265 |
| 8.2.7    | GTS management primitives.....                            | 266 |
| 8.2.7.1  | MLME-GTS.request.....                                     | 266 |
| 8.2.7.2  | MLME-GTS.confirm.....                                     | 267 |
| 8.2.7.3  | MLME-GTS.indication.....                                  | 268 |
| 8.2.8    | Primitives for orphan notification.....                   | 269 |
| 8.2.8.1  | MLME-ORPHAN.indication.....                               | 269 |
| 8.2.8.2  | MLME-ORPHAN.response.....                                 | 270 |
| 8.2.9    | Primitives for resetting the MAC sublayer.....            | 271 |
| 8.2.9.1  | MLME-RESET.request.....                                   | 271 |
| 8.2.9.2  | MLME-RESET.confirm.....                                   | 271 |
| 8.2.10   | Primitives for specifying the receiver enable time.....   | 272 |
| 8.2.10.1 | MLME-RX-ENABLE.request.....                               | 272 |
| 8.2.10.2 | MLME-RX-ENABLE.confirm.....                               | 273 |
| 8.2.11   | Primitives for channel scanning.....                      | 274 |
| 8.2.11.1 | MLME-SCAN.request.....                                    | 274 |
| 8.2.11.2 | MLME-SCAN.confirm.....                                    | 277 |
| 8.2.12   | Primitives for updating the superframe configuration..... | 279 |
| 8.2.12.1 | MLME-START.request.....                                   | 279 |
| 8.2.12.2 | MLME-START.confirm.....                                   | 282 |



|          |                                                         |     |
|----------|---------------------------------------------------------|-----|
| 8.2.13   | Primitives for synchronizing with a coordinator .....   | 283 |
| 8.2.13.1 | MLME-SYNC.request .....                                 | 283 |
| 8.2.13.2 | MLME-SYNC-LOSS.indication .....                         | 284 |
| 8.2.14   | Primitives for requesting data from a coordinator ..... | 286 |
| 8.2.14.1 | MLME-POLL.request .....                                 | 286 |
| 8.2.14.2 | MLME-POLL.confirm .....                                 | 287 |
| 8.2.15   | Primitives for specifying dynamic preamble .....        | 287 |
| 8.2.15.1 | MLME-DPS.request .....                                  | 288 |
| 8.2.15.2 | MLME-DPS.confirm .....                                  | 288 |
| 8.2.15.3 | MLME-DPS.indication .....                               | 289 |
| 8.2.16   | Primitives for channel sounding .....                   | 289 |
| 8.2.16.1 | MLME-SOUNDING.request .....                             | 289 |
| 8.2.16.2 | MLME-SOUNDING.confirm .....                             | 289 |
| 8.2.17   | Primitives for ranging calibration .....                | 290 |
| 8.2.17.1 | MLME-CALIBRATE.request .....                            | 291 |
| 8.2.17.2 | MLME-CALIBRATE.confirm .....                            | 291 |
| 8.2.18   | Primitives for Beacon Generation .....                  | 292 |
| 8.2.18.1 | MLME-BEACON.request .....                               | 292 |
| 8.2.18.2 | MLME-BEACON.confirm .....                               | 294 |
| 8.2.18.3 | MLME-BEACON-REQUEST.indication .....                    | 295 |
| 8.2.19   | Primitives for TSCH .....                               | 296 |
| 8.2.19.1 | MLME-SET-SLOTFRAME.request .....                        | 296 |
| 8.2.19.2 | MLME-SET-SLOTFRAME.confirm .....                        | 297 |
| 8.2.19.3 | MLME-SET-LINK.request .....                             | 297 |
| 8.2.19.4 | MLME-SET-LINK.confirm .....                             | 299 |
| 8.2.19.5 | MLME-TSCH-MODE.request .....                            | 300 |
| 8.2.19.6 | MLME-TSCH-MODE.confirm .....                            | 301 |
| 8.2.19.7 | MLME-KEEP-ALIVE.request .....                           | 301 |
| 8.2.19.8 | MLME-KEEP-ALIVE.confirm .....                           | 302 |
| 8.2.20   | Primitives for DSME GTS management .....                | 302 |
| 8.2.20.1 | MLME-DSME-GTS.request .....                             | 302 |
| 8.2.20.2 | MLME-DSME-GTS.indication .....                          | 305 |
| 8.2.20.3 | MLME-DSME-GTS.response .....                            | 306 |
| 8.2.20.4 | MLME-DSME-GTS.confirm .....                             | 308 |
| 8.2.21   | Primitives for reporting the link status .....          | 309 |
| 8.2.21.1 | MLME-DSME-LINK-REPORT.request .....                     | 309 |
| 8.2.21.2 | MLME-DSME-LINK-REPORT.indication .....                  | 310 |
| 8.2.21.3 | MLME-DSME-LINK-REPORT.confirm .....                     | 311 |
| 8.2.22   | Operating parameter change primitives .....             | 311 |
| 8.2.22.1 | MLME-PHY-OP-SWITCH.request .....                        | 311 |
| 8.2.22.2 | MLME-PHY-OP-SWITCH.indication .....                     | 313 |
| 8.2.22.3 | MLME-PHY-OP-SWITCH.confirm .....                        | 315 |
| 8.2.23   | TMCTP DBS allocation primitives .....                   | 316 |
| 8.2.23.1 | MLME-DBS.request .....                                  | 316 |
| 8.2.23.2 | MLME-DBS.indication .....                               | 317 |
| 8.2.23.3 | MLME-DBS.response .....                                 | 318 |
| 8.2.23.4 | MLME-DBS.confirm .....                                  | 319 |
| 8.2.24   | Primitives for device announcement .....                | 320 |
| 8.2.24.1 | MLME-DA.request primitive .....                         | 320 |
| 8.2.24.2 | MLME-DA.indication primitive .....                      | 321 |
| 8.2.24.3 | MLME-DA.confirm primitive .....                         | 321 |
| 8.2.25   | RIT data commands .....                                 | 322 |
| 8.2.25.1 | MLME-RIT-REQ.indication .....                           | 322 |
| 8.2.25.2 | MLME-RIT-RES.request .....                              | 324 |

|           |                                                                                |     |
|-----------|--------------------------------------------------------------------------------|-----|
| 8.2.25.3  | MLME-RIT-RES.indication .....                                                  | 325 |
| 8.2.25.4  | MLME-RIT-RES.confirm .....                                                     | 327 |
| 8.3       | MAC data service .....                                                         | 328 |
| 8.3.1     | MCPS-DATA.request .....                                                        | 328 |
| 8.3.2     | MCPS-DATA.confirm .....                                                        | 332 |
| 8.3.3     | MCPS-DATA.indication .....                                                     | 335 |
| 8.3.4     | MCPS-PURGE.request .....                                                       | 338 |
| 8.3.5     | MCPS-PURGE.confirm .....                                                       | 338 |
| 8.4       | MAC constants and PIB attributes .....                                         | 339 |
| 8.4.1     | MAC constants .....                                                            | 339 |
| 8.4.2     | MAC PIB attributes .....                                                       | 340 |
| 8.4.2.1   | General MAC PIB attributes for functional organization .....                   | 345 |
| 8.4.2.2   | TSCH-specific MAC PIB attributes .....                                         | 348 |
| 8.4.2.2.1 | TSCH MAC PIB attributes for macSlotframeTable .....                            | 348 |
| 8.4.2.2.2 | TSCH MAC PIB attributes for macLinkTable .....                                 | 349 |
| 8.4.2.2.3 | TSCH MAC PIB attributes for macTimeslotTemplate .....                          | 350 |
| 8.4.2.3   | MAC PIB attributes for hopping sequence .....                                  | 351 |
| 8.4.2.4   | DSME specific MAC PIB attributes .....                                         | 352 |
| 8.4.2.5   | LE specific MAC PIB attributes .....                                           | 355 |
| 8.4.2.6   | MAC performance metrics specific MAC PIB attributes .....                      | 357 |
| 8.4.2.7   | Enhanced Beacon Request command specific MAC PIB attributes .....              | 358 |
| 8.4.2.8   | Enhanced Beacon frame specific MAC PIB attributes .....                        | 359 |
| 9.        | Security .....                                                                 | 360 |
| 9.1       | Overview .....                                                                 | 360 |
| 9.2       | Functional description .....                                                   | 360 |
| 9.2.1     | Outgoing frame security procedure .....                                        | 360 |
| 9.2.2     | KeyDescriptor lookup procedure .....                                           | 362 |
| 9.2.3     | Incoming frame security procedure, Security Enabled field is set to one .....  | 362 |
| 9.2.4     | Incoming frame security procedure, Security Enabled field is set to zero ..... | 364 |
| 9.2.5     | DeviceDescriptor lookup procedure .....                                        | 365 |
| 9.2.6     | SecurityLevelDescriptor lookup procedure .....                                 | 366 |
| 9.2.7     | Incoming IE security level checking procedure .....                            | 366 |
| 9.2.8     | Incoming IE key usage policy checking procedure .....                          | 367 |
| 9.2.9     | Incoming security level checking procedure .....                               | 367 |
| 9.2.10    | Incoming key usage policy checking procedure .....                             | 367 |
| 9.3       | Security operations .....                                                      | 368 |
| 9.3.1     | Integer and octet representation .....                                         | 368 |
| 9.3.2     | CCM* nonce .....                                                               | 368 |
| 9.3.2.1   | CCM* nonce for non-TSCH mode .....                                             | 368 |
| 9.3.2.2   | CCM* nonce for TSCH mode .....                                                 | 368 |
| 9.3.2.3   | CCM* nonce for Fragment frames .....                                           | 369 |
| 9.3.3     | CCM* prerequisites .....                                                       | 369 |
| 9.3.4     | CCM* transformation data representation .....                                  | 370 |
| 9.3.4.1   | Key and nonce data inputs .....                                                | 370 |
| 9.3.4.2   | a data and m data .....                                                        | 370 |
| 9.3.4.3   | c data output .....                                                            | 370 |
| 9.3.5     | CCM* inverse transformation data representation .....                          | 371 |
| 9.3.5.1   | Key and nonce data inputs .....                                                | 371 |
| 9.3.5.2   | c data and a data .....                                                        | 371 |
| 9.3.5.3   | m data output .....                                                            | 372 |
| 9.4       | Auxiliary security header .....                                                | 372 |
| 9.4.1     | Security Control field .....                                                   | 372 |

|             |                                                                  |     |
|-------------|------------------------------------------------------------------|-----|
| 9.4.1.1     | Security Level field.....                                        | 372 |
| 9.4.1.2     | Key Identifier Mode field .....                                  | 373 |
| 9.4.1.3     | Frame Counter Suppression field .....                            | 374 |
| 9.4.1.4     | ASN in Nonce .....                                               | 374 |
| 9.4.2       | Frame Counter field .....                                        | 374 |
| 9.4.3       | Key Identifier field.....                                        | 375 |
| 9.4.3.1     | Key Source field .....                                           | 375 |
| 9.4.3.2     | Key Index field .....                                            | 375 |
| 9.5         | Security-related MAC PIB attributes.....                         | 375 |
| 10.         | General PHY requirements .....                                   | 381 |
| 10.1        | General requirements and definitions .....                       | 381 |
| 10.1.1      | Operating frequency range.....                                   | 382 |
| 10.1.2      | Channel assignments.....                                         | 387 |
| 10.1.2.1    | Channel numbering for 780 MHz band .....                         | 387 |
| 10.1.2.2    | Channel numbering for 868 MHz, 915 MHz, and 2450 MHz bands ..... | 387 |
| 10.1.2.3    | Channel numbering for CSS PHY .....                              | 388 |
| 10.1.2.4    | Channel numbering for HRP UWB PHY .....                          | 388 |
| 10.1.2.5    | Channel numbering for MSK PHY 433 MHz band .....                 | 389 |
| 10.1.2.6    | Channel numbering for MSK PHY 2450 MHz band .....                | 390 |
| 10.1.2.7    | Channel numbering for LRP UWB PHY .....                          | 392 |
| 10.1.2.8    | Channel numbering for SUN and TVWS PHYs .....                    | 392 |
| 10.1.2.9    | Channel numbering for 2380 MHz band .....                        | 395 |
| 10.1.2.10   | Channel numbering for LECIM PHYs .....                           | 395 |
| 10.1.2.10.1 | Channel numbering for LECIM DSSS PHY .....                       | 396 |
| 10.1.2.10.2 | Channel numbering for LECIM FSK PHY .....                        | 396 |
| 10.1.2.11   | Channel numbering for RCC PHYs.....                              | 397 |
| 10.1.3      | Minimum LIFS and SIFS periods.....                               | 398 |
| 10.1.4      | RF power measurement .....                                       | 399 |
| 10.1.5      | Transmit power .....                                             | 399 |
| 10.1.6      | Out-of-band spurious emission.....                               | 399 |
| 10.1.7      | Receiver sensitivity definitions.....                            | 399 |
| 10.1.8      | Common signaling mode (CSM) for SUN PHY .....                    | 400 |
| 10.2        | General radio specifications.....                                | 400 |
| 10.2.1      | TX-to-RX turnaround time .....                                   | 400 |
| 10.2.2      | RX-to-TX turnaround time .....                                   | 400 |
| 10.2.3      | Error-vector magnitude (EVM) definition.....                     | 400 |
| 10.2.4      | Receiver maximum input level of desired signal.....              | 401 |
| 10.2.5      | Receiver ED .....                                                | 401 |
| 10.2.6      | Link quality indicator (LQI) .....                               | 402 |
| 10.2.7      | Clear channel assessment (CCA).....                              | 402 |
| 11.         | PHY services .....                                               | 404 |
| 11.1        | Overview.....                                                    | 404 |
| 11.2        | PHY constants.....                                               | 404 |
| 11.3        | PHY PIB attributes .....                                         | 404 |
| 12.         | O-QPSK PHY .....                                                 | 411 |
| 12.1        | PPDU format.....                                                 | 411 |
| 12.1.1      | SHR field format.....                                            | 411 |
| 12.1.1.1    | Preamble field .....                                             | 411 |

|          |                                                     |     |
|----------|-----------------------------------------------------|-----|
| 12.1.1.2 | SFD field.....                                      | 411 |
| 12.1.2   | PHR field format.....                               | 411 |
| 12.1.2.1 | Frame Length field.....                             | 411 |
| 12.1.2.2 | PHY Payload field.....                              | 411 |
| 12.2     | Modulation and spreading.....                       | 412 |
| 12.2.1   | Data rate.....                                      | 412 |
| 12.2.2   | Reference modulator diagram.....                    | 412 |
| 12.2.3   | Bit-to-symbol mapping.....                          | 412 |
| 12.2.4   | Symbol-to-chip mapping.....                         | 412 |
| 12.2.5   | O-QPSK modulation.....                              | 414 |
| 12.2.6   | Pulse shape.....                                    | 414 |
| 12.2.7   | Chip transmission order.....                        | 415 |
| 12.3     | O-QPSK PHY RF requirements.....                     | 415 |
| 12.3.1   | Operating frequency range.....                      | 415 |
| 12.3.2   | Transmit power spectral density (PSD) mask.....     | 416 |
| 12.3.3   | Symbol rate.....                                    | 416 |
| 12.3.4   | Receiver sensitivity.....                           | 416 |
| 12.3.5   | Receiver interference rejection.....                | 416 |
| 12.3.6   | TX-to-RX turnaround time.....                       | 417 |
| 12.3.7   | RX-to-TX turnaround time.....                       | 417 |
| 12.3.8   | EVM.....                                            | 417 |
| 12.3.9   | Transmit center frequency tolerance.....            | 417 |
| 12.3.10  | Transmit power.....                                 | 417 |
| 12.3.11  | Receiver maximum input level of desired signal..... | 417 |
| 12.3.12  | Receiver ED.....                                    | 417 |
| 12.3.13  | LQI.....                                            | 417 |
| 13.      | Binary phase-shift keying (BPSK) PHY.....           | 418 |
| 13.1     | PPDU format.....                                    | 418 |
| 13.2     | Modulation and spreading.....                       | 418 |
| 13.2.1   | BPSK PHY data rates.....                            | 418 |
| 13.2.2   | Reference modulator.....                            | 418 |
| 13.2.3   | Differential encoding.....                          | 418 |
| 13.2.4   | Bit-to-chip mapping.....                            | 419 |
| 13.2.5   | BPSK modulation.....                                | 419 |
| 13.2.5.1 | Pulse shape.....                                    | 419 |
| 13.2.5.2 | Chip transmission order.....                        | 419 |
| 13.3     | BPSK PHY RF requirements.....                       | 419 |
| 13.3.1   | Operating frequency range.....                      | 419 |
| 13.3.2   | 915 MHz band transmit PSD mask.....                 | 419 |
| 13.3.3   | Symbol rate.....                                    | 420 |
| 13.3.4   | Receiver sensitivity.....                           | 420 |
| 13.3.5   | Receiver interference rejection.....                | 420 |
| 13.3.6   | TX-to-RX turnaround time.....                       | 420 |
| 13.3.7   | RX-to-TX turnaround time.....                       | 420 |
| 13.3.8   | EVM.....                                            | 421 |
| 13.3.9   | Transmit center frequency tolerance.....            | 421 |
| 13.3.10  | Transmit power.....                                 | 421 |
| 13.3.11  | Receiver maximum input level of desired signal..... | 421 |
| 13.3.12  | Receiver ED.....                                    | 421 |
| 13.3.13  | LQI.....                                            | 421 |
| 14.      | Amplitude shift keying (ASK) PHY.....               | 422 |

|         |                                                                               |     |
|---------|-------------------------------------------------------------------------------|-----|
| 14.1    | Status of ASK PHY .....                                                       | 422 |
| 14.2    | PPDU format.....                                                              | 422 |
| 14.2.1  | Preamble field for ASK PHY .....                                              | 422 |
| 14.2.2  | SFD for ASK PHY .....                                                         | 422 |
| 14.3    | Modulation and spreading .....                                                | 422 |
| 14.3.1  | ASK PHY data rates .....                                                      | 422 |
| 14.3.2  | Reference modulator.....                                                      | 423 |
| 14.3.3  | Bit-to-symbol mapping.....                                                    | 423 |
| 14.3.4  | Symbol-to-chip mapping .....                                                  | 423 |
| 14.3.5  | ASK modulation .....                                                          | 424 |
| 14.3.6  | Pulse shape.....                                                              | 426 |
| 14.3.7  | Chip transmission order.....                                                  | 426 |
| 14.4    | ASK PHY RF requirements.....                                                  | 426 |
| 14.4.1  | Operating frequency range.....                                                | 426 |
| 14.4.2  | 915 MHz band transmit PSD mask.....                                           | 426 |
| 14.4.3  | Symbol rate.....                                                              | 426 |
| 14.4.4  | Receiver sensitivity.....                                                     | 427 |
| 14.4.5  | Receiver interference rejection .....                                         | 427 |
| 14.4.6  | TX-to-RX turnaround time .....                                                | 427 |
| 14.4.7  | RX-to-TX turnaround time .....                                                | 427 |
| 14.4.8  | EVM.....                                                                      | 427 |
| 14.4.9  | Transmit center frequency tolerance.....                                      | 427 |
| 14.4.10 | Transmit power .....                                                          | 427 |
| 14.4.11 | Receiver maximum input level of desired signal.....                           | 428 |
| 14.4.12 | Receiver ED .....                                                             | 428 |
| 14.4.13 | LQI.....                                                                      | 428 |
| 14.4.14 | Example of PSSS encoding .....                                                | 428 |
| 15.     | Chirp spread spectrum (CSS) PHY .....                                         | 430 |
| 15.1    | CSS PPDU format .....                                                         | 430 |
| 15.1.1  | Preamble field.....                                                           | 430 |
| 15.1.2  | SFD field.....                                                                | 430 |
| 15.1.3  | PHR field .....                                                               | 431 |
| 15.1.4  | PHY Payload field .....                                                       | 431 |
| 15.2    | Modulation and spreading .....                                                | 431 |
| 15.2.1  | Data rates .....                                                              | 431 |
| 15.2.2  | Reference modulator.....                                                      | 431 |
| 15.2.3  | De-multiplexer (DEMUX).....                                                   | 431 |
| 15.2.4  | Serial-to-parallel mapping .....                                              | 431 |
| 15.2.5  | Data-symbol-to-bi-orthogonal-codeword mapping .....                           | 432 |
| 15.2.6  | Parallel-to-serial converter and QPSK symbol mapping.....                     | 436 |
| 15.2.7  | DQPSK coding .....                                                            | 436 |
| 15.2.8  | DQPSK-to-DQCSK modulation.....                                                | 437 |
| 15.2.9  | CSK generator.....                                                            | 437 |
| 15.2.10 | Bit interleaver .....                                                         | 437 |
| 15.3    | Waveform and subchirp sequences.....                                          | 438 |
| 15.3.1  | Graphical presentation of chirp symbols (subchirp sequences).....             | 438 |
| 15.3.2  | Active usage of time gaps .....                                               | 438 |
| 15.3.3  | Mathematical representation of the continuous time CSS base-band signal ..... | 439 |
| 15.3.4  | Raised cosine window for chirp pulse shaping.....                             | 441 |
| 15.3.5  | Subchirp transmission order .....                                             | 441 |
| 15.3.6  | Example of CSK signal generation.....                                         | 442 |
| 15.4    | CSS RF requirements.....                                                      | 443 |

|          |                                                                      |     |
|----------|----------------------------------------------------------------------|-----|
| 15.4.1   | Transmit power spectral density (PSD) mask and signal tolerance..... | 443 |
| 15.4.2   | Symbol rate .....                                                    | 444 |
| 15.4.3   | Receiver sensitivity .....                                           | 444 |
| 15.4.4   | Receiver interference rejection .....                                | 444 |
| 15.4.5   | TX-to-RX turnaround time .....                                       | 444 |
| 15.4.6   | RX-to-TX turnaround time .....                                       | 444 |
| 15.4.7   | Transmit center frequency tolerance.....                             | 444 |
| 15.4.8   | Transmit power .....                                                 | 445 |
| 15.4.9   | Receiver maximum input level of desired signal.....                  | 445 |
| 15.4.10  | Receiver ED .....                                                    | 445 |
| 15.4.11  | LQI.....                                                             | 445 |
| 16.      | HRP UWB PHY .....                                                    | 446 |
| 16.1     | General.....                                                         | 446 |
| 16.2     | HRP UWB PPDU format .....                                            | 446 |
| 16.2.1   | PPDU encoding process.....                                           | 447 |
| 16.2.2   | Symbol structure .....                                               | 449 |
| 16.2.3   | PSDU timing parameters .....                                         | 450 |
| 16.2.4   | Preamble timing parameters .....                                     | 452 |
| 16.2.5   | SHR field .....                                                      | 454 |
| 16.2.5.1 | SYNC field .....                                                     | 454 |
| 16.2.5.2 | SFD field.....                                                       | 457 |
| 16.2.6   | PHR field .....                                                      | 457 |
| 16.2.7   | PHY Payload field .....                                              | 458 |
| 16.3     | Modulation.....                                                      | 459 |
| 16.3.1   | Modulation mathematical framework.....                               | 459 |
| 16.3.2   | Spreading .....                                                      | 459 |
| 16.3.3   | FEC .....                                                            | 461 |
| 16.3.3.1 | Reed-Solomon encoding.....                                           | 461 |
| 16.3.3.2 | Systematic convolutional encoding .....                              | 462 |
| 16.4     | RF requirements.....                                                 | 463 |
| 16.4.1   | Operating frequency bands .....                                      | 463 |
| 16.4.2   | Channel assignments.....                                             | 464 |
| 16.4.3   | Regulatory compliance .....                                          | 464 |
| 16.4.4   | Operating temperature range .....                                    | 464 |
| 16.4.5   | Baseband impulse response .....                                      | 464 |
| 16.4.6   | Transmit PSD mask .....                                              | 466 |
| 16.4.7   | Chip rate clock and chip carrier alignment.....                      | 467 |
| 16.4.8   | TX-to-RX turnaround time .....                                       | 467 |
| 16.4.9   | RX-to-TX turnaround time .....                                       | 467 |
| 16.4.10  | Transmit center frequency tolerance.....                             | 467 |
| 16.4.11  | Receiver maximum input level of desired signal.....                  | 467 |
| 16.4.12  | Receiver ED .....                                                    | 467 |
| 16.4.13  | LQI.....                                                             | 467 |
| 16.4.14  | CCA .....                                                            | 467 |
| 16.5     | HRP UWB PHY optional pulse shapes .....                              | 467 |
| 16.5.1   | HRP UWB PHY optional chirp on UWB (CoU) pulses .....                 | 468 |
| 16.5.2   | HRP UWB PHY optional continuous spectrum (CS) pulses .....           | 469 |
| 16.5.3   | HRP UWB PHY linear combination of pulses (LCP).....                  | 470 |
| 16.6     | Extended preamble for optional CCA mode 6.....                       | 471 |
| 16.7     | Ranging.....                                                         | 472 |
| 16.7.1   | Ranging counter .....                                                | 472 |
| 16.7.2   | Crystal characterization .....                                       | 472 |

|          |                                                             |     |
|----------|-------------------------------------------------------------|-----|
| 16.7.2.1 | Ranging tracking offset.....                                | 472 |
| 16.7.2.2 | Ranging tracking interval.....                              | 472 |
| 16.7.3   | Ranging FoM .....                                           | 473 |
| 17.      | GFSK PHY .....                                              | 475 |
| 17.1     | PPDU formats .....                                          | 475 |
| 17.2     | Modulation.....                                             | 475 |
| 17.2.1   | GFSK PHY data rates .....                                   | 475 |
| 17.2.2   | Reference modulator diagram.....                            | 475 |
| 17.2.3   | Data whitening .....                                        | 475 |
| 17.2.4   | GFSK modulation .....                                       | 476 |
| 17.3     | GFSK PHY RF requirements .....                              | 476 |
| 17.3.1   | Operating frequency range.....                              | 476 |
| 17.3.2   | Transmit PSD mask .....                                     | 476 |
| 17.3.3   | Symbol rate .....                                           | 477 |
| 17.3.4   | Receiver sensitivity .....                                  | 477 |
| 17.3.5   | Receiver interference rejection .....                       | 477 |
| 17.3.6   | TX-to-RX turnaround time .....                              | 477 |
| 17.3.7   | RX-to-TX turnaround time .....                              | 477 |
| 17.3.8   | Transmit center frequency tolerance.....                    | 477 |
| 17.3.9   | Transmit power .....                                        | 477 |
| 17.3.10  | Receiver maximum input level of desired signal.....         | 477 |
| 17.3.11  | Receiver ED .....                                           | 478 |
| 17.3.12  | LQI.....                                                    | 478 |
| 18.      | MSK PHY .....                                               | 479 |
| 18.1     | PPDU formats .....                                          | 479 |
| 18.2     | Data rate .....                                             | 479 |
| 18.3     | SFD for the MSK PHY .....                                   | 479 |
| 18.4     | MSK modulation.....                                         | 480 |
| 18.4.1   | Reference modulator diagram.....                            | 480 |
| 18.4.2   | Data whitening .....                                        | 480 |
| 18.4.3   | Bit-to-symbol mapping .....                                 | 480 |
| 18.4.4   | Signal modulation .....                                     | 480 |
| 18.5     | MSK PHY requirements .....                                  | 481 |
| 18.5.1   | Operating frequency range.....                              | 481 |
| 18.5.2   | Transmit PSD mask .....                                     | 481 |
| 18.5.3   | Symbol rate .....                                           | 481 |
| 18.5.4   | Transmit center frequency tolerance.....                    | 482 |
| 18.5.5   | Transmit power .....                                        | 482 |
| 18.5.6   | Receiver maximum input level of desired signal.....         | 482 |
| 18.5.7   | Modulation frequency deviation tolerance .....              | 482 |
| 18.5.8   | Zero crossing tolerance .....                               | 482 |
| 19.      | LRP UWB PHY specification .....                             | 483 |
| 19.1     | Overview.....                                               | 483 |
| 19.2     | LRP UWB PHY symbol structure.....                           | 483 |
| 19.2.1   | Base mode LRP UWB PHY symbol structure .....                | 483 |
| 19.2.1.1 | Base mode LRP UWB PHY PSDU synchronization signal.....      | 484 |
| 19.2.2   | Extended mode LRP UWB PHY symbol structure .....            | 484 |
| 19.2.2.1 | Extended mode LRP UWB PHY PSDU synchronization signal ..... | 485 |

|          |                                                               |     |
|----------|---------------------------------------------------------------|-----|
| 19.2.3   | Long-range mode LRP UWB PHY symbol structure .....            | 486 |
| 19.2.3.1 | Long-range mode LRP UWB PHY PSDU synchronization signal ..... | 486 |
| 19.3     | LRP UWB SHR .....                                             | 487 |
| 19.3.1   | LRP UWB SHR preamble .....                                    | 487 |
| 19.3.1.1 | LRP UWB base mode SHR preamble .....                          | 487 |
| 19.3.1.2 | RP UWB extended mode SHR preamble .....                       | 487 |
| 19.3.1.3 | LRP UWB long-range mode SHR preamble .....                    | 487 |
| 19.3.2   | LRP UWB SHR SFD .....                                         | 487 |
| 19.4     | LRP UWB PHR .....                                             | 487 |
| 19.4.1   | Encoding Type field .....                                     | 488 |
| 19.4.2   | Header Extension field .....                                  | 488 |
| 19.4.3   | SECDED field .....                                            | 489 |
| 19.4.4   | Frame Length field .....                                      | 489 |
| 19.4.5   | LEIP Length field .....                                       | 489 |
| 19.4.6   | LEIP Position field .....                                     | 489 |
| 19.5     | LRP UWB PSDU .....                                            | 490 |
| 19.6     | LRP UWB location enhancing information postamble .....        | 490 |
| 19.7     | LRP UWB transmitter specification .....                       | 490 |
| 19.7.1   | Pulse shape .....                                             | 490 |
| 19.7.2   | Pulse timing .....                                            | 491 |
| 19.7.3   | Transmit PSD mask .....                                       | 491 |
| 19.8     | LRP UWB receiver specification .....                          | 492 |
| 20.      | SUN FSK PHY .....                                             | 493 |
| 20.1     | Introduction .....                                            | 493 |
| 20.2     | PPDU format for SUN FSK .....                                 | 493 |
| 20.2.1   | SHR field format .....                                        | 494 |
| 20.2.1.1 | Preamble field .....                                          | 494 |
| 20.2.1.2 | SFD .....                                                     | 494 |
| 20.2.2   | PHR field format .....                                        | 495 |
| 20.2.3   | Mode Switch PHR .....                                         | 495 |
| 20.2.4   | PHY Payload field .....                                       | 497 |
| 20.3     | Modulation and coding for SUN FSK .....                       | 497 |
| 20.3.1   | Reference modulator .....                                     | 499 |
| 20.3.2   | Bit-to-symbol mapping .....                                   | 500 |
| 20.3.3   | Modulation quality .....                                      | 501 |
| 20.3.3.1 | Frequency deviation tolerance .....                           | 501 |
| 20.3.3.2 | Zero crossing tolerance .....                                 | 502 |
| 20.3.4   | FEC .....                                                     | 502 |
| 20.3.5   | Code-symbol interleaving .....                                | 505 |
| 20.4     | Data whitening for SUN FSK .....                              | 506 |
| 20.5     | Mode switch mechanism for SUN FSK .....                       | 506 |
| 20.6     | SUN FSK PHY RF requirements .....                             | 509 |
| 20.6.1   | Operating frequency range .....                               | 509 |
| 20.6.2   | Regulatory compliance .....                                   | 509 |
| 20.6.3   | Radio frequency tolerance .....                               | 509 |
| 20.6.4   | Channel switch time .....                                     | 510 |
| 20.6.5   | Transmitter symbol rate .....                                 | 510 |
| 20.6.6   | Transmit spectral mask .....                                  | 510 |
| 20.6.7   | Receiver sensitivity .....                                    | 510 |
| 20.6.8   | Receiver interference rejection .....                         | 511 |
| 20.6.9   | TX-to-RX turnaround time .....                                | 511 |
| 20.6.10  | RX-to-TX turnaround time .....                                | 511 |



|          |                                                      |     |
|----------|------------------------------------------------------|-----|
| 20.6.11  | Transmit power .....                                 | 511 |
| 20.6.12  | Receiver maximum input level of desired signal ..... | 511 |
| 20.6.13  | Receiver ED .....                                    | 511 |
| 20.6.14  | LQI .....                                            | 511 |
| 21.      | SUN OFDM PHY .....                                   | 512 |
| 21.1     | Introduction .....                                   | 512 |
| 21.2     | PPDU format for SUN OFDM .....                       | 512 |
| 21.2.1   | Short Training field (STF) .....                     | 512 |
| 21.2.1.1 | Frequency domain STF .....                           | 512 |
| 21.2.1.2 | Time domain STF generation .....                     | 515 |
| 21.2.1.3 | Time domain STF repetition .....                     | 515 |
| 21.2.1.4 | STF normalization .....                              | 516 |
| 21.2.2   | Long Training field (LTF) .....                      | 516 |
| 21.2.2.1 | Frequency domain LTF .....                           | 516 |
| 21.2.2.2 | Time domain LTF generation .....                     | 519 |
| 21.2.2.3 | LTF normalization .....                              | 519 |
| 21.2.3   | PHR .....                                            | 519 |
| 21.2.4   | PSDU field .....                                     | 520 |
| 21.3     | Data rates for SUN OFDM .....                        | 520 |
| 21.4     | Modulation and coding for SUN OFDM .....             | 521 |
| 21.4.1   | Reference modulator diagram .....                    | 521 |
| 21.4.2   | Bit-to-symbol mapping .....                          | 521 |
| 21.4.3   | PIB attribute values for phySymbolsPerOctet .....    | 523 |
| 21.4.4   | FEC .....                                            | 523 |
| 21.4.5   | Interleaver .....                                    | 524 |
| 21.4.6   | Frequency spreading .....                            | 526 |
| 21.4.6.1 | Frequency spreading by 2x .....                      | 526 |
| 21.4.6.2 | Frequency spreading by 4x .....                      | 526 |
| 21.4.6.3 | No spreading .....                                   | 527 |
| 21.4.7   | Pilot tones/null tones .....                         | 527 |
| 21.4.8   | Cyclic prefix (CP) .....                             | 530 |
| 21.4.9   | PPDU Tail field .....                                | 530 |
| 21.4.10  | Pad field .....                                      | 530 |
| 21.4.11  | Scrambler and scrambler seeds .....                  | 531 |
| 21.5     | SUN OFDM PHY RF requirements .....                   | 532 |
| 21.5.1   | Operating frequency range .....                      | 532 |
| 21.5.2   | Transmit power spectral density (PSD) mask .....     | 532 |
| 21.5.3   | Receiver sensitivity .....                           | 532 |
| 21.5.4   | Adjacent channel rejection .....                     | 532 |
| 21.5.5   | Alternate channel rejection .....                    | 533 |
| 21.5.6   | TX-to-RX turnaround time .....                       | 533 |
| 21.5.7   | RX-to-TX turnaround time .....                       | 533 |
| 21.5.8   | EVM definition .....                                 | 533 |
| 21.5.9   | Transmit center frequency and symbol tolerance ..... | 535 |
| 21.5.10  | Transmit power .....                                 | 535 |
| 21.5.11  | Receiver maximum input level of desired signal ..... | 535 |
| 21.5.12  | Receiver ED .....                                    | 535 |
| 21.5.13  | LQI .....                                            | 535 |
| 22.      | SUN O-QPSK PHY .....                                 | 536 |
| 22.1     | Introduction .....                                   | 536 |

|          |                                                                                   |     |
|----------|-----------------------------------------------------------------------------------|-----|
| 22.2     | PPDU format for SUN O-QPSK .....                                                  | 536 |
| 22.2.1   | SHR field format .....                                                            | 536 |
| 22.2.1.1 | Preamble field format .....                                                       | 536 |
| 22.2.1.2 | SFD field format .....                                                            | 537 |
| 22.2.2   | PHR field format .....                                                            | 537 |
| 22.2.3   | PHY Payload field .....                                                           | 538 |
| 22.3     | Modulation and coding for SUN O-QPSK .....                                        | 538 |
| 22.3.1   | Reference modulator .....                                                         | 538 |
| 22.3.2   | SHR coding and spreading .....                                                    | 539 |
| 22.3.3   | PHR coding and spreading .....                                                    | 540 |
| 22.3.4   | PSDU coding and spreading for DSSS .....                                          | 540 |
| 22.3.5   | PSDU coding and spreading for MDSSS .....                                         | 542 |
| 22.3.6   | FEC .....                                                                         | 543 |
| 22.3.7   | Code-bit interleaving .....                                                       | 545 |
| 22.3.8   | Bit differential encoding (BDE) .....                                             | 546 |
| 22.3.9   | DSSS bit-to-chip mapping .....                                                    | 547 |
| 22.3.10  | MDSSS bit-to-chip mapping .....                                                   | 551 |
| 22.3.11  | Chip whitening .....                                                              | 555 |
| 22.3.12  | Pilot insertion .....                                                             | 556 |
| 22.3.13  | Modulation parameters for O-QPSK .....                                            | 557 |
| 22.4     | Support of legacy devices of the 780 MHz, 915 MHz, and 2450 MHz O-QPSK PHYs ..... | 558 |
| 22.5     | SUN O-QPSK PHY RF requirements .....                                              | 558 |
| 22.5.1   | Operating frequency range .....                                                   | 558 |
| 22.5.2   | Transmit power spectral density (PSD) mask .....                                  | 558 |
| 22.5.3   | Receiver sensitivity .....                                                        | 559 |
| 22.5.4   | Adjacent channel rejection .....                                                  | 559 |
| 22.5.5   | TX-to-RX turnaround time .....                                                    | 560 |
| 22.5.6   | RX-to-TX turnaround time .....                                                    | 560 |
| 22.5.7   | EVM definition .....                                                              | 560 |
| 22.5.8   | Transmit center frequency and symbol tolerance .....                              | 560 |
| 22.5.9   | Transmit power .....                                                              | 561 |
| 22.5.10  | Receiver maximum input level of desired signal .....                              | 561 |
| 22.5.11  | Receiver ED .....                                                                 | 561 |
| 22.5.12  | LQI .....                                                                         | 561 |
| 22.5.13  | CCA .....                                                                         | 561 |
| 23.      | LECIM DSSS PHYs .....                                                             | 562 |
| 23.1     | PPDU format for DSSS .....                                                        | 562 |
| 23.2     | Modulation and spreading .....                                                    | 562 |
| 23.2.1   | Data rate .....                                                                   | 562 |
| 23.2.2   | Reference modulator diagram .....                                                 | 563 |
| 23.2.3   | Convolutional FEC encoding .....                                                  | 563 |
| 23.2.4   | Interleaver .....                                                                 | 564 |
| 23.2.4.1 | 256-bit fragment size .....                                                       | 564 |
| 23.2.4.2 | 384-bit fragment size .....                                                       | 564 |
| 23.2.4.3 | 512-bit fragment size .....                                                       | 565 |
| 23.2.5   | Differential encoding .....                                                       | 565 |
| 23.2.6   | Bit-to-symbol and symbol-to-chip encoding .....                                   | 566 |
| 23.2.6.1 | Gold code generator .....                                                         | 566 |
| 23.2.6.2 | Orthogonal variable spreading factor (OVSF) code generator .....                  | 567 |
| 23.2.7   | BPSK/O-QPSK modulation .....                                                      | 570 |
| 23.2.7.1 | BPSK modulation .....                                                             | 570 |
| 23.2.7.2 | O-QPSK modulation .....                                                           | 570 |

|          |                                                  |     |
|----------|--------------------------------------------------|-----|
| 23.3     | PSDU fragmentation .....                         | 570 |
| 23.3.1   | Configuration .....                              | 571 |
| 23.3.2   | Fragmentation .....                              | 571 |
| 23.3.3   | Fragment packet .....                            | 572 |
| 23.3.4   | Calculating FICS field using MIC .....           | 572 |
| 23.3.5   | Fragment acknowledgment and retransmission ..... | 573 |
| 23.3.6   | Frak .....                                       | 573 |
| 23.3.6.1 | Frak policy .....                                | 573 |
| 23.3.6.2 | Frak format .....                                | 574 |
| 23.3.7   | Reassembly .....                                 | 575 |
| 23.4     | DSSS PHY RF requirements .....                   | 575 |
| 23.4.1   | Radio frequency tolerance .....                  | 575 |
| 23.4.2   | Channel switch time .....                        | 575 |
| 23.4.3   | Transmit spectral mask .....                     | 575 |
| 23.4.4   | Receiver sensitivity .....                       | 575 |
| 23.4.5   | Receiver interference rejection .....            | 576 |
| 23.4.6   | TX-to-RX turnaround time .....                   | 577 |
| 23.4.7   | RX-to-TX turnaround time .....                   | 577 |
| 23.4.8   | Transmit power .....                             | 577 |
| 24.      | LECIM FSK PHY specification .....                | 578 |
| 24.1     | General .....                                    | 578 |
| 24.2     | PPDU format for LECIM FSK PHY .....              | 578 |
| 24.2.1   | SHR field format .....                           | 578 |
| 24.2.1.1 | Preamble field format .....                      | 578 |
| 24.2.1.2 | SFD field format .....                           | 578 |
| 24.2.2   | PHR field format .....                           | 578 |
| 24.2.3   | PHY Payload field .....                          | 579 |
| 24.3     | Modulation and coding for LECIM FSK PHY .....    | 579 |
| 24.3.1   | Reference modulator .....                        | 579 |
| 24.3.2   | Bit-to-symbol mapping .....                      | 580 |
| 24.3.3   | Modulation quality .....                         | 581 |
| 24.3.3.1 | Frequency deviation tolerance .....              | 581 |
| 24.3.3.2 | Zero crossing tolerance .....                    | 581 |
| 24.3.4   | FEC .....                                        | 581 |
| 24.3.5   | Code-bit interleaving .....                      | 582 |
| 24.3.6   | Spreading .....                                  | 583 |
| 24.4     | Data whitening for LECIM FSK PHY .....           | 584 |
| 24.5     | LECIM FSK PHY RF requirements .....              | 585 |
| 24.5.1   | Operating frequency range .....                  | 585 |
| 24.5.2   | Radio frequency tolerance .....                  | 585 |
| 24.5.3   | Channel switch time .....                        | 585 |
| 24.5.4   | Transmit spectral mask .....                     | 585 |
| 24.5.5   | Receiver sensitivity .....                       | 585 |
| 24.5.6   | TX-to-RX turnaround time .....                   | 586 |
| 24.5.7   | RX-to-TX turnaround time .....                   | 586 |
| 24.5.8   | Transmit power .....                             | 586 |
| 25.      | TVWS-FSK PHY .....                               | 587 |
| 25.1     | PPDU format for TVWS-FSK .....                   | 587 |
| 25.1.1   | SHR field format .....                           | 587 |
| 25.1.1.1 | Preamble field format .....                      | 587 |

|          |                                                      |     |
|----------|------------------------------------------------------|-----|
| 25.1.1.2 | SFD field format .....                               | 587 |
| 25.1.2   | PHR field format .....                               | 587 |
| 25.1.3   | PHY Payload field .....                              | 588 |
| 25.2     | Modulation and coding for TVWS-FSK .....             | 588 |
| 25.2.1   | Reference modulator .....                            | 589 |
| 25.2.2   | FEC and interleaving .....                           | 589 |
| 25.2.3   | Data whitening .....                                 | 589 |
| 25.2.4   | Spreading .....                                      | 589 |
| 25.2.5   | Bit-to-symbol mapping .....                          | 590 |
| 25.2.6   | Modulation quality .....                             | 590 |
| 25.2.7   | Values for phySymbolsPerOctet .....                  | 590 |
| 25.3     | TVWS-FSK RF requirements .....                       | 591 |
| 25.3.1   | Operating frequency range .....                      | 591 |
| 25.3.2   | Clock frequency and timing accuracy .....            | 591 |
| 25.3.3   | Channel switch time .....                            | 591 |
| 25.3.4   | Receiver sensitivity .....                           | 591 |
| 25.3.5   | TX-to-RX turnaround time .....                       | 591 |
| 25.3.6   | RX-to-TX turnaround time .....                       | 591 |
| 25.3.7   | Receiver maximum input level of desired signal ..... | 591 |
| 25.3.8   | Receiver ED .....                                    | 591 |
| 25.3.9   | LQI .....                                            | 591 |
| 26.      | TVWS-OFDM PHY .....                                  | 592 |
| 26.1     | General .....                                        | 592 |
| 26.2     | PPDU format for TVWS-OFDM .....                      | 592 |
| 26.2.1   | STF .....                                            | 592 |
| 26.2.1.1 | Frequency domain STF .....                           | 592 |
| 26.2.1.2 | Time domain STF generation .....                     | 593 |
| 26.2.1.3 | Time domain STF repetition .....                     | 594 |
| 26.2.1.4 | STF power boosting .....                             | 594 |
| 26.2.2   | LTF .....                                            | 594 |
| 26.2.2.1 | Frequency domain LTF .....                           | 594 |
| 26.2.2.2 | Time domain LTF generation .....                     | 595 |
| 26.2.3   | PHR field format .....                               | 596 |
| 26.2.4   | PSDU field .....                                     | 596 |
| 26.3     | System parameters for TVWS-OFDM .....                | 597 |
| 26.4     | Modulation and coding for TVWS-OFDM .....            | 597 |
| 26.4.1   | Reference modulator .....                            | 597 |
| 26.4.2   | Bit-to-symbol mapping .....                          | 598 |
| 26.4.3   | FEC .....                                            | 599 |
| 26.4.4   | Interleaver .....                                    | 599 |
| 26.4.5   | Pilot tones/null tones .....                         | 601 |
| 26.4.6   | CP .....                                             | 602 |
| 26.4.7   | PPDU Tail field .....                                | 602 |
| 26.4.8   | Pad field .....                                      | 602 |
| 26.4.9   | Scrambler and scrambler seeds .....                  | 602 |
| 26.5     | TVWS-OFDM RF requirements .....                      | 603 |
| 26.5.1   | Operating frequency range .....                      | 603 |
| 26.5.2   | Pulse shaping .....                                  | 603 |
| 26.5.3   | Transmit power spectral density (PSD) mask .....     | 603 |
| 26.5.4   | Receiver sensitivity .....                           | 603 |
| 26.5.5   | TX-to-RX turnaround time .....                       | 603 |
| 26.5.6   | RX-to-TX turnaround time .....                       | 603 |

|          |                                                         |     |
|----------|---------------------------------------------------------|-----|
| 26.5.7   | EVM definition .....                                    | 603 |
| 26.5.8   | Transmit center frequency and symbol tolerance .....    | 605 |
| 27.      | TVWS-NB-OFDM PHY .....                                  | 606 |
| 27.1     | PPDU format for TVWS-NB-OFDM .....                      | 606 |
| 27.1.1   | Short Training field (STF) .....                        | 606 |
| 27.1.1.1 | Frequency domain STF .....                              | 606 |
| 27.1.1.2 | Time domain STF generation .....                        | 609 |
| 27.1.1.3 | Time domain STF repetition .....                        | 609 |
| 27.1.1.4 | STF normalization .....                                 | 610 |
| 27.1.2   | Long training field (LTF) .....                         | 610 |
| 27.1.2.1 | Frequency domain LTF generation .....                   | 610 |
| 27.1.2.2 | Time domain LTF generation .....                        | 613 |
| 27.1.2.3 | Time domain LTF repetition .....                        | 614 |
| 27.1.2.4 | LTF normalization .....                                 | 614 |
| 27.1.3   | PHR .....                                               | 615 |
| 27.1.4   | PHY Payload field .....                                 | 615 |
| 27.2     | System parameters for TVWS-NB-OFDM .....                | 615 |
| 27.3     | Modulation and coding parameters for TVWS-NB-OFDM ..... | 616 |
| 27.3.1   | Reference modulator .....                               | 616 |
| 27.3.2   | Scrambler and scrambler seed .....                      | 617 |
| 27.3.3   | Outer encoding .....                                    | 617 |
| 27.3.4   | Inner encoding .....                                    | 618 |
| 27.3.5   | Pad bit insertion .....                                 | 620 |
| 27.3.6   | Spreader .....                                          | 620 |
| 27.3.7   | Bit interleaving .....                                  | 621 |
| 27.3.8   | Subcarrier mapping .....                                | 621 |
| 27.3.9   | Frequency interleaving .....                            | 623 |
| 27.3.10  | Pilot tones .....                                       | 624 |
| 27.3.11  | Cyclic prefix .....                                     | 624 |
| 27.3.12  | Pulse shaping .....                                     | 624 |
| 27.3.13  | PIB attribute values for phySymbolsPerOctet .....       | 625 |
| 27.4     | Channel aggregation for TVWS-NB-OFDM .....              | 625 |
| 27.5     | TVWS-NB-OFDM RF requirements .....                      | 625 |
| 27.5.1   | Operating frequency range .....                         | 625 |
| 27.5.2   | Receiver sensitivity .....                              | 625 |
| 27.5.3   | TX-to-RX turnaround time .....                          | 626 |
| 27.5.4   | RX-to-TX turnaround time .....                          | 626 |
| 27.5.5   | EVM definition .....                                    | 626 |
| 27.5.6   | Transmit center frequency and symbol tolerance .....    | 627 |
| 28.      | RCC LMR PHY .....                                       | 628 |
| 28.1     | RCC PHY overview .....                                  | 628 |
| 28.2     | PPDU format .....                                       | 628 |
| 28.2.1   | SHR .....                                               | 628 |
| 28.2.2   | PHR .....                                               | 629 |
| 28.2.3   | PHY payload .....                                       | 629 |
| 28.2.4   | Tail bits .....                                         | 629 |
| 28.3     | FEC .....                                               | 629 |
| 28.4     | Interleaver .....                                       | 630 |
| 28.5     | Data whitening .....                                    | 630 |
| 28.6     | Modulation .....                                        | 631 |

|         |                                                                                   |     |
|---------|-----------------------------------------------------------------------------------|-----|
| 28.6.1  | GMSK .....                                                                        | 631 |
| 28.6.2  | 4-FSK .....                                                                       | 631 |
| 28.6.3  | QPSK .....                                                                        | 632 |
| 28.6.4  | p/4 DQPSK .....                                                                   | 633 |
| 28.6.5  | DSSS DPSK .....                                                                   | 633 |
| 28.7    | Reference modulator .....                                                         | 635 |
| 28.8    | LMR PHY RF requirements .....                                                     | 635 |
| 28.8.1  | Transmitter symbol rate tolerance .....                                           | 635 |
| 28.8.2  | Channel switching time .....                                                      | 635 |
| 28.8.3  | Error vector magnitude .....                                                      | 635 |
| 28.8.4  | Receiver sensitivity .....                                                        | 635 |
| 28.8.5  | Receiver interference rejection .....                                             | 635 |
| 28.8.6  | Receiver maximum input level of desired signal .....                              | 636 |
| 28.8.7  | TX-to-RX turnaround time .....                                                    | 636 |
| 28.8.8  | RX-to-TX turnaround time .....                                                    | 636 |
| 28.8.9  | Receiver ED .....                                                                 | 636 |
| 28.8.10 | LQI .....                                                                         | 636 |
| 29.     | RCC DSSS BPSK PHY .....                                                           | 637 |
| 29.1    | Overview .....                                                                    | 637 |
| 29.2    | RCC DSSS BPSK PHY specification .....                                             | 637 |
| Annex A | (informative) Bibliography .....                                                  | 638 |
| Annex B | (normative) CCM* mode of operation .....                                          | 640 |
| Annex C | (informative) Test vectors for cryptographic building blocks .....                | 646 |
| Annex D | (informative) Protocol implementation conformance statement (PICS) proforma ..... | 659 |
| Annex E | (informative) MPSK PHY .....                                                      | 678 |
| Annex F | (normative) Time-slot relaying based link extension (TRLE) .....                  | 683 |

# IEEE Standard for Low-Rate Wireless Networks

**IMPORTANT NOTICE:** IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

## 1. Overview

### 1.1 Scope

This standard 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. In addition, the standard provides modes that allow for precision ranging. PHYs are defined for devices operating various license-free bands in a variety of geographic regions.

### 1.2 Purpose

The standard provides for ultra low complexity, ultra low cost, ultra low power consumption, and low data rate wireless connectivity among inexpensive devices. In addition, one of the alternate PHYs provides precision ranging capability that is accurate to one meter. Multiple PHYs are defined to support a variety of frequency bands.

## 2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

ANSI X3.66-1979, Advanced Data Communication Control Procedures.<sup>1</sup>

Code of Federal Regulations, Title 47: Telecommunication, Part 90—Private Land Mobile Radio Services, Subpart S—Regulations Governing Licensing and Use of Frequencies in the 806–824, 851–869, 896–901, and 935–940 MHz Bands (47 CFR, Part 90, Subpart S).<sup>2</sup>

Code of Federal Regulations, Title 47: Telecommunication, Part 90—Private Land Mobile Radio Services, Subpart Y—Regulations Governing Licensing and Use of Frequencies in the 4940–4990 MHz Band (47 CFR, Part 90, Subpart Y).

FIPS Pub 197, Advanced Encryption Standard (AES).<sup>3</sup>

IEEE Std 802<sup>®</sup>-2014, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.<sup>4, 5</sup>

IETF RFC 6225, Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information, Internet Engineering Task Force.<sup>6</sup>

<sup>1</sup>ANSI publications are available from the American National Standards Institute (<http://www.ansi.org/>)

<sup>2</sup>CFR publications are available from the U.S. Government Printing Office (<http://www.gpo.gov/>).

<sup>3</sup>FIPS publications are available from the National Technical Information Service (NTIS) (<http://www.ntis.org/>).

<sup>4</sup>IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

<sup>5</sup>The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

<sup>6</sup>IETF RFCs are available from the Internet Engineering Task Force (<http://www.ietf.org/>).



### 3. Definitions, acronyms, and abbreviations

#### 3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary* should be consulted for terms not defined in this clause.<sup>7</sup>

**association:** The service used to establish membership for a device in a network.

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

**channel hopping:** Periodically switching the channel using a sequence known to both sending and receiving devices where the entire frame is sent on a single channel.

**channel offset:** A number used in the channel calculation of a slotted channel hopping system to allow for different channels to be used in the same slot.

**chirp:** Linear frequency sweep (frequency may either increase or decrease).

**dependent device:** A device that operates without direct Internet access to the television white space (TVWS) database and depends on another device for channel availability information.

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

**fragment:** An individual subset of a MAC protocol data unit.

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

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

**independent device:** A device that has direct access to the television white space (TVWS) database via the Internet.

**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.

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

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

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

**nonce:** A nonrepeating value, such as an increasing counter, a sufficiently long random string, or a time stamp.

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

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

<sup>7</sup>IEEE Standards Dictionary subscription is available at  
[http://www.ieee.org/portal/innovate/products/standard/standards\\_dictionary.html](http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html).

**radio frequency identification (RFID):** The use of electromagnetic or inductive coupling in the radio frequency (RF) portion of the spectrum to communicate to or from a tag through a variety of modulation and encoding schemes to uniquely read the identity of an RF tag.

**slotframe:** A collection of timeslots repeating in time, analogous to a superframe in that it defines periods of communication opportunities.

**smart utility network (SUN):** a principally outdoor, low data rate wireless network that supports two-way communications among sensing, measurement, and control devices in the smart grid.

**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.

**timeslot:** A defined period of time during which a frame and an acknowledgment may be exchanged between devices.

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

### 3.2 Acronyms and abbreviations

|         |                                                                                            |
|---------|--------------------------------------------------------------------------------------------|
| Ack     | acknowledgment                                                                             |
| AES     | advanced encryption standard                                                               |
| AIFS    | acknowledgment interframe spacing                                                          |
| AR      | acknowledgment request                                                                     |
| ASK     | amplitude shift keying                                                                     |
| ASN     | absolute slot number                                                                       |
| ATI     | allowed transmission interval                                                              |
| BCH     | Bose Chaudhuri Hocquenghem                                                                 |
| BDE     | bit differential encoding                                                                  |
| BLE     | battery life extension                                                                     |
| BOP     | beacon only period                                                                         |
| BPM     | burst position modulation                                                                  |
| 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 mode encryption and cipher block chaining message authentication code              |
| CCM*    | extension of counter mode encryption and cipher block chaining message authentication code |
| CFP     | contention-free period                                                                     |
| CID     | company identifier                                                                         |
| CP      | cyclic prefix                                                                              |
| CoU     | chirp on ultra-wide band                                                                   |
| CRC     | cyclic redundancy check                                                                    |
| CS      | continuous spectrum                                                                        |
| CSK     | chirp-shift keying                                                                         |
| CSL     | coordinated sampled listing                                                                |

|         |                                                        |
|---------|--------------------------------------------------------|
| CSM     | common signaling mode                                  |
| CSMA-CA | carrier sense multiple access with collision avoidance |
| CSS     | chirp spread spectrum                                  |
| DA      | device announcement                                    |
| DAA     | detect and avoid                                       |
| DBS     | dedicated beacon slot                                  |
| DEMUX   | de-multiplexer                                         |
| DFT     | discrete Fourier transform                             |
| DPS     | dynamic preamble selection                             |
| DPSK    | differential phase-shift keying                        |
| DQCSK   | differential quadrature chirp-shift keying             |
| DQPSK   | differential quadrature phase-shift keying             |
| DSME    | deterministic and synchronous multi-channel extension  |
| DSN     | data sequence number                                   |
| DSSS    | direct sequence spread spectrum                        |
| EBSN    | enhanced beacon sequence number                        |
| ED      | energy detection (also in 6.2.8, extended duration)    |
| EIRP    | effective isotropic radiated power                     |
| Enh-Ack | enhanced acknowledgment                                |
| ESDU    | encapsulated service data unit                         |
| EUI-64  | 64-bit extended unique identifier                      |
| EVM     | error-vector magnitude                                 |
| FCC     | Federal Communications Commission                      |
| FCS     | frame check sequence                                   |
| FEC     | forward error correction                               |
| FFD     | full-function device                                   |
| FICS    | fragment integrity check sequence                      |
| FoM     | figure of merit                                        |
| Frak    | fragment acknowledgment                                |
| FSCD    | fragment sequence context description                  |
| FSK     | frequency shift keying                                 |
| GDB     | geolocation database                                   |
| GFSK    | Gaussian frequency-shift keying                        |
| GMSK    | Gaussian-filtered minimum shift keying                 |
| GTS     | guaranteed time slot                                   |
| HCS     | header check sequence                                  |
| HRP     | high rate pulse repetition frequency                   |
| ID      | identifier                                             |
| IDFT    | inverse discrete Fourier transform                     |
| IE      | information element                                    |
| IFS     | interframe space or spacing                            |
| Imm-Ack | immediate acknowledgment                               |
| ISR     | interference-to-signal ratio                           |
| I-RIT   | implicit receiver initiated transmission               |
| LBT     | listen before talk                                     |
| LCP     | linear combination of pulses                           |

|          |                                                     |
|----------|-----------------------------------------------------|
| LE       | low energy                                          |
| LECIM    | low-energy, critical infrastructure monitoring      |
| LEIP     | location enhancing information postamble            |
| LFSR     | linear feedback shift register                      |
| LIFS     | long interframe spacing                             |
| LMR      | land mobile radio                                   |
| LRP      | low rate pulse repetition frequency                 |
| LQI      | link quality indication                             |
| LR-WPAN  | low-rate wireless personal area network             |
| LSB      | least significant bit                               |
| LTF      | long training field                                 |
| MAC      | medium access control                               |
| MBAN     | medical body area network                           |
| MCPS     | MAC common part sublayer                            |
| MCPS-SAP | MAC common part sublayer service access point       |
| MCS      | modulation and coding scheme                        |
| MD       | multi-superframe duration                           |
| MDSSS    | multiplexed direct sequence spread spectrum         |
| MFR      | MAC footer                                          |
| MHR      | MAC header                                          |
| MIC      | message integrity code                              |
| MLME     | MAC sublayer management entity                      |
| MLME-SAP | MAC sublayer management entity service access point |
| MPM      | multi-PHY management                                |
| MSB      | most significant bit                                |
| MSK      | minimum shift keying                                |
| MPDU     | MAC protocol data unit                              |
| MPSK     | m-ary phase shift keying                            |
| MSDU     | MAC service data unit                               |
| NRNSC    | nonrecursive and nonsystematic code                 |
| OOK      | on-off keying                                       |
| O-QPSK   | offset quadrature phase-shift keying                |
| OUI      | organizationally unique identifier                  |
| OVSF     | orthogonal variable spreading factor                |
| PAN      | personal area network                               |
| PC       | personal computer                                   |
| PCA      | priority channel access                             |
| PD       | physical layer data                                 |
| PD-SAP   | physical layer data service access point            |
| PER      | packet error rate                                   |
| P-FSK    | position-based frequency shift keying               |
| PHR      | PHY header                                          |
| PHY      | physical layer                                      |
| PIB      | personal area network 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                                   |
| PPDU     | PHY protocol data unit                                |
| PPM      | pulse position modulation                             |
| PRBS     | pseudo-random binary sequence                         |
| PRF      | pulse repetition frequency                            |
| PSD      | power spectral density                                |
| PSDU     | PHY service data unit                                 |
| PSSS     | parallel sequence spread spectrum                     |
| QAM      | quadrature amplitude modulation                       |
| QPSK     | quadrature phase-shift keying                         |
| RCC      | rail communications and control                       |
| RCCN     | rail communications and control network               |
| RDEV     | ranging-capable device                                |
| RF       | radio frequency                                       |
| RFD      | reduced-function device                               |
| RFD-RX   | reduced function device—receive only                  |
| RFD-TX   | reduced function device—transmit only                 |
| RFID     | radio frequency identification                        |
| RFRAME   | ranging frame                                         |
| RIT      | receiver initiated transmission                       |
| RIV      | remainder initialization value                        |
| RMARKER  | ranging marker                                        |
| RSC      | recursive and systematic code                         |
| RSSI     | receive signal strength indicator                     |
| RX       | receive or receiver                                   |
| SAB      | slot allocation bitmap                                |
| SD       | superframe duration                                   |
| SF       | spreading factor                                      |
| SFD      | start-of-frame delimiter                              |
| SHR      | synchronization header                                |
| SIFS     | short interframe spacing                              |
| SNR      | signal-to-noise ratio                                 |
| SPC      | super PAN coordinator                                 |
| STF      | short training field                                  |
| SUN      | smart utility network                                 |
| TID      | transaction identifier                                |
| TMCTP    | TVWS multichannel cluster tree PAN                    |
| TPC      | turbo product code                                    |
| TRLE     | time-slot relaying based link extension               |
| TSCH     | timeslotted channel hopping                           |
| TVWS     | television white space                                |
| TX       | transmit or transmitter                               |
| UWB      | ultra-wide band                                       |
| WPAN     | wireless personal area network                        |

## 4. Format conventions

### 4.1 General

Throughout this standard, unless otherwise stated, data structures exposed in interfaces are represented using the conventions and formats defined in this clause.

NOTE—It is important to note that interfaces are depicted in this standard as bit serial by convention but may be implemented in other forms, e.g., 4- or 8-bit parallel.<sup>8</sup>

Within the PHY, data structures are passed to the reference modulator input in bit-serial convention. PHY specific coding may change the bit-order and encode multiple information bits into symbols. Hence the over-the-air interface may express the data structure content in a manner different than the conventions defined here.

After all applicable expansion, the size of a MAC frame shall be less than the maximum PSDU size supported by the PHY in use.

### 4.2 Fields

The general format of a data structure is shown in Figure 4-1. Each field is represented by a column in the figure giving the size of the field and the name and/or type of the datum encoded in the field. The convention of a data structure is recursive in that a field may contain a data structure which is in turn composed of fields.

| Octets: 2                         | Bits: 0–5                             | 6–7                                   | ... |
|-----------------------------------|---------------------------------------|---------------------------------------|-----|
| Field Name/Type<br>(Number datum) | Field Name/Type<br>(Bit string datum) | Field Name/Type<br>(Bit string datum) | ... |

**Figure 4-1—General data structure format**

The form <Octets:  $n$ > means the field is  $n$  octets long. The form <Bits:  $n$ > or <Bits:  $m$ – $n$ > means bit number  $n$  or bits  $m$  to  $n$  of the data structure. The terms octet and bit may also be written as octets or bits. In a figure, if a field length appears without either Octets or Bits, then the preceding units apply.

The form <Octets:  $n/m/l$ > means the field can be  $n$ ,  $m$  or  $l$  octets long.

Fields are concatenated into larger data structures, e.g., Figure 4-1 represents a data structure with 3 fields of length 2 octets, 6 bits and 2 bits for a total size of 3 octets.

Order of representation is strict with the leftmost field as shown in any figure occurring before the next field to its right. The second field from the left occurs before the third field and so on.

Significance, as expressed in the terms LSB and MSB, only applies to numbers as defined in 4.3. All other fields are treated as strings.

The convention of 'processing a data structure' or 'transmitting a data structure' mean treatment of the content of the data structure in the order defined in the following subclauses.

<sup>8</sup>Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

### 4.3 Numbers

Numbers are represented in decimal (nnn), binary (0bnnn), or hexadecimal (0xnnn) notation with the number of digits indicated. For example, 0b001 is a 3 digit binary number of value 1 and 0x001 is a 3 digit hexadecimal number of value 1. Numbers encoded in fields may be signed or unsigned integers. Other formats (e.g., a field containing a floating point number) are stated in the field definition where applicable.

When represented in fields, numbers are expressed with LSB leftmost and MSB rightmost as shown in Figure 4-2 for an  $n$ -bit unsigned integer. (Note that this is the inverse of the number representations 0bnnn and 0xnnn and of common number representations used in mathematical notations.) For example, the number 0x47 would be represented in an 8-bit field as 1110 0010.

|       |     |   |                   |       |
|-------|-----|---|-------------------|-------|
| Bit:0 | ... | 7 | Bits 8-( $n-2$ )  | $n-1$ |
| LSB   | ... | — | Subsequent octets | MSB   |

**Figure 4-2—General number field format**

Numbers of size greater than 1 octet occur with the octet containing the least significant bits first (leftmost) followed by octets containing bits of increasing significance to the right.

Unless otherwise stated in the definition of a field, a number in a field shall be formatted as an unsigned integer.

### 4.4 Strings

A string of length  $k$  bits is represented as a bit sequence numbered from 0 to  $k-1$ . Bit 0 occurs first and is represented as the leftmost bit of a string field. Bit 1 occurs to the right of bit 0 and so on until bit ( $k-1$ ). This is illustrated in Figure 4-3.

|              |     |              |                        |
|--------------|-----|--------------|------------------------|
| Bits:0       | ... | 7            | Bits 8-( $k-1$ )       |
| String bit 0 | ... | String bit 7 | String bits 8 to $k-1$ |

**Figure 4-3—General string field format**

A string of length greater than 1 octet is represented by the octet containing the lowest numbered bits first and leftmost, followed by octets containing increasing bit numbers to the right.

### 4.5 Reserved fields and values

Each bit within any Reserved field shall be set to zero on transmission and shall be ignored on reception.

No decision should be made on the contents of any Reserved field or field containing a reserved value.

## 5. General description

### 5.1 Introduction

A low-rate wireless personal area network (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, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol.

This standard defines multiple PHYs operating in a variety of frequency bands, as described in 10.1.1.

Two different device types can participate in an IEEE 802.15.4 network: a full-function device (FFD) and a reduced-function device (RFD). An FFD is a device that is capable of serving as a personal area network (PAN) coordinator or a coordinator, as defined in 6.1. An RFD is a device that is not capable of serving as either a PAN coordinator or a coordinator. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; it does not have the need to send large amounts of data and only associates with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources and memory capacity.

Supplemental information for different industrial domains is described in “Applications of IEEE Std 802.15.4” [B3].<sup>9</sup>

### 5.2 Special application spaces

Although this standard is intended address many diverse application spaces, some application spaces have unique requirements which required specific elements added to this standard. These elements are described in this subclause.

#### 5.2.1 Smart utility network (SUN)

SUNs enable multiple applications to operate over shared network resources, providing monitoring and control of a utility system. SUN devices are designed to operate in very large-scale, low-power wireless applications and often require using the maximum transmit power available under applicable regulations, in order to provide long-range, point-to-point connections. Frequently, SUNs are required to cover geographically widespread areas containing a large number of outdoor devices. In these cases, SUN devices are able to employ mesh or peer-to-peer multihop techniques to communicate with an access point.

#### 5.2.2 Rail communications and control (RCC)

RCC refers to a wireless information exchange and sensor or control communications deployed in areas such as the following:

- A wireless link between trains, locomotives, or other mobile rolling stock to fixed trackside or network infrastructure
- A link between connected fixed, remote trackside infrastructure and fixed network infrastructure
- A link between vehicles in the same train or between two or more trains

RCC devices are intended to support mobile rail vehicle communications at high speeds with data rates that enable connections at distances of over 50 km. The RCC PHYs are designed to take advantage of relatively small amounts of spectrum where spectrum is costly or scarce.

<sup>9</sup>The numbers in brackets correspond to the numbers of the bibliography in Annex A.



### 5.2.3 Television white space (TVWS)

TVWS operation has the requirement to determine which TVWS frequency allocations are available for use at a given time and geographic location. TVWS devices need to have access to TVWS channel availability information, for example, via a central database that is accessed over the Internet. A TVWS device that has no connection to the Internet must depend upon another TVWS device that has access to the TVWS database via the Internet to acquire channel availability information.

### 5.2.4 Radio frequency identification (RFID)

Active RFID devices are used to identify and often locate people or objects in industrial or commercial environments. Typical applications include asset management, inventory management, process control and automation, safety and accountability.

In its simplest form an active RFID system comprises a number of transmit-only tags that periodically transmit a packet containing a unique ID and a small amount of data. The packet is received by one or more readers that may simply register the tag as present, may employ further processing to determine the location of the tag, or forward data to an application server. More complex active RFID systems might employ two-way communications with the tag for control, communication, and coordination.

### 5.2.5 Low-energy, critical infrastructure monitoring (LECIM)

The LECIM portions of this standard are designed to implement a minimal network infrastructure that enables the collection of scheduled and event data from a large number of non-mains powered end points that are widely dispersed, or are in challenging propagation environments. To facilitate low energy operation necessary for multi-year battery life, MAC protocols that minimize network maintenance traffic and device wake durations have been defined.

### 5.2.6 Medical body area network (MBAN) services

Some countries have allocated spectrum for MBAN services on a secondary basis such that MBAN devices are required to protect all primary users and accept possible interference from those users. MBAN devices operating within this allocated spectrum conform to a set of rules which restrict use of the band to only medical, non-voice use under direction of a healthcare practitioner, among other requirements. When a primary user is making use of a portion of the band, MBAN devices vacate that portion of the band. Use of the band by the primary user is, in general, scheduled well in advance allowing MBAN users to share the band in an orderly manner.

## 5.3 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 has a single radio interface that implements an IEEE Std 802.15.4 MAC and PHY. Two or more devices communicating on the same physical channel constitute a wireless personal area network (WPAN). A WPAN includes at least one FFD, operating as the PAN coordinator.

## 5.4 Multi-PHY management (MPM) of the SUN WPAN

Multiple, different SUN PHYs can operate in the same location and within the same frequency band. In order to mitigate interference, an MPM scheme is specified for SUNs to facilitate inter-PHY coexistence. For this purpose, the MPM scheme facilitates interoperability and negotiation among potential coordinators with different PHYs by permitting a potential coordinator to detect an operating network during its discovery phase using the common signaling mode (CSM) appropriate to the band being used. The MPM

procedure can be used in conjunction with the clear channel assessment (CCA) mechanism to provide coexistence.

## 5.5 Network topologies

Depending on the application requirements, an IEEE 802.15.4 LR-WPAN operates in either of two topologies: the star topology or the peer-to-peer topology. Both are shown in Figure 5-1. 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. The PAN coordinator is the primary controller of the PAN. All devices operating on a network of either topology have unique addresses, referred to as extended addresses. In addition, a device can be assigned a short address during the association process. A device will use either the extended address or the short address for communication within the PAN. The transmission of the extended and short address fields is optional for RFD-TX devices. The PAN coordinator will often be mains powered, while the devices will likely be battery powered. Applications that benefit from a star topology include home automation, personal computer (PC) peripherals, games, and personal health care.

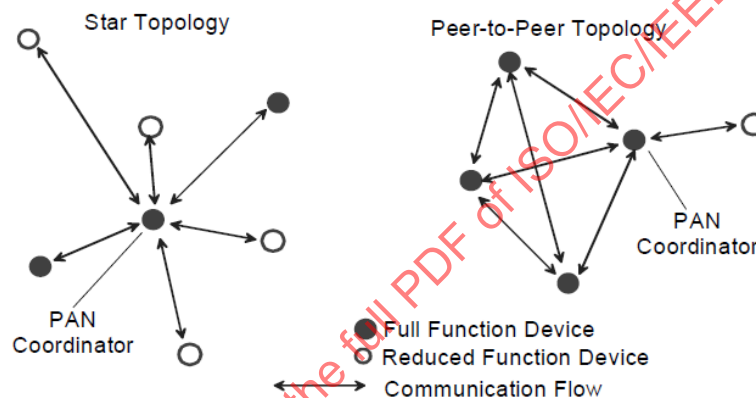


**Figure 5-1—Star and peer-to-peer topology examples**

The peer-to-peer topology also has a PAN coordinator; however, it differs from the star topology in that any device is able to 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 allows 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 they are not part of this standard.

Each independent PAN selects a unique identifier. This PAN identifier (ID) 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.

### 5.5.1 Star network formation

The basic structure of a star network is illustrated in Figure 5-1. 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 ID that is not currently used by any other network within the radio communications range. Once the PAN ID is chosen, the PAN coordinator

allows other devices, potentially both FFDs and RFDs, to join its network. The higher layer can use the procedures described in 6.3 and 6.4 to form a star network. An RFD-RX can also serve as a PAN coordinator termination point for RFD-TXs.

Timeslotted channel hopping (TSCH) PANs are topology independent and can be used in star topologies as well as partial or full mesh topologies.

LECIM networks primarily operate in a star topology. LECIM networks are typically asymmetric in energy supply and capability, having a PAN coordinator that is mains powered (or otherwise provided a substantial power source) and energy and/or cost constrained devices. The PAN coordinator typically monitors the channel more often than a device. A device sleeps unless it has a Data frame to send.

### 5.5.2 Peer-to-peer network formation

In a peer-to-peer topology, each device is capable of communicating with any other device within its radio communications range. 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 FFD is able to act as a coordinator and provide synchronization services to other devices or other coordinators. Only one of these coordinators is the overall PAN coordinator, potentially because it has greater computational resources than any other device in the PAN. The PAN coordinator forms the first cluster by choosing an unused PAN ID 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 is able to 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 are able to 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 6.3 and 6.4.

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 instructs 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 5-2. The lines in Figure 5-2 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.

A TMCTP is a form of cluster tree network where the SPC is the overall PAN coordinator providing synchronization services to other PAN coordinators in the cluster. The SPC also has access to the geolocation database (GDB) server to provide TVWS channel availability information to other PAN coordinators. An example is shown in Figure 5-3. In the TMCTP, collisions between clusters can be reduced because each cluster uses its own channel; in addition, the coverage area is increased through the TMCTP parent-child structure. Each TMCTP-parent PAN coordinator, including the SPC, communicates with its TMCTP-child PAN coordinators during the CAP or CFP of the TMCTP-parent PAN coordinator superframe and receives beacon frames of its TMCTP-child PAN coordinators on a dedicated channel during the dedicated beacon slots (DBS) assigned to them in the BOP, as shown with an asterisk (\*) in Figure 5-3.

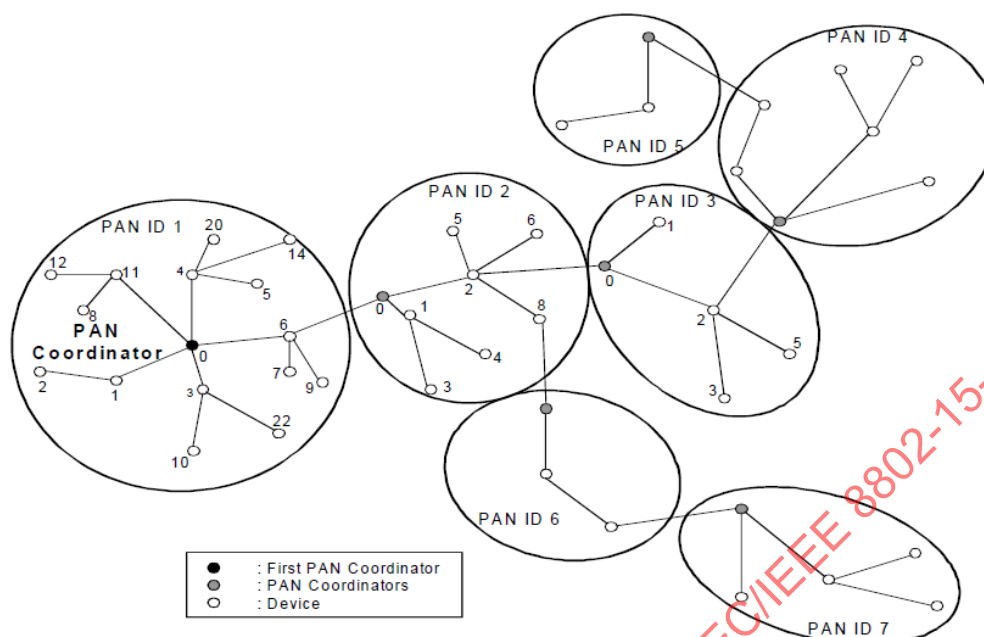


Figure 5-2—Cluster tree network

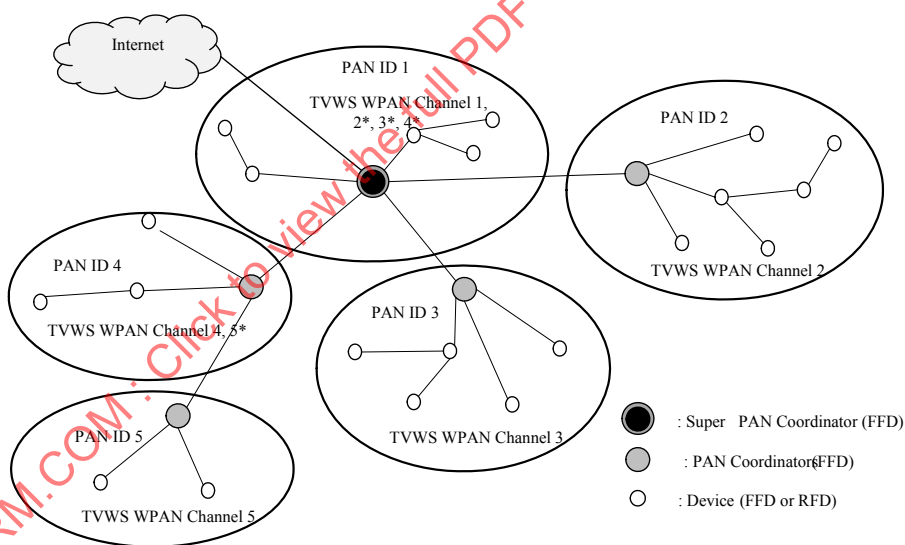


Figure 5-3—Example of TVWS multichannel cluster tree PAN

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

An LR-WPAN device comprises at least one 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 5-4 shows these blocks in a graphical representation.

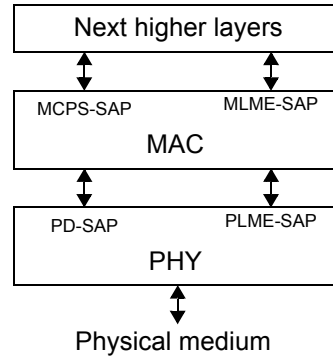


**Figure 5-4—LR-WPAN device architecture**

The next higher layers, shown in Figure 5-4, 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 layers is outside the scope of this standard.

### 5.6.1 PHY

The features of the PHY are activation and deactivation of the radio transceiver, energy detection (ED), link quality indication (LQI), channel selection, CCA, ranging and transmitting as well as receiving packets across the physical medium. The HRP UWB PHY also has the feature of precision ranging.

Coexistence with other wireless standards has been analyzed for the PHYs defined in this standard. These analyses can be found in the following documents:

- For the O-QPSK PHY, BPSK PHY, ASK PHY, CSSS PHY, HRP UWB PHY, and GFSK PHY: “Coexistence analysis of IEEE Std 802.15.4 with other IEEE standards and proposed standards” [B5].
- For the MSK PHY and LRP UWB PHY: “TG4f Coexistence Assurance Document” [B17].
- For TVWS-FSK PHY, TVWS-OFDM PHY, and TVWS-NB-OFDM PHY: Chang and Seibert [B4].

### 5.6.2 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.

## 5.7 Functional overview

### 5.7.1 Superframe structure

#### 5.7.1.1 Beacon superframe

The format of the superframe is defined by the coordinator. The superframe is bounded by beacons (either Beacon frames or Enhanced Beacon Frames) sent by the coordinator, as illustrated in Figure 5-5a). Optionally, the superframe can have an active and an inactive portion, as illustrated in Figure 5-5b). During the inactive portion, the coordinator is able to enter a low-power mode. The beacon frame transmission starts at the beginning of 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 devices, to identify the PAN, and to describe the structure of the superframes.

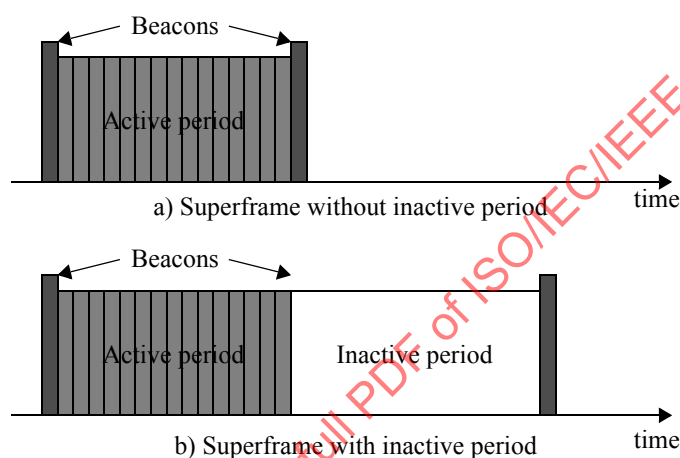


**Figure 5-5—Superframe structure**

Any device wishing to communicate during the contention access period (CAP) between two beacons competes with other devices using a slotted CSMA-CA or ALOHA mechanism, as appropriate. For low-latency applications or applications requiring specific data bandwidth, the PAN coordinator dedicates 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 5-6. The PAN coordinator allocates up to seven of these GTSs, and a GTS is allowed to 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 are 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 6.2.1.

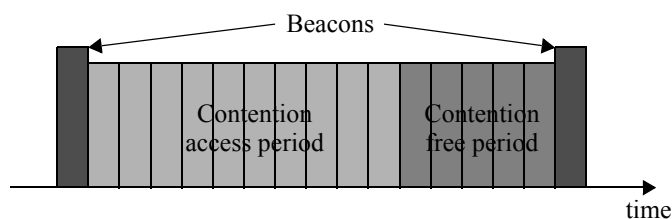


**Figure 5-6—Structure of the active periods with GTSs**

### 5.7.1.2 DSME multi-superframe structure

PANs with deterministic and synchronous multi-channel extension (DSME) enabled use the DSME multi-superframe structure. The format of the DSME multi-superframe structure is defined by coordinators that periodically transmit an Enhanced Beacon frame with the DSME PAN descriptor IE. A multi-superframe is a cycle of repeated superframes, each of which consists of an Enhanced Beacon frame, a CAP and a CFP. An example of a multi-superframe structure is shown in Figure 5-7.

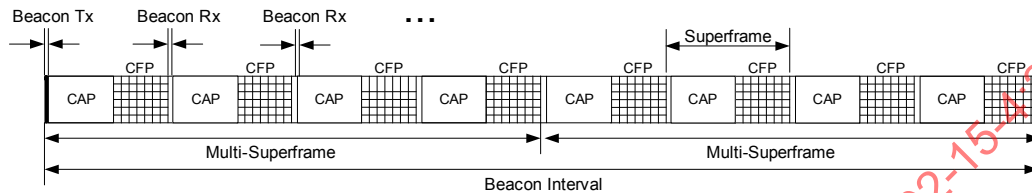


Figure 5-7—General DSME Multi-superframe Structure

The single common channel, the logical channel number used in the successful association, is used to transmit the enhanced beacon frames and the frames transmitted during the CAP. The channel diversity method, either channel adaptation or channel hopping as selected by *macChannelDiversityMode*, is applied to transmit Data frames and acknowledge frames during the CFP. Beacons are scheduled as defined in 6.11.6. Frames sent during the CFP are transmitted using the allocated channel for DSME GTS. A DSME GTS can be allocated on any of the available channels in the current ChannelPage.

Details on the DSME multi-superframes structure are described in 6.11.2.

### 5.7.1.3 Slotframes

In a TSCH PAN, the concept of the superframe is replaced with a slotframe. The slotframe also contains defined periods of communications between peers that are either CSMA-CA or guaranteed, but the slotframe automatically repeats based on the participating devices' shared notion of time. Unlike the superframe, slotframes and a device's assigned timeslot(s) within the slotframe can be initially communicated by beacon, but are typically configured by a higher layer as the device joins the network. Because all devices share common time and channel information, devices hop over the entire channel space to minimize the negative effects of multipath fading and interference, and do so in a slotted way to avoid collisions, minimizing the need for retransmissions. Both of these features are desirable for operation in a harsh industrial environment.

### 5.7.1.4 TMCTP superframe

TVWS allows the optional use of a superframe structure in a TMCTP that is extended by the addition of a BOP to the active portion of the superframe. The format of the TMCTP superframe is defined by the SPC which sends an enhanced beacon containing a TMCTP Specification IE, as in 7.4.4.28. The TMCTP superframe is bounded by beacons sent by the SPC. The active portion of the TMCTP superframe is composed of a beacon, a CAP, a CFP, and a BOP. An example of a TMCTP superframe including a BOP is illustrated in Figure 5-8. The BOP is composed of one or more DBSs. A DBS is used to communicate beacons between a TMCTP-parent PAN coordinator and one of its TMCTP-child PAN coordinator(s). More information on the TMCTP superframe structure can be found in 6.2.8.



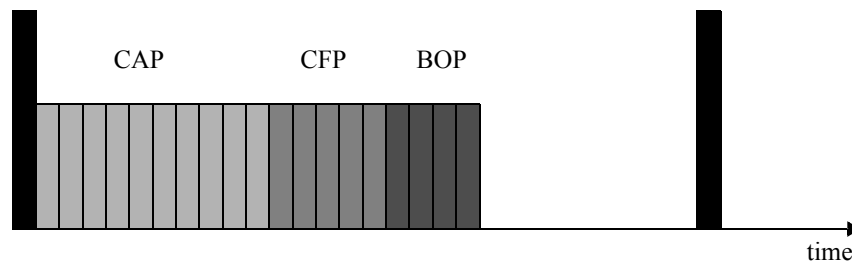


Figure 5-8—TMCTP superframe extension

## 5.7.2 Data transfer model

### 5.7.2.1 Types of data transfer models

The data transfer models used in this standard are as follows:

- Transfer to a coordinator in which a device transmits the data
- Transfer from a coordinator in which the device receives the data
- Transfer between two peer devices

If a frame that is correctly received requests an acknowledgment, the device that received the frame will respond with one of the acknowledgment (Ack) frames: immediate acknowledgment (Imm-Ack), enhanced acknowledgment (Enh-Ack) or fragment acknowledgment (Frak) depending on the state of the network and the version number of the frame that was received.

#### 5.7.2.2 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 beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate time, the device transmits its Data frame to the coordinator.

When a device wishes to transfer data in a nonbeacon-enabled PAN, it transmits its Data frame to the coordinator.

#### 5.7.2.3 Data transfer from a coordinator

When the coordinator wishes to transfer data to a device in a beacon-enabled PAN, it indicates in the beacon that the data message is pending. The device periodically listens to the beacon and, if a message is pending, transmits a Data Request command. The pending Data frame is then sent by the coordinator. Upon successful completion of the data transaction, the message is removed from the list of pending messages in the beacon.

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 requests data by transmitting a Data Request command to its coordinator. If a Data frame is pending, the coordinator transmits the Data frame. If a Data frame is not pending, the coordinator indicates this fact either in the Ack frame following the Data Request command, if an acknowledgment was requested, or in a Data frame with a zero-length payload, as defined in 6.7.3.



#### 5.7.2.4 Peer-to-peer data transfers

For peer-to-peer transfers, a device will either receive constantly or synchronize with each the other device. In the former case, the device attempts to transmit the data when it gains access to the channel. In the latter case, other measures need to be taken in order to achieve synchronization. Such measures are beyond the scope of this standard.

#### 5.7.3 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.

The MAC frames are passed to the PHY as the PHY service data unit (PSDU), which becomes the PHY payload. The PPDU is illustrated in Figure 5-9.

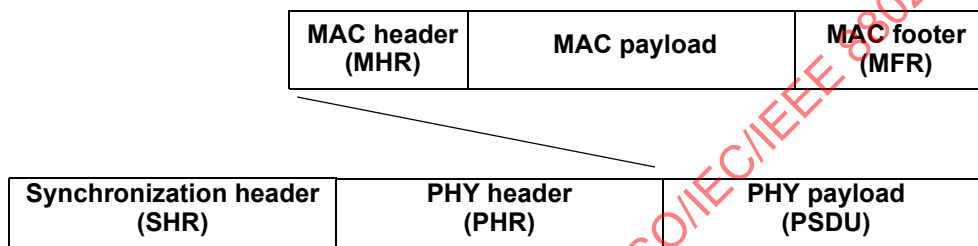


Figure 5-9—Schematic view of the PPDU

The format of the SHR and PHR is defined for each of the PHYs in their respective clause.

The MHR and MFR are defined in 7.2.

This standard makes use of IEs to transfer formatted data between layers and between devices. IEs consist of an identification, a length, and the IE content. Devices can accept or discard a particular element if the ID is known, and skip over unknown ID elements.

#### 5.7.4 Access methods

The access methods defined in this standard are as follows:

- Unslotted CSMA-CA used in nonbeacon-enabled PANs, as described in 6.2.5.1
- Slotted CSMA-CA used in beacon-enabled PANs, as described in 6.2.5.1
- TSCH CCA used in non-shared slots in a TSCH PAN, as described in 6.2.5.2
- TSCH CSMA-CA used for shared slots in a TSCH PAN, as described in 6.2.5.3
- CSMA-CA with PCA in for critical events, as described in 6.2.5.4
- LECIM ALOHA with PCA, as defined in 6.2.5.5

##### 5.7.4.1 Frame acknowledgment

A successful reception and validation of a frame is optionally confirmed with an acknowledgment, as described in 6.7.4.

The receiving device can insert additional content in an Enh-Ack frame encapsulated as IEs. If the originator does not understand the IE content of the Enh-Ack frame, it is ignored, but the transmission is considered successful.

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.

#### 5.7.4.2 Frak

The Frak is used during the fragment sequence transfer to determine which fragments have been received successfully and which fragments need to be retransmitted. A Frak includes the status of one or more fragments. The format of the Frak is given in 23.3.6.2.

#### 5.7.4.3 Data verification

A cyclic redundancy check (CRC) is used to detect errors in every PSDU, as defined in 7.2.10.

To accommodate individual fragment acknowledgments, a fragment integrity check sequence (FICS) is included with each fragment. The recipient uses the FICS and fragment number to determine which fragments of the sequence have been received correctly and which are missing. The FICS is described in 23.3.3.

#### 5.7.5 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.

In addition to the power saving features of the LR-WPAN system, the HRP UWB PHY also provides a hybrid modulation that enables very simple, noncoherent receiver architectures to further minimize power consumption and implementation complexity.

##### 5.7.5.1 Low-energy mechanisms

Low-energy mechanisms, coordinated sampled listening (CSL), as defined in 6.12.2, and receiver initiated transmission (RIT), as described in 6.12.3, are provided to further reduce energy consumption by allowing devices to communicate while maintaining low duty cycles.

CSL allows receiving devices to periodically sample the channel(s) for incoming transmissions at low duty cycles. The receiving device and the transmitting device are coordinated to reduce transmitting overhead.

RIT allows receiving devices to periodically broadcast data request frames, and transmitting devices only transmit to a receiving device upon receiving a data request frame. RIT is suitable for the following application scenarios:

- Low data traffic rate and loose latency requirement, where a few seconds of latency is allowable by application.
- Local regulations restricting the duration of continuous radio transmissions.

### 5.7.6 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 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 never have communicated before. These constraints severely limit the choice of cryptographic algorithms and protocols and 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, therefore, are 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 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 uses either 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.

Implementations should include mechanisms to prevent unauthorized access to locally stored keys.

For more detailed information on the cryptographic security mechanisms used for protected MAC frames following this standard, refer to Clause 9.

## 5.8 Concept of primitives

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 5-10, showing the service hierarchy and the relationship of the two correspondent users and their associated layer (or sublayer) peer protocol entities.

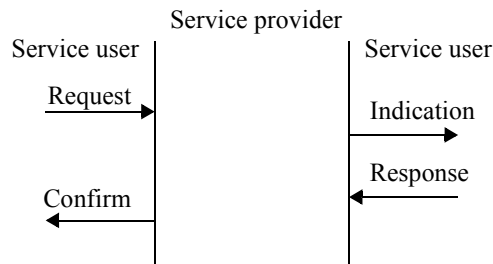


Figure 5-10—Service primitives

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 primitive can be one of four generic types:

- *Request*: The request primitive is used to request that a service is initiated.
- *Indication*: The indication primitive is used to indicate to the user an internal event.
- *Response*: The response primitive is used to complete a procedure previously invoked by an indication primitive.
- *Confirm*: The confirm primitive is used to convey the results of one or more associated previous service requests.

## 5.9 Deprecation of features

Certain features that were in prior versions of this standard have either not seen wide use or are no longer in active development. In this case, these features are deprecated and can be removed in future versions of the standard. In the version of the standard, the following items are marked as deprecated:

- Clause 14, Amplitude shift keying (ASK) PHY
- Subclause 20.2.3, Mode Switch PHR
- Subclause 20.5, Mode switch mechanism for SUN FSK

## 6. MAC functional description

### 6.1 Device types and conventions

There are two device types:

- The FFD may operate as a PAN coordinator, a coordinator, a device, RFD-TX device, RFD-RX device, or any combination thereof.
- An RFD shall not operate as a PAN coordinator nor as a coordinator but may operate as a device, RFD-TX device, or RFD-RX device.

Constants and PAN information base (PIB) attributes that are specified and maintained by the MAC sublayer or PHY layer are written in the text in italics. Constants have a general prefix of “a”, e.g., *aBaseSlotDuration*, and the MAC constants are listed in Table 8-80, while the PHY constants are listed in Table 11-1. MAC PIB attributes have a general prefix of “mac”, e.g., *macExtendedAddress*, and are listed in Table 8-81, while the security attributes are listed in Table 9-8. PHY PIB attributes have a general prefix of “phy”, e.g., *phyCurrentChannel*, and are listed in Table 11-2.

Some of the timing parameters in definition of the MAC are in units of PHY symbols. For PHYs that have multiple symbol periods, the duration to be used for the MAC parameters is defined in that PHY clause.

The broadcast short address is defined to be 0xffff. The broadcast PAN ID is defined to be 0xffff. The term broadcast address refers to either a broadcast short address or the 64-bit broadcast MAC address which is defined in IEEE Std 802-2014 to be 0xffff ffff ffff ffff.

Acknowledgment frames and timers are not shown in the message sequence charts unless they affect the behavior of that procedure.

### 6.2 Channel access

#### 6.2.1 Superframe structure

A coordinator of a PAN may use a superframe structure. A superframe is bounded by the transmission of a Beacon frame and may have an active portion and an inactive portion. The coordinator may enter a low-power (sleep) mode during the inactive portion.

For LE-applications an optional superframe structure may be used, as described in 6.2.7.

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* and the beacon interval (BI) are related as follows:

$$BI = aBaseSuperframeDuration \times 2^{macBeaconOrder}$$

for

$$0 \leq macBeaconOrder \leq 14$$

If *macBeaconOrder* = 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 *macBeaconOrder* = 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*, and the superframe duration (SD) are related as follows:

$$SD = aBaseSuperframeDuration \times 2^{macSuperframeOrder}$$

for

$$0 \leq macSuperframeOrder \leq macBeaconOrder \leq 14$$

If *macSuperframeOrder* = 15, the superframe shall not remain active after the beacon. If *macBeaconOrder* = 15, the superframe shall not exist (the value of *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  $2^{macSuperframeOrder} \times aBaseSlotDuration$  and is composed of 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.

PANs that wish to use the superframe structure (referred to as beacon-enabled PANs) 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 nonbeacon-enabled PANs) 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 Ack frames and any Data frame that quickly follows the acknowledgment of a Data Request command, as described in 6.7.3, 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 6-1. In this case, the beacon interval, *BI*, is twice as long as the active SD, and the CFP contains two GTSS.

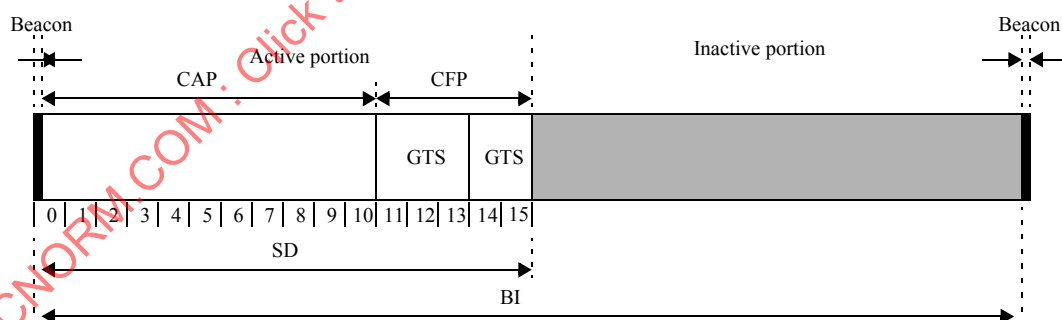


Figure 6-1—An example of the superframe structure

When operating as a TMCTP, the superframe includes the BOP, as described in 6.2.1.3, and the structure of the superframe is described in 6.2.8.

**6.2.1.1 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*, unless additional space is needed to temporarily accommodate the increase in the Beacon frame length needed to perform GTS maintenance, as described in 7.3.1.4, and shall shrink or grow dynamically to accommodate the size of the CFP.

All frames, except Ack frames and any Data frame that quickly follows the acknowledgment of a Data Request command, as described in 6.7.3, transmitted in the CAP shall use a slotted CSMA-CA mechanism to access the channel. A device transmitting within the CAP shall ensure that its transaction is complete (i.e., including the reception of a minimum length acknowledgment) one interframe spacing (IFS) period, as described in 6.2.4, before the end of the CAP. If this is not possible, the device shall defer its transmission until the CAP of the following superframe.

**6.2.1.2 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 duration of the CFP, therefore, changes depending on the total duration 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, as described in 6.2.4, before the end of its GTS.

**6.2.1.3 BOP**

When present, the BOP shall commence on the slot boundary immediately following CFP, if the CFP is present. When there is no CFP, the BOP shall commence on the slot boundary immediately following the CAP. The BOP shall complete before the end of the active portion of the superframe. The BOP duration depends on the number of DBSs, one of which is allocated to each TMCTP-child PAN coordinator. All DBSs shall be located within the BOP and occupy contiguous slots. The BOP, therefore, grows or shrinks depending on the total length of all of the combined DBSs. Multiple base slots in the BOP can be allocated to a DBS according to the length of the beacon to be sent by the TMCTP-child coordinator which will occupy the DBS.

CSMA-CA is not used for beacon transmissions in the BOP. A TMCTP-child PAN coordinator transmitting in the BOP shall ensure that its beacon transmission is complete prior to one IFS period before the end of its DBS. The IFS period is described in 6.2.4.

**6.2.2 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, as described in 8.2.12.1 and 6.3.4. The relationship between incoming and outgoing superframes is illustrated in Figure 6-2. If a device receives a Coordinator Realignment command from its coordinator indicating that the coordinator will begin using a new superframe configuration, the device shall ensure that its own beacons do not overlap with the beacons transmitted by the coordinator. If the new superframe configuration causes the incoming and outgoing superframes to overlap, the device shall stop transmitting its beacons immediately and notify the next higher layer via the MLME-SYNC-LOSS.indication primitive, as described in 8.2.13.2, with the *LossReason* parameter set to SUPERFRAME\_OVERLAP.



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.

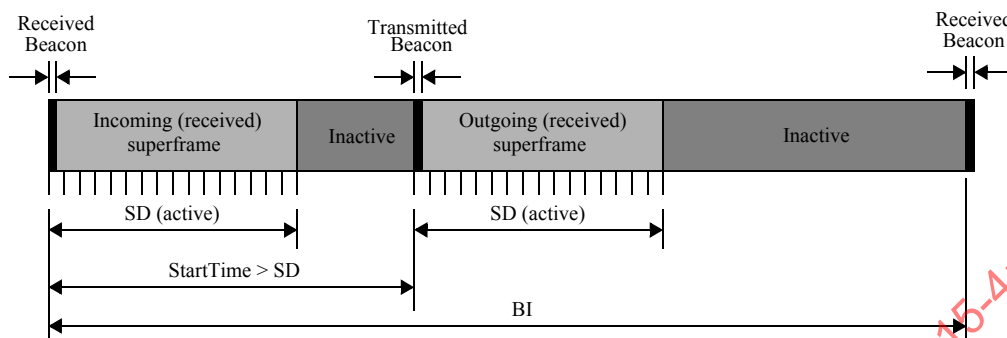


Figure 6-2—The relationship between incoming and outgoing beacons

### 6.2.3 Enhanced Beacon frame timing for MPM procedure

In a beacon-enabled PAN, a SUN device operating as a coordinator transmits an Enhanced Beacon frame containing a Coexistence Specification IE at fixed intervals, in addition to the usual periodic beacons. Figure 6-3 shows the Enhanced Beacon frame timing for beacon-enabled PANs.

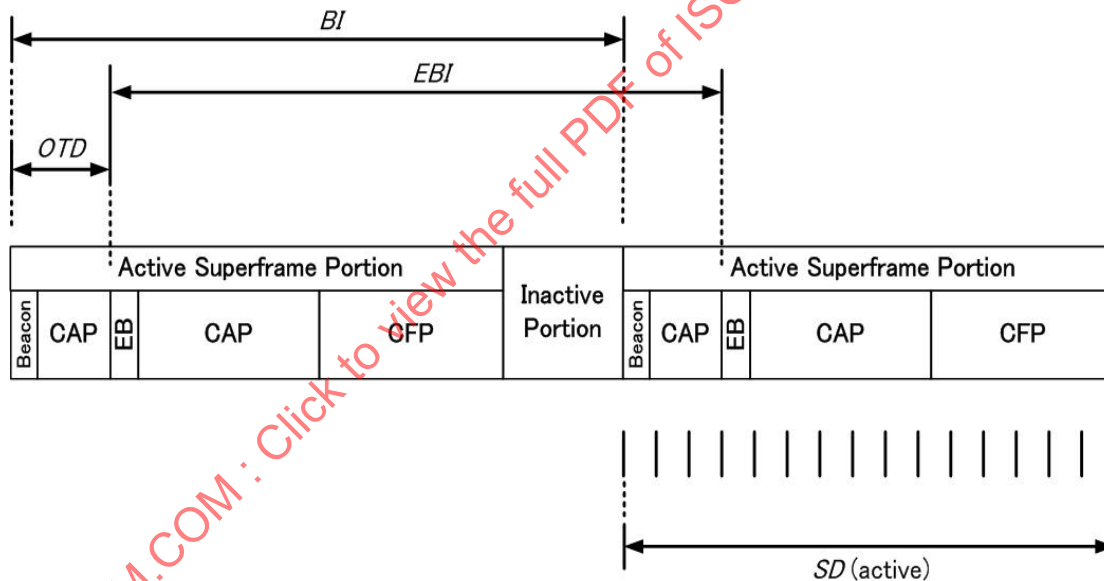


Figure 6-3—Timing information for Enhanced Beacon frames

The interval at which the coordinator shall start transmissions of its Enhanced Beacon frames is defined by *macEnhancedBeaconOrder*. The values of *macEnhancedBeaconOrder* and the enhanced beacon interval, *EBI*, are related as follows:

$$EBI = aBaseSuperframeDuration \times 2^{macEnhancedBeaconOrder}$$

The value of *macEnhancedBeaconOrder* should not be larger than the value of *macBeaconOrder*.



The time offset between the start of the periodic beacon transmission and the start of the following Enhanced Beacon frame transmission is described by *macOffsetTimeSlot*. The values of *macOffsetTimeSlot* and offset time duration, *OTD*, are related as follows:

$$OTD = aBaseSlotDuration \times macOffsetTimeSlot$$

In a nonbeacon-enabled PAN, the time offset between the starts of two Enhanced Beacon frames,  $EBI_{NB PAN}$ , is described by *macNbPanEnhancedBeaconOrder*. The resolution of time shall be *aBaseSlotDuration*. The values of *macNbPanEnhancedBeaconOrder* and  $EBI_{NB PAN}$  are related as follows:

$$EBI_{NB PAN} = aBaseSlotDuration \times macNbPanEnhancedBeaconOrder$$

## 6.2.4 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 Ack frame and the second transmission shall be at least the acknowledgment interframe spacing (AIFS). 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* shall be followed by a short interframe space (SIFS) period of a duration of at least  $\max(macSifsPeriod, aTurnaroundTime)$ . Frames (i.e., MPDUs) with lengths greater than *aMaxSifsFrameSize* shall be followed by the maximum of long interframe spacing (LIFS) of a duration of at least  $\max(macLifsPeriod, aTurnaroundTime)$ . These concepts are illustrated in Figure 6-4.

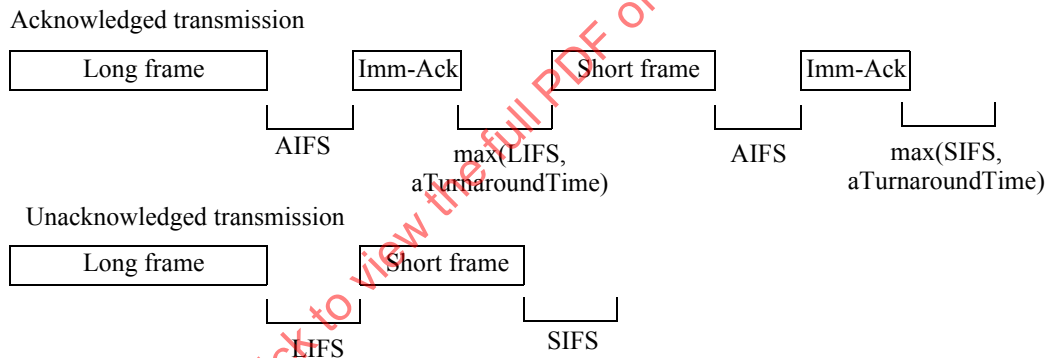


Figure 6-4—IFS usage

In order to ensure that there is sufficient time for the PHY to transition from TX to RX or RX to TX, the following relation holds:

$$AIFS \leq SIFS < LIFS$$

The CSMA-CA algorithm shall take this requirement into account for transmissions in the CAP. The timing of the transmission of an Ack frame, AIFS, is defined in 6.7.4.2.

## 6.2.5 Random access methods

### 6.2.5.1 CSMA-CA algorithm

The CSMA-CA algorithm shall be used before the transmission of data or MAC commands transmitted within the CAP, unless the frame can be quickly transmitted following the acknowledgment of a Data Request command, as defined in 6.7.3. The CSMA-CA algorithm shall not be used for the transmission of

Beacon frames in a beacon-enabled PAN, Imm-Ack frames, Enh-Ack 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  $aUnitBackoffPeriod$ .

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 back off 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. The value of  $CW_0$  shall be initialized to two before each transmission attempt and reset to  $CW_0$  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 battery life extension (BLE) field, as defined in Figure 7-7, set to zero,  $BE$  shall be initialized to the value of  $macMinBe$ . In slotted systems with the received BLE field 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.

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 6-5 illustrates the steps of the CSMA-CA algorithm. If the algorithm ends in “Success,” the MAC is allowed to begin transmission of the frame. Otherwise, the algorithm terminates with a channel access failure.

In a slotted CSMA-CA system with the BLE field 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. 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, or a single continuous CCA analysis of at least  $phyCCADuration$  for the regulatory domains that require listen before talk (LBT) such as the 920 MHz band in Japan, as described in “Applications of IEEE Std 802.15.4” [B3], 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 before evaluating whether it can proceed again.

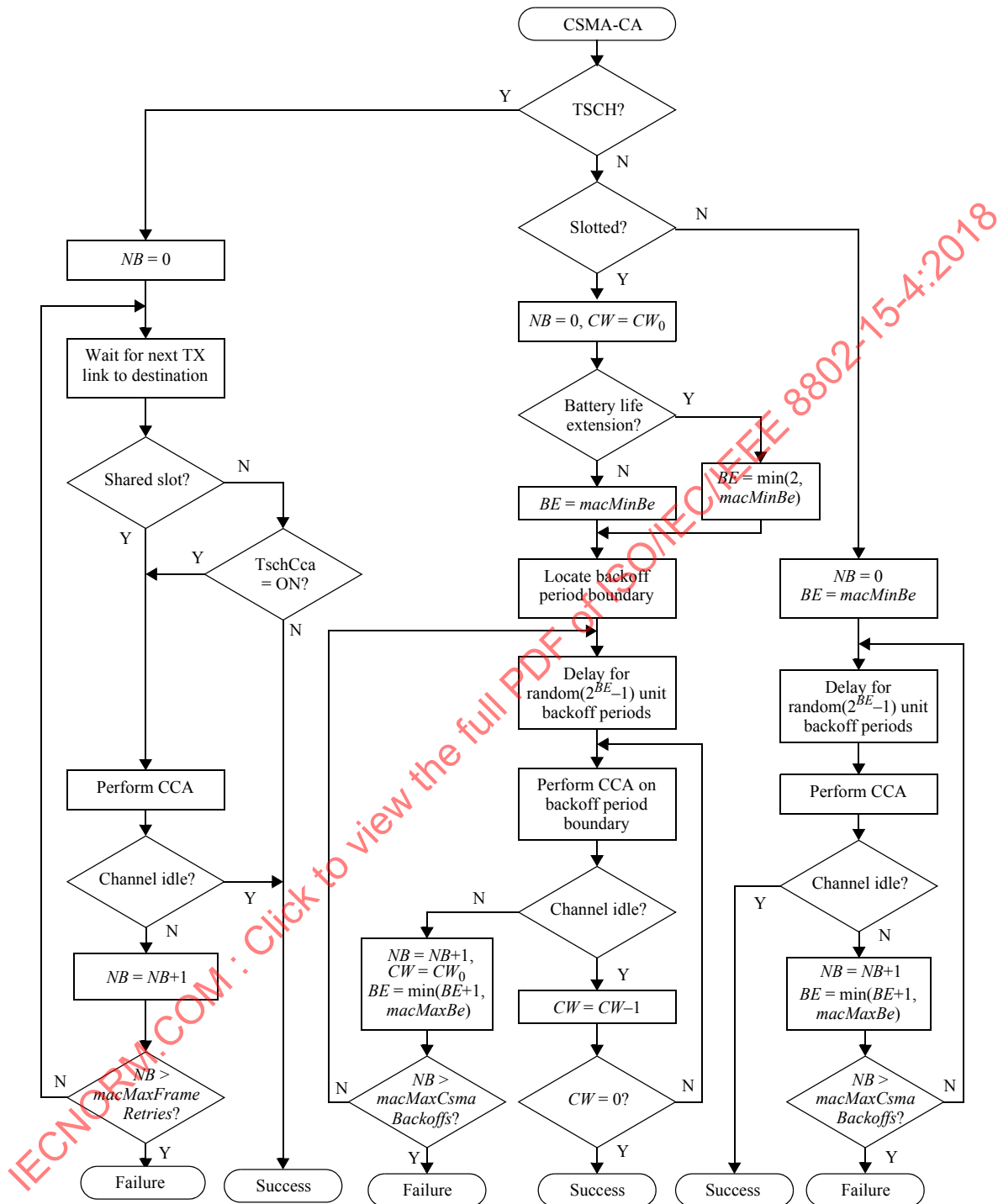


Figure 6-5—CSMA-CA algorithm

In a slotted CSMA-CA system with the BLE field 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, i.e., two CCA analyses, or a single continuous CCA analysis of at least *phyCCADuration* for the regulatory domains that require listen LBT, 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 before evaluating whether it can proceed again.

#### 6.2.5.2 TSCH CCA algorithm

When a device is operating in TSCH mode as described in 6.3.6, CCA may be used to promote coexistence with other users of the radio channel. For other devices in the same network, the start time of transmissions, *macTsTxOffset*, is closely aligned making intra-network collision avoidance using CCA ineffective. TSCH devices also use channel hopping so there is no backoff period used when CCA prevents a transmission.

As illustrated in Figure 6-5, when a TSCH device has a packet to transmit, it shall wait for a link to the destination device. If *TschCca* was set to ON in the MLME-TSCH-MODE.request primitive, the MAC requests the PHY to perform a CCA at the designated time in the timeslot, *macTsCcaOffset*, without any backoff delays.

#### 6.2.5.3 TSCH CSMA-CA retransmission algorithm

Shared links (links with the linkOptions Bitmap set to shared transmission) are intentionally assigned to more than one device for transmission. This can lead to collisions and result in a transmission failure detected by not receiving an acknowledgment. To reduce the probability of repeated collisions when the packets are retransmitted, the retransmission backoff algorithm shown in Figure 6-6 shall be implemented for shared links.

When a packet is transmitted on a shared link for which an acknowledgment is expected and none is received, the transmitting device shall invoke the TSCH retransmission algorithm. Subsequent retransmissions may be in either shared links or dedicated links as retransmission occurs in the next link to the destination. This backoff algorithm has the following properties:

The retransmission backoff wait applies only to the transmission on shared links. There is no waiting for transmission on dedicated links. The retransmission backoff is calculated in the number of shared transmission links. The backoff window increases for each consecutive failed transmission in a shared link. A successful transmission in a shared link resets the backoff window to the minimum value. The backoff window does not change when a transmission is a failure in a dedicated link.

The backoff window does not change when a transmission is successful in a dedicated link and the transmission queue is still not empty afterwards.

The backoff window is reset to the minimum value if the transmission in a dedicated link is successful and the transmit queue is then empty.

In the TSCH mode, backoff is calculated in shared links, so the CSMA-CA *aUnitBackoffPeriod* is not used.

For the *macMaxBe* and *macMinBe* values when the device is in TSCH mode refer to Table 8-83.

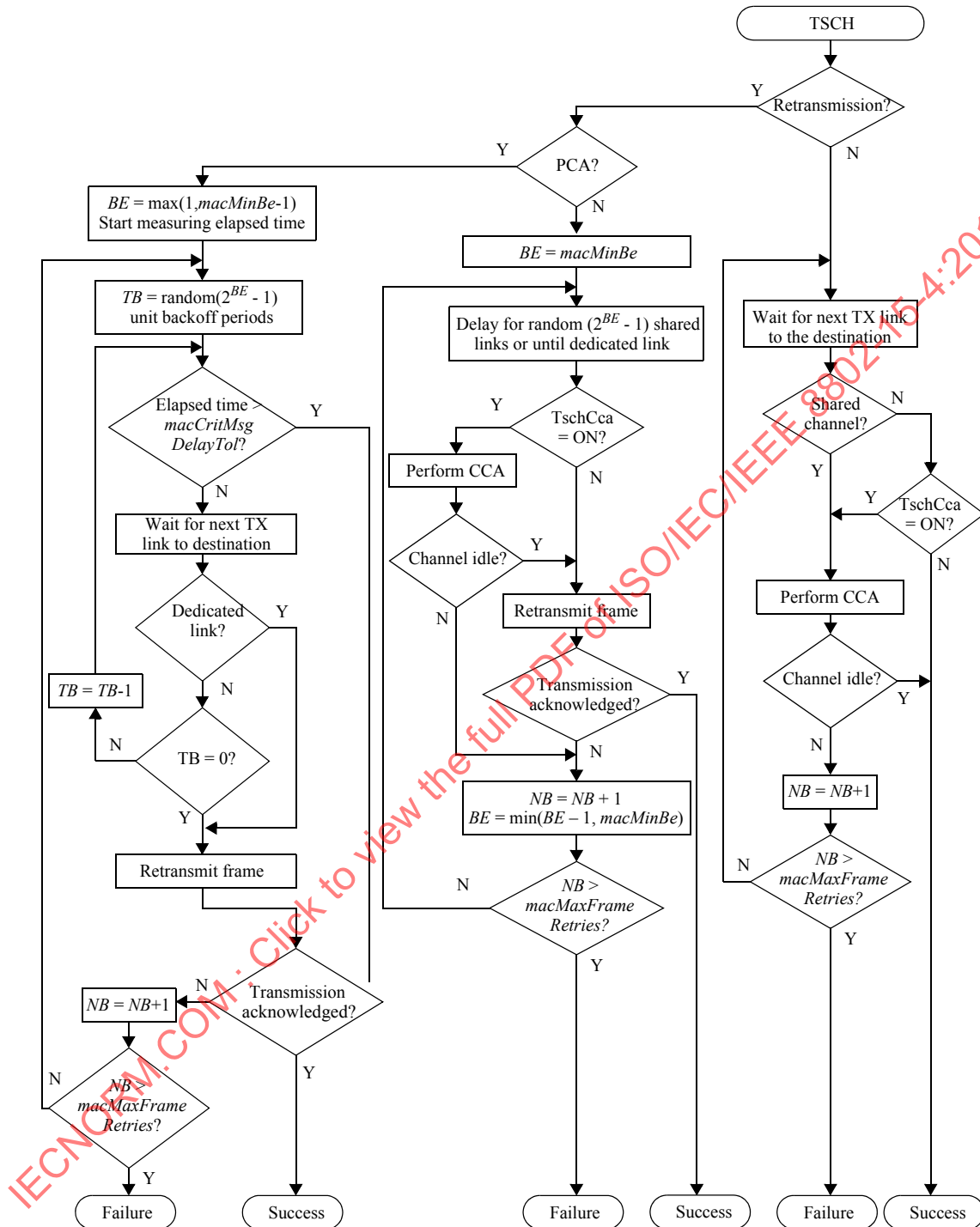


Figure 6-6—TSCH Retransmission backoff algorithm

The device shall use an exponential backoff mechanism analogous to that described in 6.2.5.1. A device upon encountering a transmission failure in a shared link shall initialize the  $BE$  to  $macMinBe$ . The MAC sublayer shall delay for a random number in the range 0 to  $(2^{BE} - 1)$  shared links (on any slotframe) before attempting a retransmission on a shared link. Retransmission on a dedicated link may occur at any time. For each successive failure on a shared link, the device should increase the backoff exponent until the backoff exponent =  $macMaxBe$ . Successful transmission on a shared link resets the backoff exponent to  $macMinBe$ .

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.

A device in a TSCH PAN may also use the PCA backoff method for critical event messages, as defined in 6.2.5.4.

#### 6.2.5.4 CSMA-CA with PCA

An MSDU or MSDU fragment in which the CriticalEventMessage parameter in the MCPS-DATA.request primitive is TRUE is referred to as a critical event message. The CSMA-CA with PCA backoff algorithm is used before the transmission of a critical event message during the CAP.

In a beacon-enabled PAN when PCA is enabled ( $macPriorityChannelAccess$  is set to TRUE), the PCA Allocation IE shall be included in Enhanced Beacon frames that are sent at every beacon interval.

In a beacon-enabled PAN, the MAC sublayer employs the slotted version of the CSMA-CA with PCA backoff algorithm for transmissions in the CAP of the superframe. Conversely, in a nonbeacon-enabled PAN or if a beacon could not be located in a beacon-enabled PAN, the MAC sublayer may transmit using the unslotted version of the CSMA-CA with PCA backoff algorithm.

CSMA-CA with PCA using a CCA Mode not equal to 4 is illustrated in Figure 6-7.

CSMA-CA with PCA using a CCA Mode equal to 4 is illustrated in Figure 6-8.

The variable  $NB$  is not used in CSMA-CA with PCA backoff algorithm. The PCA backoff algorithm is used during the transmission of a priority message: the backoff exponent  $BE$  is initialized to the value of  $(macMinBe - 1)$  or 1, whichever is larger, prior to the first transmission attempt, and  $BE$  remains constant for subsequent retransmissions. The MAC sublayer is responsible for maintaining a variable called  $TB$ , which indicates the number of remaining backoff periods since the start of the CSMA-CA with PCA backoff algorithm.  $TB$  is initialized to a random value between 0 and  $(2^{BE} - 1)$ . The PCA backoff algorithm follows a persistent CSMA mechanism, meaning that the device continues to monitor the channel and decrements  $TB$  by one any time the channel is sensed idle in a backoff period, in order to gain access to the channel in a timely manner.

In the slotted CSMA-CA with a PCA backoff algorithm, the MAC sublayer shall ensure that, after the persistent random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. If  $TB$  is greater than the remaining number of backoff periods in the CAP, the MAC sublayer pauses the  $TB$  countdown at the end of the CAP and resumes it at the start of the CAP in the next superframe. If  $TB$  is less than or equal to the remaining number of backoff periods in the CAP, the MAC sublayer applies the PCA backoff algorithm one CCA attempt further and then again evaluates whether there is sufficient time to proceed. The MAC sublayer may proceed if the remaining CSMA-CA algorithm steps, the frame transmission, and any acknowledgment can be completed before the end of the CAP. If there is not sufficient time to proceed, the MAC sublayer waits until the start of the CAP in the next superframe before continuing to apply the PCA backoff algorithm.

When operating a LECIM PHY in a nonbeacon-enabled PAN using unslotted CSMA-CA, the critical event message transmission may be initiated at any time.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

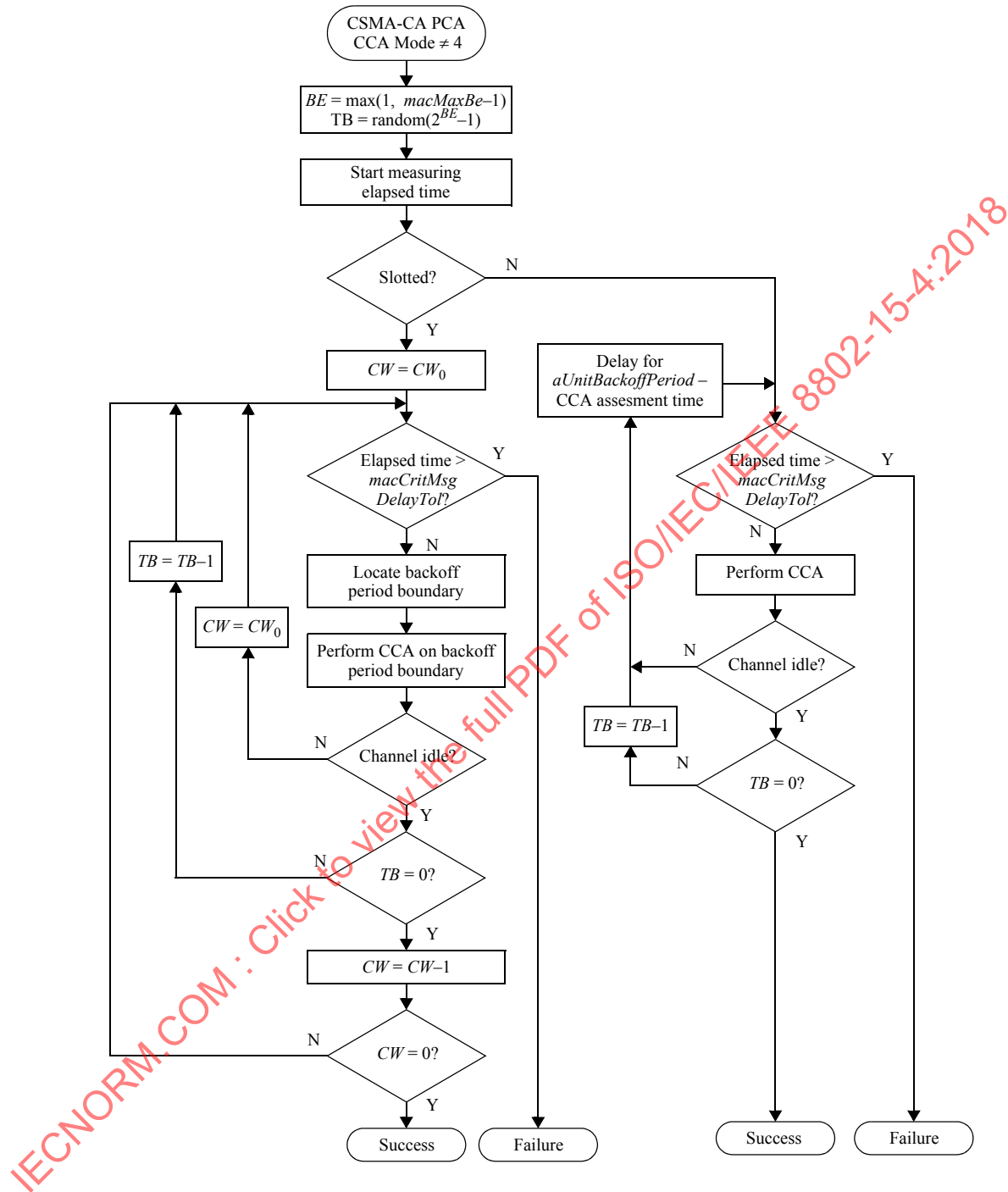


Figure 6-7—PCA CSMA-CA, non-Aloha

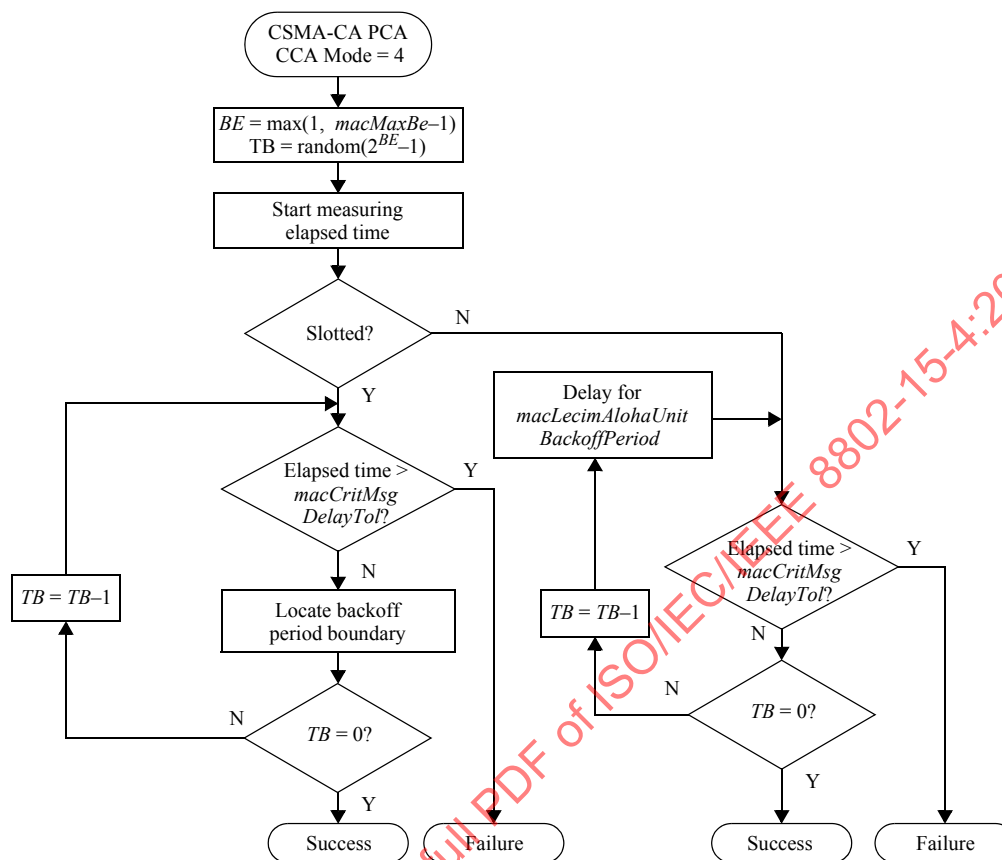


Figure 6-8—PCA CSMA-CA, Aloha

In a beacon-enabled PAN, the length of a PCA allocation shall be at least 880 symbol periods. When *macPriorityChannelAccess* is TRUE, the minimum number of PCA allocations in a superframe is defined by the MAC personal area network information base (PIB) attributes *macPcaAllocationSuperRate*, *macPcaAllocationRate*, and *macCritMsgDelayTol*. The relations of the parameters are illustrated in Table 6-1.

Table 6-1—PCA MAC PIB attribute relations

| Value of <i>macPcaAllocationSuperRate</i> | SD                                                  | <i>macPcaAllocationRate</i>                                         |
|-------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------|
| FALSE                                     | $SD \leq macCritMsgDelayTol/3$                      | Maximum value<br>$\lfloor macCritMsgDelayTol/(3 \times SD) \rfloor$ |
| TRUE                                      | $macCritMsgDelayTol/3 < SD \leq macCritMsgDelayTol$ | Minimum value<br>1                                                  |
| TRUE                                      | $macCritMsgDelayTol < SD$                           | Minimum value<br>$\lceil SD/macCritMsgDelayTol \rceil$              |



In Table 6-1,  $\lfloor \cdot \rfloor$  indicates the closest integer less than or equal to its argument and  $\lceil \cdot \rceil$  indicates the closest integer larger than or equal to its argument. When *macPcaAllocationSuperRate* is FALSE, *macPcaAllocationRate* is interpreted as a substrate, and it indicates the maximum number of consecutive superframes for which only one PCA allocation is required. In this case, the PCA allocations need only occur within the superframes having *macBsn* that are integer divisible by the *macPcaAllocationRate* value. When  $SD \leq \text{macCritMsgDelayTol}/3$ , *macPcaAllocationSuperRate* shall be set to FALSE; otherwise, it shall be set to TRUE. When *macPcaAllocationSuperRate* is TRUE, *macPcaAllocationRate* indicates the minimum number of PCA allocations required per superframe.

If there are multiple PCA allocations per superframe, the first allocation occurs at the start of the CAP. The remaining PCA allocations are distributed throughout the superframe, but no PCA allocation shall occur outside the CAP.

When a critical event message transmission is initiated within the CAP during a time that is not a PCA allocation, CSMA-CA, as defined in 6.2.5, with the previously described PCA backoff algorithm may be used.

If DSME is utilized with *macCapReduction* set to TRUE and the multi-superframe duration (MD) is longer than *macCritMsgDelayTol*, then *macPriorityChannelAccess* shall be set to FALSE.

When *macPriorityChannelAccess* is TRUE, a PCA allocation shall not occur if the CAP length duration is less than *aMinCapLength* plus the time required for a single PCA allocation.

#### 6.2.5.5 LECIM ALOHA PCA

When PCA is in use with CCA Mode 4 (ALOHA), PCA is achieved using a modified version of the PCA backoff algorithm described in 6.2.5.4. Instead of one backoff period being equal to *aUnitBackoffPeriod*, a backoff period is defined as *macLecimAlohaUnitBackoffPeriod*.

When MPDU fragmentation is in use, the value of *macLecimAlohaUnitBackoffPeriod* should be long enough to accommodate the transmission of a single MPDU fragment with the associated IFS period and a Frak frame. When MPDU fragmentation is not in use, the value of *macLecimAlohaUnitBackoffPeriod* should be long enough to accommodate the transmission of a maximum-sized MPDU with the associated IFS and an Ack frame.

When *macPriorityChannelAccess* is set as TRUE, each PCA allocation shall be at least four consecutive *macLecimAlohaUnitBackoffPeriod* in duration. The first PCA allocation shall begin at the start of the CAP. The number of PCA allocations per superframe is described by Table 6-1.

A PCA allocation cannot occur if the CAP length duration is less than *aMinCapLength* plus the time required for a single PCA.

### 6.2.6 TSCH slotframe structure

#### 6.2.6.1 General

A slotframe is a collection of timeslots repeating in time. Each timeslot allows enough time for a pair of devices to exchange a frame and an acknowledgment. It is possible, although usually undesirable, to define a timeslot that is not long enough for a pair of devices to exchange a maximum length frame and an Ack frame. The number of timeslots in a given slotframe (slotframe size) determines how often each timeslot repeats, thus setting a communication schedule for nodes that use the timeslots. When a slotframe is created, it is associated with a slotframe handle (*macSlotframeHandle*) for identification. Each slotframe repeats on a cycle dependent on its length. Each timeslot is an opportunity for a device to send or receive a single frame,

and optionally receive or transmit an acknowledgment to that frame. Slotframes and timeslots are configured by a higher layer.

### 6.2.6.2 Absolute slot number (ASN)

The total number of timeslots that has elapsed since the start of the network or an arbitrary start time determined by the PAN coordinator is called the Absolute Slot Number (ASN). It increments globally in the network every *macTsTimeslotLength*, as defined in Table 8-86. It may be beacons by devices already in a TSCH PAN, allowing new devices to synchronize. It is used globally by devices in a TSCH PAN as the frame counter (thus allowing for time-dependent security) and is used to compute the channel for any given pairwise communication as described in 6.2.6.3.

### 6.2.6.3 Links

Figure 6-9 illustrates an example of nodes communicating in a sample three-timeslot slotframe. Nodes A and B communicate during timeslot 0, nodes B and C communicate during timeslot 1, and timeslot 2 is not being used. Every three timeslots, the schedule repeats, but note that ASN increments continuously. The pairwise assignment of a directed communication between devices for a given *macSlotFrameHandle*, as defined in 8.4.2.2.1, in a given timeslot on a given *macChannelOffset* is a link.

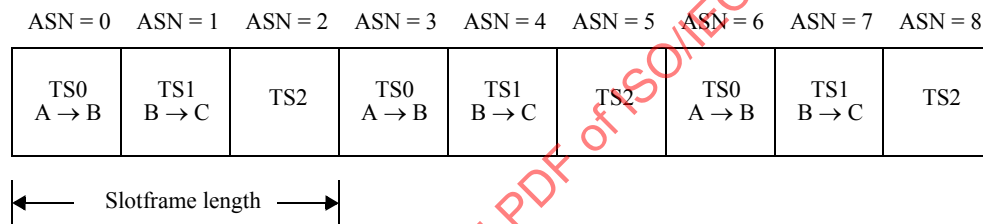


Figure 6-9—Example of a three time-slot slotframe

The physical channel, *CH*, in a link is calculated as follows:

$$CH = \text{macHoppingSequenceList}[(\text{macAsn} + \text{macChannelOffset}) \% \text{macHoppingSequenceLength}]$$

where  $a \% b$  indicates  $a$  modulo  $b$ .

Use of a *macChannelOffset* allows for different channels to be used at a given *macAsn* for a given *macHoppingSequenceList*. There are *macNumberOfChannels* channel offsets that will result in a unique channel for that combination of *macAsn* and *macHoppingSequenceList*.

### 6.2.6.4 Multiple slotframes

A given network using timeslot-based access may contain several concurrent slotframes of different sizes. Multiple slotframes may be used to define a different communication schedule for various groups of nodes or to run the entire network at different duty cycles by giving some devices many active timeslots in a slotframe, and others few or none.

A network device may participate in one or more slotframes simultaneously, and not all devices need to participate in all slotframes. By configuring a network device to participate in multiple overlapping slotframes of different sizes, it is possible to establish different communication schedules and connectivity matrices that all work at the same time.

Slotframes can be added, removed, and modified while the network is running. Even though this is the case, all slotframes are aligned to timeslot boundaries, and timeslot 0 of the first repetition of every slotframe is projected back to *macAsn* = 0, which is determined by the PAN coordinator (or other network device that starts the network). Because of this, timeslots in different slotframes are always aligned, even though the beginning and end of a particular repetition of that slotframe may not be as illustrated in Figure 6-10. When, for any given timeslot, a device has links in multiple slotframes, transmissions take precedence over receives, and lower *macSlotframeHandle* slotframes takes precedence over higher *macSlotframeHandle* slotframes.

|                        | ASN = 0 | ASN = 1 | ASN = 2 | ASN = 3 | ASN = 4 | ASN = 5 | ASN = 6 | ASN = 7 |
|------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| Slotframe 1<br>5 slots | TS0     | TS1     | TS2     | TS3     | TS4     | TS0     | TS1     | TS2     |
| Slotframe 2<br>3 slots | TS0     | TS1     | TS2     | TS0     | TS1     | TS2     | TS0     | TS1     |

Figure 6-10—Multiple slotframes in the network

### 6.2.7 LE functional description

#### 6.2.7.1 LE contention access period (LE CAP)

When *macCslPeriod* is nonzero, CSL is deployed in CAP. CSL behavior is defined in 6.12.2. The *macRitPeriod* shall be set to zero in a beacon-enabled PAN.

When *macLowEnergySuperframeSupported* is TRUE, the transaction shall be ensured to be completed one IFS period before the end of the inactive period. If a device senses a frame in the CAP that does not end within the CAP when *macLowEnergySuperframeSupported* is TRUE, the device may continue receiving the frame until it ends before the end of the inactive period. When *macLowEnergySuperframeSupported* is TRUE, the coordinator shall not allocate GTSS in order to avoid the interference from the frames exceeding the CAP and going into the CFP. When *macLowEnergySuperframeSupported* is TRUE, the coordinator shall notify the devices that already associated or intend to associate the condition of *macLowEnergySuperframeSupported* in the Beacon frames.

#### 6.2.7.2 LE superframe structure

If *macLowEnergySuperframeSupported* is TRUE and *macLowEnergySuperframeSyncInterval* is not zero, the coordinator shall transmit Beacon frames not in every beacon interval, but once in every beacon interval time *macLowEnergySuperframeSyncInterval*, except when requested to do so. If *macLowEnergySuperframeSupported* is TRUE and *macLowEnergySuperframeSyncInterval* is zero, the coordinator shall transmit Beacon frames only when it is requested to do so.

#### 6.2.7.3 LE-incoming and outgoing superframe timing

If a device supports *macLowEnergySuperframeSupported*, the beacon order and superframe order may be equal for all superframes on a PAN.

#### 6.2.7.4 LE scan

When *macCslPeriod* is nonzero, CSL is deployed in channel scans. When *macCslMaxPeriod* is nonzero, each coordinator broadcasts Beacon frames with wake-up frame sequence. This allows devices to perform channel scans with low duty cycles.

### 6.2.8 Superframe use for TMCTP operation

The TMCTP superframe is an extension of the basic superframe defined in 6.2.1. The active portion of the TMCTP superframe is composed of four parts, which is illustrated in Figure 5-8.

- The Beacon frame, as described in 5.2.2.1, which is used to set the timing allocations and to communicate management information for the PAN.
- The CAP, as described in 6.2.1.1, which is used to communicate commands and/or data.
- The CFP, as described in 6.2.1.2, which is composed of GTSS. No transmissions within the CFP shall use a CSMA-CA mechanism to access the channel.
- The BOP, as described in 6.2.1.3, which is composed of one or more DBSs. A DBS is used to communicate beacons between a TMCTP-parent PAN coordinator (including the SPC) and one of its TMCTP-child PAN coordinator(s) in a TMCTP.

The SD and BI of the TMCTP superframe are the same as described in 6.2.1. The MAC PIB attribute *macTmctpExtendedOrder* describes the extended length of the active portion of the superframe. The value of *macTmctpExtendedOrder* and the extended duration, ED, are related as follows:

$$\begin{aligned}
 ED &= aBaseSuperframeDuration \times 2^{macTmctpExtendedOrder} \\
 &= aBaseSlotDuration \times aNumSuperframeSlots \times 2^{macTmctpExtendedOrder}
 \end{aligned}$$

for

$$0 \leq macTmctpExtendedOrder \leq (macBeaconOrder - macSuperframeOrder) \leq macBeaconOrder \leq 14$$

The ED of each TMCTP superframe shall be divided into  $aNumSuperframeSlots \times 2^{macTmctpExtendedOrder}$  equally spaced slots of duration *aBaseSlotDuration* in a BOP. The BOP consists of multiple DBSs. Each DBS is composed of one or more base slots, which are *aBaseSlotDuration* in length.

The total duration of the active portion of each TMCTP superframe consists of the SD and the ED.

An example of a TMCTP superframe structure is shown in Figure 6-11, according to various values of the *macBeaconOrder* (BO), the *macSuperframeOrder* (SO), and the *macTmctpExtendedOrder* (EO).

### 6.2.9 Rail communications and control network (RCCN) superframe structure

Support for the RCCN superframe structure is optional for an RCC device. For typical usage, refer to “Applications of IEEE Std 802.15.4” [B3].

The RCCN superframe structure is shown in Figure 6-12.

The duration of an RCCN superframe slot is as follows:

$$(k+1) \times aRccnBaseSlotDuration$$

where *k* is the value of the Slot Size Multiplier field of the RCCN descriptor IE, as described in 7.4.2.14.

The superframe begins with an enhanced beacon containing the RCCN descriptor IE. The management downlink slots are used by the RCCN PAN coordinator to send frames to RCCN endpoints, while the management uplink slots are used by RCCN endpoints to send frames to the RCCN PAN coordinator. Channel access in the management uplink slots is performed using slotted CSMA-CA. Following the uplink slots is the CAP, in which communication between any devices in the RCCN may occur; channel access is gained using slotted CSMA-CA, optionally with PCA, as described in 6.2.5.4. Following the CAP is a CFP containing one or more GTSS allocated by the RCCN PAN coordinator.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

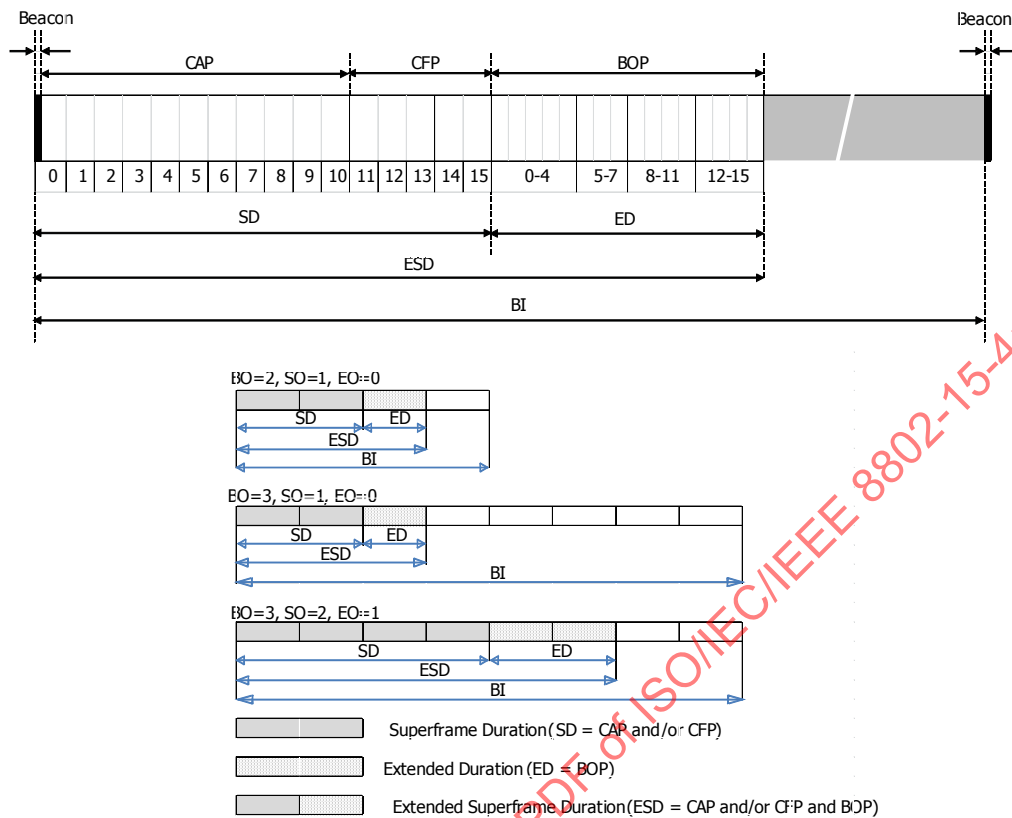


Figure 6-11—An example of the TMCTP superframe structure

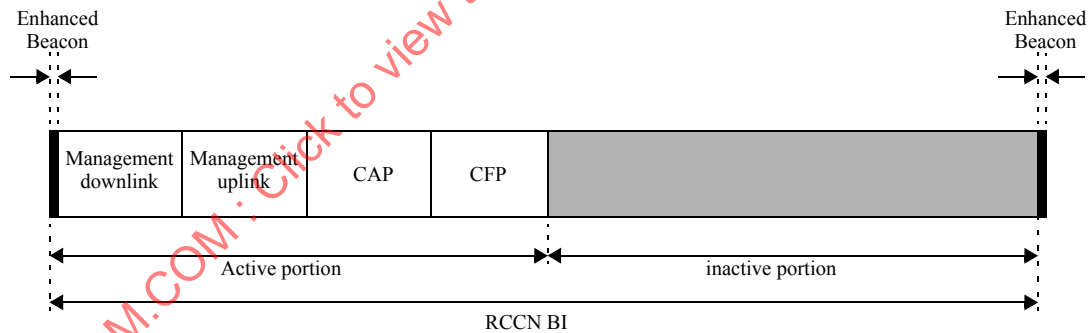


Figure 6-12—RCCN superframe structure

### 6.2.10 Channel hopping

Networks may support channel hopping using any multichannel PHY. Devices may hop in a slotted mode (e.g., TSCH or DSME) or in an unslotted mode.

The slotted mode uses network coordination within a superframe, for DSME, or slotframe, for TSCH, via a shared hop sequence to which devices participating in the network synchronize. Because the hop dwell time is usually one slot time, the network synchronization covers the needs of hopping and time slots. This mechanism allows a node to communicate with one or many other nodes. The unslotted mode often does not

use network synchronization for hopping and time slots. This mechanism allows a node to communicate with one or many other nodes. The unslotted mode often does not use network synchronization for hopping, e.g., networks have many sequences without any global clock. For neighbor devices to communicate, a node needs to know the other nodes's hop sequence and timing. Devices may advertise their hop sequences and timing via Channel Hopping IE and the Hopping Timing IE in Enhanced Beacon frames.

A specific hopping sequence is specified by its Hopping Sequence ID (*macHoppingSequenceId*) with ID = 0 denoting the default hop sequence for a particular PHY (or PHY configuration if the PHY supports more than one channel list). The hopping sequence characteristics are defined in Table 8-87.

This default hopping sequence is a pseudo-randomly shuffled set of all of the channels available to the PHY. The mechanism to generate the default sequence is defined as follows:

- SHUFFLE is a *macHoppingSequenceLength*-sized array. The contents of this array are equivalent to the first *macHoppingSequenceLength* outputs of a 9-bit linear feedback shift register (LFSR) with polynomial  $x^9 + x^5 + 1$  and a starting seed of 255. Each LFSR output is modulo *macHoppingSequenceLength*, so that each entry of SHUFFLE is between 0 and (*macHoppingSequenceLength* - 1), inclusive.
- CHANNELS is a *macHoppingSequenceLength*-sized array that is initially populated with the monotonically increasing set of channels available to the PHY.
- CHANNELS is shuffled as per Figure 6-13. Elements may wind up being swapped multiple times in this process.
- The default sequence (i.e., *macHoppingSequenceList* for *macHoppingSequenceId* = 0) is equivalent to the shuffled CHANNELS array.

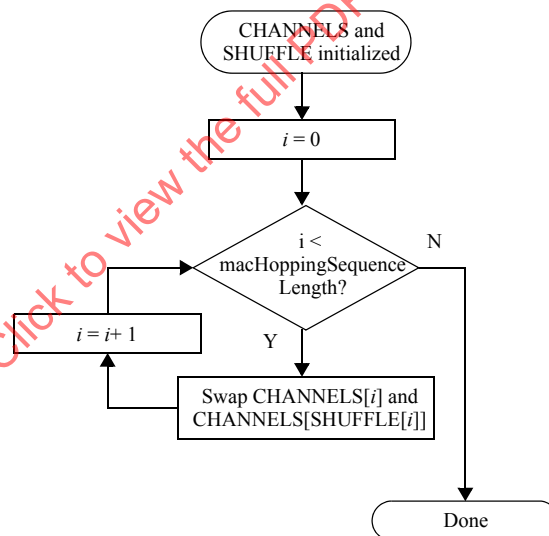


Figure 6-13—CHANNELS shuffle algorithm

The use of other sequences (*macHoppingSequenceId* > 0) may be defined by a particular channel hopping system. The *macHoppingSequenceList* for a *macHoppingSequenceId* > 0 may be longer or shorter than the default sequence and may be specified algorithmically or set as a predefined channel list. Two hopping devices cannot communicate unless their PHYs support the same number of channels, and they either use the default hopping sequence or agree upon the hopping sequence being used, either through carrying this information in an Enhanced Beacon frame, or through pre-configuration.

For cases where *macHoppingSequenceLength* is greater than the number of channels available to the PHY, this implies that some channels will appear multiple times in the array. For cases where *macHoppingSequenceLength* is less than or equal to the number of channels available to the PHY, some channels available to the PHY may be excluded from the array. The selection of channels (the subset of available PHY channels, and which, if any, channels are used multiple times in the hopping sequence) is implementation-specific.

In general, a device can calculate the channel as follows:

$$CH = \text{macHoppingSequenceList} [\text{COUNTER} \% \text{macHoppingSequenceLength}]$$

where the COUNTER is some appropriate shared counter for a pair of devices communicating using that mode, and % indicates modular division.

## 6.3 Starting and maintaining PANs

### 6.3.1 Scanning through channels

All devices, except for the RFD-RX and RFD-TX devices, shall be capable of performing passive and orphan scans across a specified set of channels. In addition, an FFD shall be able to perform energy detection (ED) and active scans. Optionally an FFD may be able to perform an enhanced active scan.

A device is instructed to begin a channel scan through the MLME-SCAN.request primitive, as described in 8.2.11.1. 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. For HRP UWB and CSS PHYs, each preamble code appropriate to the specified channel is scanned. Upon the conclusion of the scan, the coordinator device in a beacon-enabled PAN shall recommence beacon transmissions. The results of the scan shall be returned via the MLME-SCAN.confirm primitive as described in 8.2.11.2.

#### 6.3.1.1 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 channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an ED scan. For each channel, the MLME shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and then repeatedly perform an ED measurement for  $[aBaseSuperframeDuration \times (2^n + 1)]$ , where  $n$  is the value of the ScanDuration parameter in the MLME-SCAN.request primitive. 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 at least one channel ED measurement.

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 channels.

#### 6.3.1.2 Active and passive channel scan

An active or passive channel scan allows a device to locate any coordinator transmitting Beacon frames within its radio communications range. An active scan uses the Beacon Request command to extract the beacon from a coordinator. In a passive scan, the Beacon Request command is not transmitted. A message sequence chart for active scan is illustrated in Figure 6-14 and for passive scan in Figure 6-15.



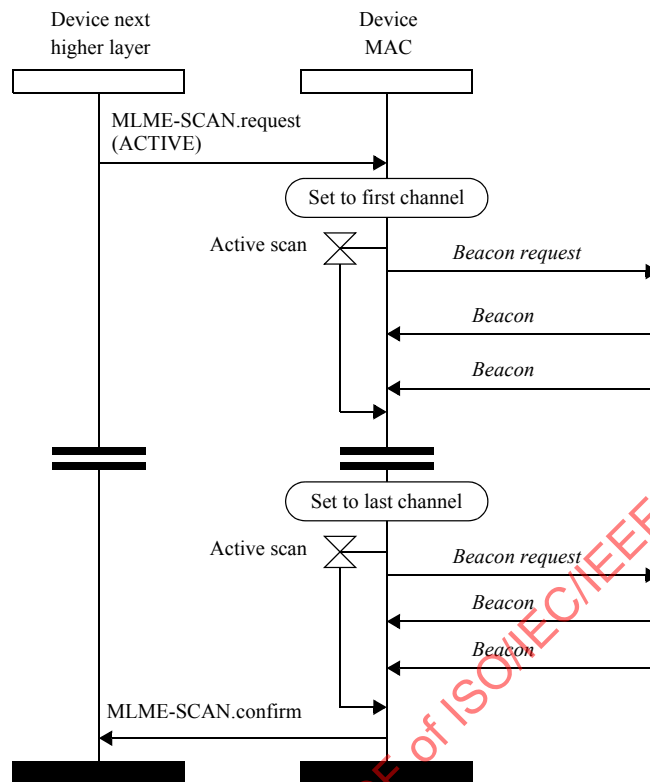


Figure 6-14—Active scan message sequence chart

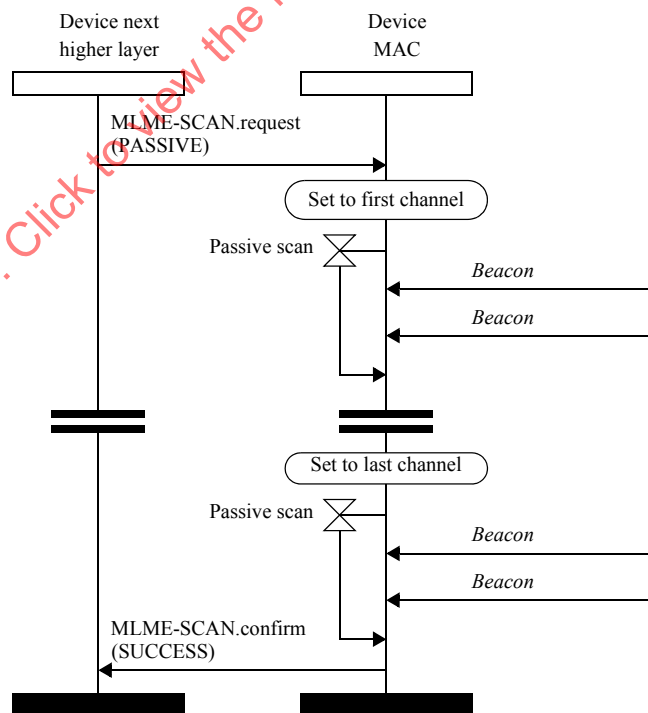


Figure 6-15—Passive scan message sequence chart



During an active or 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 set of pending addresses, the scanning device shall not attempt to extract the pending data.

A passive scan or an active scan or an enhanced active scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate passive scan, an active scan or enhanced active scan respectively. For each channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly. For an active scan, the device shall send a Beacon Request command, as described in 7.5.8. For HRP UWB and CSS PHYs, the scan process shall be repeated for each mandatory preamble code, setting the *phyCurrentCode* appropriately. For an enhanced active scan the device shall send an Enhanced Beacon Request command as described in 7.5.9. Upon successful transmission of the Beacon Request command for an active scan or after switching to the channel for a passive scan, the device shall enable its receiver for at most  $[aBaseSuperframeDuration \times (2^n + 1)]$ , where  $n$  is the value of the ScanDuration parameter. During this time, the device shall reject all frames that are not Beacon frames and record the information contained in all unique beacons in a PAN descriptor structure, as described in Table 8-12 including the channel information and, if required, the preamble code.

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, optional for RFD-RX and RFD-TX devices, shall be able to store at least one PAN descriptor. A Beacon frame shall be assumed to be unique if it contains both a PAN ID 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 as described in 8.2.5.1. A received Beacon frame containing a nonzero-length payload shall also cause the PAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY.indication primitive. Once the scan with *macAutoRequest* set to FALSE is complete, the MLME-SCAN.confirm shall be issued to the next higher layer with a null PANDescriptorList.

For HRP UWB and CSS PHYs, the beacon request is repeated for each preamble code.

If a protected Beacon frame is received, i.e., the Security Enabled field is set to one, the device shall attempt to unsecure the Beacon frame using the unsecuring process described in 9.2.3.

The security-related elements of the PAN descriptor corresponding to the beacon, as defined in Table 8-12, shall be set to the corresponding parameters returned by the unsecuring process. The SecurityStatus 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 frame to be unsecured 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 6.3.1.2. 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.

If a coordinator capable of responding to the Enhanced Beacon Request command, as described in 7.5.9, and capable of filtering receives an Enhanced Beacon Request command, it shall perform the filtering as indicated in the Enhanced Beacon Request command. If the filter conditions are satisfied, it shall transmit the appropriate Enhanced Beacon frame as indicated in Table 6-2.

**Table 6-2—Channel access for response to an Enhanced Beacon Request command**

| Mode of operation | Access method     | When to respond                              |
|-------------------|-------------------|----------------------------------------------|
| Beacon PAN        | Slotted CSMA-CA   | Next available CAP                           |
| Nonbeacon PAN     | Unslotted CSMA-CA | As soon as possible                          |
| DSME beacon       | Slotted CSMA-CA   | At the beginning of the next beacon interval |
| DSME non-beacon   | Unslotted CSMA-CA | As soon as possible                          |
| TSCH              | Slotted CSMA-CA   | Next available time slot on same channel     |

The messages exchanged when using an enhanced active scan with an Enhanced Beacon Request are shown in Figure 6-16.

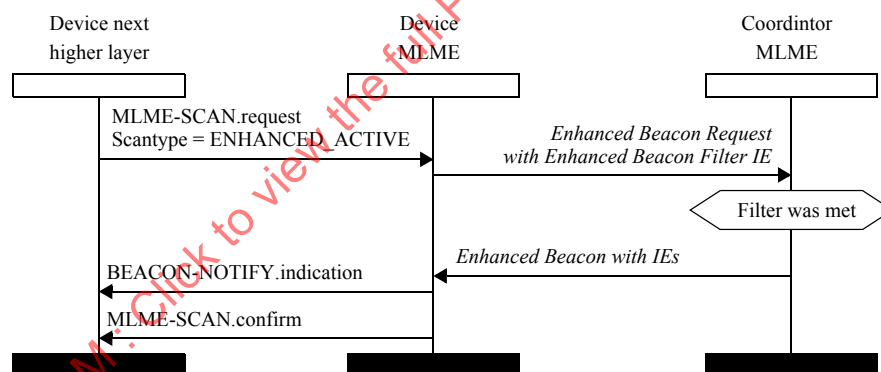


**Figure 6-16—Enhanced Active scan with Enhanced Beacon request**

### 6.3.1.3 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 commands.

An orphan scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an orphan scan. For each channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and for HRP UWB and CSS PHYs, setting the preamble code *phyCurrentCode* appropriately, and then send an Orphan Notification command, as described in 7.5.7. Upon successful transmission of the Orphan Notification command, the device shall enable its receiver for at most *macResponseWaitTime*. If the device successfully receives a

Coordinator Realignment command, as described in 7.5.10, within this time, the device shall terminate the scan. For the HRP UWB and CSS PHYs, if the Coordinator Realignment command is not received, the process shall be repeated for each preamble code until a Coordinator Realignment command is received or all preamble codes for the PHY have been used.

The orphan scan shall terminate when the device receives a Coordinator Realignment command or the specified set of channels has been scanned.

An example message sequence chart for orphan scan and realignment is shown in Figure 6-17.

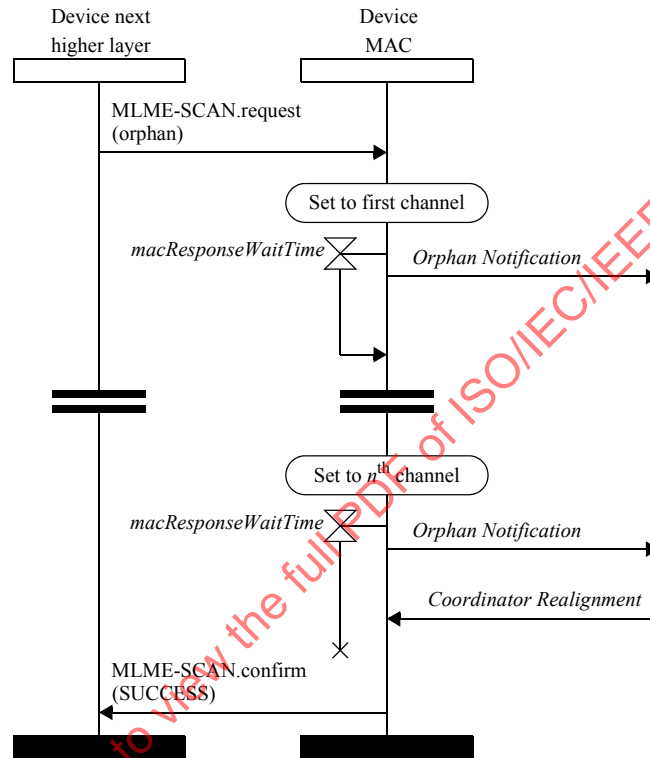


**Figure 6-17—Orphaned device realignment message sequence chart**

If a coordinator receives the Orphan Notification command, the MLME shall send the MLME-ORPHAN.indication primitive, as described in 8.2.8.1, 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 send a Coordinator Realignment command to the orphaned device using the MLME-ORPHAN.response primitive, as described in 8.2.8.2, with the AssociatedMember parameter set to TRUE and the ShortAddress parameter set to the corresponding short address allocated to the orphaned device. The process of searching for the device and sending the Coordinator Realignment command shall occur within *macResponseWaitTime*. The Coordinator Realignment command shall contain its current PAN ID, *macPanId*, its current channel and channel page, and the short address of the orphaned device. If the next higher layer of the coordinator finds no record of the device, it should send the MLME-ORPHAN.response primitive to the MLME with the AssociatedMember parameter set to FALSE.

Figure 6-18 illustrates the sequence of messages necessary for a coordinator to issue a notification of an orphaned device.

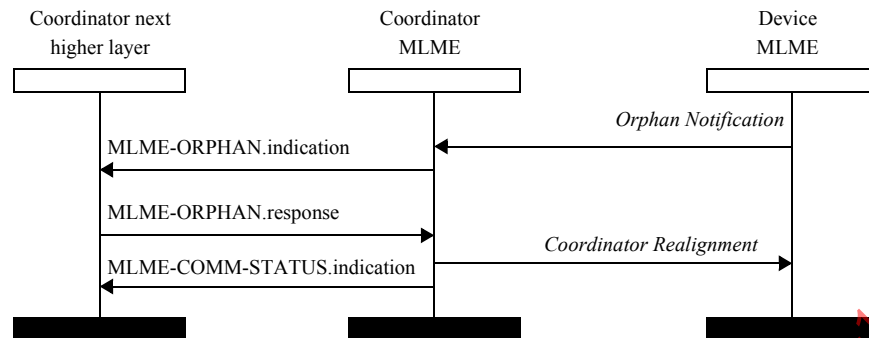


Figure 6-18—Message sequence chart for orphan notification

#### 6.3.1.4 RIT passive channel scan

The RIT passive scan is essentially the passive scan as described 6.3.1.2 but rather than discarding all frames that are not Beacon frames, the RIT passive scan discards all frames that are not RIT Data Request command frames.

A RIT passive scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the Valid Range of the ScanType parameter set to indicate RIT\_PASSIVE. For each channel, the device shall set *phyCurrentChannel* and *phyCurrentPage* as per the MLME-SCAN.request primitive. After switching to the channel for an RIT passive scan, the device shall enable its receiver for at most  $[macRitPeriod \times n]$ , where  $n$  is the value of ScanDuration parameter. During this time, the device shall reject all frames that are not RIT Data Request command frames and record the information contained in all unique RIT Data Request command frames in a PAN descriptor structure, as described in Table 8-12, including the channel information. A RIT Data Request command frame shall be assumed to be unique if it contains both a PAN ID and a source address that has not been seen before during the scan of the current channel.

If an RIT Data Request command 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. When the scan is complete, the list shall be sent to the next higher layer in the PANDescriptorList parameter of the MLME-SCAN.confirm primitive.

If an RIT Data Request command 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, as described in 8.2.5.1. A received RIT Data Request Command frame containing a nonzero-length payload shall also cause the PAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY.indication primitive. Once the scan with *macAutoRequest* set to FALSE is complete, the MLME-SCAN.confirm shall be issued to the next higher layer with a null PANDescriptorList.

The RIT passive scan procedure is illustrated in Figure 6-19.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

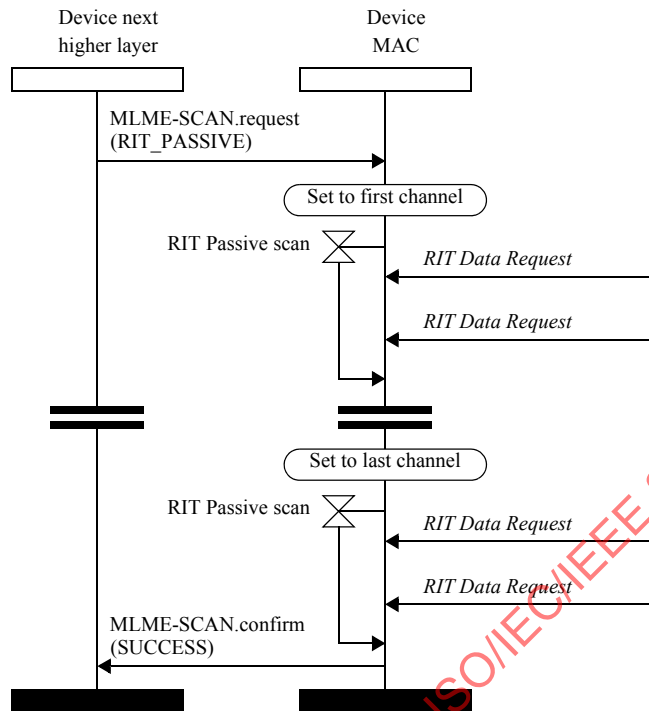


Figure 6-19—RIT Passive scan message sequence chart

### 6.3.2 PAN ID conflict resolution

In some instances a situation could occur in which two PANs exist in the same radio communications range with the same PAN ID. If this conflict happens, the PAN coordinator and its devices shall perform the PAN ID conflict resolution procedure.

This procedure is optional for an RFD.

#### 6.3.2.1 Detection

The PAN coordinator shall conclude that a PAN ID conflict is present if either of the following apply:

- A Beacon frame is received by the PAN coordinator with the PAN Coordinator field, as defined in 7.3.1.3, set to one and the PAN ID equal to *macPanId*.
- A PAN ID Conflict Notification command, as defined in 7.5.6, 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 ID conflict is present if the following applies:

- A Beacon frame is received by the device with the PAN Coordinator field set to one, the PAN ID 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 ID conflict.

### 6.3.2.2 Resolution

On the detection of a PAN ID conflict by a device, it shall generate the PAN ID Conflict Notification command, as defined in 7.5.6, and send it to its PAN coordinator. Because the PAN ID Conflict Notification command has the Acknowledgment Request (AR) field set to request an acknowledgment, the PAN coordinator shall confirm its receipt by sending an Ack frame. Once the device has received the Ack frame from the PAN coordinator, the MLME shall issue an MLME-SYNC-LOSS.indication primitive, as described in 8.2.13.2, with the LossReason parameter set to PAN\_ID\_CONFLICT. If the device does not receive an Ack frame, the MLME shall not inform the next higher layer of the PAN ID conflict.

On the detection of a PAN ID 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 ID. The algorithm for selecting a suitable PAN ID is outside the scope of this standard. If the next higher layer does select a new PAN ID, it may then issue an MLME-START.request with the CoordRealignment parameter set to TRUE in order to realign the PAN, as described in 6.3.3.

### 6.3.3 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.

#### 6.3.3.1 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, as described in 8.2.9.1, with the SetDefaultPIB parameter set to TRUE, an ED channel scan, as described in 6.3.1.1, an active channel scan, as described in 6.3.1.2, and selecting a PAN ID distinct from PAN IDs received during the active channel scan. If the device is a SUN device operating as a coordinator, a passive scan for an Coexistence Specification IE should take place prior to the active channel scan, as described in 6.14. The algorithm for selecting a suitable PAN ID 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, as defined in 8.2.12.1, 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 6.3.3.4. After completing this, the MAC sublayer shall issue the MLME-START.confirm primitive, as described in 8.2.12.2, with a Status of SUCCESS and begin operating as the PAN coordinator.

An example of a message sequence chart for starting a PAN is illustrated in Figure 6-20.

#### 6.3.3.2 Realigning a PAN

If a coordinator receives the MLME-START.request primitive, as defined in 8.2.12.1, with the CoordRealignment parameter set to TRUE, the coordinator shall attempt to transmit a Coordinator Realignment command containing the new parameters for PanId, ChannelNumber, 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 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.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

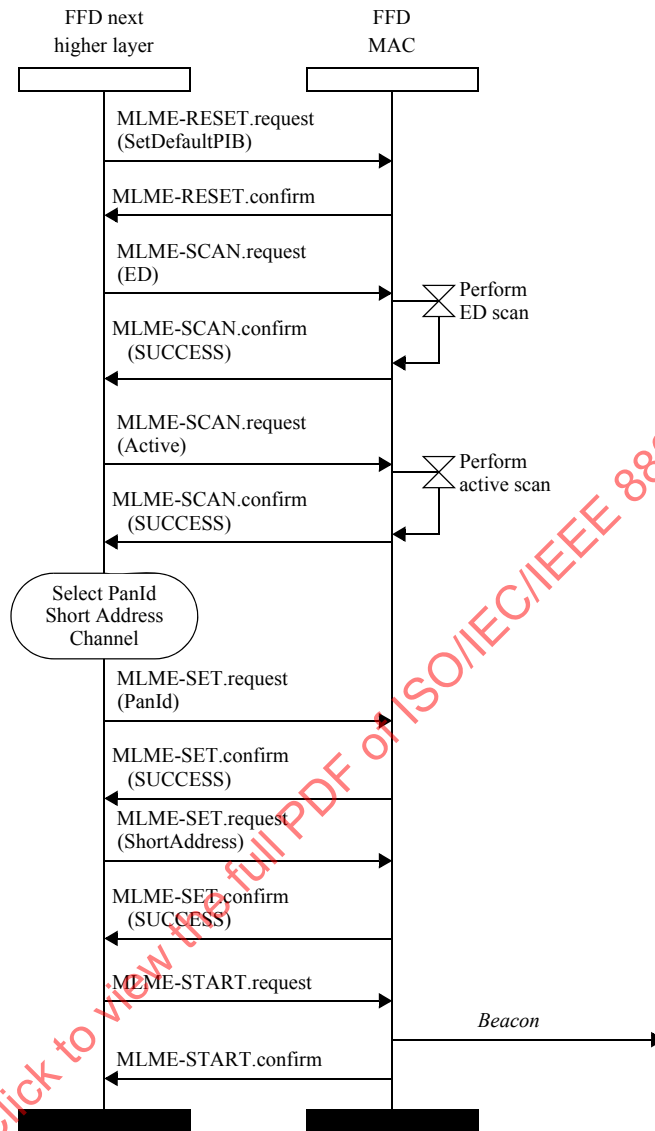


**Figure 6-20—PAN start message sequence chart—PAN coordinator**

When the coordinator is not already transmitting beacons and the CoordRealignement parameter is set to TRUE, the Coordinator Realignement command shall be transmitted immediately on the current channel using CSMA-CA.

If the transmission of the Coordinator Realignement 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 Realignement command, the new superframe configuration and channel parameters shall be put into operation as described in 6.3.3.4 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.



### 6.3.3.3 Realignment in a PAN

If a device has received the Coordinator Realignment command, as defined in 7.5.10, 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, ChannelNumber, 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 device that is not a coordinator may instead change the superframe configuration or channel parameters through use of the MLME-SET.request primitive, as described in 8.2.6.3.

TSCH devices shall not send a Coordinator Realignment command. TSCH devices that receive a Coordinator Realignment command shall ignore the command.

### 6.3.3.4 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 ChannelNumber parameters, respectively.

### 6.3.4 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, as defined in 8.2.12.1. 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 period 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 calculated time. 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*, as defined in 6.3.3.4, 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 ID in *macPanId* and use this value in the Source PAN ID field of the Beacon frame. The address used in the Source Address field of the Beacon frame shall contain the value of *macExtendedAddress* if *macShortAddress* is equal to 0xffff or *macShortAddress* otherwise.

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 6.5.2.

All Beacon frames, as defined in 7.3.1, shall be transmitted at the beginning of each superframe at an interval equal to  $aBase\text{-}SuperframeDuration \times 2^n$ , where  $n$  is the value of *macBeaconOrder*.

The Enhanced Beacon Request command allows a device to request that a coordinator respond with an Enhanced Beacon containing the requested IEs. Either header or payload IEs may be requested. Header IEs, Payload IEs or Nested IEs are requested by sending the desired IE with a length set to zero. If the Attribute ID field is empty in the Enhanced Beacon Filter IE and the IEs are not included in the Enhanced Beacon Request, then the responding coordinator shall return an Enhanced Beacon frame. If either are present and the filter criteria are met, the responding coordinator shall return an Enhanced Beacon frame containing the IEs as appropriate. If necessary to check filtering or return requested IEs, the Enhanced Beacon Filter IE and other IEs may be passed to a higher layer using the MLME-BEACON-REQUEST.indication primitive provided that the recipient has *macBeaconAutoRespond* set to FALSE.

When an Enhanced Beacon frame is generated in response to an Enhanced Beacon Request command that contained Attribute IDs in the Enhanced Beacon Filter IE, as described in 7.4.4.6, the content of the Enhanced Beacon frame shall include the IEs corresponding to the requested Attribute IDs shown in Table 6-3. A device supporting filtering by the Attribute ID listed in the first column shall also support the IEs in the IEs to include column in Table 6-3. When generated in response to an Enhanced Beacon Request command that contained a request for IEs, as described in 7.5.9, the Enhanced Beacon frame shall also contain the requested IEs.

**Table 6-3—Enhanced Beacon Request command IEs per enabled attribute**

| Attribute ID | PIB attribute            | IEs to include                                                                                                                     |
|--------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| 0            | <i>macTrleEnabled</i>    | TRLE Descriptor IE (F.5.1.1)                                                                                                       |
| 1            | <i>macTschEnabled</i>    | TSCH Synchronization IE (7.4.4.2), TSCH Slotframe and Link IE (7.4.4.3), TSCH Timeslot IE (7.4.4.4), Channel hopping IE (7.4.4.31) |
| 2            | <i>macDsmeEnabled</i>    | DSME PAN descriptor IE (7.4.2.5)                                                                                                   |
| 3            | <i>macLeEnabled</i>      | CSL IE (7.4.2.3), RIT IE (7.4.2.4)                                                                                                 |
| 4            | <i>macHoppingEnabled</i> | Hopping timing IE (7.4.4.5), Channel hopping IE (7.4.4.31)                                                                         |
| 5            | <i>macDaEnabled</i>      | DA IE (7.4.2.16)                                                                                                                   |
| 6            | <i>macMpmIe</i>          | Coexistence Specification IE (7.4.4.9)                                                                                             |
| 7–23         | —                        | Reserved                                                                                                                           |

For devices operating in beacon-enabled mode in the Japanese 920 MHz band, a coordinator may precede beacon transmission with listen before talk (LBT) without random backoff. The MAC shall ensure that the beacon is transmitted at the beginning of the superframe with accurate timing.

Beacon transmissions shall be given priority over all other transmit and receive operations.

### 6.3.5 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, as described in 6.3.4, using the identifier of the PAN with which the device has associated, *macPanId*, and its extended address, *macExtendedAddress*, if *macShortAddress* is equal to 0xffff, or its short address, *macShortAddress*, otherwise. A Beacon frame shall be transmitted at a rate of one Beacon frame every  $aBaseSuperframeDuration \times 2^n$ , where  $n$  is the value of *macBeaconOrder*.

### 6.3.6 TSCH PAN formation

A TSCH PAN is formed when a device, referred to as an advertising device, advertises the presence of the network by sending Enhanced Beacon frames upon receipt of a MLME-BEACON.request from a higher layer. In a TSCH PAN the Enhanced Beacon frames contain the following IEs:

- TSCH Synchronization IE, as described in 7.4.4.2, containing timing information so new devices can synchronize to the network.
- Channel hopping IE, as described in 7.4.4.31, containing channel hopping information, as described in 6.2.10.
- TSCH Timeslot IE, as described in 7.4.4.4, containing timeslot information describing when to expect a frame to be transmitted and when to send an acknowledgment.
- TSCH Slotframe and Link IE, as described in 7.4.4.3, containing initial link and slotframe information so new devices know when to listen for transmissions from the advertising device and when they can transmit to the advertising device.

Enhanced Beacon frames in TSCH mode shall not be encrypted, but may be authenticated (i.e., security level 1, 2 or 3).

NOTE—If Enhanced Beacon frames were encrypted, the TSCH Synchronization IE used to transmit the ASN to joining devices would be encrypted as well. A joining device or a device that has lost synchronization with the network would be unable to decrypt the Enhanced Beacon frame as the current ASN is required for the generate the nonce. Thus, these devices would be unable to join the network or resynchronize with the network.

The device wishing to join a TSCH network begins passively (preferred) or actively scanning for the network as the result of receiving an MLME-SCAN.request from a higher layer. Once the listening device has heard a valid Enhanced Beacon, it generates an MLME-BEACON-NOTIFY.indication to a higher layer. The higher layer may wait for additional MLME-BEACON-NOTIFY.indication primitives before selecting a TSCH network based upon the value of the Join Metric field in the TSCH Synchronization IE. The higher layer may initialize the slotframe and links contained in the Enhanced Beacon from the preferred TSCH network and switch the device into TSCH mode with a MLME-TSCH-MODE.request.

NOTE—A lower value in the Join Metric field indicates that connection of the beaconing device to a specific network device determined by the higher layer is a shorter route.

At this point the device is synchronized to the network and may optionally send an Association Request command. If the device uses association, it may request a short address. The sequence of messages exchanged to synchronize a device to the networks is shown in Figure 6-21, and the process of synchronization is described in 6.5.4.

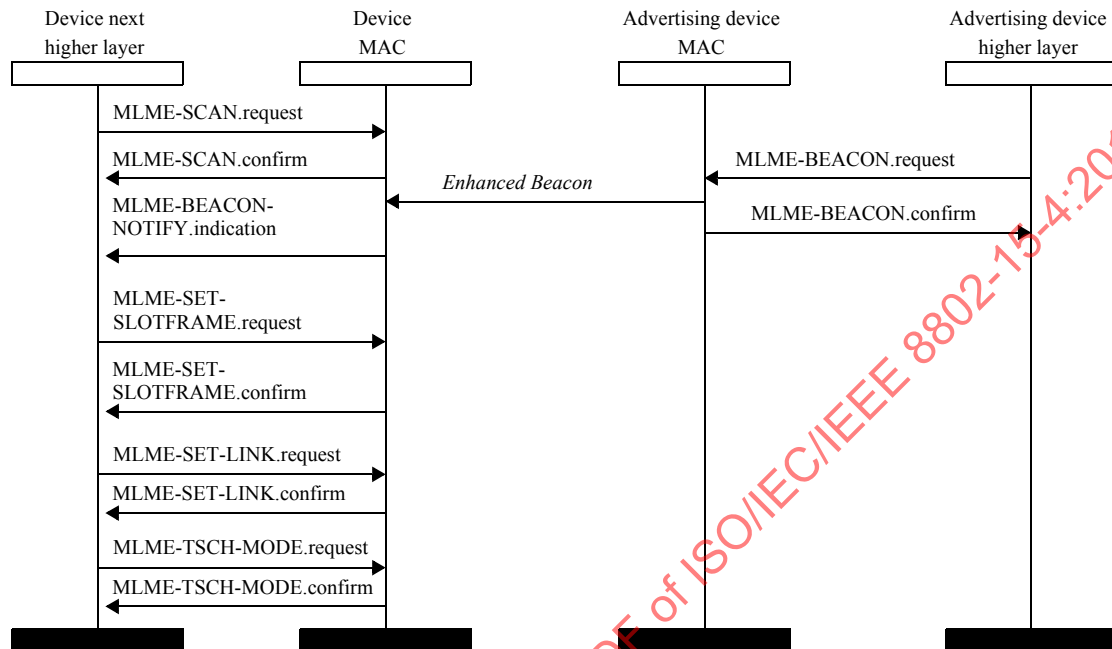


**Figure 6-21—Message sequence chart for TSCH procedure to find an advertising device**

Typically at this point the device will go through a procedure to allocate additional communication resources (slotframes and links) to the joining device. This procedure may include a security handshake to mutually authenticate the joining device, configure encryption keys, and configure routing information. The mechanism and rules for setting up these additional communication links needs to be defined in a higher layer standard.

Once synchronized and configured by a higher layer to do so, all FFDs that are already part of the network may send Enhanced Beacon frames announcing the presence of the network. The advertising rate and content is configured by a higher layer as appropriate to the density of devices, the desired rate of network formation, and the energy devoted to network formation.

After joining, the device may receive additional slotframes and links from a higher layer management entity or peer. Likewise, the device may be instructed to remove certain slotframes and links obtained from the Enhanced Beacon.

## 6.4 Association and disassociation

### 6.4.1 Association

The next higher layer shall attempt to associate only after having first performed a MAC sublayer reset, by issuing the MLME-RESET.request primitive with the SetDefaultPIB set to TRUE, and then having completed either an active or a passive channel scan, as defined in 6.3.1.2. 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 outside 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, as described in 8.2.3.1, that the MLME configures the following PHY and MAC PIB attributes to the values necessary for association:

- *phyCurrentChannel* shall be set equal to the *ChannelNumber* 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 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.

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, as described in 7.5.2, to the coordinator of an existing PAN. This is optional for RFD-RX and RFD-TX devices. If the Association Request command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer.

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*. 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 replaced. If sufficient resources are available, the next higher layer should allocate a short address to the device that is unique within the PAN, and the MAC sublayer shall generate an Association Response command, as described in 7.5.3, containing the new address and a Status field 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 field indicating a failure, as defined in Table 7-50. The Association Response command shall be sent to the device requesting association using indirect transmission; i.e., the Association Response command 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 6.7.3.

If the Allocate Address field of the Capability Information field, as described in 7.5.2, of the Association Request command is set to one, the next higher layer of the coordinator shall allocate a address with a range depending on the addressing mode supported by the coordinator, as described in Table 6-4. If the Allocate Address field of the Association Request command is set to zero, the short address shall be equal to 0xffff. A short address of 0xffff 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 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* for the coordinator to make its association decision; *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*. If the device does not receive an Association Response command from the coordinator within *macResponseWaitTime*, the MLME shall issue the MLME-ASSOCIATE.confirm primitive, as described in 8.2.3.4, 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, as described in 8.2.13.1, with the TrackBeacon parameter set to FALSE.

If the Association Status field of the Association Response command indicates that the association was successful, the device shall store the address contained in the 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 6-4. 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, shall be stored in *macCoordExtendedAddress*.

**Table 6-4—Usage of the short address**

| Value of <i>macShortAddress</i> | Description                                                                                                                                                                                                   |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0000–0xffffd                  | If a source address is included, the device shall use short source addressing mode for Beacon frames and Data frames and the appropriate source addressing mode specified in 7.3.4 for MAC command frames.    |
| 0xffffe                         | If a source address is included, the device shall use extended source addressing mode for Beacon frames and Data frames and the appropriate source addressing mode specified in 7.3.4 for MAC command frames. |
| 0xfffff                         | 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 7.3.4 for MAC commands.                 |

If the value of the Association Status field of the command is not “Association successful,” if there were a communication failure during the association process due to a missed Ack frame, or if the Association Response command were not received, the device, optional for RFD-RX and RFD-TX devices, shall set *macPanId* to the default value.

A message sequence chart for association is illustrated in Figure 6-22.

Figure 6-23 illustrates a sequence of messages that may be used by a device that is not tracking the beacon of the coordinator to successfully associate with a PAN.

Association is optional for devices operating in TSCH mode.

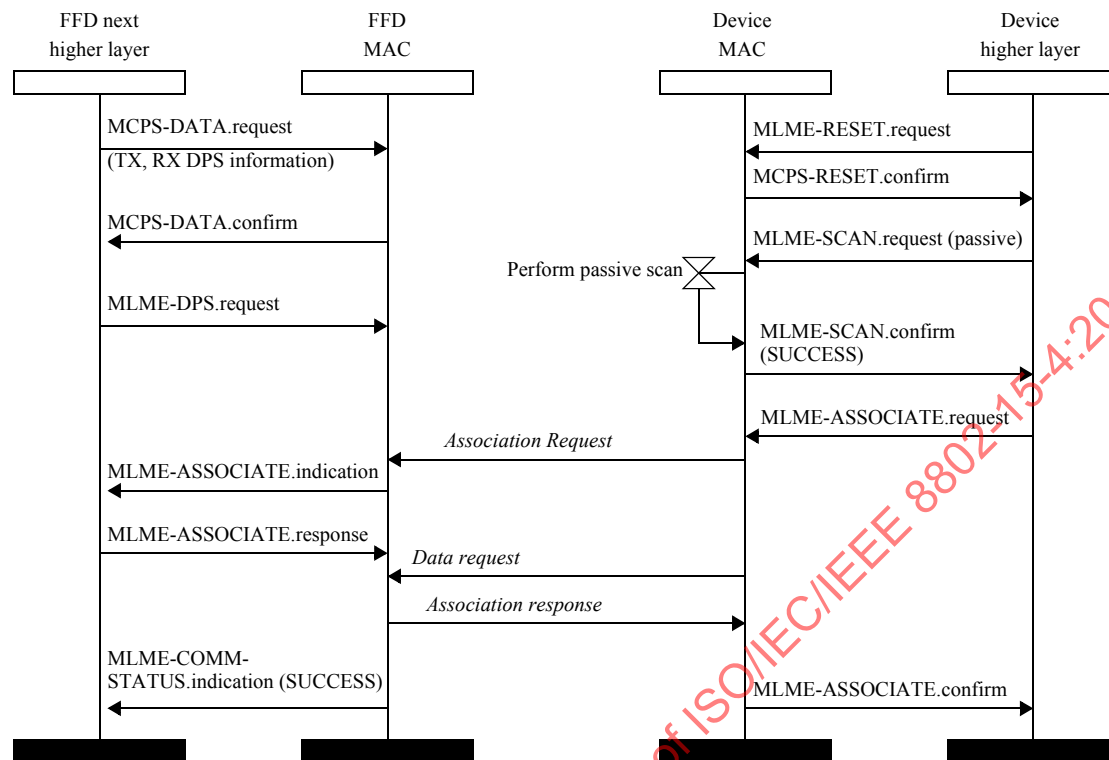


Figure 6-22—Association message sequence chart

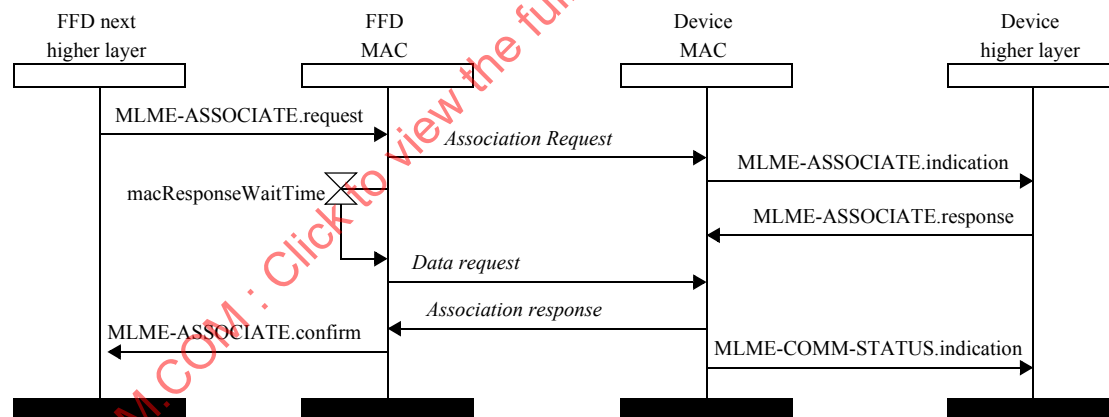


Figure 6-23—Association for a device that is not tracking beacons

#### 6.4.2 Disassociation

The disassociation procedure is initiated by the next higher layer by issuing the MLME-DISASSOCIATE.request primitive, as described in 8.2.4.1, 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 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 6.7.3. If the command 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.

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. If the acknowledgment to Disassociation Notification command is not received, the device should consider itself disassociated.

If the source address contained in the Disassociation Notification command is equal to *macCoordExtendedAddress*, 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 corresponds to one of its associated devices; if so, the coordinator should consider the device disassociated. If none of these conditions are satisfied, the Disassociation Notification 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 *macCoordExtendedAddress* 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, as described in 8.2.4.3.

Figure 6-24 illustrates the sequence of messages for a device to disassociate itself from the PAN.

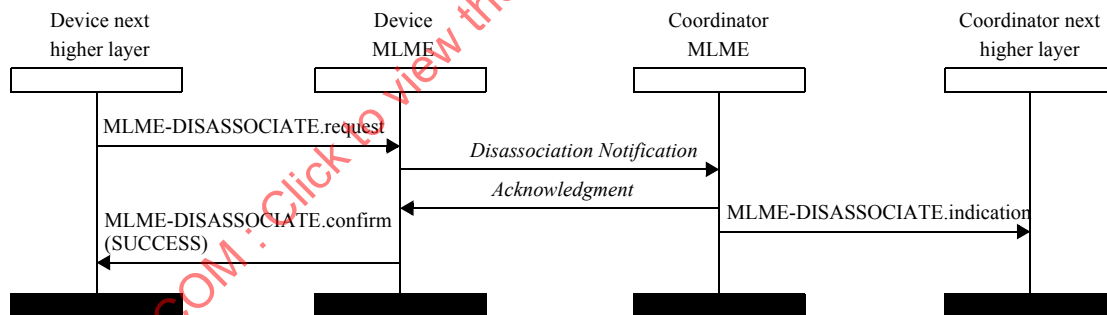


**Figure 6-24—Message sequence chart for disassociation initiated by a device**

Figure 6-25 illustrates the sequence necessary for a coordinator in a beacon-enabled PAN to successfully disassociate a device from its PAN using indirect transmission.

For devices using the optional TSCH mode, additional disassociation behavior is required. A device shall only disassociate from the PAN if it receives a Disassociation Notification command from either the PAN coordinator, or all of its time source neighbors, as defined in 6.5.4.

Upon determining that it should disassociate from the PAN, the device shall transmit Disassociate Notification commands to all its neighbors on any available link for *macDisconnectTime* timeslots, after which it should clear all synchronization information and leave the PAN.



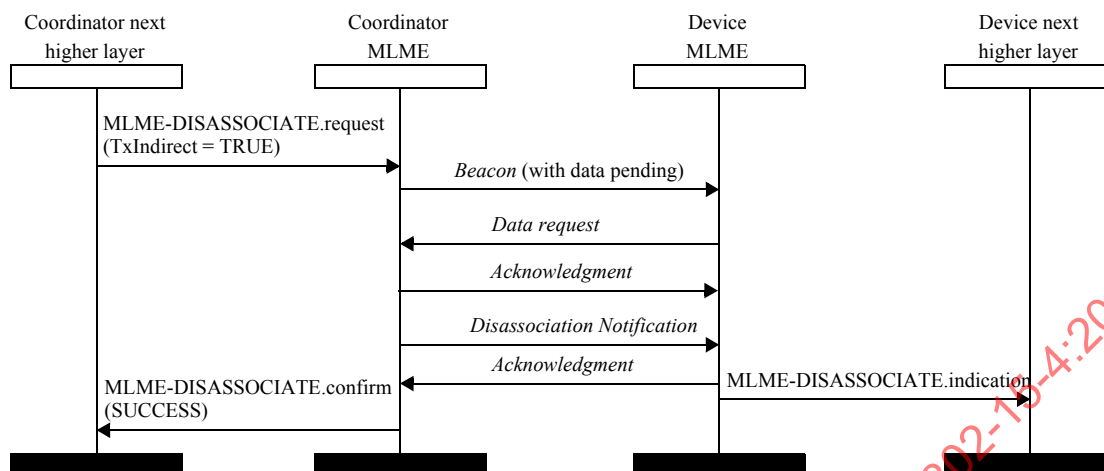


**Figure 6-25—Message sequence chart for disassociation initiated by a coordinator, using indirect transmission, in a beacon-enabled PAN**

### 6.4.3 Fast association

Fast association is optional.

A device is instructed to fast associate with a PAN through the MLME-ASSOCIATE.request primitive.

The MAC sublayer of an unassociated device initiates the fast association procedure by sending an Association Request command with the Association Type field of the Capability Information field set to one to the coordinator of an existing PAN.

If the device does not receive an Association Response command from the coordinator within *macResponseWaitTime*, the MLME shall issue the MLME-ASSOCIATE.confirm primitive, as described in 8.2.3.4, with a Status of NO\_DATA, and the association attempt shall be deemed a failure.

If the coordinator next higher layer accepts the association request, it send an MLME-ASSOCIATE.response to the MAC sublayer allocating a short address to the device that is unique within the PAN. If the coordinator next higher layer rejects the association request, it will send an MLME-ASSOCIATE.response with a Status parameter indicating the reason for the rejection. Upon receipt of an MLME-ASSOCIATE.response primitive, the MAC sublayer shall generate an Association Response command. If the request was successful, the Association Response command contains the new address and a Status field indicating a successful fast association. If the request failed, the Association Response command with a Status field set to indicate the reason the request failed.

If the Association Type field of the Capability Information field of the Association Request command is set to one, the MAC sublayer of the coordinator shall send the Association Response command to the device directly. If the Association Type field of the Capability Information field of the Association Request command is set to zero, the Association Response command shall be sent as 6.4.1.

If the Association Status field of the command indicates “fast association successful,” the device shall store the address contained in the Short Address field of the command in *macShortAddress* for its communication use in the PAN.

Figure 6-26 illustrates a sequence of messages for fast association.



IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

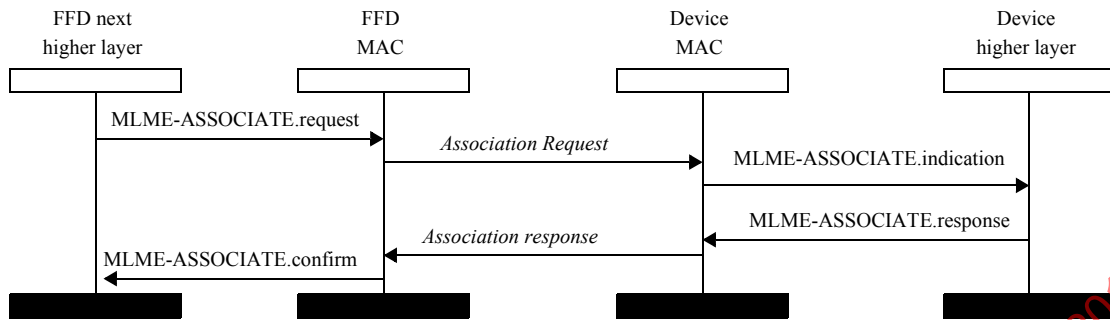


Figure 6-26—Fast Association message sequence chart

## 6.5 Synchronization

### 6.5.1 General

This subclause specifies the procedures for coordinators to generate Beacon frames and for devices to synchronize with a coordinator. For PANs supporting Beacon frames, synchronization is performed by receiving and decoding the Beacon frames. For PANs not supporting Beacon frames, synchronization is performed by polling the coordinator for data.

For devices using the optional TSCH mode, initial synchronization is performed by the use of Enhanced Beacon frames, and synchronization is maintained by slotted communication with other devices in the PAN.

### 6.5.2 Synchronization with beacons

All devices operating on a beacon-enabled PAN (i.e.,  $macBeaconOrder < 15$ ) shall be able to acquire beacon synchronization 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 ID specified in *macPanId*. If *macPanId* specifies the broadcast PAN ID, 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  $[aBaseSuperframeDuration \times (2^n + 1)]$ , where  $n$  is the value of *macBeaconOrder*. If a Beacon frame containing the current PAN ID 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 field is set to one), the device shall attempt to unsecure the Beacon frame using the unsecuring process described in 9.2.3.

If the Status from the unsecuring process is not SUCCESS, the MLME shall issue an MLME-COMM-STATUS.indication primitive, as described in 8.2.5.2, with the Status parameter set to the Status from the unsecuring process, indicating the error.

The security-related elements of the PAN descriptor corresponding to the beacon, as defined in Table 8-12, shall be set to the corresponding parameters returned by the unsecuring process. The SecurityStatus 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 ID 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 ID 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 short address or extended address of the device and the source PAN ID matches *macPanId*, the MLME shall follow the procedure for extracting pending data from the coordinator, as described in 6.7.3.

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.

In Figure 6-27 the next higher layer issues a synchronization request with TrackBeacon set to FALSE. 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 6.7.3.

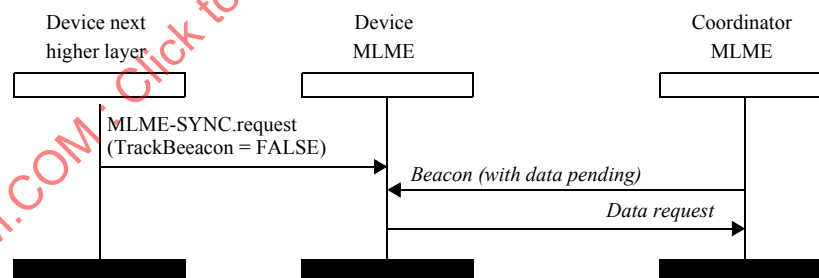


**Figure 6-27—Synchronizing to a coordinator in a beacon-enabled PAN without tracking beacons**

In Figure 6-28, the next higher layer issues a synchronization request with TrackBeacon set to TRUE. 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.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

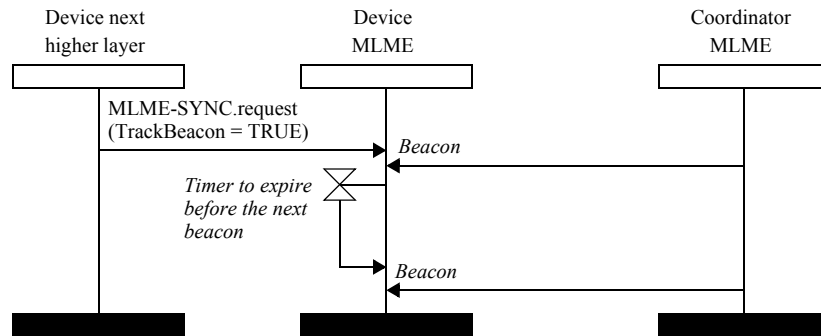


**Figure 6-28—Synchronizing to a coordinator in a beacon-enabled PAN while tracking beacons**

### 6.5.3 Synchronization without beacons

All devices, except for RFD-RX and RFD-TX 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, as described in 8.2.14.1. On receipt of this primitive, the MLME shall follow the procedure for extracting pending data from the coordinator, as described in 6.7.3.

### 6.5.4 Synchronization in TSCH PAN

In a TSCH PAN, all communication happens in timeslots as described in 6.5.4.1. To remain synchronized, the devices should have the same notion of when each timeslot begins and ends, within  $\pm \text{macTsRxWait}/2$ . In a typical TSCH PAN, time propagates outwards from the PAN coordinator. A device shall periodically synchronize its network time to at least one other network device, which is a time source neighbor. A time source neighbor is another device for which the *macLinkTimekeeping* is TRUE for the *macLinkHandle* a link. The device may also provide its network time to one or more network devices. A higher layer may add or change time source neighbors at any time.

Note that a device sending Enhanced Beacons to advertise a TSCH PAN should set the Timekeeping bit in the Link Option field, as described in 7.4.4.3, for the joining devices' receive link so that joining devices can maintain time synchronization until additional time source neighbors are configured by a higher layer.

A network device may have more than one neighbor as its time source. In such cases, the device shall synchronize its time to all of the neighbors that are acting as its time source, synchronizing to the relative drift of all its time source neighbors.

Figure 6-29 shows an example of time propagation in a TSCH PAN. The arrows indicate the direction of time distribution. In this example, the PAN coordinator acts as the time source for the entire network. Device 1 synchronizes to the PAN coordinator only, and is the time source for device 3. Device 2 synchronizes its time to both 1 and the PAN coordinator, and device 4 synchronizes to the PAN coordinator, device 2, and device 3.

Synchronization is possible whenever a device exchanges a frame with a time source neighbor. This can either come from receiving an acknowledgment with time correction information, as described in 7.4.2.7, or from the arrival time of a frame from the time source neighbor. Both methods are described in 6.5.4.2.

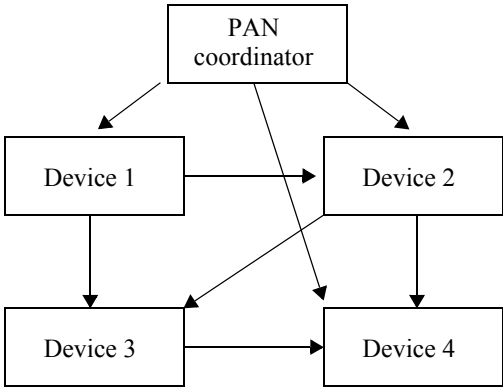


Figure 6-29—Example of possible time propagation in a TSCH PAN

In order to ensure that it remains synchronized with the TSCH PAN, a network device shall ensure that it communicates with each of its timekeeping neighbors, as defined in 8.2.19.7.

6.5.4.1 Timeslot communication

During a timeslot in a slotframe, one node typically sends a frame, and another sends back an Enh-Ack frame containing the Time Correction IE, as described in 7.4.2.7, if it successfully receives that frame. A positive acknowledgment indicates that the receiver has successfully received the frame and has taken ownership of it for further routing. A negative acknowledgment indicates that the receiver cannot accept the frame at this time, but has received it with no errors. The Time Correction IE includes timing information used by nodes to maintain network synchronization. Frames sent to a unicast node address require that a link-layer acknowledgment be sent in response during the same timeslot as shown in Figure 6-30. If an acknowledgment is requested and not received within the time-out period, retransmission of the frame waits until the next assigned transmit timeslot to that address occurs.

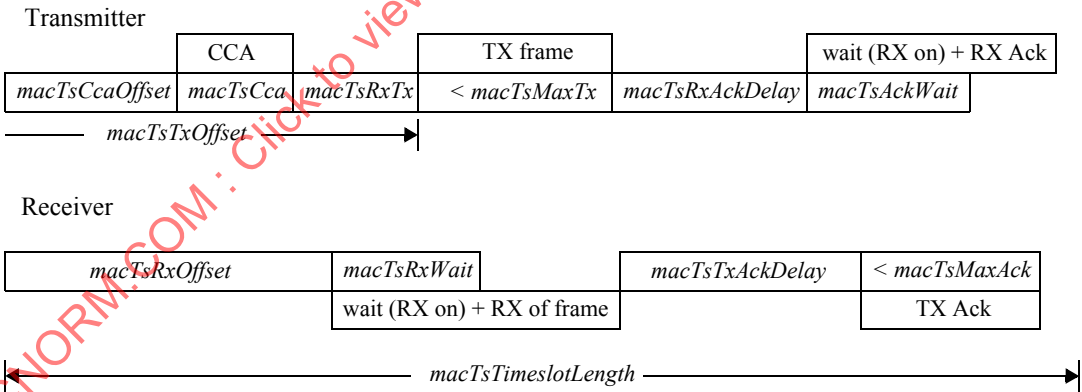


Figure 6-30—Timeslot diagram of acknowledged transmission

As shown in Figure 6-30, the timeslot starts at time  $T = 0$  from the transmitting device's perspective. The transmitter waits  $macTsCcaOffset$  and then performs CCA (if active). At  $macTsTxOffset$ , the device begins transmitting the frame. If an acknowledgment is expected, the device waits  $macTsRxAckDelay$  and then enables the receiver to await the acknowledgment. If the acknowledgment does not arrive within the expected time the device may idle the radio and consider the transmission a failure. If no acknowledgment is expected, the transmitter may idle the radio after sending the frame.

On the receiver's side, at its estimate of  $T = 0$  it waits *macTsRxOffset* and then goes into receive mode for *macTsRxWait*. If the frame has not started by that time, it may idle the receiver. Otherwise, once the frame has been received, the receiver waits *macTsTxAckDelay* and then sends an acknowledgment.

The transmitter or receiver may resynchronize clocks as described in 6.5.4.2.

#### 6.5.4.2 Node synchronization

Device-to-device synchronization is necessary to maintain connection with neighbors in a slotframe-based network. There are two methods for a device to synchronize to the network, Acknowledgment-based and Frame-based. Originator in this context is the device sending a frame, and receiver is the device receiving that frame and sending back an acknowledgment as is appropriate. Since timestamps are required to maintain synchronization in a TSCH PAN, all devices shall have *macTimestampSupported* = TRUE.

Acknowledged communication provides a basic method of time synchronization through the exchange of Data frames and Ack frames. The algorithm involves the receiver calculating the delta between the expected time of frame arrival and its actual arrival, and providing that information to the transmitting node in the subsequent acknowledgment.

The acknowledgment-based synchronization algorithm can be described as follows:

- Originator sends a frame, timing the start symbol to be sent at *macTsTxOffset* according to its clock, which would correspond to  $\text{macTsRxOffset} + \text{macTsRxWait}/2$  in the receiver's clock if both clocks were perfectly synchronized.
- Receiver records the timestamp when it receives the start symbol of the frame.
- Receiver calculates a time correction =  $\text{macTsRxOffset} + \text{macTsRxWait}/2 - \text{arrival timestamp}$ .
- Receiver sends back the time correction in the IE field in the corresponding Enh-Ack frame (assuming the incoming frame passes validation).
- Originator receives the acknowledgment. If the receiver node is a time source neighbor, the originator adjusts its own clock by incorporating the difference into an average of the drift to all its time source neighbors. The averaging method is implementation dependent. If the receiver is not a clock source, the time correction is ignored.

In Frame-based synchronization a node may synchronize its own network clock if it receives a frame from a time source neighbor. The receiver calculates the delta between expected time of frame arrival and its actual arrival time to use that information to adjust its own clock.

The Frame-based synchronization algorithm can be described as follows:

- Receiver records the timestamp when it receives the start symbol of the frame.
- The receiver calculates a time correction =  $\text{macTsRxOffset} + \text{macTsRxWait}/2 - \text{arrival timestamp}$ .
- If the originator was a time source neighbor, the receiver adjusts its own clock by incorporating the information from all of its time source neighbors in an implementation-dependent manner. If the originator is not a time source neighbor, the time correction shall be ignored.

Figure 6-31 illustrates both time synchronization mechanisms. In both cases, the receiver calculates its time adjustment to either send back to the transmitting device or to use locally.

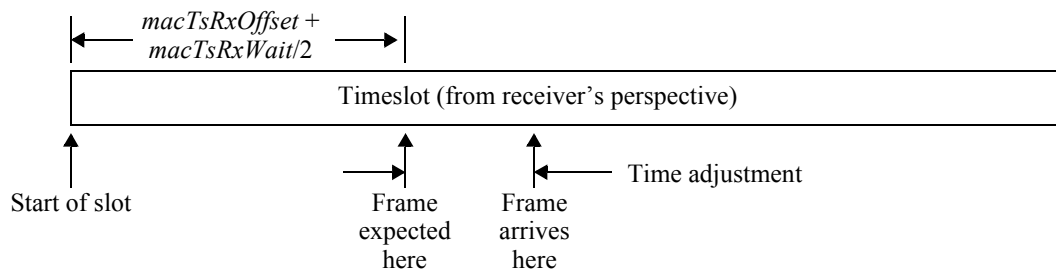


Figure 6-31—Time synchronization

### 6.5.5 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 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 set of channels to be scanned. Upon receiving this primitive, the MAC sublayer shall begin an orphan scan, as described in 6.3.1.3.

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, as described in 7.5.10.

## 6.6 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 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, as described in 8.2.3.3. 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 except for RFD-RX devices serving as PAN coordinator termination points, 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 field 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 set of pending addresses in the Beacon frame. Instead it shall transmit the transaction in the GTS allocated for the device, as described in 6.8.4.

If there is a transaction pending for the broadcast address, the Frame Pending field in the Beacon frame shall be set to one, and the pending message shall be transmitted immediately following the Beacon frame using the CSMA-CA algorithm. TSCH mode indirect behavior is further described in 7.2.1.3. In a beacon-enabled PAN, if there is a second message pending for the broadcast address, its transmission shall be delayed until the following superframe.

If a device receives a Beacon frame with the Frame Pending field set to one, it shall leave its receiver enabled to receive the broadcast Data frame from the coordinator.

In a nonbeacon-enabled PAN, upon receipt of the MLME-POLL.request primitive, a device shall attempt to extract the data from the coordinator, as defined in 6.7.3.

In a beacon-enabled PAN, a device that receives a Beacon frame containing its address in the set of pending addresses shall attempt to extract the data from the coordinator, as defined in 6.7.3.

## 6.7 Transmission, reception, and acknowledgment

### 6.7.1 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 outside the scope of this standard. Each time a Data frame or a MAC command 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 data sequence number (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 beacon sequence number (BSN) value in the MAC PIB attribute *macBsn* and initialize it to a random value; the algorithm for choosing a random number is outside 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.



Each coordinator shall store its current enhanced beacon sequence number (EBSN) value in the MAC PIB attribute *macEbsn* and initialize it to a random value; the algorithm for choosing a random number is outside the scope of this standard. Each time an Enhanced Beacon frame is generated, the MAC sublayer shall copy the value of *macEbsn* into the Sequence Number field of the MHR of the outgoing frame and then increment it by one. The value of *macEbsn* shall be permitted to roll over.

NOTE—The DSN, BSN, and EBSN 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 short address (i.e., *macShortAddress* is not equal to 0xffff or 0xfffff), it shall use that address in preference to its extended address (i.e., *macExtendedAddress*) wherever possible. When a device has not yet associated to a PAN, it shall use its 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 or RFD-TX device, and the Destination Address field shall contain the address of the recipient, optional for RFD-TX devices.

The Destination Address field, if present, shall contain the address of the intended recipient of the frame, which may be either a short address or an 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.

The PAN ID compression field, the Source PAN ID field and the Destination PAN ID field are set as indicated in 7.2.1.5.

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, as described in 6.5.2, and hence the device does not know where the beacon will appear, it shall enable its receiver and search for at most  $[aBaseSuperframeDuration \times (2^n + 1)]$ , 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, as described in 6.2.5. 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, as described in 6.2.5, and transmissions in a GTS shall not use CSMA-CA. When the LECIM direct sequence spread spectrum (DSSS) PHY is in use in a beacon-enabled PAN, transmissions after the beacon shall commence *phyLecimDsssPpduTxAt* following the reception of the last symbol of the Beacon frame.

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, as described in 6.2.5.

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 9.2.1.

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, and shall not transmit the frame.

If the Status from the outgoing frame security procedure is SUCCESS, the MAC sublayer shall transmit the frame.



If security is enabled in TSCH mode, then only one frame shall be sent in a given slot to prevent the same nonce being used for more than one frame. The ASN is used in the nonce in TSCH mode.

### 6.7.2 Reception and rejection

When PSDU fragmentation is in use, the acknowledgment of fragments uses the procedure described in 23.3.

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 radio communications range, 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 next higher layer.

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, as described in 7.2.10. 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 next higher layer 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), the third layer of filtering shall be dependent on whether the MAC sublayer is currently performing a scan. If the MAC sublayer is currently performing a scan, the MAC sublayer shall process all frames received as described in the relevant subclause of 6.3.1. If the MAC sublayer is not currently performing a scan, it shall accept only frames that satisfy all of the following fourth-level filtering requirements:

- a) The Frame Type field shall not contain a reserved frame type.
- b) The Frame Version field shall not contain a reserved value.
- c) If a destination PAN ID is included in the frame, it shall match *macPanId* or shall be the broadcast PAN ID.
- d) The Destination Address field shall satisfy one of the following conditions:
  - 1) A short destination address is included in the frame and it matches either *macShortAddress* or the broadcast address.
  - 2) An extended destination address is included in the frame and matches either *macExtendedAddress* or, if *macGroupRxMode* is set to TRUE, an EUI-64 group address, as defined in IEEE Std 802.
  - 3) The Destination Address field and the Destination PAN ID field are not included in the frame and *macImplicitBroadcast* is TRUE.

- 4) The device is the PAN coordinator, only source addressing fields are included in a Data frame or MAC command and the source PAN ID matches *macPanId*.
- 5) The device is the PAN coordinator, only source addressing fields are included in a Multipurpose frame and the destination PAN ID matches *macPanId*.
- e) If the frame type indicates that the frame is a Beacon frame, the source PAN ID shall match *macPanId* unless *macPanId* is equal to the broadcast PAN ID, in which case the Beacon frame shall be accepted regardless of the source PAN ID.

If any of the fourth-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 field indicates a Data frame or MAC command with the Frame Version field set to 0b00–0b01 and the AR field is set to request an acknowledgment, the MAC sublayer shall send an Imm-Ack frame. Prior to the transmission of the Imm-Ack frame, the value of the Sequence Number field included in the received Data frame or MAC command shall be used for the value of the Sequence Number field of the Imm-Ack frame. This step will allow the transaction originator to know that it has received the appropriate Imm-Ack frame.

For valid frames that are not broadcast, if the Frame Type field indicates one of a Multipurpose frame, a Data frame with the Frame Version set to 0b10 or a MAC command with the Frame Version field set to 0b10, and the AR field is set to request an acknowledgment, the MAC sublayer shall send an Enh-Ack frame unless the device performs the incoming frame security procedure, as defined in 9.2.3. If the device performs the incoming frame security procedure and the Status is not SUCCESS, the device is not required to send an Enh-Ack frame. If the Enh-Ack frame contains IEs and/or a Frame Payload and it is in response to a secured frame, then the Enh-Ack frame shall be secured. If the Enh-Ack frame is in response to a secured frame and does not contain either IEs or a Frame Payload, then the Enh-Ack may be secured.

If the Source PAN ID field is not included in the frame and the Destination PAN ID field is included in the frame, the MAC sublayer shall use the value of Destination PAN ID field as the source PAN ID.

The device shall process the frame using the incoming frame security procedure described in 9.2.3, if the Security Enabled field is set to one, or 9.2.4, if the Security Enabled field is set to zero.

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.

All IEs received in a valid frame that are marked as PASSED by the incoming frame security procedure shall be processed by the MAC sub-layer, and if required, passed to the next higher layer.

If the valid frame is a Data frame or Multipurpose frame and the status from the incoming frame security procedure is SUCCESS, the MAC sublayer shall pass the MSDU to the next higher layer.

### 6.7.3 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 6.5.2. 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, as described in 7.5.5, to the coordinator during the CAP with the AR field set to request an acknowledgment; the only exception to this is if the Beacon frame is received while performing an active or passive scan, as described in 6.3.1. There are two other cases for which the MLME 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* after the acknowledgment to a MAC command, 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 9.2.1. Otherwise, the MLME shall perform the security process on the Data Request command based on the *secAutoRequestSecurityLevel*, *secAutoRequestKeyIdMode*, *secAutoRequestKeySource*, and *secAutoRequestKeyIndex* PIB attributes, according to 9.2.1.

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 Ack frame, as described in 6.7.4.2, it shall set the Frame Pending field of the Ack 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 field of the Ack frame to one.

On receipt of the Ack frame with the Frame Pending field set to zero, the device shall conclude that there are no data pending at the coordinator.

On receipt of the Ack frame with the Frame Pending field set to one, a device shall enable its receiver 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 *macSifsPeriod* and (*macSifsPeriod* + *aUnitBackoffPeriod*), on a backoff period boundary, and there is time remaining in the CAP for the message, appropriate IFS, and acknowledgment. If a requested Ack 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 the expected time 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 Ack frame, if requested, thus confirming receipt.

If the Frame Pending 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.

In Figure 6-32 a poll request is issued to the MLME, which then sends a Data Request command to the coordinator. The corresponding Ack frame has the Frame Pending (FP) field set to zero and the MLME issues the poll request confirmation immediately.

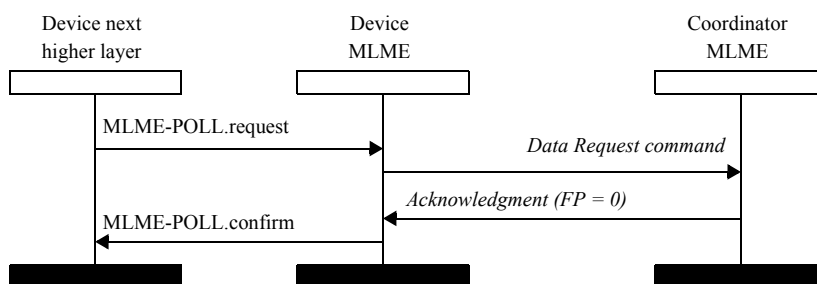


**Figure 6-32—Message sequence chart for requesting data from the coordinator when the coordinator does not have data pending**

In Figure 6-33 a poll request is issued to the MLME, which then sends a Data Request command to the coordinator. The corresponding Ack frame has the Frame Pending field 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.

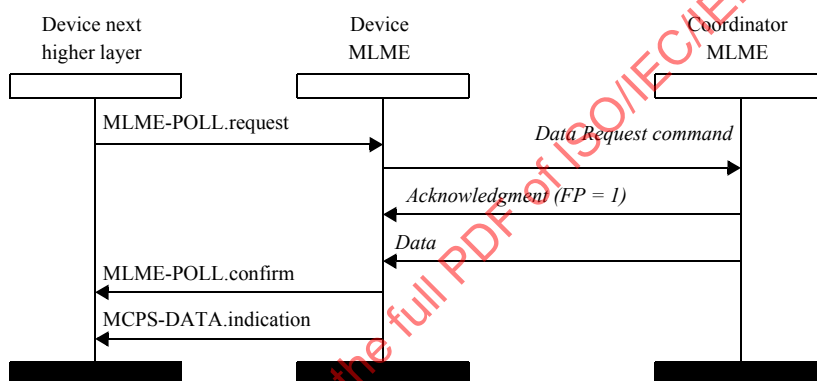


**Figure 6-33—Message sequence chart for requesting data from the coordinator when the coordinator has data pending**

#### 6.7.4 Use of acknowledgments and retransmissions

A Data frame or MAC command shall be sent with the AR field set appropriately for the frame. A Beacon frame or Ack frame shall always be sent with the AR field set to indicate no acknowledgment requested. Similarly, any frame that is broadcast or has a group address as the extended destination address, as defined in IEEE Std 802, shall be sent with its AR field set to indicate no acknowledgment requested.

##### 6.7.4.1 No acknowledgment

A frame transmitted with its AR field set to indicate no acknowledgment requested, as defined in 7.2.1.4, shall not be acknowledged by its target. The originating device shall assume that the transmission of the frame was successful.

The message sequence chart in Figure 6-34 shows the scenario for transmitting a single frame of data from an originator to a target without requiring an acknowledgment.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

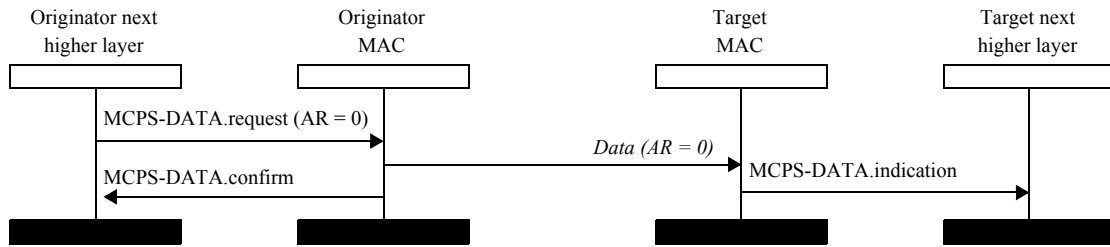


Figure 6-34—Successful data transmission without an acknowledgment

#### 6.7.4.2 Acknowledgment

If the intended recipient received a valid frame, as defined in 6.7.2, with the AR field set to request an acknowledgment, it shall generate and send an Ack frame, as defined in 6.7.2.

The AR field shall be set to no acknowledgment when using Frak.

The transmission of an Ack frame in a nonbeacon-enabled PAN or in the CFP shall commence AIFS after the reception of the last symbol of the Data frame or MAC command. The transmission of an Ack frame in the CAP shall commence either AIFS after the reception of the last symbol of the Data frame or MAC command or at a backoff period boundary. In the latter case, the transmission of an Ack frame shall commence between AIFS and  $(AIFS + aUnitBackoffPeriod)$  after the reception of the last symbol of the Data frame or MAC command. The value of AIFS is 1 ms for the SUN PHYs, LECIM PHYs, or TVWS PHYs. The value of AIFS is equal to *macSifsPeriod* for all other PHYs.

The receiving device may include additional content in an Enh-Ack frame using IEs. If the originator does not understand a specific IE in the Enh-Ack frame, that IE is ignored, but the transmission is considered successful. The number and the content of the IEs included in the Enh-Ack should be limited to only those IEs with the minimal content that is required.

In the case where the Ack frame may violate the transmission timing restriction defined in 6.7.5, the device shall use the default SHR and PHR lengths defined for that PHY when sending the Ack frame.

The Time Correction IE shall be used in all Enh-Ack frames if *macTschEnabled* is TRUE. When returning Time Correction IE, as described in 7.4.2.7, in the Enh-Ack frame the receiving device may indicate a negative acknowledgment to indicate that the frame successfully passed FCS check, but that the MAC discarded the frame.

The message sequence chart in Figure 6-35 shows the scenario for transmitting a single Data frame from an originator to a recipient with an acknowledgment requested.

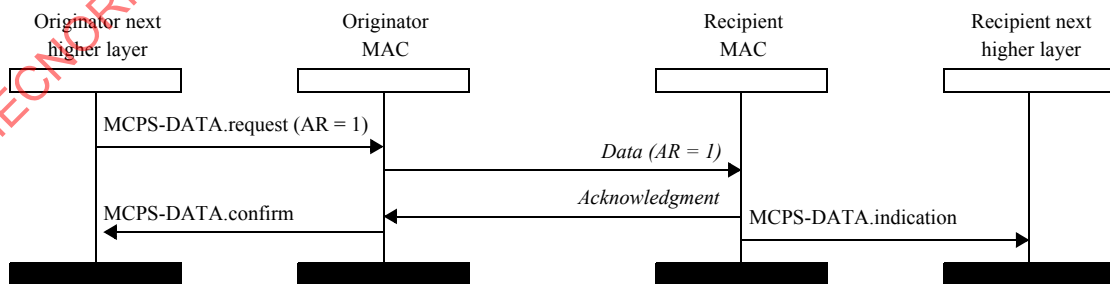


Figure 6-35—Successful data transmission with an acknowledgment

When in TSCH mode, incoming frames are acknowledged using the Enh-Ack frame as described in 7.3.3. Security of the Enh-Ack frame shall match that of the incoming frame. When operating in the TSCH mode the Enh-Ack frame is sent at the time specified by the *macTimeslotTemplate*.

For TVWS RDEVs, the Enh-Ack frame shall include the Timestamp Difference IE as defined in 7.4.4.27 if the Ranging field of the PHR is set to one in the frame being acknowledged.

#### 6.7.4.3 Retransmissions

A device that sends a frame with the AR field set to indicate no acknowledgment requested may assume that the transmission was successfully received and shall not perform the retransmission procedure.

A device that sends a frame with its AR field set to acknowledgment requested shall wait for the corresponding Ack frame to be received. If an Ack frame is received within the expected time 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 Ack frame is not received within the expected time or the Ack frame that is received contains 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 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 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.

If a single transmission attempt failed and the device is operating in TSCH mode, the retransmission process is defined in 6.2.5.3.

When not using TSCH mode and a frame with the Security Enabled field set to one is retransmitted, the frame shall be retransmitted without changes and without passing through the outgoing frame security procedure, as defined in 9.2.1.

When using TSCH mode, and a frame with the Security Enabled field set to one is retransmitted, the frame shall follow the outgoing frame security procedure, as defined in 9.2.1.

NOTE—In TSCH mode, the security processing needs to be performed again because the ASN is used in the nonce and the retransmitted frame is sent in a slot with a different ASN.

#### 6.7.5 Transmission timing restrictions

In beacon-enabled PANs, TSCH PANs, and DSME PANs, certain times may be allocated to devices for transmission. These times are referred to as the allowed transmission interval (ATI). Examples of these include the following:



- GTS: start and end time defined by GTS Info field in the beacon
- Beacon transmission: start time defined by the superframe duration in a beacon-enabled PAN
- TSCH timeslot: start and end time defined the *macSlotframeHandle*, *macTimeslot* and *macTsTimeslotLength*
- DSME timeslot: start and end time defined in the DSME GTS Response command

The transmission of frames and the associated Ack frames need to be completed at least one SIFS time before the end of the ATI. A SIFS is required prior to the end of the ATI for the device transmitting or receiving the final frame to switch to receiving or transmitting, respectively, in the event that the device is required to switch.

Devices shall also allow for the guard time, as defined in 6.7.6, when calculating the end of the ATI.

Accordingly, the following rules apply:

- 1) For frames that do not request acknowledgment, the transmitting device shall not send the frame if the time required for the frame transmission plus a SIFS is greater than the time remaining before the end of the allowed transmission time.
- 2) For frames that request acknowledgment, the transmitting device shall not send the frame if the time required for frame transmission plus the AIFS plus the minimum duration Ack frame plus a SIFS is greater than the time remaining before the end of the allowed transmission time. Note that the minimum required Ack frame duration is for the appropriate Ack frame (Imm-Ack or Enh-Ack) including only the required IEs (e.g., the Time Synchronization IE in TSCH mode), security overhead (if appropriate), and the SHR and PHR duration. If the SHR and/or PHR of a PHY is variable, the PHY defines a default SHR and PHR duration and that value shall be used in the calculation.

### 6.7.6 Guard time

Guard times are required to keep transmissions in adjacent ATIs, as defined in 6.7.5, from colliding. In addition, as described in 6.7.5, a SIFS time is required to ensure sufficient turnaround time between ATIs. Guard time is the time between the end of one ATI and the next ATI. Figure 6-36 is an illustration of the allocation of the guard time such that the ATIs are separated by at least a SIFS if the devices allowed to transmit in adjacent ATIs drift in time towards the each other.

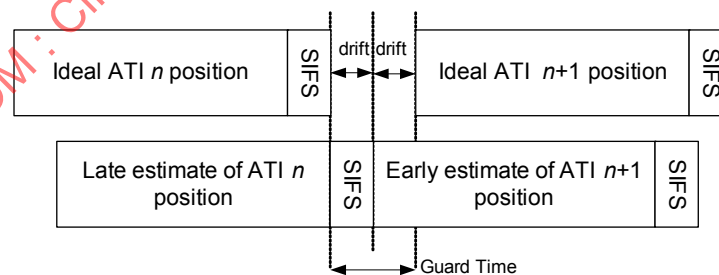


**Figure 6-36—Guard time**

The required guard time depends on the maximum drift between a device's local time and the ideal time. This drift is a function of the time elapsed since a synchronizing reference event. In a beacon-enabled PAN or DSME PAN the synchronizing event is the start of the preamble of a beacon. In a TSCH PAN the synchronizing event is a Timing Correction IE received from a timekeeping neighbor.

The maximum drift,  $MaxDrift$ , may be calculated as follows:

$$MaxDrift = [\text{Clock accuracy}] \times \text{time elapsed since the last synchronizing event}$$

NOTE—While propagation delay also affect timing uncertainty, for this standard it is ignored when calculating the guard time.

Figure 6-37 illustrates the example of a device transmitting frames with no acknowledgment requested including the guard time.

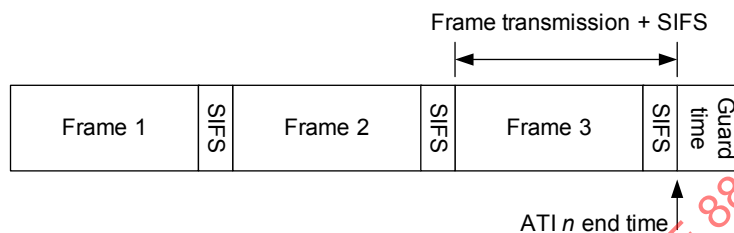


**Figure 6-37—SIFS and Guard time at the end of a ATI—no Ack requested**

Figure 6-38 illustrates the example of a device sending multiple frames with the Ack frame following an AIFS after the frame requesting the Ack frame including the guard time.

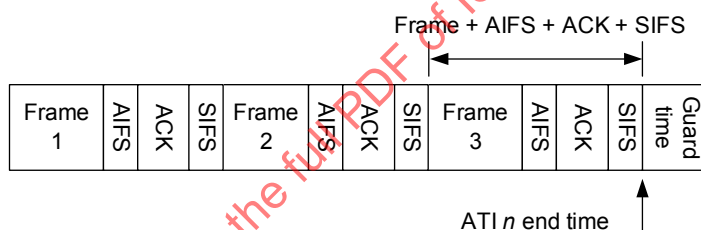


**Figure 6-38—AIFS, ACK, SIFS, and Guard time at the end of a ATI—Ack requested**

Figure 6-39 illustrates the example of a device transmitting frames just prior to scheduled beacon transmission including the guard time.

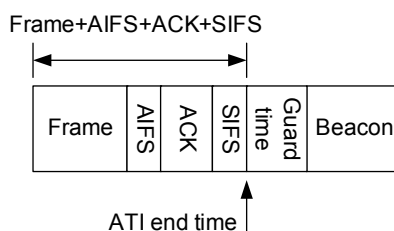


**Figure 6-39—Guard time at the end of a superframe**



### 6.7.7 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.

When in promiscuous mode, the MAC sublayer shall process received frames according to 6.7.2 and pass all frames correctly received to the next higher layer using the MCPS-DATA.indication primitive. The only valid parameters of the MCPS-DATA.indication primitive are Msdu, MpduLinkQuality, Timestamp and Rssi. The Msdu parameter shall contain the MHR concatenated with the MAC payload, as illustrated in Figure 7-1.

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*.

### 6.7.8 Transmission scenarios

Due to the imperfect nature of the radio medium, a transmitted frame does not always reach its intended destination. There are three different transmission scenarios:

- *Successful data transmission.* The originator MAC sublayer transmits the Data frame to the recipient via the PHY data service. 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 within the expected time. The data transfer is now complete, and the originator MAC sublayer issues a success confirmation to the next higher layer. This sequence of messages for successful data transmission is illustrated in Figure 6-40.

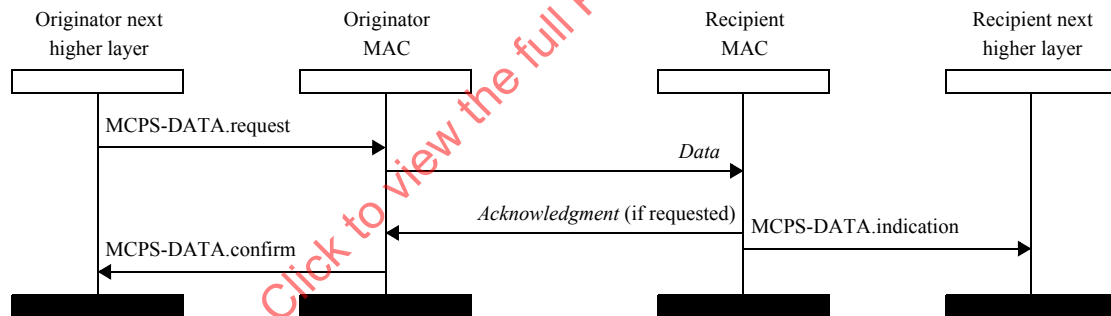


Figure 6-40—Successful data transmission sequence

- *Lost Data frame.* The originator MAC sublayer transmits the Data frame to the recipient via the PHY data service. The recipient MAC sublayer does not receive the Data frame and so does not respond with an acknowledgment. The acknowledgment is not received within the expected time and 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  $(1 + \text{macMaxFrameRetries})$  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. The sequence of messages for a lost Data frame is illustrated in Figure 6-41.

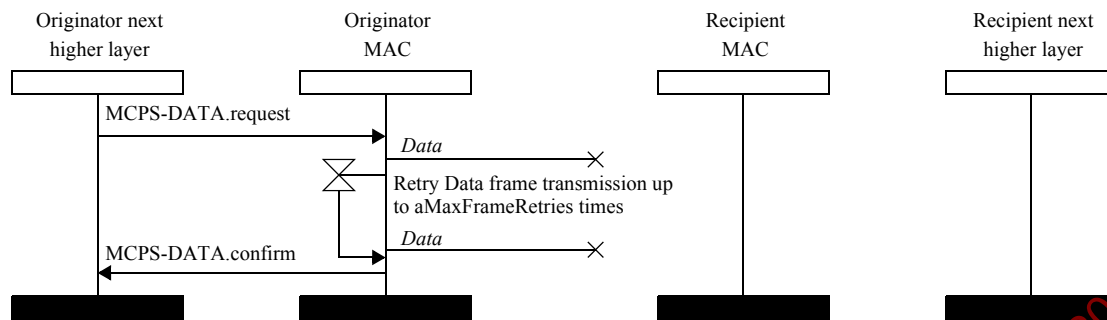


Figure 6-41—Lost Data frame message sequence

- *Lost Ack frame.* The originator MAC sublayer transmits the Data frame to the recipient via the PHY data service. 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 Ack frame within the expected time. Therefore, the data transfer has failed. If the transmission was direct, the originator retransmits the Data frame, and this entire sequence may be repeated up to a maximum of *macMaxFrameRetries* times. If a data transfer attempt fails a total of  $(1 + \text{macMaxFrameRetries})$  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. The message sequence for a lost Ack frame is illustrated in Figure 6-42.

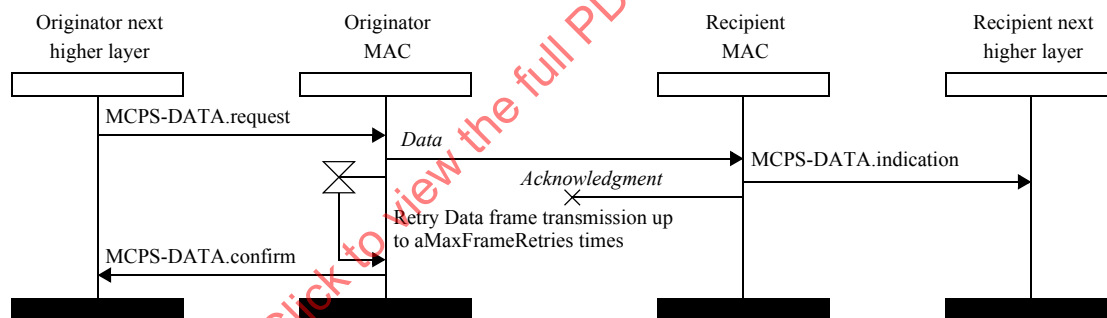


Figure 6-42—Lost acknowledgment message sequence

### 6.7.9 Device announcement

To facilitate data transfer effectively between two or more peer devices, a device announces its address and its neighbors' addresses to its neighbor devices by broadcasting beacons with a Device Announcement (DA) IE, as defined in 7.4.2.16. A DA IE shall only be sent in an Enhanced Beacon frame.

A device shall broadcast an Enhanced Beacon frame with a DA IE upon receiving a MLME-DA.request primitive, as defined in 8.2.24.1, from the next higher layer. It may also broadcast beacons with a DA IE at other times. After transmitting an Enhanced Beacon frame with the DA IE requested by an MLME-DA.request, the device shall send an MLME-DA.confirm, as described in 8.2.24.3, to the next higher layer.

Upon receiving an Enhanced Beacon frame with a DA IE, a device shall indicate the address of the transmitting device and the addresses list in the DA IE to its next higher layer using the MLME-DA.indication primitive 8.2.24.2. A device may check whether or not its address is known at the

transmitting device by tracking the received Enhanced Beacon frames with a DA IE. If not, the device may broadcast an Enhanced Beacon with a DA IE to announce its address at the appropriate time.

The message sequence chart for an Enhanced Beacon with a DA IE to announce the address of a device and its neighbors' addresses is illustrated in Figure 6-43.

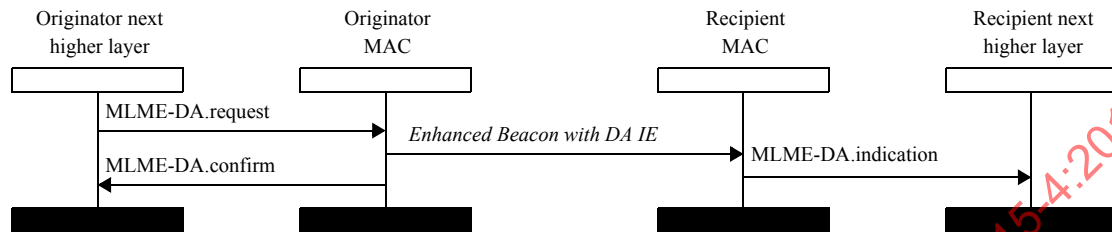


Figure 6-43—Sending DA IE message sequence chart

## 6.8 GTS allocation and management

### 6.8.1 GTS general requirements

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. The use of GTSs is optional.

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 command 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. 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 Ack frame as usual. Similarly, a device shall be able to receive an Ack 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.

### 6.8.2 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

### 6.8.3 GTS allocation

A device is instructed to request the allocation of a new GTS through the MLME-GTS.request primitive, as described in 8.2.7.1, 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, as described in 7.5.11, to the PAN coordinator. The Characteristics Type field of the GTS Characteristics field of the request shall be set to one (GTS allocation), and the length and direction fields shall be set according to the desired characteristics of the required GTS.

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*.

On receipt of the acknowledgment to the GTS Request command, the device shall continue to track beacons and wait for at most *aGtsDescPersistenceTime*. 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, as described in 8.2.7.2, 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 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 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, as described in 8.2.7.3, 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 field 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).

Figure 6-44 illustrates the message flow for the case in which the device requests the GTS allocation.

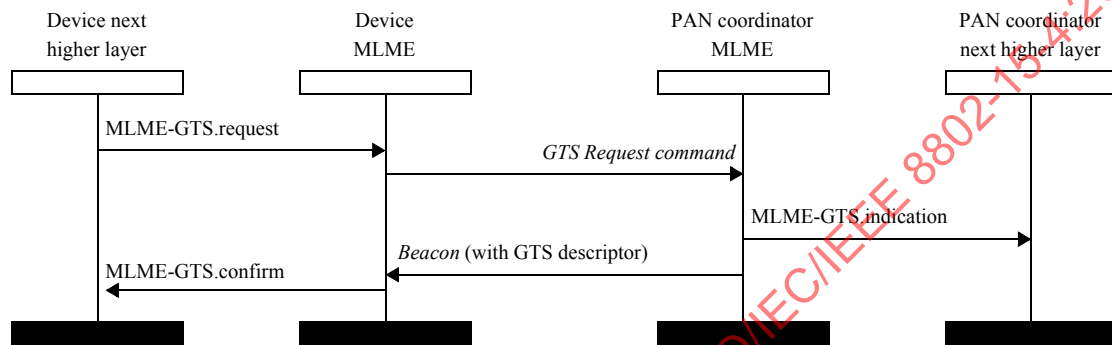


Figure 6-44—Message sequence chart for GTS allocation initiated by a device

#### 6.8.4 GTS usage

When the MAC sublayer of a device that is not the PAN coordinator receives an MCPS-DATA.request primitive, as described in 8.3.1, 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.

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 set of pending addresses in the Beacon frame, as described in 6.6. 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.

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.

#### 6.8.5 GTS deallocation

A device is instructed to request the deallocation of an existing GTS through the MLME-GTS.request primitive, as described in 8.2.7.1, 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, as described in 7.5.11, to the PAN coordinator. The Characteristics Type field of the GTS Characteristics field of the request shall be set to zero (i.e., GTS deallocation), and the length and direction fields shall be set according to the characteristics of the GTS to deallocate. 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, as described in 8.2.7.2, with a Status of SUCCESS and a GTSCharacteristics parameter with its Characteristics Type field 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 6.8.7.

On receipt of a GTS Request command with the Characteristics Type field 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, as described in 8.2.7.3, with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type field set to zero. The PAN coordinator shall also update the Final CAP Slot field 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, as described in 6.8.7, or maintenance required to maintain the minimum CAP length, *aMinCapLength*, as described in 6.8.2.

The next higher layer of the PAN coordinator initiates a GTS deallocation using an MLME-GTS.request primitive with the GTS Characteristics field of the request set to indicate a GTS deallocation and the length and direction fields set according to the characteristics of the GTS to deallocate. The MLME shall then respond with an MLME-GTS.confirm primitive with a Status of SUCCESS and the GTSCharacteristics parameter with a Characteristics Type field set to zero.

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 using the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type field 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.

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 using the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type field set to zero.

Figure 6-45 depicts the message flow for the cases in which a GTS deallocation is initiated by a device.

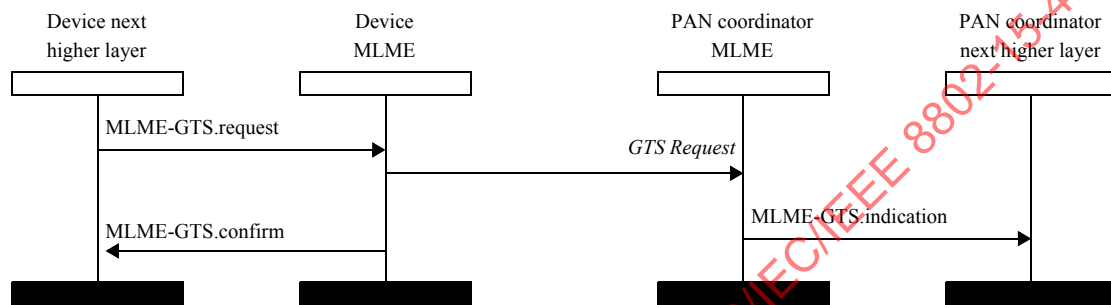


**Figure 6-45—Message sequence chart for GTS deallocation initiated by a device**

Figure 6-46 depicts the message flow for the cases in which a GTS deallocation is initiated by the PAN coordinator.

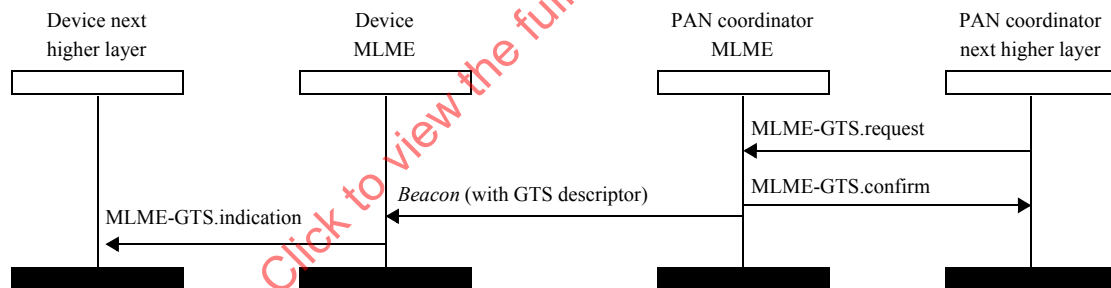


**Figure 6-46—Message sequence chart for GTS deallocation initiated by the PAN coordinator**

### 6.8.6 GTS reallocation

The deallocation of a GTS may result in the superframe becoming fragmented. For example, Figure 6-47 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).

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.

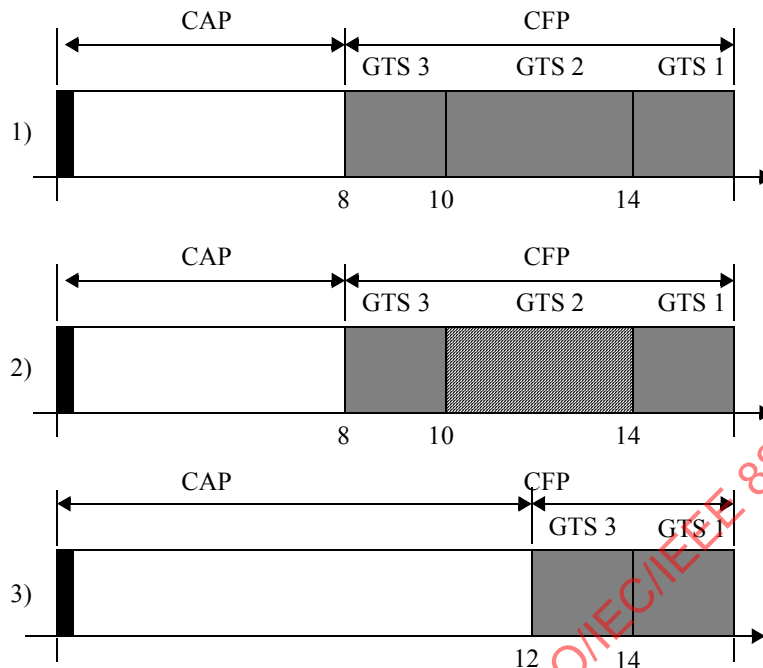


Figure 6-47—CFP defragmentation on GTS deallocations

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.

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.



### 6.8.7 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  $2 \times n$  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 Ack frame is not received from the device at least every  $2 \times n$  superframes, where  $n$  is defined below. If the Data frames sent in the GTS do not require Ack 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:

$$n = 2^{(8 - \text{macBeaconOrder})} \quad 0 \leq \text{macBeaconOrder} \leq 8$$

$$n = 1 \quad 9 \leq \text{macBeaconOrder} \leq 14$$

## 6.9 Ranging

### 6.9.1 Ranging requirements

Ranging is an optional feature. The fundamental measurements for ranging are achieved using a Data frame Ack frame sequence. The Data frame has the Ranging field set to indicated ranging and is referred to as a ranging frame (RFRAME). Ranging capabilities are enabled in a ranging-capable device (RDEV) with the MCPS-DATA.request primitive. Whenever ranging is enabled in an RDEV, the RDEV delivers timestamp reports to the next higher layer as a result of events at the device antenna. The timestamp that is reported is measured relative to the RMARKER. For all PHYs the RMARKER is defined to be the time when the beginning of the first symbol of the PHR of the RFRAME is at the local antenna.

### 6.9.2 Set-up activities before a ranging exchange

The mandatory part of ranging is limited to the generation of timestamp reports during the period that ranging is enabled in an RDEV. It is possible that an RDEV will consume more power when ranging is enabled; therefore, a natural default for an application would be to have ranging disabled. Prior to a two-way ranging exchange, both RDEVs involved in the exchange shall already have ranging enabled. Furthermore, if the optional dynamic preamble selection (DPS) capability is to be used, there shall have been some sort of coordination of preambles prior to the two-way ranging exchange. How this coordination and enabling actually is accomplished is beyond the scope of this standard. It may be perfectly acceptable to accomplish the coordination and enabling with a clock and a look-up table that says what a device should do at a particular time. Because coordination generally involves communication and because the PHYs are designed to achieve communication, it is natural to suggest that the PHY be used for coordination.

### 6.9.3 Finish-up activities after a ranging exchange

At the end of a two-way exchange, each device is in possession of a timestamp report. To accomplish anything useful, both of those timestamp reports shall eventually come to be at the same node where computations are performed. How this movement of timestamp reports is accomplished is beyond the scope of this standard. Timestamp reports are just data. Because movement of data involves communication and because the PHYs are designed to achieve communication, it is natural to suggest that the PHY be used for the final consolidation of timestamp reports.

The application is responsible for enabling the ranging mode in the RDEV before a ranging exchange. After a ranging exchange, the application is again responsible for disabling the ranging mode in the RDEV. If the application fails to disable the ranging mode in the RDEV, there will be no algorithmic harm. Ranging mode is fully compatible with other uses of the RDEV, and the only result of leaving ranging enabled when it is not really being used is that the RDEV will generate useless timestamp reports while potentially consuming more power.

#### 6.9.4 Managing DPS

Figure 6-48 shows a suggested message sequence for two-way ranging. The messages represented in the two top dotted boxes are simply suggestions showing how the communications capability of the RDEV can be used to accomplish the ranging setup activities. The messages in the bottom dotted box are suggestions showing how the communications capability of the RDEV can be used to accomplish the ranging finish-up activities.

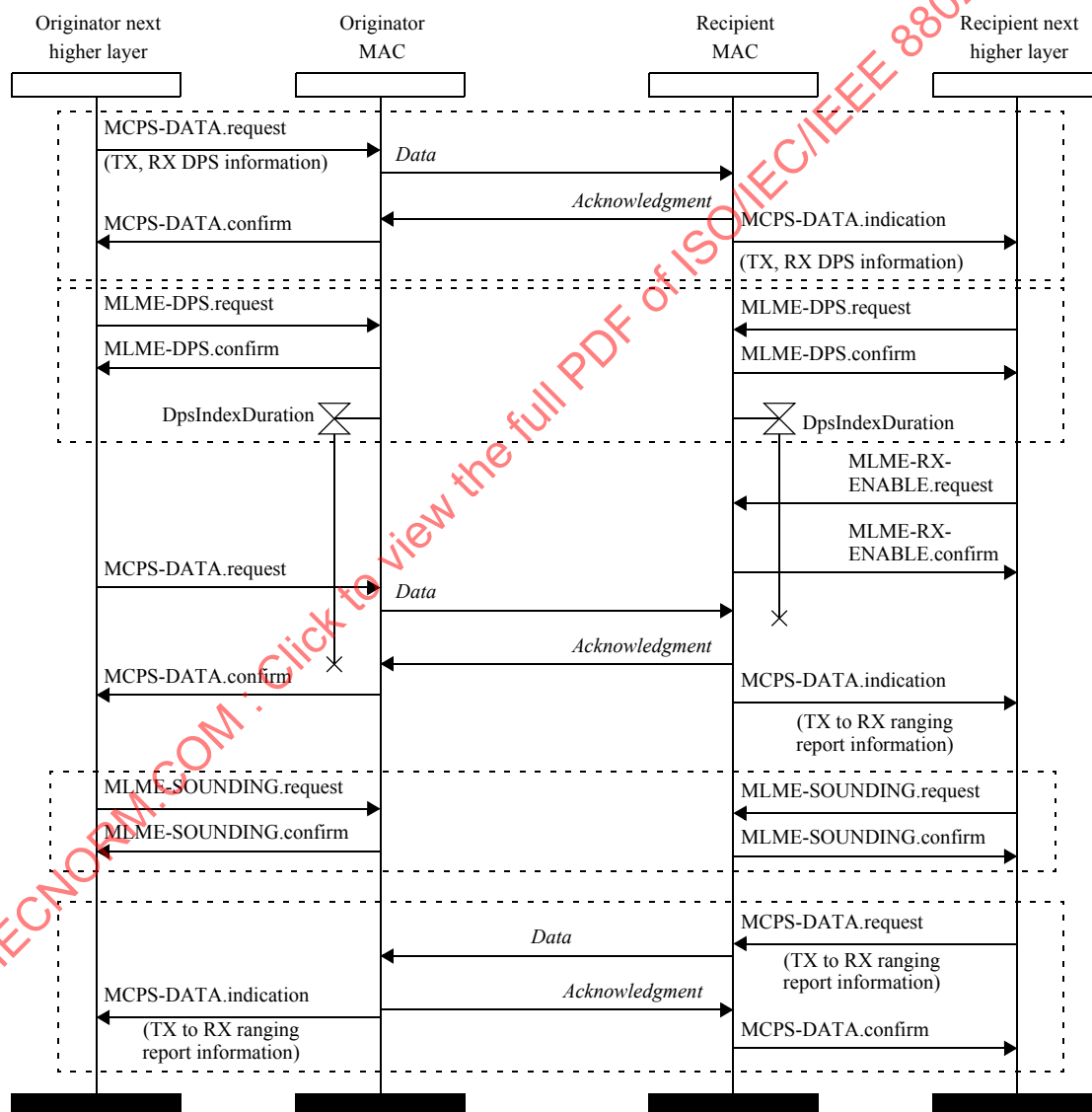


Figure 6-48—A message sequence for ranging

The top dotted box in Figure 6-48 illustrates the use of a data exchange to effect the coordination of the preambles to be used for a two-way ranging exchange. The coordination of preambles is needed only when using the optional DPS capability of the PHY. If optional DPS is not used, the communication sequence in the top box can be thought of as arranging for the recipient RDEV to become aware that a ranging exchange is desired and that the recipient next higher layer should enable ranging in the recipient PHY. The second from the top dotted box in Figure 6-48 illustrates the use of the MLME-DPS.request, as described in 8.2.15.1, and the MLME-DPS.confirm, as described in 8.2.15.2. Use of these primitives is unique to the optional DPS mode of ranging.

Upon the assertion of the MLME-DPS.confirm primitives, as illustrated in Figure 6-48, both of the PHYs have switched from the normal length preamble symbols to long preamble symbols. This is desirable behavior intended to help hide the PHYs' transmissions from malicious nodes and protect the PHYs from transmissions by malicious nodes. A side effect of this mode is that neither PHYs can communicate with the rest of the network. To prevent the PHYs from becoming lost as a result of this optional behavior, the MAC sublayers on both sides of the link shall initiate timers after sending the frame (for the originator) or receiving the frame (for the recipient). If the timer duration is exceeded before the MAC sublayer receives the MCPS-DATA.confirm (for the originator) or the MCPS-DATA.indication primitive (for the recipient), then the MAC sublayer shall initiate a MLME-DPS.indication to the next higher layer as described in 8.2.15.3. Not shown in Figure 6-48, one responsibility of the application, if the optional DPS capability is used, is to initiate the MLME-DPS.request primitive on both sides of the ranging link at the completion of the ranging exchange. Most typically, this MLME-DPS.request primitive would be part of the finish-up activities and would have both *TxDpsIndex* and *RxDpsIndex* set to zero to return the PHYs to using *phyCurrentCode* from the PIB. Also not shown in Figure 6-48, another responsibility of the application is to initiate a MLME-DPS.request primitive in response to an MLME-DPS.indication. Most typically, this MLME-DPS.request primitive would have both *TxDpsIndex* and *RxDpsIndex* set to zero and return the PHY to using *phyCurrentCode* from the PIB.

### 6.9.5 The ranging exchange

The essential core of the ranging exchange is shown in Figure 6-48 starting just below the just after the MLME-DPS exchange. The application is responsible for initiating the MLME-RX-ENABLE.request primitive, as described in 8.2.10.1, with *RangingRxControl* equal to *RANGING\_ON*. Once the RDEV has received the MLME-RX-ENABLE.request primitive with *RangingRxControl* equal to *RANGING\_ON*, all future RFRAMES received by the RDEV shall generate timestamp reports until ranging is disabled.

At the initiator, the application is responsible for initiating a MCPS-DATA.request primitive with *Ranging* equal to *ALL\_RANGING*. Upon receipt of a MCPS-DATA.request primitive with *Ranging* equal to *ALL\_RANGING*, RDEV shall generate timestamp reports for all RFRAMES after the transmit frame is transmitted. The timestamp reports will continue until ranging is disabled. The TX-to-RX turnaround enabling the originator to receive the Ack frame is necessary and is not shown in Figure 6-48. This turnaround is the normal turnaround that is done for any exchange expecting an acknowledgment. The turnaround happens without any action required by the originator next higher layer. Timestamp reports are generated to the next higher layer independent of the state of the AR field in the MAC header of received RFRAMES.

As shown in Figure 6-48, the first timestamp report to the originator next higher layer shall come back as elements of the MCPS-DATA.confirm. The first timestamp report to the recipient next higher layer shall come back as elements of the MCPS-DATA.indication primitive. All subsequent timestamp reports on either side of the link shall come back as elements of MCPS-DATA.indication primitives. The potential additional MCPS-DATA.indication primitives that would be due to unexpected stray RFRAMES are not shown in Figure 6-48 for simplicity. The timestamp reports due to any strays shall continue until ranging is disabled. The reporting of timestamps for a stream of "strays" is the behavior that enables the RDEV to be used as an infrastructure RDEVs in one-way ranging applications. One-way ranging is described in "Applications of IEEE Std 802.15.4" [B3].

For non-TVWS RDEVs, the timestamp is defined in 16.7. Use of nonzero timestamp reports is limited to RDEVs. Only devices that have *phyRanging* set to TRUE shall return a nonzero timestamp report to a next higher layer.

For TVWS RDEVs, the Timestamp IE, 7.4.4.26, and the Timestamp Difference IE, 7.4.4.27, are provided for exchanging timing information between TVWS RDEVs to support the ranging feature.

## 6.10 PHY parameter change notification procedure

### 6.10.1 Signaling using Beacon frames

This method is initiated by the reception of the MLME-PHY-OP-SWITCH.request primitive with the SignalMethod parameter value set to USE\_BEACON. The method requires that the Enhanced Beacon frames are supported and that the device is the PAN coordinator using Enhanced Beacon frames. If these conditions are not met, the device shall respond with the MLME-PHY-OP-SWITCH.confirm primitive having the appropriate Status parameter value indicating the reason for the request failure.

A PHY Parameter Change IE, as defined in 7.4.4.13, shall be generated and inserted in the next outgoing Enhanced Beacon frame. The Effective Time of Change field of the IE shall be set to the value of the TargetTime parameter of the MLME-PHY-OP-SWITCH.request primitive. The Notification Time field shall be updated with the local time of the device each time it is transmitted. The appropriate operating mode description IE shall be generated according to the values in the PhyParameterList and inserted in the same Enhanced Beacon frame following the PHY Parameter Change IE.

If the value of the RepeatCount parameter of the MLME-PHY-OP-SWITCH.request primitive is nonzero, then the generated IEs shall be included in each Enhanced Beacon frame subsequently generated until the repeat count is exhausted or until the value in the TargetTime parameter has elapsed. If the RepeatCount parameter is zero, the generated IEs shall be included in only the next Enhanced Beacon frame.

### 6.10.2 Signaling using multipurpose frames

This method is initiated by the reception of the MLME-PHY-OP-SWITCH.request primitive with the SignalMethod parameter value set to USE\_MP. The method requires that the multipurpose frame, as defined in 7.3.5, is supported. If this condition is not met, the device shall respond with the MLME-PHY-OP-SWITCH.confirm primitive having the appropriate Status parameter value.

The device shall generate a PHY Op Mode Switch frame, which is a Multipurpose frame containing a PHY Parameter Change IE and an Operating Mode Description IE. The addressing fields shall be set according to the DeviceAddrMode and DeviceAddr parameter values in the MLME-PHY-OP-SWITCH.request primitive. If the DeviceAddress parameter contains the broadcast address, then only the PAN ID addressing field shall be included, and it shall be set to the broadcast PAN ID. The PHY Parameter Change IE and an Operating Mode Description IE shall be generated using the TargetTime, RepeatCount, and PhyParameterList parameters, as described in 6.10.1.

For a directed PHY Op Mode Switch frame, the frame shall be generated with the AR field in the MHR set to request an acknowledgment and transmitted according to 6.7.

If the RepeatCount parameter value is greater than zero, the MLME shall repeat transmission of the frame after a delay equal to the value of the RepeatInterval parameter until the RepeatCount parameter value is exhausted. The Notification Time field shall be updated with the local time of the device each time it is transmitted.

When the TxIndirect parameter is set to TRUE, the PHY Op Mode Switch frame shall be sent using indirect transmission, as described in 6.6.

## 6.11 Deterministic and synchronous multi-channel extension (DSME)

### 6.11.1 DSME command requirements

An FFD device in a DSME-enabled PAN shall be capable of transmitting and receiving the following MAC Command frames:

- DSME Association Request command, as defined in 7.5.12
- DSME Association Response command, as defined in 7.5.13
- DSME GTS Request command, as defined in 7.5.14, which shall only be sent in the CAP
- DSME GTS Response command, as defined in 7.5.15, which shall only be sent in the CAP
- DSME GTS Notify command, as defined in 7.5.16, which shall only be sent in the CAP
- DSME Information Request command, as defined in 7.5.17
- DSME Information Response command, as defined in 7.5.18
- DSME Beacon Allocation Notification command, as defined in 7.5.19
- DSME Beacon Collision Notification command, as defined in 7.5.20
- DSME Link Report command, as defined in 7.5.21

### 6.11.2 DSME multi-superframe structure

A coordinator in a DSME-enabled PAN shall periodically transmit an Enhanced Beacon frame with DSME PAN Descriptor IE, as described in 7.4.2.5, to coordinate a DSME multi-superframe structure. A multi-superframe is a cycle of repeated superframes. A superframe consists of a Beacon frame, a CAP, and a CFP.

The structure of this multi-superframe is described by the values of *macBeaconOrder*, *macSuperframeOrder*, and *macMultisuperframeOrder*.

The MAC PIB attribute *macMultiSuperframeOrder* describes the length of a multi-superframe. The value of *macSuperframeOrder*, SO, and the SD are related as follows: for  $0 \leq SO \leq BO \leq 14$ ,  $SD = aBaseSuperframeDuration \times 2^{SO}$ .

The MAC PIB attribute *macMulti superframeOrder* describes the length of a multi superframe, which is a cycle of repeated superframes. The value of *macMultisuperframeOrder*, MO, and the MD are related as follows: for  $0 \leq SO \leq MO \leq BO \leq 14$ ,  $MD = aBaseSuperframeDuration \times 2^{MO}$ . The value of *macMultisuperframeOrder* shall be ignored if *macBeaconOrder* = 15.

Each superframe shall be divided into *aNumSuperframeSlots* equally spaced slots of duration  $aBaseSlotDuration \times 2^{SO}$  and is composed of three parts: an enhanced beacon, a CAP, and a CFP. Enhanced Beacon frames and other frames transmitted during CAP shall be transmitted using the channel number used in the successful association or start. Frames during CFP shall be transmitted using the assigned channel for DSME GTS.

Enhanced beacons shall be transmitted, without the use of CSMA-CA, at the start of slot 0 if *macDeferredBeaconUsed* is FALSE. The start of slot 0 is defined as the point at which the first symbol of the beacon PPDU is transmitted. If *macDeferredBeaconUsed* is TRUE, enhanced beacons shall be transmitted following the procedure described in 6.11.8.

The CAP shall commence immediately following the beacon and ends before slot 9. The CFP follows immediately after the CAP and extends to the end of the superframe. Any allocated DSME GTSs shall be located within the CFP.

An example of a multi-superframe structure is shown in Figure 6-49.

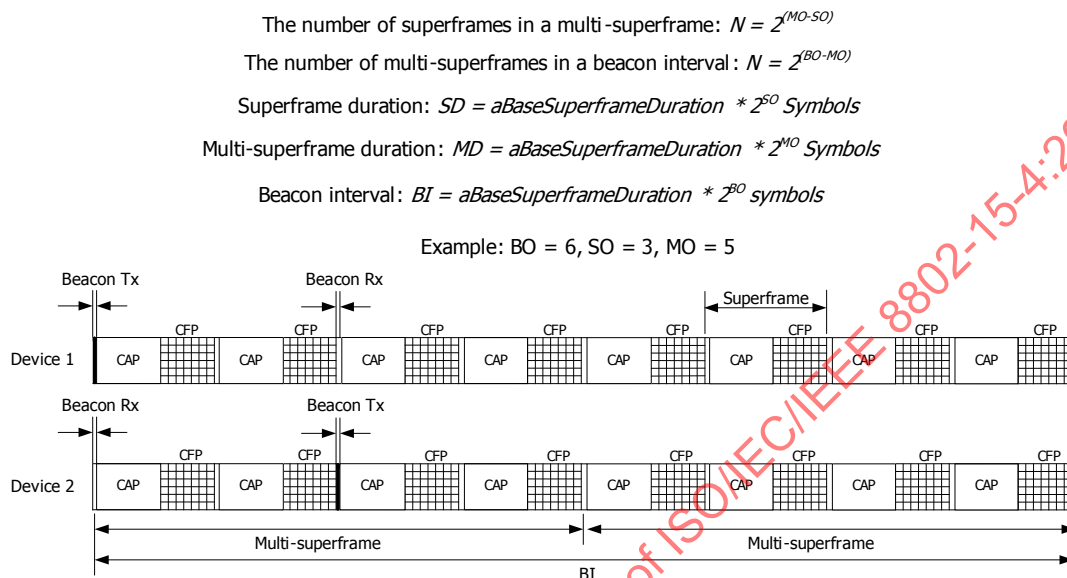


Figure 6-49—DSME multi-superframe structure

When *macExtendedDsmeEnabled* is TRUE, the Extended DSME PAN Descriptor IE shall be included in Enhanced Beacon frames that are sent every beacon interval in an Extended DSME-enabled PAN.

When *macExtendedDsmeEnabled* is TRUE, the values of the multi-superframe order, MO, the superframe order, SO, and the beacon order, BO, are related as follows:

$$SO \leq MO \leq [BO + \text{len}(\text{BSN})]$$

where  $\text{len}(\text{BSN})$  is 8 when the Enhanced Beacon frame contains the Sequence Number field and 0 when the Sequence Number field is not present.

Because the value of MO may be larger than that of BO, there may be multiple beacon intervals BIs within an MD. An example of a multi-superframe structure with the value of MO larger than that of BO is shown in Figure 6-50.

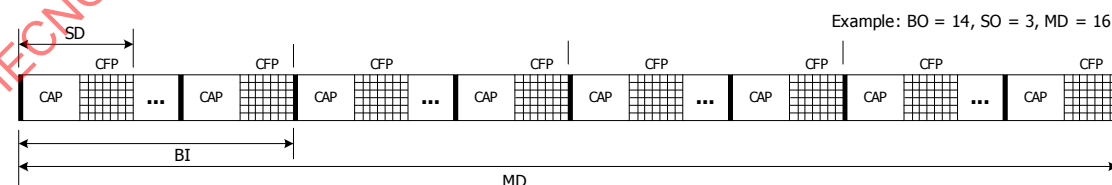


Figure 6-50—Example of DSME multi-superframe structure ( $MO > BO$ )

### 6.11.3 Channel diversity

DSME GTS service is provided in one of the two channel diversity methods, namely, Channel Adaptation and Channel Hopping.

#### 6.11.3.1 Channel adaptation

In channel adaptation mode, the Source device may allocate DSME GTSs in a single channel or in different channels to a destination device based on the knowledge of current channel quality. If DSME GTSs in different slots in different channels are successfully allocated for a pair of a source device and a destination device, the source device shall transmit Data frames according to the scheduled timeslots and channels specified in *macDsmeAct*.

An example of the schedule of channels and DSME GTSs in channel adaptation mode is illustrated in Figure 6-51. In this example, the device uses channel 11 from slot 0 to 2, and then it switches to channel 13 on slot 5.

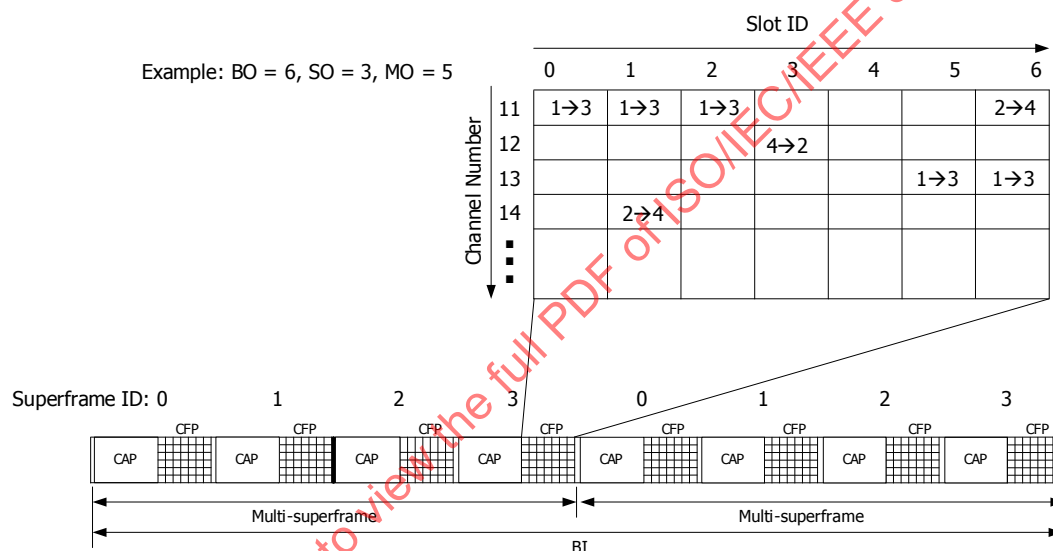


**Figure 6-51—Channel usage of DSME GTSs in Channel Adaptation**

If the link quality of allocated DSME GTSs has degraded, it is recommended that the DSME GTSs should be deallocated and new DSME GTSs with better link quality should be allocated.

Figure 6-52 illustrates the sequence of messages necessary for DSME link report initiated by a source device. The message sequence of DSME link report initiated by destination device is the same as sequence initiated by source device.

In Channel Adaptation mode, the SAB Sub-block is as illustrated in Figure 6-53.



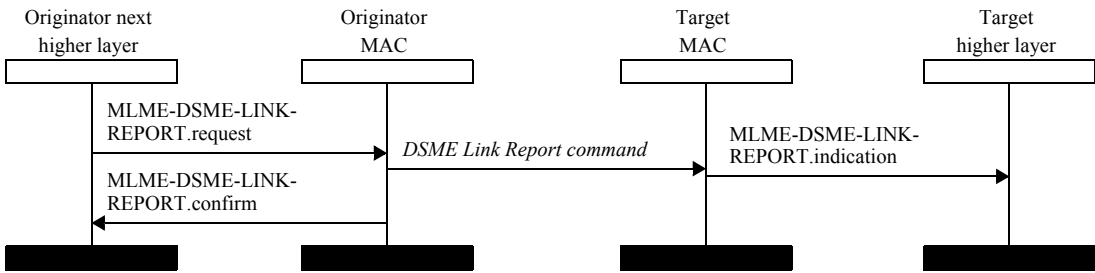


Figure 6-52—Message sequence chart for DSME link report

DSME-GTS multi-superframe slot identifier  
= (channel, time slot)

Column = channel  
Row = timeslot  
0 = vacant  
1 = occupied

If CAP reduction is off,  
SAB sub-block unit = 14 octets  
(7 timeslots × 16 channels)

If CAP reduction is on,  
SAB sub-block unit = 30 octets  
(15 timeslots × 16 channels)

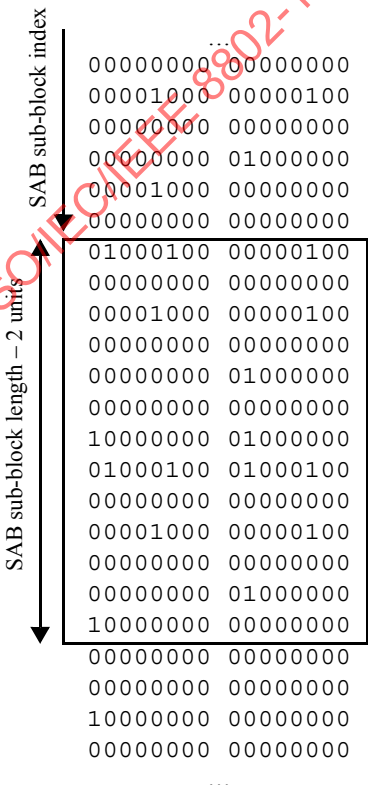


Figure 6-53—SAB sub-block in channel adaptation

6.11.3.2 Channel hopping

In channel hopping mode, each DSME GTS hops over predefined frequency channels to receive. The series of channels used at each DSME GTS is referred to as a hopping sequence. The same hopping sequence shall be repeated over whole DSME GTSS.

An example of the schedule of channels and DSME GTSS in channel hopping mode is illustrated in Figure 6-54.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

Example: BO = 6, SO = 3, MO = 5

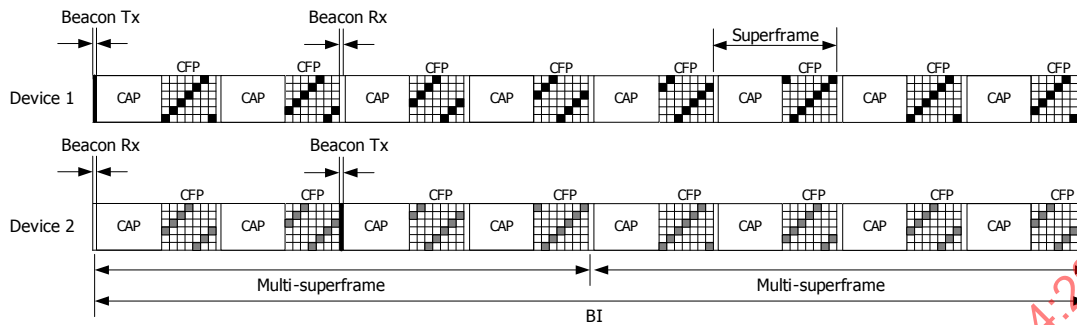


Figure 6-54—Channel usage of DSME GTSSs in Channel Hopping

In this example, hopping sequence is {1, 2, 3, 4, 5, 6} and the channel offset values of two devices are 0 and 2, respectively. For the device with channel offset value of 0, DSME GTSSs (timeslot, channel) for this device are (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 1), (8, 2), (9, 3), and so on. Similarly, for the device with channel offset value of 2, DSME slots are given as (1, 3), (2, 4), (3, 5), (4, 6), (5, 1), (6, 2), and so on. The transmitting device shall switch to the channel used by the receiving device in order to send a Data frame. If the receiving device receives the Data frame successfully, it sends an Enh-Ack frame to the transmitting device on the same channel. The transmission of an Enh-Ack frame shall commence between AIFS and (AIFS + aUnitBackoffPeriod) after the reception of the last symbol of the Data frame.

Channel number  $C$  at the given DSME GTS Slot ID  $i$  in SdIndex  $j$ , shall be determined as follows:

$$C(i) = \text{macHoppingSequenceList}[(j \times l + i + \text{macChannelOffset} + \text{macPanCoordinatorBsn}) \% \text{macHoppingSequenceLength}]$$

where  $l$  is 15 if *macCapReduction* is TRUE and SdIndex  $j$  is not zero, and 7 otherwise, *macHoppingSequenceList*[ $m$ ] represents the ( $m$ )<sup>th</sup> channel number in *macHoppingSequenceList*, *macChannelOffset* is the channel offset value of the receiver device, *macHoppingSequenceLength* is the length of Hopping Sequence and *macPanCoordinatorBsn* is an EBSN of a PAN coordinator.

In channel hopping mode, the SAB represents the usage of corresponding DSME GTS, a bit shall be set to one if the corresponding slot is allocated to transmit or receive, or zero if the slot is available. Similarly in channel adaptation, DSME SAB sub-block Index and DSME SAB Sub-block length shall indicate the start position and the length of SAB Sub-block. Thus, only a sub-block of whole SAB is exchanged for scheduling. This is illustrated in Figure 6-55.

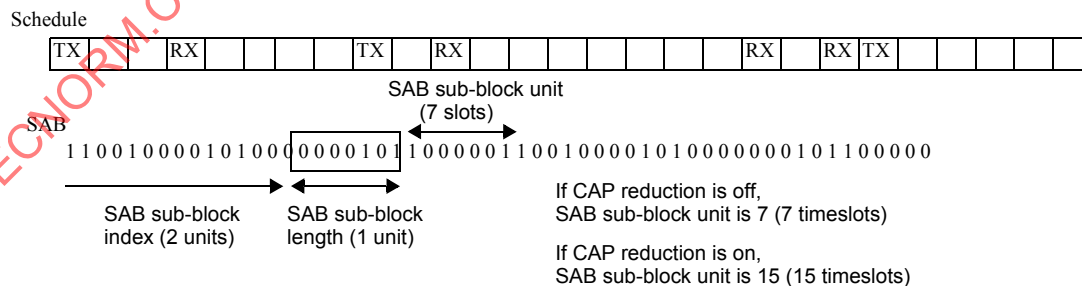


Figure 6-55—SAB sub-block in channel hopping

### 6.11.4 CAP reduction

If *macCapReduction* is TRUE, then CAP reduction shall be enabled. When the CAP reduction is enabled, the first superframe (superframe ID 0) in the multi-superframe has the CAP and all other superframes do not have a CAP.

Figure 6-56 shows an example of the multi-superframe structure when CAP reduction is enabled. As shown in Figure 6-56, every device has the CAP at the same time, which is the first superframe in a multi-superframe. The devices do not have the CAP in all other superframes.

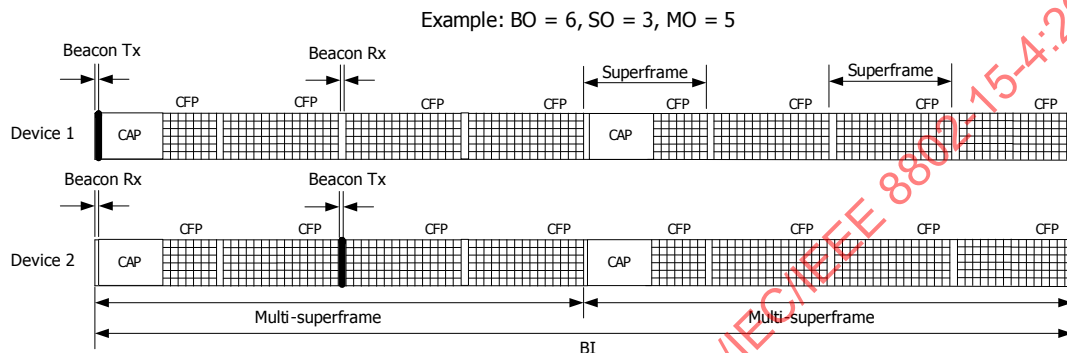


**Figure 6-56—CAP reduction in DSME multi-superframe structure**

### 6.11.5 DSME GTS allocation and management

DSME GTS functionality allows a DSME device to operate on the channel within a portion of the superframe that is dedicated (on the PAN) exclusively to that device. A DSME GTS shall be allocated by the destination device, and it shall be used only for communications between the source device and the destination device.

A DSME GTS allows a pair of device to exchange a data frame and an Ack frame. A DSME GTS shall be allocated between a pair of devices prior to its use. Allocation of a DSME GTS is initiated by a source device by transmitting a DSME GTS Request command during a CAP. The destination device determines if a DSME GTS will be allocated based on the requirements of the DSME GTS Request command and the current slot availability as described in 6.11.5.1.

DSME GTSs shall be allocated on a first-come-first-served basis, and all DSME GTSs shall be placed contiguously at the end of the superframe and after the CAP (or after the beacon slot if CAP reduction is enabled). Each DSME GTS shall be deallocated when the DSME GTS is no longer required, and a DSME GTS may be deallocated at any time by the destination device or the source device that originally requested the DSME GTS.

A Data frame transmitted in an allocated DSME GTS shall use short addressing.

The management of DSME GTSs shall be undertaken by both the destination device and the source device. To facilitate DSME GTS management, the destination device and the source device shall be able to store all the information necessary to manage DSME GTSs. For each DSME, the destination device and the source device shall be able to store its starting slot, length, and associated device address.

If a device has been allocated a DSME GTS for the data reception, it shall enable its receiver for the entirety of the DSME GTS. If a Data frame is received during a DSME GTS and an acknowledgment is requested, the destination device shall transmit the Enh-Ack frame as usual. Similarly, the source device shall be able to receive an Enh-Ack frame during the DSME GTS it requested.

**6.11.5.1 DSME GTS allocation**

A DSME-enabled device is instructed to request the allocation of new DSME GTSs through the MLME-DSME-GTS.request primitive with the ManagementType parameter set to ALLOCATION.

On receipt of the primitive, the device shall send a DSME GTS Request command, as described in 7.5.14, to the device identified by the DeviceAddress parameter of the primitive. The DSME SAB Sub-block field shall contain a bitmap of a sub-block of *macDsmeSab*. The values of the DsmeSabSubBlockIndex parameter and DsmeSabSubBlockLength parameter indicate the sub-block that shall be contained in the command.

If the device successfully transmits the DSME GTS Request command and an acknowledgment is not received, the device shall issue the MLME-DSME-GTS.confirm primitive with a Status of NO\_ACK.

If no DSME GTS Response command is received within the expected time, the device shall notify the higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a Status of NO\_DATA.

On receipt of a DSME GTS Request command with the Management Type set to ALLOCATION, the device shall issue an MLME-DSME-GTS.indication to the higher layer. The higher layer will make the decision on the allocation using the value of DsmeSabSpecification parameter as the slot availability information. It is recommended that the preferred superframe ID and the preferred slot ID are considered in allocating slots. If the preferred slot is not available, the next slot may be used. Following this approach, slot allocation of a multihop flow can be allocated sequentially on each hop to reduce the end-to-end delay. After making the allocation decision, next higher layer issues an MLME-DSME-GTS.response primitive. The DsmeSabSpecification parameter indicates the newly allocated slots.

On receipt of MLME-DSME-GTS.reponse primitive with Status parameter value of SUCCESS, the device shall send a DSME GTS Response command to the requesting device with the destination address set to the broadcast short address and the DSME GTS destination address set to the short address of the requesting device. The Management Type field shall be set to indicate allocation. The Status field in the command shall be set to indicate approved. The DSME SAB Specification field shall be set to the same as the DsmeSabSpecification parameter of the primitive.

On receipt of MLME-DSME-GTS.reponse primitive with Status parameter value of DENIED or INVALID\_PARAMETER, the device shall send a DSME GTS Response command to the requesting device. The Management Type field shall be set to allocation. The Status field shall be set to either disapproved: lack of availability or disapproved: unknown GTS. The DSME SAB Specification field shall be set to the same as the DsmeSabSpecification parameter of the primitive.

On receipt of a DSME GTS Response command with management type allocation, the device shall check the DSME GTS Destination Address field of the received command.

If the DSME GTS Destination address is the same as the *macShortAddress*, the device shall inform the next higher layer of the result using MLME-DSME-GTS.confirm. If the value of the Status field in the command is approved, the device shall broadcast a DSME GTS Notify command with the destination address set to the broadcast short address, and the DSME GTS destination address is set to the short address of the source device of the received command. The Management Type field shall be set to allocation. The DSME SAB Specification field shall be set to the same value as the DSME SAB Specification field in the received command. Also, the device shall update *macDsmeAct* according to the DSME SAB Specification field in the received command.

If the DSME GTS destination address of the DSME GTS Response command is not the same as the *macShortAddress*, the device shall check if the slots marked as one in the command is conflicting with the readily allocated slots in *macDsmeAct*. If there is no conflict, the device shall update *macDsmeSab*

according to the DSME SAB Specification field in this command to reflect the neighbor's newly allocated DSME GTSs. If there is a conflict, the device shall send a DSME GTS Request command with Management Type field set to duplicated allocation notification to the source device of the received command.

On receipt of a DSME GTS Notify command with management type one allocation, the device shall check the DSME GTS Destination Address field of the received command. If the DSME GTS Destination address is the same as the *macShortAddress*, the device shall notify the next higher layer of the receipt of the DSME GTS Notify command using MLME-COMM-STATUS.indication. The device shall update *macDsmeAct* according to the value of DSME SAB Specification field of the DSME GTS Notify command.

If a DSME GTS Notify command is not received at the destination device within the expected time, the device shall notify the higher layer of the failure by the MLME-COMM-STATUS.indication primitive with a Status of TRANSACTION\_EXPIRED.

If the DSME GTS destination address of the DSME GTS Notify command is not the same as the *macShortAddress*, the device shall check if the slots marked as one in this command is conflicting with the readily allocated slots in *macDsmeAct*. If there is no conflict, the device shall update *macDsmeSab* according to the DSME SAB Specification field to reflect the neighbor's newly allocated DSME GTSs. If there is a conflict, the device shall send a DSME GTS Request command with the Management Type field set to duplicated allocation notification to the source of the received command.

On receipt of a DSME GTS Request command indicating a DSME GTS duplicate allocation notification, the device shall deallocate the duplicated DSME GTSs as described in 6.11.5.2.

Figure 6-57 depicts the message flow for the case in which a Originator device initiates the DSME GTS allocation, deallocation, duplicated allocation notification, reduce, or restart.

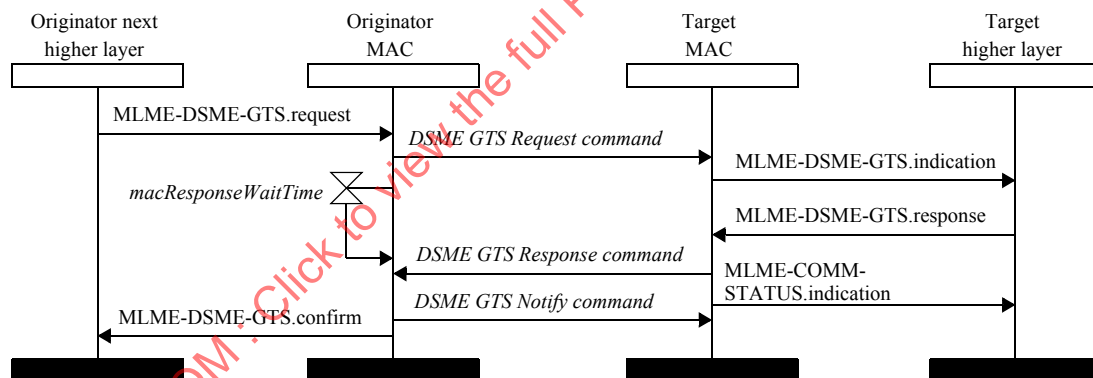


**Figure 6-57—Message sequence chart for DSME GTS allocation and management**

An example of a DSME GTS allocation, consider device 3 is requesting 2 slots to transmit data to device 1. Assuming time slot 0, channel 15 is already assigned by device 4 for transmitting data from device 4 to device 3 and channel 19 is a bad channel, then the messages would be as follows:

- 1) Device 3 sends a unicast DSME GTS Request command to device 1 with 2 slots request, Direction field set to TX, Preferred Superframe ID field set to 1, and the SAB Sub-block field set as follows:
 

```
{0000100010000000
0000000010000000
...
0000000010000000}
```

- 2) Device 1 sends a broadcast DSME GTS Reply command with the destination address field in the command set to that of device 3, Direction field set to TX, Superframe ID set to 1, and a newly allocated SAB Sub-block field set as follows:

```
{0000000000000000
0000001000000000
0000001000000000
...
0000000000000000}
```

- 3) Device 3 sends a broadcast DSME GTS Notify command with the Destination Address field in the command set to that of device 1, the Direction field set to TX and a newly allocated SAB Sub-block field set as follows:

```
{0000000000000000
0000001000000000
0000001000000000
...
0000000000000000}
```

Devices in the DSME-enabled PAN that hear either the DSME GTS Reply command or DSME GTS Notify command will update their SAB.

Devices in a DSME-enabled PAN may be allocated DSME GTSs during the association procedure. If *macExtendedDsmeEnabled* is TRUE and a device is instructed to associate with the PAN through the MLME-ASSOCIATE.request primitive having the *DsmeAssociation* parameter set to TRUE, the device requests DSME GTS allocation by sending a DSME Association Request command to a coordinator with the Extended DSME GTS Allocation field present, as described in 7.5.12. On receipt of the DSME Association Request command, the MAC sublayer of the coordinator informs the next higher layer that DSME GTS allocation is being requested through the MLME-ASSOCIATE.indication primitive with the *DsmeAssociation* parameter set to TRUE. The next higher layer of the coordinator instructs the MAC sublayer to respond to the request for DSME GTS allocation through the MLME-ASSOCIATE.response primitive. Then, the MAC sublayer of the coordinator sends a DSME Association Response command to the device containing the DSME GTS allocation information described in 7.5.13. On receipt of the DSME Association Response command, the MAC sublayer of the device allocates a DSME GTS and reports the results to the next higher layer. The DSME association process for a device that is not tracking the beacon is shown in Figure 6-58.

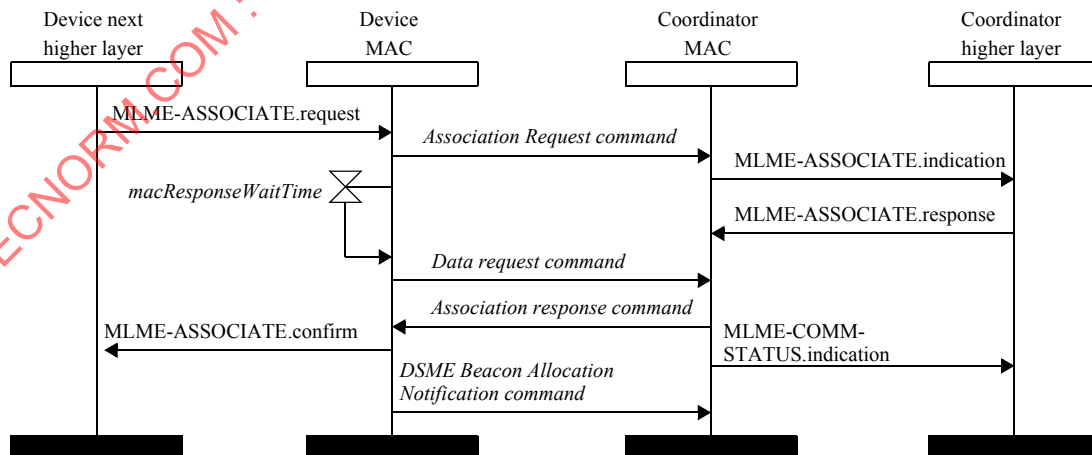


Figure 6-58—DSME association message sequence chart

**6.11.5.2 DSME GTS deallocation**

The DSME GTS source device is instructed to request the deallocation of existing DSME GTS through the MLME-DSME-GTS.request primitive. The destination device may also request the deallocation of an existing DSME GTS upon the expiration of the DSME GTS. From this point onward, the DSME GTS to be deallocated shall not be used by the device, and the *macDsmeSab* and the *macDsmeAct* shall be reset accordingly.

The higher layer of the device initiates a DSME GTS deallocation by issuing the MLME-DSME-GTS.request primitive to the MAC with the ManagementType parameter set to deallocation and the DsmeSabSpecification set to indicate the DSME GTSs to deallocate.

When a DSME GTS deallocation is due to the DSME GTS expiring, the device shall issue the MLME-DSME-GTS.indication primitive with which the ManagementType parameter set to EXPIRATION.

To deallocate an existing DSME GTSs, the device shall send the DSME GTS Request command to the corresponding device (the source or destination of which the DSME GTSs are to be deallocated). The destination address of the command shall be set to the address of the corresponding device. The Management Type field shall be set to deallocation and the DSME SAB Specification field shall be set to indicate the DSME GTSs to deallocate.

If a DSME GTS Response command with Management Type field set to deallocation is not received within the expected time, the MLME of the device shall notify the higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a Status parameter set to NO\_DATA.

On receipt of a DSME GTS Request command with Management Type field set to deallocation, the device shall check the DSME GTSs specified for deallocation in the received command.

If the DSME GTSs in the command match one or more of the allocated DSME GTSs in *macDsmeAct*, the device shall notify the receipt of the command to the higher layer using MLME-DSME-GTS.indication. The higher layer makes the decision on the deallocation, and informs MLME of its decision using MLME-DSME-GTS.response. Then, the device shall send a DSME GTS Response command to the source device of the received command (i.e., the destination address is set to the short broadcast address, and the DSME GTS destination address is set to the short address of the source device). The Management Type field of the DSME GTS Response command shall be set to deallocation, and the Status field shall be set to approved. The DSME SAB Specification field shall be set to indicate the DSME GTSs to deallocate.

If the DSME GTSs in the command do not match any of the allocated DSME GTSs in *macDsmeAct*, the device shall send a DSME GTS Response command to the source device of the received command (i.e., the destination address is set to the short address of the source device, and also the DSME GTS destination address is set to the short address of the source device). The Management Type field shall be set to deallocation, and the Status field shall be set to disapproved: unknown GTS. The DSME SAB Specification field shall be set to the same as the received command.

On receipt of a DSME GTS Response command with Management Type field set to deallocation, the device shall check the DSME GTS Destination Address field of the received command. If the DSME GTS Destination address is the same as the *macShortAddress*, the device shall inform the next higher layer of the result of the DSME GTS deallocation request using MLME-DSME-GTS.confirm. If the value of the Status field in the command is approved, the device shall send a DSME GTS Notify command to the source device of the received command (i.e., the destination address is set to short broadcast address, and the DSME GTS destination address is set to the short address of the source device). The Management Type field of the DSME GTS Notify command shall be set to deallocation, and the DSME SAB Specification field shall be set to indicate the DSME GTSs to deallocate.



If the DSME GTS Destination address is not the same as the *macShortAddress*, the device shall update *macDsmeSab* according to the DSME SAB Specification field in this command to reflect the neighbor's deallocated DSME GTSS.

On receipt of a DSME GTS Notify command with Management Type field set to deallocation, the device shall check the DSME GTS Destination Address field of the received command. If the DSME GTS Destination address is the same as the *macShortAddress*, the device shall notify the next higher layer of the receipt of the DSME GTS Notify command using MLME-COMM-STATUS.indication.

If the DSME GTS Destination address is not the same as the *macShortAddress*, the device shall update *macDsmeSab* according to the DSME SAB Specification in this command to reflect the neighbor's deallocated DSME GTSS.

#### 6.11.5.3 DSME GTS expiration

The MLME of the device shall attempt to detect idle DSME GTSS in *macDsmeAct*, described in Table 8-90, using the following rules:

- The MLME of the device shall perform DSME GTS deallocation when a DSME GTS has expired (i.e., a device has stopped using the DSME GTS).
- The MLME of the destination device of DSME GTS shall assume that the source device is no longer using its DSME GTS if a Data frame has not been received in the DSME GTS for *macDsmeGtsExpirationTime*.
- The MLME of the Source device of DSME GTS shall assume that the link quality is bad if an Enh-Ack frame has not been received within *macDsmeGtsExpirationTime* when the acknowledgment was requested.

#### 6.11.5.4 DSME GTS retrieve

A DSME-enabled device needs to maintain time synchronization and DSME GTS related information in order to use DSME GTS.

If the originator device has lost synchronization before its allocated DSME GTSS of current superframe starts, the originator device sends a DSME Information Request command to the target device. The Info Type field shall be set to the appropriate value based on the information that is being requested.

On receipt of a DSME Information Request command with Info Type field set to timestamp, the target device shall determine whether it has an allocated DSME GTS to the requesting device. If it has an allocated DSME GTS, the target device shall send a DSME Information Response command in the DSME GTS. If it does not have an allocated DSME GTS to the originator device, the target device shall ignore the request.

On receipt of a DSME Information Request command with Info Type field set to request SAB specification, the target device shall send a DSME Information Response command with the DSME SAB Specification field set according to the current allocation status of all one-hop neighborhoods of the device. The entire DSME SAB is stored in *macDsmeSab*. The DSME SAB sub-block index and the DSME SAB sub-block length indicate which part of the *macDsmeSab* shall be selected as the DSME SAB sub-block to be included in the DSME Information Response command.

On receipt of a DSME Information Request command with Info Type field set to request the PAN descriptor, the target device shall send a DSME Information Response command with the DSME PAN Descriptor.

On receipt of a DSME Information Response command containing the Timestamp field, Superframe ID field and the Slot ID field, the originator device may use this information to retrieve synchronization and continue to use DSME GTSS.

The message sequence chart for a successful exchange of DSME information is shown in Figure 6-59.

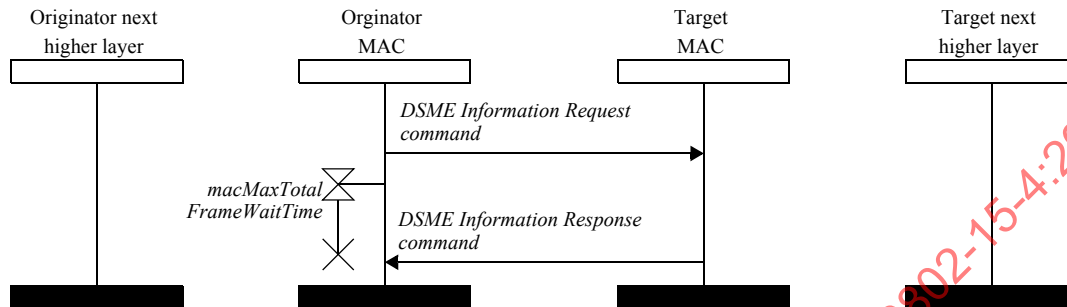


Figure 6-59—Message sequence chart for DSME information request

#### 6.11.5.5 DSME GTS change

The procedure of DSME GTS change shall be initiated either by the MAC or by the next higher layer. The next higher layer initiates a DSME GTS change by issuing an MLME-DSME-GTS.request primitive with the ManagementType parameter set to REDUCE.

To request the change of existing DSME GTSS, the MLME of the device shall send the DSME GTS Request command to the corresponding device of the DSME GTSS. The Management Type field shall be set to either reduce or restart, and DSME GTS SAB Specification field shall be set to indicate the DSME GTSS to change.

On receipt of a DSME GTS Request command for DSME GTS change, if the GTS exists, the device shall immediately change its DSME GTS according to the Management Type field in the received command. Then the MLME of the device shall issue the MLME-DSME-GTS.indication primitive as described in 8.2.20.2 to notify the next higher layer of the change. The device shall send DSME GTS Response command to the source of the DSME GTS Request command.

If the GTS does not exist, then the device sends a DSME GTS Response command to the source of the DSME GTS Request command with the Status field set to “Disapproved, unknown GTS.”

On receipt of the DSME GTS Response command, the MLME of the device shall issue the MLME-DSME-GTS.confirm primitive, as described in 8.2.20.4, with a Status of SUCCESS, the ManagementType parameter set to either REDUCE or RESTART, based on the value of the Management Type field in the command, and DSMEGTSSABSpecification parameter set to indicate the DSME GTSS to change.

#### 6.11.6 Beacon scheduling

The transmission of Enhanced Beacon frames from different devices shall be scheduled using the procedure described in this subclause.

When a new device wants to join a network, it uses the MLME-SCAN.request primitive in order to initiate a channel scan over a given set of channels. It searches for all coordinators transmitting Enhanced Beacon frames within the maximum BI period. The neighboring devices (i.e., coordinators) would share their beacon schedule information by transmitting an Enhanced Beacon frame to the new device. The beacon schedule information is expressed in a bitmap sequence that represents the schedule of Enhanced Beacon

frames transmitted from all neighboring devices. The corresponding bit in the bitmap shall be set to one if a beacon is occupied in that beacon slot. The new device shall search a vacant beacon slot from the bitmap in all the received Enhanced Beacon frames. Once the new device finds a vacant beacon slot, it uses the slot as its own beacon slot.

There can be a beacon slot collision when two or more devices are trying to compete for same beacon slot number. Figure 6-60 illustrates this circumstance; device D and E are new devices that join the network, and these devices receive the beacon bitmap from their neighboring devices. Since device A is a common neighboring device to device D and E, there can be a collision if both new devices E and D notify the use of the same vacant beacon slot within the CAP. This happens because device D and E are hidden to each other, and cannot listen to each other's transmission. DSME Beacon Allocation Notification command is used to resolve this possible collision. If device D and E send a DSME Beacon Allocation Notification command with same SdIndex value (SdIndex of 1 in Figure 6-60), device A determines which device has notified first. Device A shall respond with the Beacon Collision Notification command to the device that has notified later.

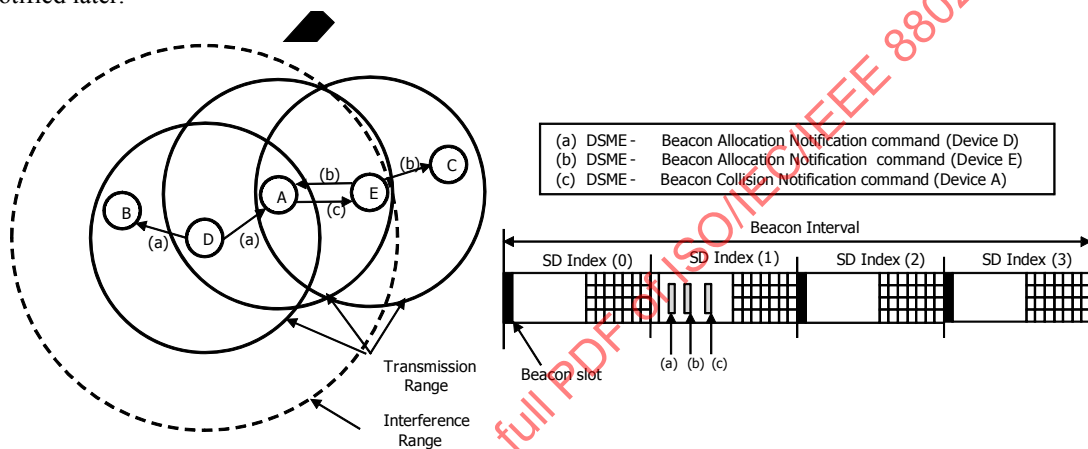


**Figure 6-60—Avoidance of beacon slot allocation collision**

### 6.11.7 Time synchronization

A new device discovers its neighboring devices via the scanning process. Upon discovering its neighbor devices, it may associate with one of them to become part of the network. The device with which the new device associates is referred to as its time synchronization parent. Since the new device tracks its time synchronization parent's Enhanced Beacon frame, it shall determine the receiving time of Enhanced Beacon frame.

Each device performs time synchronization using the values of the Timestamp fields in the Enhanced Beacon frame from its time synchronization parent in order to maintain global time synchronization in the PAN. As shown in Figure 6-61, a device locates the start time of superframe using the value of the Beacon Timestamp field in the Enhanced Beacon frame. The device knows when the Enhanced Beacon frame is actually transmitted from the time synchronization parent using value of the Beacon Offset Timestamp field. The difference in timestamp values reflects time delay due to CCA or processing time at PHY to transmit a frame. A device that has received the Enhanced Beacon frame calculates the actual reception time considering the value of the Beacon Offset Timestamp field, and then resets its local time accordingly. The PAN coordinator does not reset its local time according to the received Enhanced Beacon frame since the PAN coordinator does not have the time synchronization parent device.

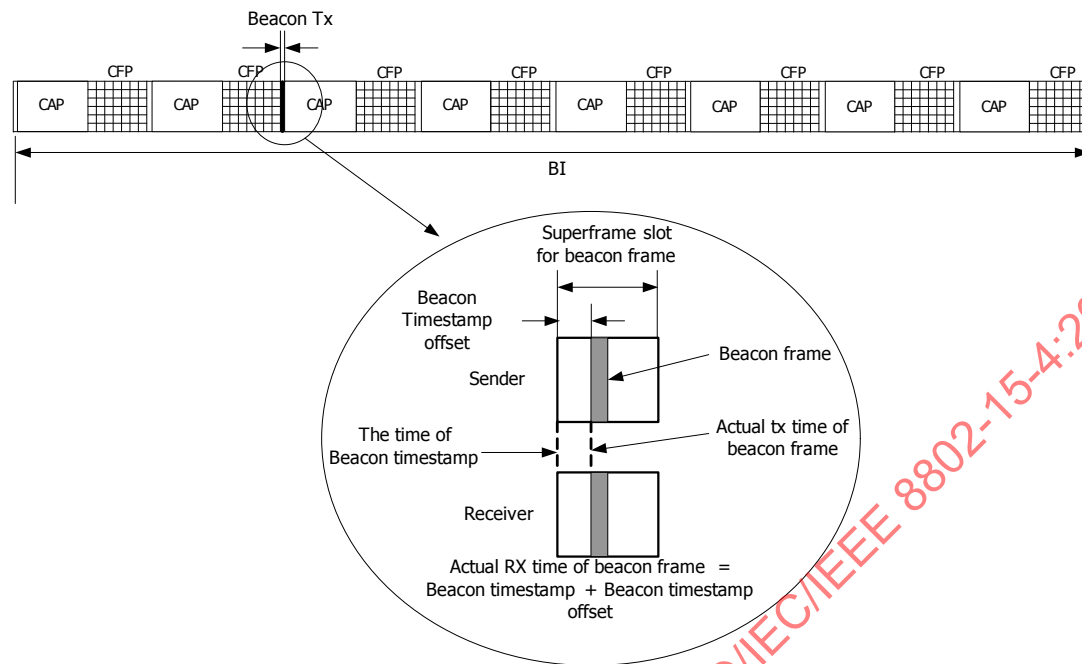


Figure 6-61—Time synchronization in a DSME-enabled PAN

### 6.11.8 Deferred beacon

Generally in a beacon-enabled PAN, Beacon frames are transmitted without CCA or backoff. However, Beacon frames can experience a collision due to the interference from inside or outside of the PAN. In such an environment, a coordinator may use deferred beacon to avoid the collision problem and improve the reliability of the beacon transmission. If a coordinator is instructed to use the deferred beacon, a coordinator shall set the Deferred Beacon field in its Enhanced Beacon frames to indicate deferred beacons. Then, the coordinator shall use CCA before sending an Enhanced Beacon frame. If CCA confirms the clear channel, the coordinator shall send the beacon. The Beacon Timestamp field of the Enhanced Beacon frame shall be set to the start time of the superframe. The Beacon Offset Timestamp field shall be set to the amount of time that is delayed due to CCA.

### 6.11.9 Passive channel scan

The Channel Hopping Specification field in the received Enhanced Beacon frame shall update the value of ChannelHoppingSpecification in the PANDescriptor. This value is sent to the next higher layer via the MLME-SCAN.confirm primitive. The value of Channel Offset field in the received beacon shall update the value of *macChannelOffsetBitmap* in the MAC PIB attributes. For instance, if ChannelOffset is set to 0x01, the value of *macChannelOffsetBitmap* corresponding channel shall set to one. Thus, the value of *macChannelOffsetBitmap* shall represent if the channel offset value is used among one hop neighbor devices.

## 6.12 LE transmission, reception and acknowledgment

### 6.12.1 LE transmission, reception, and acknowledgment with positive handshakes

When *macLeHsEnabled* is set to TRUE in the coordinator and the device, the data transmission, reception, and acknowledgment process illustrated in Figure 6-62 shall be used.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

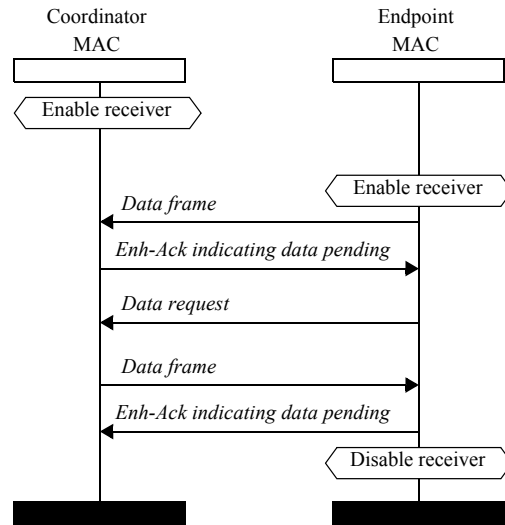


Figure 6-62—LE transmission with positive handshake

If the device received an Enh-Ack frame from the coordinator indicating that the coordinator has a pending frame, the device shall send a Data Request command to the coordinator and wait for the corresponding Data frame from the coordinator.

If the Enh-Ack frame is not received as expected, retransmission shall be performed as defined in 6.7.4.3. If, after sending the Enh-Ack frame with the Frame Pending field set, the Data Request command is not received, the coordinator waits for a retransmission and, if received, repeats the Enh-Ack frame with the Frame Pending field set.

### 6.12.2 Coordinated sampled listening (CSL)

The CSL mode is turned on when the PIB attribute *macCslPeriod* is nonzero and turned off when *macCslPeriod* is zero. In CSL mode, transmission, reception, and acknowledgment work as follows. Figure 6-63 illustrates the basic CSL operations.

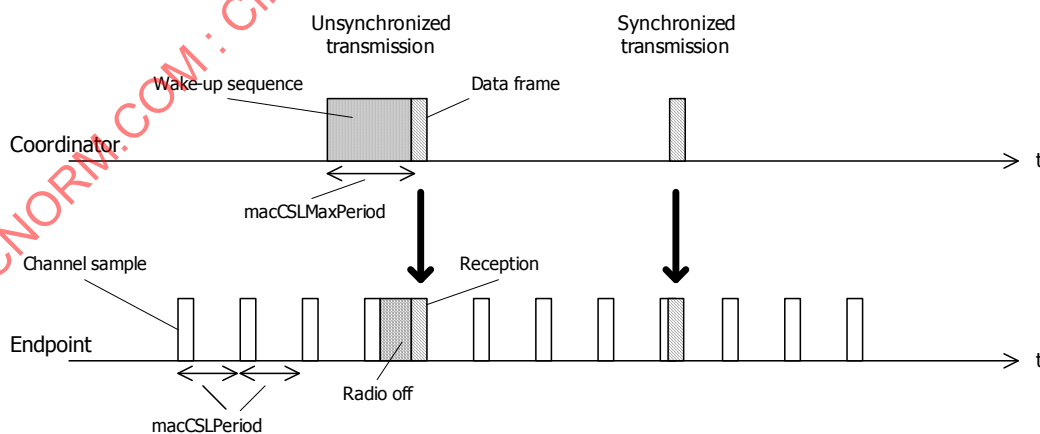


Figure 6-63—Basic CSL operations

A wake-up frame is a Multipurpose frame with the following:

- The AR field set to indicate no acknowledgment requested
- Only the Rendezvous Time IE, as defined in 7.4.2.6
- No payload

The CSL Phase is the time from the first symbol of the frame containing the CSL IE, as defined in 7.4.2.3, was transmitted until the next channel sample.

### 6.12.2.1 CSL idle listening

During idle listening, the CSL coordinator performs a channel sample every *macCslPeriod* time. If a wake-up frame is not received on the channel, the CSL coordinator disables the receiver until the next channel sample time and then performs the next channel sample. If the channel sample contains a wake-up frame, the CSL coordinator checks the destination address in the wake-up frame. Acknowledgment and retransmission are performed as described in 6.7.4.3, with the additional requirement that for each transmission, the process described in this subclause is used. If the destination address of the wake-up frame matches *macShortAddress*, the CSL coordinator checks whether the wake-up frame contains the Wake-up Interval field. If the wake-up frame does not contain the Wake-up Interval field, the CSL coordinator disables the receiver until the rendezvous time and then enables the receiver to receive the Data frame. If the wake-up frame does contain the Wake-up Interval field, and its value is nonzero, the CSL coordinator disables the receiver and transmits the data request frame with the AR field in the Data frame set to one. Then the CSL coordinator waits for the Enh-Ack frame. If the Enh-Ack frame is received, the rendezvous time is updated using the contents of the Enh-Ack frame, and the receiver remains on in order to receive the frame.

When the wake-up frame contains the Wake-up Interval field with a nonzero value, the transmission, reception, and acknowledgment operation is as illustrated in Figure 6-64.

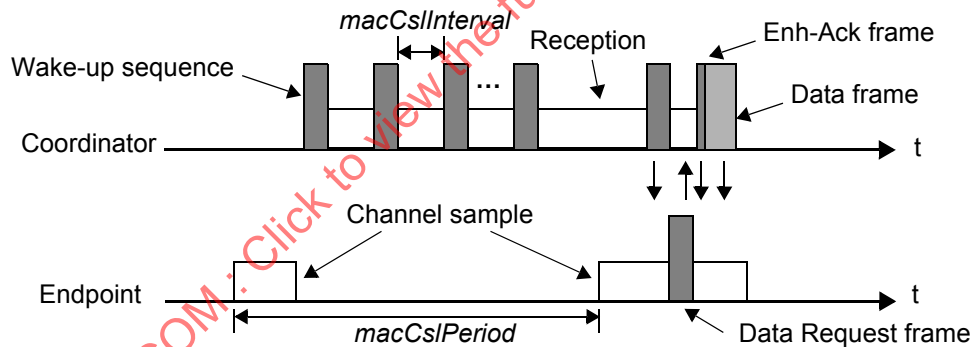


Figure 6-64—CSL operations when the wake-up frame interval is nonzero

If the destination address of the wake-up frame does not match *macShortAddress*, the CSL coordinator disables the receiver until the rendezvous time plus the transmission time of the maximum length payload frame and the secure Enh-Ack frame, and then resumes channel sampling.

### 6.12.2.2 CSL transmission

Each CSL transmission of a payload frame is preceded with a sequence of wake-up frames (wake-up sequence).

### 6.12.2.3 Unicast transmission

In unicast transmissions, the wake-up sequence length may be long or short based on the following two cases:

*Unsynchronized transmission:* This is the case when the MAC sublayer does not know the CSL phase, and period of the destination device. In this case, the maximum wake-up sequence length is *macCslMaxPeriod*.

*Synchronized transmission:* This is the case when the MAC sublayer knows the CSL phase and period of the destination device. In this case, the wake-up sequence length is only the guard time against clock drift based on the last time when CSL phase and period updated about the destination device.

If the next higher layer has multiple frames to transmit to the same destination, it may set the Frame Pending field in the Frame Control field to one in all but the last frame, in order to maximize the throughput.

CSL unicast transmission is performed in the following steps by the MAC:

- a) Perform CSMA-CA to acquire the channel.
- b) If the previous acknowledged frame to the destination has the Frame Pending field set and is within *macCslFramePendingWait*, as defined in Table 8-91, go to step e).
- c) If it is a synchronized transmission, wait until the destination device's next channel sample.
- d) For the duration of wake-up sequence length
  - 1) Construct wake-up frame with the destination short address and remaining time to payload frame transmission (at the end of wake-up sequence).
  - 2) Transmit wake-up frame.
  - 3) If *macCslInterval* is not equal to zero, wait for up to *macCslInterval* for the data request frame from the corresponding destination device. If the data request frame is received, then stop the transmission of the wake-up sequence, perform a CSMA-CA to acquire the channel, and transmit an Enh-Ack frame with the Rendezvous Time field updated to zero.
- e) Transmit the frame
- f) Wait for the Enh-Ack frame if the AR field of the received frame is set to one.
- g) If the Enh-Ack frame is received, update CSL phase and period information about the destination device from the CSL IE.
- h) If the Enh-Ack frame is not received, start retransmission process.

Retransmissions follow the same process as defined in 6.7.4.3 except that each transmission follows the process above.

### 6.12.2.4 Broadcast transmission

Broadcast transmission is the same as unicast transmission except for the following conditions:

- It is always unsynchronized transmission.
- The destination address of the wake-up frames is set to the broadcast short address.
- If *macCslInterval* is nonzero, the CSL coordinator will stop sending the wake-up sequence only after either receiving data request frames from all of the destination devices or when the *macCslMaxPeriod* expires.
- It may include a CSL IE.



### 6.12.2.5 CSL reception

When a payload frame is received, the MAC layer performs the following steps:

- Immediately sends back an Enh-Ack frame with the destination address set as the transmitting device and its own CSL phase and period filled in the CSL IE. The Enh-Ack frame may be authenticated and/or encrypted depending on the current security mode.
- If the CSL IE is present in the received frame, the CSL phase and period information about the transmitting device is updated with the information in the CSL IE.
- Frame Pending field in the received frame is set to one, the receiver is kept on for *macCslFramePendingWait* time before going back to CSL idle listening. Otherwise, CSL idle listening starts.

### 6.12.2.6 CSL over multiple channels

When *macCslChannelMask* is nonzero, the CSL operations are extended to all the channels selected in the bitmap. CSL idle listening performs a channel sample on each channel from the lowest number to the highest in a round-robin fashion. In the unsynchronized case, CSL transmission transmits a wake-up sequence of the length number of channels  $\times$  *macCslMaxPeriod* before each frame. In the synchronized case, CSL transmission calculates the next channel sample time and channel number and transmits at the next channel sample time on the right channel with a short wake-up sequence. In this case, CSL phase is the duration from now to the next channel sample on the first channel selected in *macCslChannelMask*.

### 6.12.2.7 Turning off CSL mode to reduce latency

The next higher layer has the option to turn off sampled listening and stop sending wake-up sequences to reduce latency for urgent messages. This assumes that the higher layer manages the coordination between the sender and receiver in turning on and off sampled listening. To turn off sampled listening, the next higher layer simply sets *macCslPeriod* to zero. To turn on sampled listening, the high layer restores *macCslPeriod* to its previous nonzero values. Similarly, to stop sending wake-up sequences, the next higher layer sets *macCslMaxPeriod* to zero and restores it to its previous value to return to normal CSL mode. To request a neighboring device to turn off sampled listening, the device shall send a frame to the device with frame pending bit set to one. This prevents CSL from turning off the radio before the request is processed.

## 6.12.3 RIT

### 6.12.3.1 General

RIT is an alternative low-energy MAC for nonbeacon-enabled PAN. RIT mode is turned on when PIB attribute *macRitPeriod* is nonzero and is turned off when *macRitPeriod* is zero.

*macCslPeriod* (in coordinated sample listening) and *macRitPeriod* shall not be nonzero at the same time. Figure 6-65 illustrates the basic RIT operations. Figure 6-66 illustrates the RIT operations when the RIT Data Request command payload carries schedule information, as defined in 7.5.22.

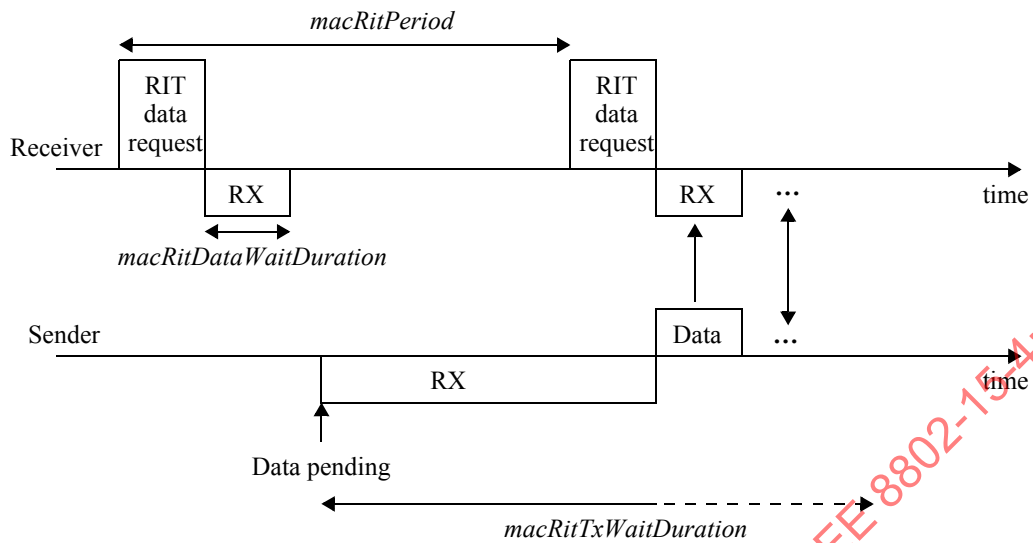


Figure 6-65—Basic RIT operations

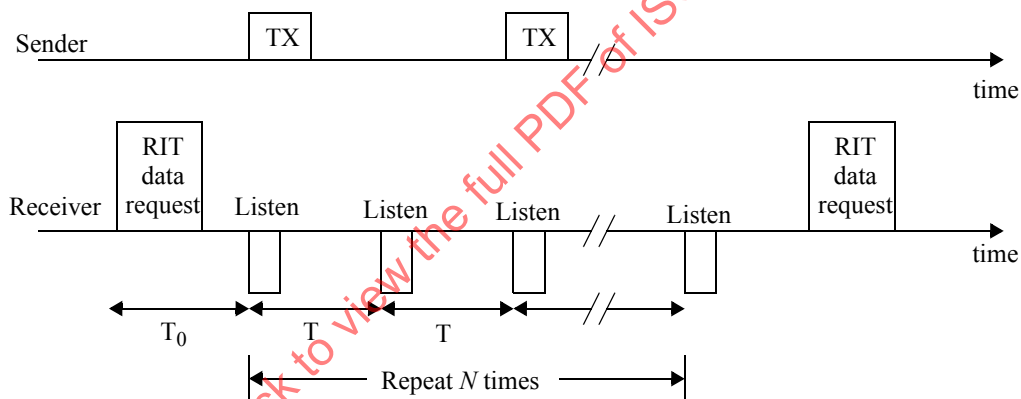


Figure 6-66—RIT operations when data request carries schedule information

### 6.12.3.2 Periodic RIT data request transmission and reception

In RIT mode, a device transmits an RIT Data Request command every *macRitPeriod* using unslotted CSMA-CA.

The destination address of the command may be broadcast short address or the address of the intended transmitter of data (associated coordinator). The command may optionally contain a payload, as defined in 7.5.22. When the command carries no payload, after the transmission of the RIT Data Request command, the device listens for *macRitDataWaitDuration* for an incoming frame (except for an RIT Data Request command) and then goes back to sleep state till the next periodic transmission of the RIT Data Request command. When a device is in the receiving state after transmission of the RIT Data Request command, an RIT Data Request command that is received from another device shall be discarded. When the RIT Data Request command carries the timing information as payload, the device goes back to sleep after the

transmission of the RIT Data Request command until the end of the Time to First Listen ( $T_0$ ) period of time. Then the device repeats a listen interval of  $macRitDataWaitDuration$  every Repeat Listen Interval ( $T$ ) period of time for Number of Repeat Listen ( $N$ ) times. The value of the Number of Repeat Listen field shall be less than  $(macRitPeriod - \text{Time to First Listen field}) / (\text{Repeat Listen Interval field})$ .

Upon reception of RIT Data Request command with RIT Request Payload field, the device shall notify the next higher layer by issuing an MLME-RIT-REQ.indication, as defined in 8.2.25.1.

Figure 6-67 shows the message sequence chart for starting RIT mode.

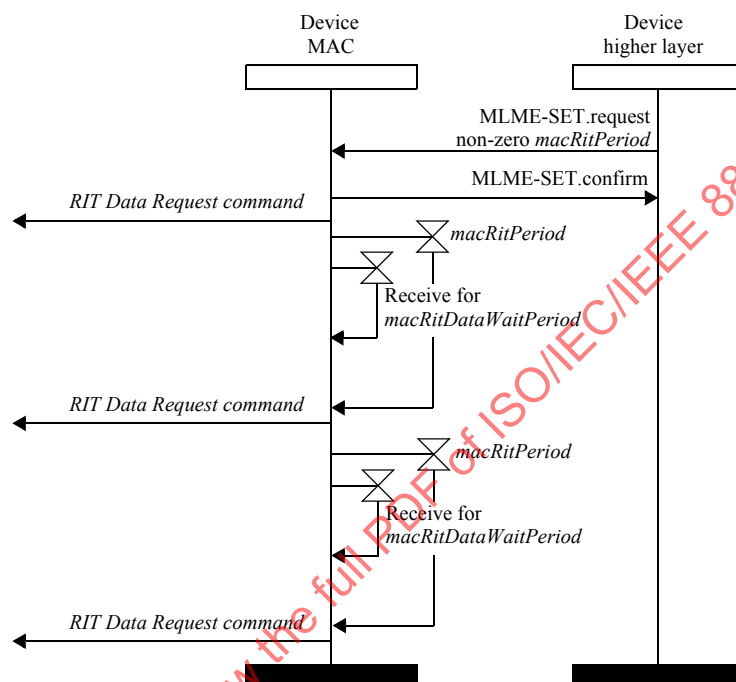


Figure 6-67—Message sequence chart for starting RIT mode

The device shall start listening slightly before each scheduled listen time based on a guard time computed from possible clock skew since the last RIT Data Request command transmission.

Upon reception of frame after the transmission of the RIT Data Request command, the device notifies its arrival to the next higher layer by initiating corresponding indication primitive. At the completion of frame reception, the next higher layer may set  $macRitPeriod$  to zero (RIT off). If this is the case, the device shall stop periodic transmission of RIT Data Request command and become always active until  $macRitPeriod$  is set to a nonzero value by the next higher layer. During this period when RIT is off all transactions shall be handled as those of normal nonbeacon-enabled PAN.

Figure 6-68 shows the message sequence chart for RIT mode when the RIT Data Request command contains a payload.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

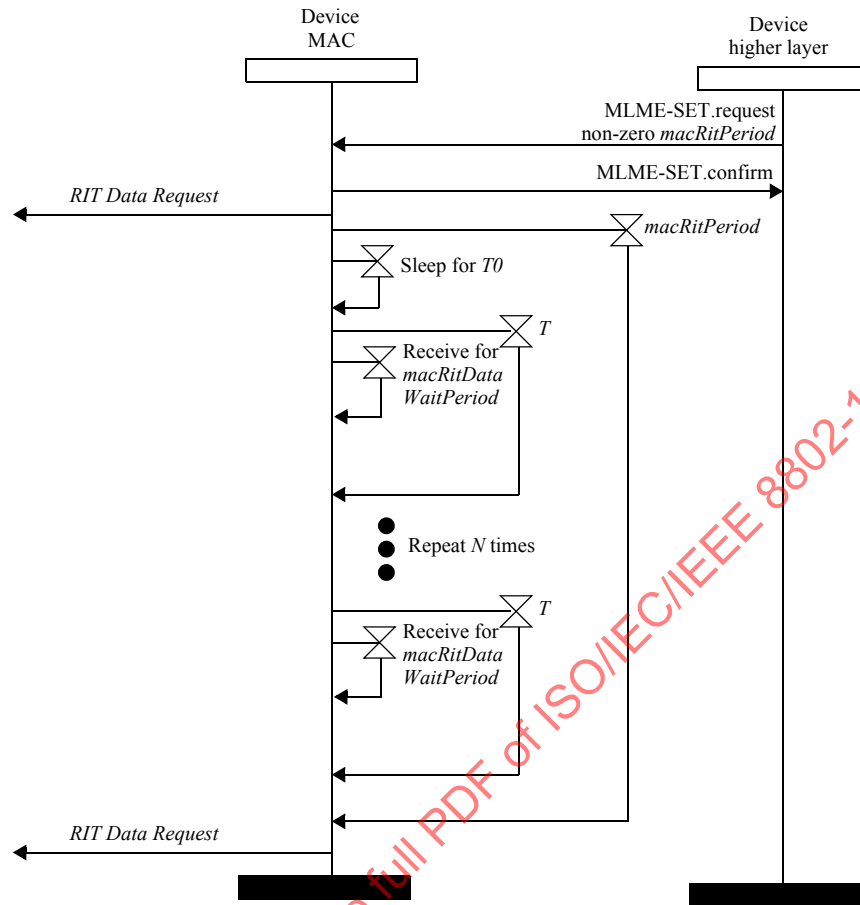


Figure 6-68—Message sequence chart for RIT with the RIT Data Request carrying a payload

### 6.12.3.3 RIT transmission

In order to transmit frame in RIT mode, the device may stop its periodic transmission of RIT Data Request commands, enable its receiver, and wait at most *macRitTxWaitDuration* for reception of an RIT Data Request command from another device.

Upon reception of RIT Data Request command, the MAC sublayer sends the pending data using unslotted CSMA-CA. The Destination PAN ID field and the Destination Address field of the outgoing Data frame shall be set as the Source PAN ID field and the Source Address field of the received RIT Data Request command, respectively. At the completion of frame transmission, the corresponding confirm primitive shall be issued by the MAC sublayer to the next higher layer. At this point, the device shall restart its transmission of periodic RIT Data Request commands. If the next higher layer sets *macRitPeriod* parameter to RIT off, the device shall stop periodic transmission of RIT Data Request command and become active.

When the RIT Data Request commands carry the Listen Information field, the device may either wait to receive a data request frame from the receiving device, or sleep until the next scheduled listen time by the receiving device and then wake up to transmit the intended frame.

Figure 6-69 shows the message sequence chart for data transmission in RIT mode.

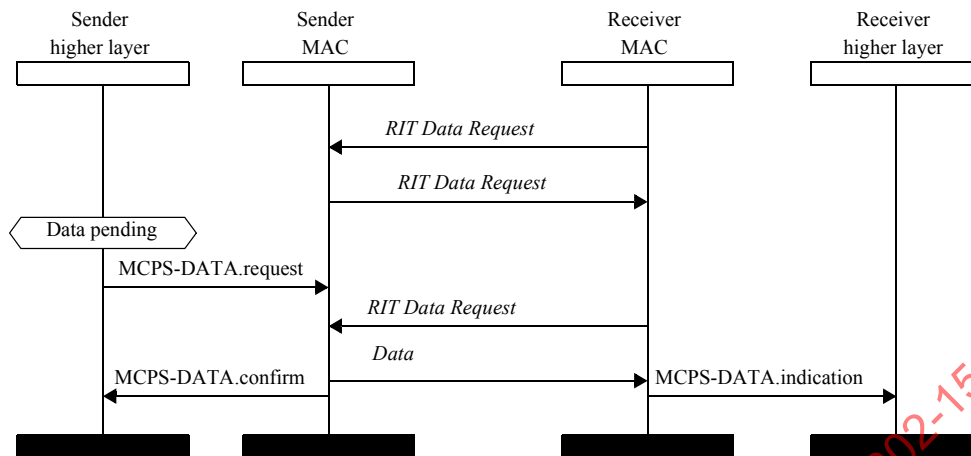


Figure 6-69—Message sequence chart for data transmission in RIT mode

When the RIT Data Request command contains an RIT Request Payload field, the MLME shall issue an MLME-RIT-REQ.indication, as defined in 8.2.25.1. The higher layer may respond with the MLME-RIT-RES.request, as defined in 8.2.25.2.

Optionally, for the devices operating in 920 MHz band, a sender device may skip doing CSMA-CA for transmission of RIT Data Response command as long as it comply with the regulatory requirements.

Figure 6-70 shows the message sequence chart for data transmission in RIT Mode with RIT Request Payload field and using an RIT Data Response command.

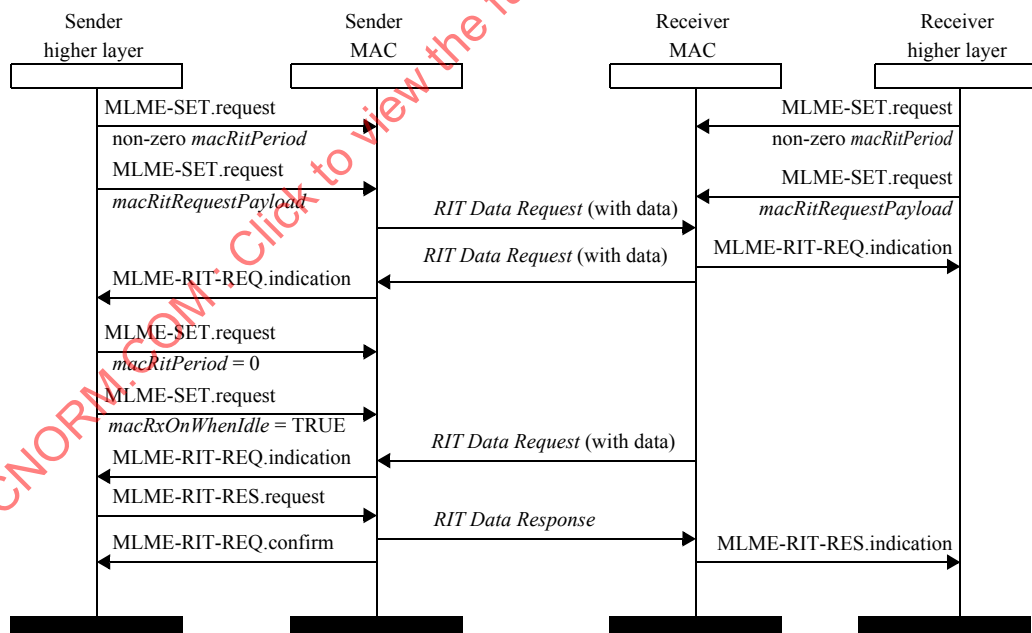


Figure 6-70—Message sequence chart for data transmission in RIT mode with RIT Response Payload field

### 6.12.4 Implicit RIT (I-RIT)

I-RIT is a low energy mode for nonbeacon-enabled PANs. I-RIT is designed to be used by end devices, such as sensors, that primarily transmit information to a coordinator but have no way of determining when they should make use of conventional RIT.

In order to enable I-RIT in an end device, the PIB attribute *macRitEnabled* is set to TRUE. When an end device has I-RIT enabled, the device shall enable its receiver *macRitOffsetInterval* after the last bit of its transmitted frame for a period of *macRitListenDuration*, in order to allow the receipt of a frame from the coordinator. The values of *macCslPeriod* and *macCslInterval* shall be ignored when *macRitEnabled* is TRUE. Transmission and reception in I-RIT mode is illustrated in Figure 6-71.

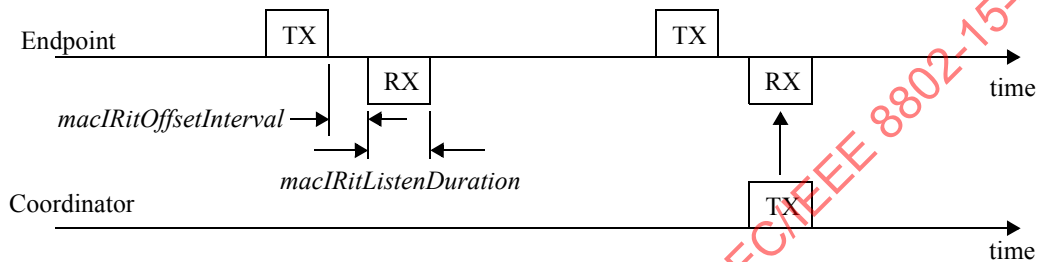


Figure 6-71—I-RIT transmission

### 6.13 Starting and maintaining TMCTPs

Figure 6-72 shows an example with a suggested message sequence for TMCTP formation between the SPC, which is the TMCTP-parent PAN coordinator, and a TMCTP-child PAN Coordinator.

In step A, the SPC obtains the set of available TVWS channels from the GDB through the Internet. The protocol used to access the GDB over the Internet is outside the scope of this standard. Alternatively, the SPC may obtain the set of available TVWS channels from another device. The SPC maps the TVWS channels to corresponding PHY channels, selects one of the available PHY channels, and transmits its Beacon frame in that channel. The TMCTP-child PAN coordinator completes the scan procedure over all PHY channels.

In Step B, the SPC transmits an Enhanced Beacon frame containing a TMCTP Specification IE, as described in 7.4.4.28. The TMCTP Specification IE shall be included in all Enhanced Beacon frames that are sent in a TMCTP-enabled PAN. Upon successful reception of the Enhanced Beacon frame from the SPC, the TMCTP-child PAN coordinator may request a DBS allocation or a DBS deallocation by sending a DBS Request command, as described in 7.5.23, to the SPC. Upon receiving the DBS Request command, the SPC will allocate or deallocate a DBS slot and a channel and generate a DBS Response command, as described in 7.5.24, to report the slot and the channel allocated or deallocated. The SPC may generate the DBS Response command for the deallocation without the request of the TMCTP-child PAN coordinator.

In Step C, the SPC indicates pending data for the TMCTP-child PAN coordinator in its Beacon frame. The TMCTP-child PAN coordinator sends the Data Request command. Upon receiving the Data Request command, the SPC replies with the DBS Response command generated in Step B. The SPC then sends its own Beacon frame.

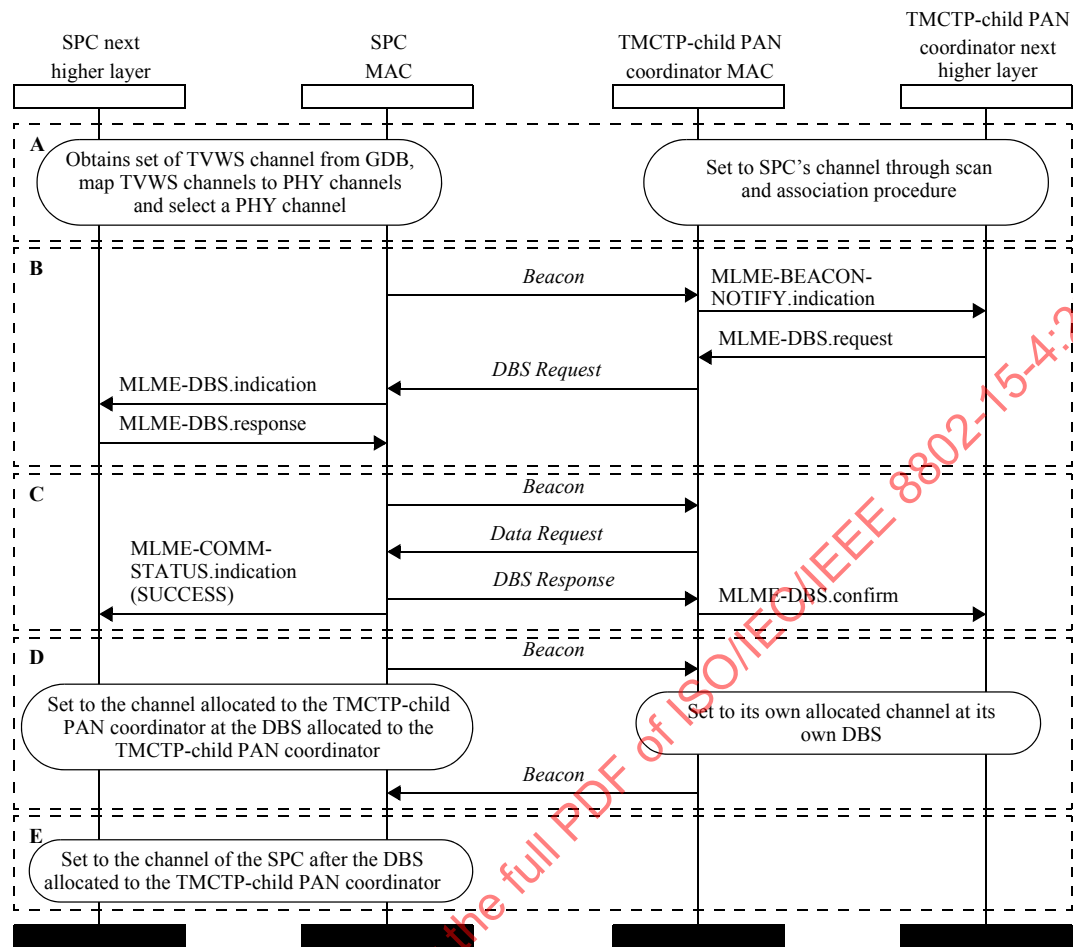


**Figure 6-72—Example message sequence chart between SPC and TMCTP-child PAN coordinator**

In Step D, the SPC switches to the channel allocated to the TMCTP-child PAN coordinator and listens for the Beacon frame from the TMCTP-child PAN coordinator. If the SPC does not receive a Beacon frame from the TMCTP-child PAN coordinator within three BIs, the SPC switches to the SPC channel and sends a DBS Response command to the TMCTP-child PAN coordinator in next superframe of the SPC.

In Step E, upon receiving the Beacon frame during the slot allocated to the TMCTP-child PAN coordinator on the channel allocated to the TMCTP-child PAN coordinator, the SPC switches to its own dedicated channel.

During the CAP of the SPC TMCTP superframe, each TMCTP-child PAN coordinator sends a DBS Request command to the SPC and receives a DBS Response command from the SPC. The SPC switches to the channel allocated to the TMCTP-child PAN coordinator before the DBS slot time allocated to the TMCTP-child PAN coordinator. Each TMCTP-child PAN coordinator forms an independent PAN by transmitting its beacon during the allocated DBS slot.

Figure 6-73 provides another example for TMCTP formation between two PAN coordinators, where one is the TMCTP-parent PAN coordinator and the other is a TMCTP-child PAN Coordinator.



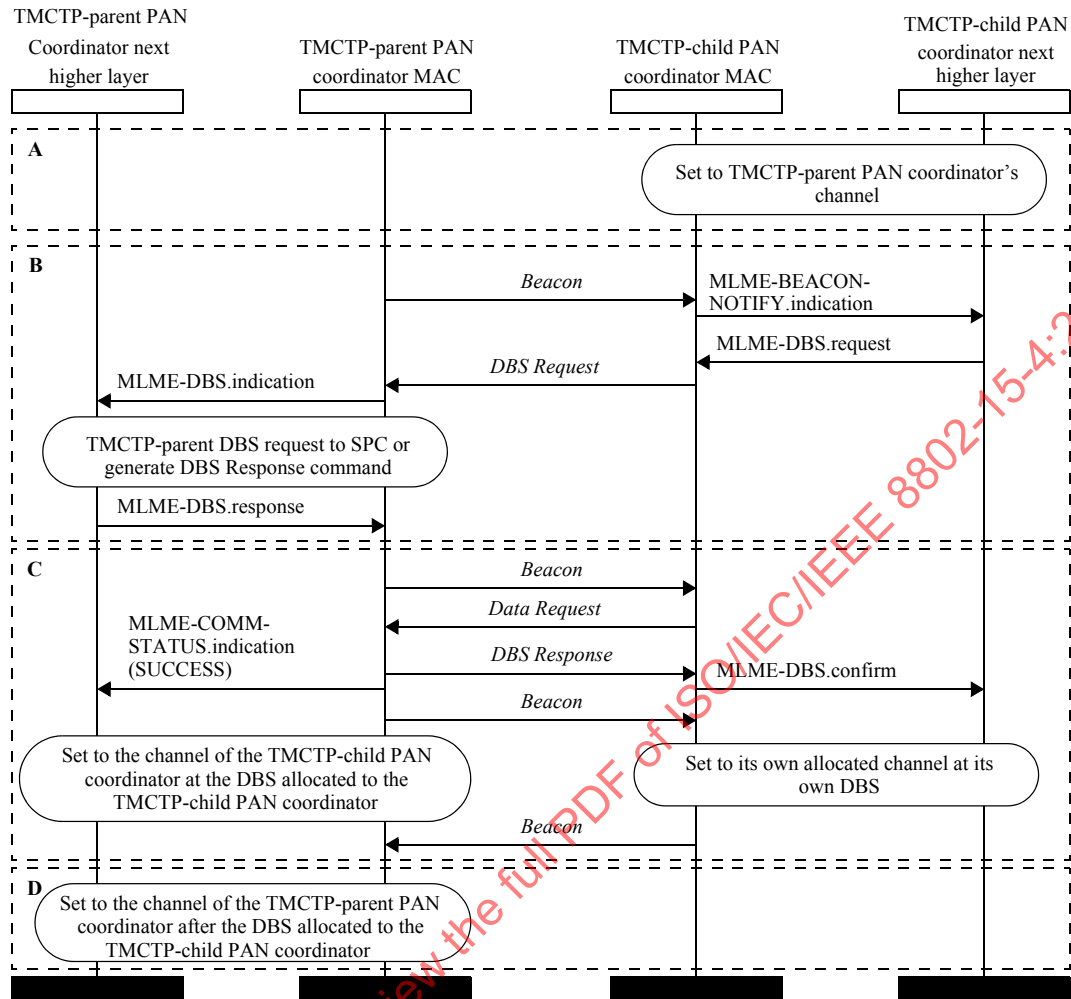


**Figure 6-73—Example message sequence chart between TMCTP PAN coordinators**

In Step A, the TMCTP-child PAN coordinator performs a scan procedure and waits for the Beacon frame of the TMCTP-parent PAN coordinator.

In Step B, the TMCTP-parent PAN coordinator sends an Enhanced Beacon frame containing a TMCTP Specification IE, as defined in 7.4.4.28. Upon successful reception of the Enhanced Beacon frame from the TMCTP-parent PAN coordinator, the TMCTP-child PAN coordinator requests a channel and a slot by using the DBS Request command sent to the TMCTP-parent PAN coordinator. Upon receiving the DBS Request command, the TMCTP-parent PAN coordinator either directly generates the DBS Response command reporting the slot and the channel allocated or deallocated, or sends the DBS Request command to the SPC and then receives the DBS Response command from the SPC. The TMCTP-parent PAN coordinator may generate the DBS Response command for the deallocation without a request of the TMCTP-child PAN coordinator.

In Step C, the TMCTP-parent PAN coordinator sends a Beacon frame. The TMCTP-parent PAN coordinator switches to the channel allocated to the TMCTP-child PAN coordinator and receives the Beacon frame from the TMCTP-child PAN coordinator.

In Step D, upon receiving the Beacon frame during the slot allocated to the TMCTP-child PAN coordinator on the channel allocated to the TMCTP-child PAN coordinator, the TMCTP-parent PAN coordinator switches to its own dedicated channel.

During CAP of the TMCTP-parent PAN coordinator, which has a relay capability or a channel allocation capability, each TMCTP-child PAN coordinator sends a DBS Request command to the TMCTP-parent PAN coordinator and receives the DBS Response command from the TMCTP-parent PAN coordinator. The TMCTP-parent PAN coordinator switches to the channel allocated to the TMCTP-child PAN coordinator during the DBS slot allocated to each TMCTP-child PAN coordinator. Each TMCTP-child PAN coordinator manages its own WPAN by transmitting a Beacon frame during the allocated DBS slot time.

Figure 6-74 shows an example of the multichannel allocation for the network topology as presented in Figure 5-3. In this case, the SPC operates on the dedicated channel, which is Channel 1 in Figure 6-74, and switches to the dedicated channels of TMCTP-child PAN coordinators 2, 3, and 4 during their DBSs. Similarly TMCTP-child PAN coordinator 4 operates on the dedicated channel, which is Channel 4, and switches to the dedicated channel of TMCTP-child PAN coordinator 5 during its DBS to communicate with TMCTP-child PAN coordinator 5.

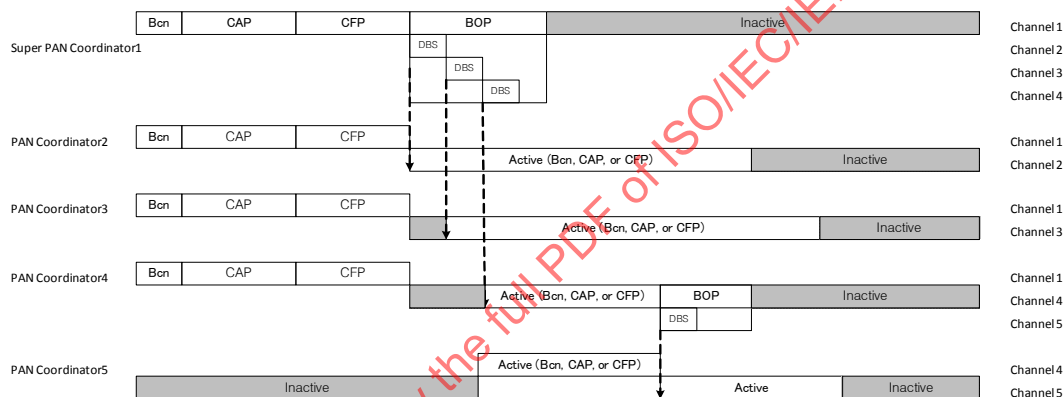


Figure 6-74—Example TMCTP BOP allocation

#### 6.14 MPM procedure for inter-PHY coexistence

To facilitate interference avoidance among multiple SUNs utilizing different PHYs in the same location, all SUN coordinators operating at a duty cycle of more than 1% shall support the MPM procedures. In the MPM scheme, Enhanced Beacon frames are sent using the CSM, as defined in Table 10-17. Enhanced Beacon frames used in the MPM procedures described here are Enhanced Beacon frames containing a Coexistence Specification IE.

The transmission of Enhanced Beacon frames should take place in all the channels defined for CSM, as described in Table 10-17, that overlap with the channel(s) in operation. The scanning for Enhanced Beacon frames and the transmission of Enhanced Beacon Request commands should take place in all the channels defined for CSM that overlap with the channel of interest or at least two channels for PHY modes where the CSM requires frequency hopping.

In a beacon-enabled PAN, an existing coordinator<sup>10</sup> shall transmit an Enhanced Beacon frame at a fixed interval by using CSM. Any intending coordinator<sup>11</sup> shall first scan for an Enhanced Beacon frame until the expiration of the enhanced beacon interval or until an Enhanced Beacon frame is detected, whichever occurs first. If an intending coordinator detects an Enhanced Beacon frame, it shall either occupy another channel, achieve synchronization with the existing PAN, or stop communication. While specific mechanisms to achieve synchronization between two PANs utilizing different PHY modes are implementation dependent, the timing information applicable for synchronization purposes is specified in the Enhanced Beacon frame. Figure 6-75 illustrates the MPM procedure.

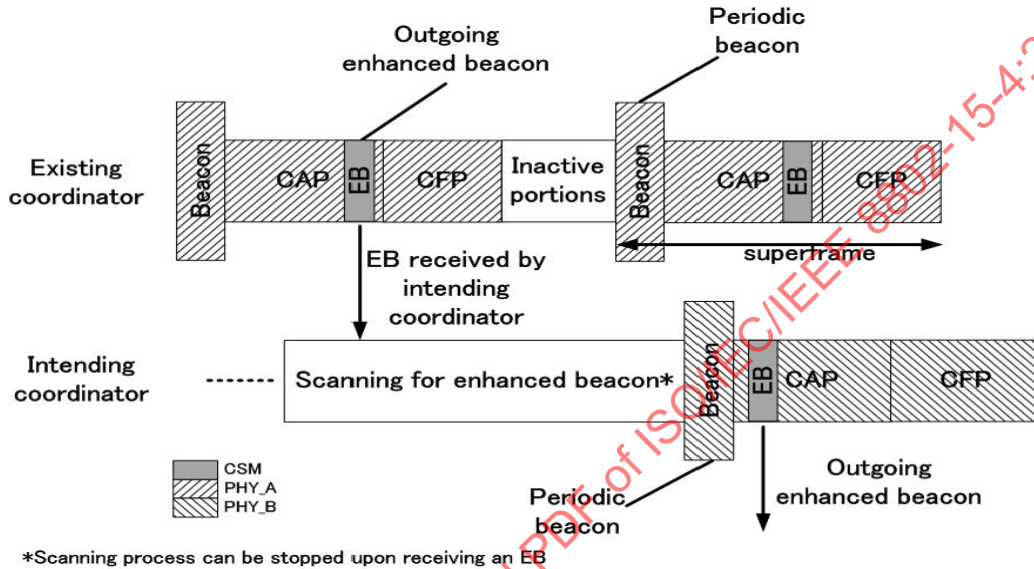


Figure 6-75—Inter-PHY mode coexistence in a beacon-enabled PAN

The Enhanced Beacon frame shall only be sent in the CAP.

In a nonbeacon-enabled PAN, an existing coordinator should transmit an Enhanced Beacon frame periodically using the CSM. Any intending coordinator shall first scan for an Enhanced Beacon frame until the expiration of  $EBI_{NB PAN}$  or until an Enhanced Beacon frame is detected, whichever occurs first. The illustration of the procedure is given in Figure 6-76.

Alternatively, an Enhanced Beacon frame may be obtained in an on-demand manner. In this case, an Enhanced Beacon Request command containing the ID of the Coexistence Specification IE in the set of IE IDs is sent by the intending coordinator requesting an Enhanced Beacon frame from the existing coordinator. Upon receiving an Enhanced Beacon Request command, the existing coordinator (or any other device within the same area that is capable of receiving and transmitting an Enhanced Beacon Request command or Enhanced Beacon frame using the CSM) may respond by sending an Enhanced Beacon frame to the intending coordinator. The intending coordinator should transmit an Enhanced Beacon Request command at least once every  $EBI_{NB PAN}$ . To increase the probability of receiving an Enhanced Beacon Request command, the existing coordinator may periodically allocate a fraction of the CAP time to scan for the Enhanced Beacon Request command in CSM. The illustration of the procedure is given in Figure 6-77.

<sup>10</sup>An existing coordinator is a coordinator currently operating a network.

<sup>11</sup>An intending coordinator is a coordinator intending to start a separate network.

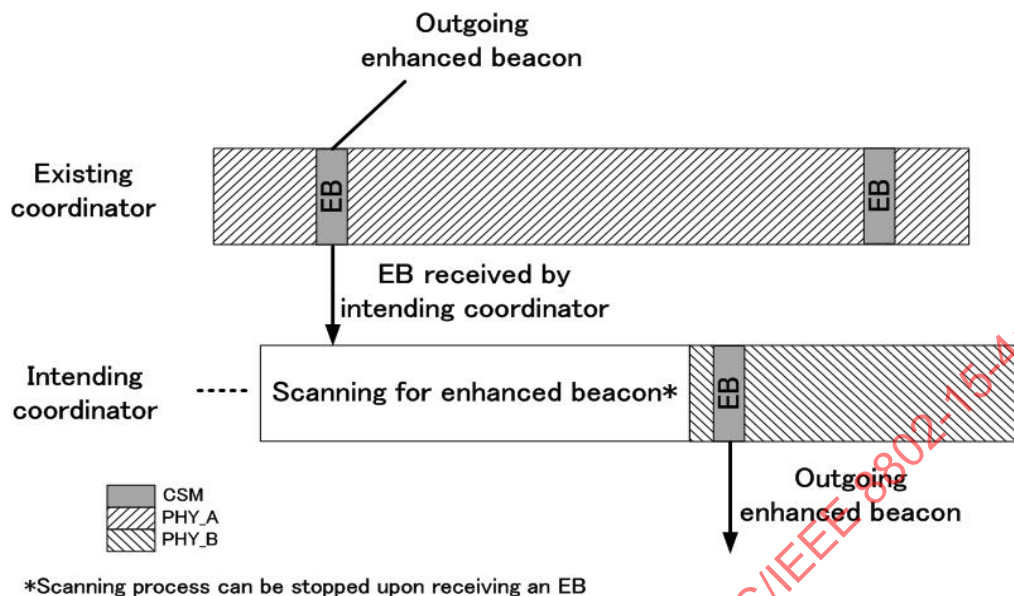


Figure 6-76—Inter-PHY mode coexistence in a nonbeacon-enabled PAN

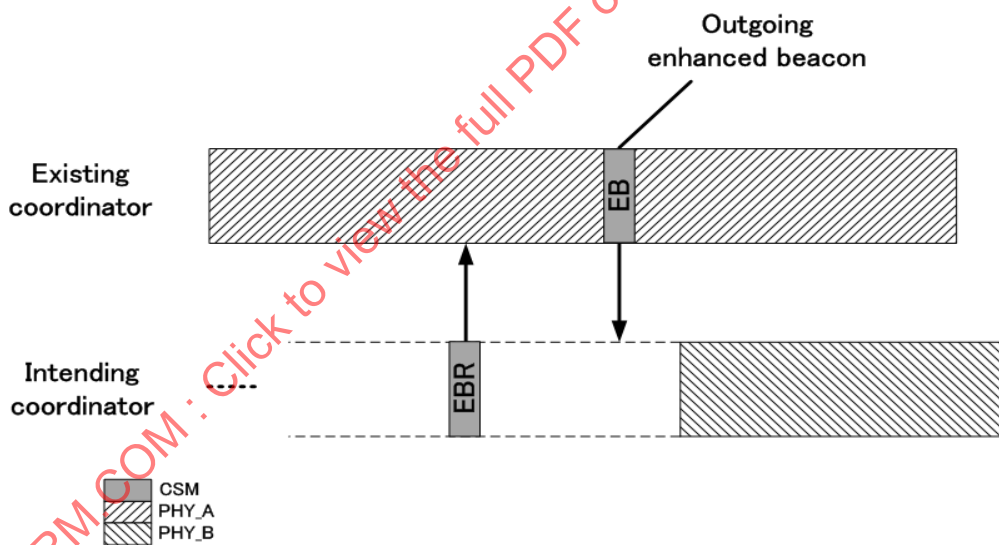


Figure 6-77—Alternative method for inter-PHY mode coexistence in a nonbeacon-enabled PAN

## 6.15 TVWS access procedures

In certain regulatory domains, an independent device operating in TVWS is required to communicate with a database which stores information on operation of primary incumbent systems to obtain permission and radio resource information, prior to starting communications. This database to protect primary systems is typically, but not limited to, a GDB. When a GDB is employed as the database, an independent device first communicates with the GDB to obtain permission to operate in TVWS. The communication between the independent device and the GDB is outside the scope of this standard. In this case, the independent device determines its geolocation to be reported to the GDB. The GDB then provides available channels and relevant operating information. Upon receiving permission from the GDB, the independent device may start a network and begin enabling other devices. Figure 6-78 shows the state transition diagram of the enabling procedure for a dependent device.

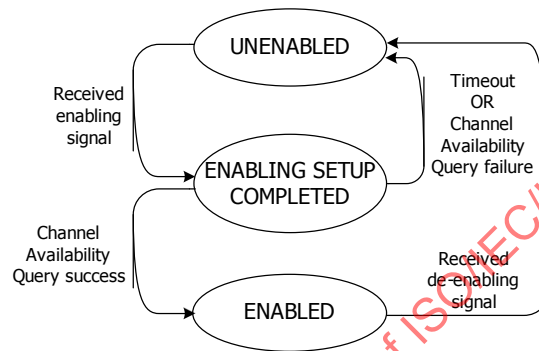


Figure 6-78—State transition of a dependent device

A dependent device, prior to receiving channel availability information (such as at power on or reset condition) begins in the UNENABLED state. The device may perform a channel scan or other procedure to detect transmissions that are active on the channel and determine a suitable source of channel availability data (e.g., an independent device advertisement). Upon receiving a frame with the TVWS Channel Information Source IE, as defined in 7.4.4.24, from an independent device, the state transitions to ENABLING SETUP COMPLETED. From the ENABLING SETUP COMPLETED state, the dependent device will initiate the exchange of information in order to be enabled, as required by the particular regulatory domain in which it is operating. The information exchange is facilitated by several IEs as specified in 7.4.4.20 and 7.4.4.21. Information on the list of locations of particular valid channel is contained in the IE specified in 7.4.4.22. Information on the set of available channels is contained in the IE specified in 7.4.4.23. A device that has access to channel availability information advertises that fact using the TVWS Channel Information Source IE as specified in 7.4.4.24. Upon successfully completing a channel availability query the state transitions to ENABLED. In this state, the dependent device is able to conduct data communications.

## 6.16 Channel timing management (CTM)

CTM facilitates assessment of the available timing schedule when a channel is available. CTM is used by employing the CTM IE as in 7.4.4.25. Upon receipt of MLME-CTM.request, the MLME shall schedule transmission for a CTM IE as in 7.4.4.25. The CTM IE may be contained in an enhanced beacon frame.

A device transmits a CTM IE to a PAN coordinator that is capable of database access as indicated in the enabling signal. The PAN coordinator capable of database access visits the database and obtains the channel timing information for available channels. The PAN coordinator may respond to a request by sending the CTM IE with CTM Control field set to Request Declined, to indicate that it has no capability to provide schedule information.

A device may transmit a CTM IE to request channel information from another device. A device shall respond to a CTM IE request using a CTM IE with the CTM Control field set to Success if it is capable of providing channel timing information on WPAN channels obtained from a database. Otherwise it shall respond with the CTM Control field set to Request Declined, to indicate that it has no capability to provide schedule information on available channels due to various reasons.

When the information in a CTM IE is identical to the information in the most recently transmitted CTM IE to the same requesting device, the responding device may set the CTM Control field value in a query response to Successful with no channel schedule changes from the last query and leave the Channel Timing Information field empty.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018

## 7. MAC frame formats

### 7.1 Device extended address

A device's extended address shall be a 64-bit extended universal identifier (EUI-64), as defined by IEEE Std 802-2014 and assigned by the IEEE Registration Authority.<sup>12</sup>

The EUI-64 shall be sent in the canonical form defined in IEEE Std 802-2014.

### 7.2 General MAC frame format

The general MAC frame for Frame Type values other than fragment and extended, as defined in Table 7-1, shall be formatted as illustrated in Figure 7-1.

| Octets:<br>1/2   | 0/1                | 0/2                   | 0/2/8                  | 0/2              | 0/2/8             | variable                        | variable   |             | variable         | 2/4 |             |
|------------------|--------------------|-----------------------|------------------------|------------------|-------------------|---------------------------------|------------|-------------|------------------|-----|-------------|
| Frame<br>Control | Sequence<br>Number | Destination<br>PAN ID | Destination<br>Address | Source<br>PAN ID | Source<br>Address | Auxiliary<br>Security<br>Header | IE         |             | Frame<br>Payload | FCS |             |
|                  |                    | Addressing fields     |                        |                  |                   |                                 | Header IEs | Payload IEs |                  |     |             |
|                  |                    | MHR                   |                        |                  |                   |                                 |            |             |                  |     | MAC Payload |

Figure 7-1—General MAC frame format

The fields of the MHR appear in a fixed order; however, some fields may not be included in all frames.

If the Frame Type field indicates a Fragment packet or Frak then the frame format shall be as formatted as defined in 23.3.3 and 23.3.6.2, respectively.

If the Frame Type field indicates an Extended frame, then the frame format shall be formatted as illustrated in 7.3.6.

#### 7.2.1 Frame Control field

The Frame Control field for frames other than the Multipurpose frame, Fragment frame and Extended frame shall be formatted as illustrated in Figure 7-2. The Frame Control fields for the Multipurpose frame and Extended frame are specified in 7.3.5 and 7.3.6, respectively.

| Bits: 0–2     | 3                   | 4                | 5  | 6                     | 7        | 8                                 | 9             | 10–11                             | 12–13            | 14–15                        |
|---------------|---------------------|------------------|----|-----------------------|----------|-----------------------------------|---------------|-----------------------------------|------------------|------------------------------|
| Frame<br>Type | Security<br>Enabled | Frame<br>Pending | AR | PAN ID<br>Compression | Reserved | Sequence<br>Number<br>Suppression | IE<br>Present | Destination<br>Addressing<br>Mode | Frame<br>Version | Source<br>Addressing<br>Mode |

Figure 7-2—Format of the Frame Control field

<sup>12</sup>Interested applicants should contact the IEEE Registration Authority, <http://standards.ieee.org/develop/regauth/>.



The Fragment frame type does not have a Frame Control field.

#### 7.2.1.1 Frame Type field

The Frame Type field shall be set as defined in Table 7-1.

**Table 7-1—Values of the Frame Type field**

| Frame type value<br>b2 b1 b0 | Description                   |
|------------------------------|-------------------------------|
| 000                          | Beacon                        |
| 001                          | Data                          |
| 010                          | Acknowledgment                |
| 011                          | MAC command                   |
| 100                          | Reserved                      |
| 101                          | Multipurpose                  |
| 110                          | Fragment or Frak <sup>a</sup> |
| 111                          | Extended                      |

<sup>a</sup>The Fragment and Frak formats are defined in 23.3.3 and 23.3.6.2, respectively.

Frame formats for each of the Frame Type field values are specified in 7.3.

#### 7.2.1.2 Security Enabled field

The Security Enabled field 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 field is set to one.

#### 7.2.1.3 Frame Pending field

The Frame Pending field shall be set to one if the device sending the frame has more data for the recipient, as described in 6.7.3. This field shall be set to zero otherwise.

The Frame Pending field 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.

When operating in low-energy (LE) CSL mode, the frame pending bit may be set to one to indicate that the transmitting device has pending frames to send to the same recipient and expects the recipient to keep the radio on until the frame pending bit is reset to zero.

When operating in TSCH mode, the frame pending bit can be set to one to indicate that the recipient should stay on in the next timeslot and on the same channel if there is no link scheduled.

At all other times, it shall be set to zero on transmission and ignored on reception.

**7.2.1.4 AR field**

The AR field specifies whether an acknowledgment is required from the recipient device on receipt of a Data frame or MAC command. If this field is set to one, the recipient device shall send an Ack frame or only if, upon reception, the frame passes the filtering described in 6.7.2. If this field is set to zero, the recipient device shall not send an Ack frame.

**7.2.1.5 PAN ID Compression field**

The PAN ID Compression field is used to indicate the presence of the PAN ID field.

When the frame version field value is 0b00 or 0b01, the PAN ID compression field is interpreted as follows:

- If both destination and source addressing information is present, the MAC sublayer shall compare the destination and source PAN identifiers. If the PAN IDs are identical, the PAN ID Compression field shall be set to one, and the Source PAN ID field shall be omitted from the transmitted frame. If the PAN IDs are different, the PAN ID Compression field shall be set to zero, and both Destination PAN ID field and Source PAN ID fields shall be included in the transmitted frame.
- If only either the destination or the source addressing information is present, the PAN ID Compression field shall be set to zero, and the PAN ID field of the single address shall be included in the transmitted frame.

When the frame version field value is 0b10, the PAN ID Compression Field for Beacon frame, Data frame, MAC Command frame and Ack frame shall be set based on the addressing fields present as defined in Table 7-2. Combinations of destination and source address with destination and source PAN ID and PAN Compression not shown in Table 7-2 shall not be generated.

**Table 7-2—PAN ID Compression field value for frame version 0b10**

| Destination Address | Source Address     | Destination PAN ID | Source PAN ID | PAN ID Compression |
|---------------------|--------------------|--------------------|---------------|--------------------|
| Not Present         | Not Present        | Not Present        | Not Present   | 0                  |
| Not Present         | Not Present        | Present            | Not Present   | 1                  |
| Present             | Not Present        | Present            | Not Present   | 0                  |
| Present             | Not Present        | Not Present        | Not Present   | 1                  |
| Not Present         | Present            | Not Present        | Present       | 0                  |
| Not Present         | Present            | Not Present        | Not Present   | 1                  |
| Extended            | Extended           | Present            | Not Present   | 0                  |
| Extended            | Extended           | Not Present        | Not Present   | 1                  |
| Short <sup>a</sup>  | Short <sup>a</sup> | Present            | Present       | 0                  |
| Short <sup>a</sup>  | Extended           | Present            | Present       | 0                  |
| Extended            | Short <sup>a</sup> | Present            | Present       | 0                  |
| Short <sup>a</sup>  | Extended           | Present            | Not Present   | 1                  |
| Extended            | Short <sup>a</sup> | Present            | Not Present   | 1                  |
| Short <sup>a</sup>  | Short <sup>a</sup> | Present            | Not Present   | 1                  |

<sup>a</sup>If both the destination and source addressing information is present and either is a short address, the MAC sublayer shall compare the destination and source PAN IDs and the PAN ID Compression field shall be set to zero if and only if the PAN identifiers are identical.

NOTE 1—In IEEE Std 802.15.4-2003, i.e., frame version 0b00, the PAN ID Compression field is named Intra-PAN, but it is in the same position and has the same effect as the PAN ID Compression field in 802.15.4-2006.

NOTE 2—The PAN ID Compression field is not present in Multipurpose frames.

### 7.2.1.6 Sequence Number Suppression

When set to one, this field indicates suppression of the Sequence Number field in the frame, and the sequence number shall be omitted. When set to zero, the Sequence Number field is present. If the Frame Version field is 0b00 or 0b01, the Sequence Number Suppression field shall be zero.

### 7.2.1.7 IE Present field

The IE Present field shall be set to one if IEs are contained in the frame. This field shall be set to zero otherwise. If the Frame Version field is 0b00 or 0b01, the IE Present field shall be zero.

### 7.2.1.8 Destination Addressing Mode field

The Destination Addressing Mode field shall be set to one of the values listed in Table 7-3.

If this field is equal to zero and the Frame Type field specifies a Data frame or MAC command and the Frame Version field is set to 0b00 or 0b01, the Source Addressing Mode field shall be nonzero, implying that the frame is directed to the PAN coordinator with the PAN ID as specified in the Source PAN ID field.

For frames with a Frame Version of 0b10, the destination address or destination PAN ID or both may be omitted.

**Table 7-3—Valid values of the Destination Addressing Mode and Source Addressing Mode fields**

| Addressing mode value<br>b1 b0 | Description                                          |
|--------------------------------|------------------------------------------------------|
| 00                             | PAN ID and address fields are not present.           |
| 01                             | Reserved                                             |
| 10                             | Address field contains a short address (16 bit).     |
| 11                             | Address field contains an extended address (64 bit). |

### 7.2.1.9 Frame Version field

The Frame Version field is an unsigned integer that specifies the version number corresponding to the frame. For all frame types, the Frame Version field shall be set as described in that subclause.

A summary of the frame versions, and their applicability, for each of the frame types is shown in Table 7-4.

**Table 7-4—Frame Version field values**

| Frame type     | Frame Version field value                |                        |                   |          |
|----------------|------------------------------------------|------------------------|-------------------|----------|
|                | 0b00                                     | 0b01                   | 0b10              | 0b11     |
| Beacon         | IEEE Std 802.15.4-2003                   | IEEE Std 802.15.4-2006 | IEEE Std 802.15.4 | Reserved |
| Data           | IEEE Std 802.15.4-2003                   | IEEE Std 802.15.4-2006 | IEEE Std 802.15.4 | Reserved |
| Acknowledgment | IEEE Std 802.15.4-2003                   | IEEE Std 802.15.4-2006 | IEEE Std 802.15.4 | Reserved |
| MAC Command    | IEEE Std 802.15.4-2003                   | IEEE Std 802.15.4-2006 | IEEE Std 802.15.4 | Reserved |
| Reserved       | —                                        | —                      | —                 | —        |
| Multipurpose   | IEEE Std 802.15.4                        | Reserved               | Reserved          | Reserved |
| Fragment       | Frame Version field not present in frame |                        |                   |          |
| Extended       | Frame Version field not present in frame |                        |                   |          |

**7.2.1.10 Source Addressing Mode field**

The Source Addressing Mode field shall be set to one of the values listed in Table 7-3.

If this field is equal to zero and the Frame Type field specifies a Data frame or MAC command, and the Frame Version field is set to 0b00 or 0b01, the Destination Addressing Mode field shall be nonzero, implying that the frame has originated from the PAN coordinator with the PAN ID as specified in the Destination PAN ID field.

**7.2.2 Sequence Number field**

The Sequence Number field specifies the sequence identifier for the frame.

For a Beacon frame, the Sequence Number field shall specify a BSN. For other frames, the Sequence Number field shall specify a DSN.

**7.2.3 Destination PAN ID field**

The Destination PAN ID field, when present, is an unsigned integer that specifies the unique PAN ID of the intended recipient of the frame. A value of the broadcast PAN ID, as defined in 6.1, in the Destination PAN ID field shall be accepted as a valid PAN ID by all devices currently listening to the channel.

**7.2.4 Destination Address field**

The Destination Address field, when present, with a length specified in the Destination Addressing Mode field of the Frame Control field as described in 7.2.1.8, specifies the address of the intended recipient of the frame. A value the broadcast short address, as defined in 6.1, in the Destination Address field shall be accepted as a valid address by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode field is nonzero.

### 7.2.5 Source PAN ID field

The Source PAN ID field, when present as described in 7.2.1.5, specifies the unique PAN ID of the originator of the frame.

The PAN ID of a device is initially determined during association on a PAN but may change following a PAN ID conflict resolution, as described in 6.3.2.

### 7.2.6 Source Address field

The Source Address field, when present, specifies the address of the originator of the frame. This field shall be included in the MAC frame only if the Source Addressing Mode field is nonzero.

### 7.2.7 Auxiliary Security Header field

The Auxiliary Security Header field specifies information required for security processing. This field shall be present only if the Security Enabled field is set to one. The formatting of the Auxiliary Security Header field is described in 9.4.

### 7.2.8 IE field

The IE field is variable length and contains one or more IE. This field is comprised of the Header IE and Payload IE subfields. This field shall be present only if the IE Present field in the Frame Control field is set to one. The format of the IE field is shown in Figure 7-3. Each IE consists of a descriptor and an optional payload as described in 7.4. This standard does not limit the number of IEs within the IE field.

Header IEs, if present, follow the Auxiliary Security Header and are part of the MHR. Header IEs, if present, may require termination, as defined in 7.4.1.

Payload IEs, if present, follow the MHR and are considered part of the MAC payload, i.e., they may be encrypted. A set of payload IEs may require termination, as described in 7.4.1.

| Octets: variable | ... | variable        | 0/2                      | variable         | ... | variable         | 0/2                       |
|------------------|-----|-----------------|--------------------------|------------------|-----|------------------|---------------------------|
| Header IE<br>#1  | ... | Header IE<br>#n | Header<br>Termination IE | Payload IE<br>#1 | ... | Payload IE<br>#m | Payload<br>Termination IE |

Figure 7-3—Format of IE field

### 7.2.9 Frame Payload field

The Frame Payload field contains information specific to individual frame types. If the Security Enabled field is set to one, the frame payload the frame may be cryptographically protected, as described in Clause 9.

### 7.2.10 FCS field

The FCS field contains a 16-bit ITU-T CRC or a 32-bit CRC equivalent to ANSI X3.66-1979.<sup>13</sup> The FCS is calculated over the MHR and MAC payload parts of the frame; these parts together are referred to as the calculation field. Devices compliant with one or more of the SUN PHYs or TVWS PHYs shall implement the 4-octet FCS.

<sup>13</sup>Information on references can be found in Clause 2.

The 2-octet FCS shall be calculated using the following standard generator polynomial:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1$$

The 2-octet FCS shall be calculated for transmission using the following algorithm:

- Let  $M(x) = b_0x^{k-1} + b_1x^{k-2} + \dots + b_{k-2}x + b_{k-1}$  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,  $R(x) = r_0x^{15} + r_1x^{14} + \dots + r_{14}x + r_{15}$ .
- 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.

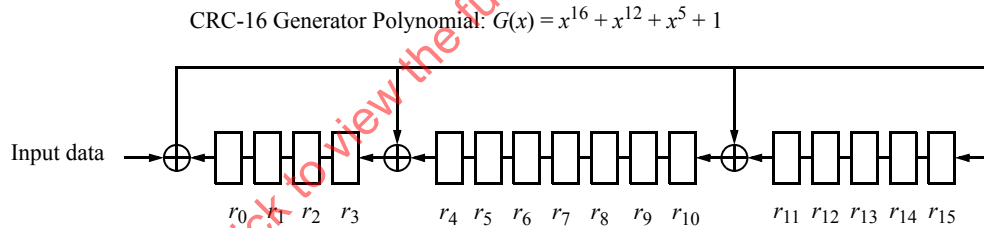
As an example, consider an Ack frame with no payload and the following 3 octet MHR:

0100 0000 0000 0000 0101 0110 [leftmost bit ( $b_0$ ) passed to the PHY first in time]  
 $b_0$ ..... $b_{23}$

The FCS for this case would be the following:

0010 0111 1001 1110 [leftmost bit ( $r_0$ ) passed to the PHY first in time]  
 $r_0$ ..... $r_{15}$

A typical implementation is depicted in Figure 7-4.



1. Initialize the remainder register ( $r_0$  through  $r_{15}$ ) to zero.
2. Shift MHR and payload into the divider in the order of that they are passed to the PHY.
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 passed to the PHY first

**Figure 7-4—Typical 2-octet FCS implementation**

The 4-octet FCS is calculated using the following standard generator polynomial of degree 32:

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

The 4-octet FCS is the one's complement of the (modulo 2) sum of the two remainders in a) and b):

- a) The remainder resulting from  $[(x^k \times (x^{31} + x^{30} + \dots))] \text{ divided (modulo 2) by } G_{32}(x)$ , where the value  $k$  is the number of bits in the calculation field.
- b) The remainder resulting from the calculation field contents, treated as a polynomial, is multiplied by  $x^{32}$  and then divided by  $G_{32}(x)$ .

At the transmitter, the initial remainder of the division shall be preset to all ones and then modified via division of the calculation field by the generator polynomial  $G_{32}(x)$ . The one's complement of this remainder is the 4-octet FCS field. The FCS field is passed to the PHY commencing with the coefficient of the highest order term.

At the receiver, the initial remainder shall be preset to all ones. The serial incoming bits of the calculation field and FCS, when divided by  $G_{32}(x)$  in the absence of transmission errors, result in a unique nonzero remainder value. The unique remainder value is the polynomial shown:

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

Upon transmission, if the length of the calculation field is less than 4 octets, the FCS computation shall assume padding the calculation field by appending zero value octets to the most significant bits to make the calculation field length exactly 4 octets; however, these pad bits shall not be transmitted. Upon reception, if the length of the calculation field is less than 4 octets, the received calculation field shall be appended with zero value octets to the most significant bits to make the calculation field length exactly 4 octets prior to computing the FCS for validation.

As an example, consider an Ack frame with no payload and the following 3-byte MHR:

0100 0000 0000 0000 0101 0110 [leftmost bit ( $b_0$ ) passed to the PHY first in time]  
 $b_0$ ..... $b_{23}$

Prior to FCS computation, the zero padded calculation field is given as follows:

0100 0000 0000 0000 0101 0110 0000 0000  
 $b_0$ ..... $b_{31}$

The 4-octet FCS for this case would be the following:

0101 1101 0010 1001 1111 1010 0010 1000 [leftmost bit ( $r_0$ ) passed to the PHY first in time]  
 $r_0$ ..... $r_{31}$

## 7.3 Format of individual frame types

### 7.3.1 Beacon frame format

The Beacon frame shall be formatted as illustrated in Figure 7-5.

The format of the Beacon frame for Frame Versions 0b00 or 0b01 shall be as illustrated in Figure 7-5. If the Frame Version is 0b00 or 0b01, the Superframe Specification, GTS, and Pending Address fields are always present. If the frame version is 0b10, the IE Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 7.2.1.7, 7.2.8, and 7.4.



IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

|               |                 |                   |                           |                          |          |                 |                |     |
|---------------|-----------------|-------------------|---------------------------|--------------------------|----------|-----------------|----------------|-----|
| Octets: 2     | 1               | 4/10              | variable                  | 2                        | variable | variable        | variable       | 2/4 |
| Frame Control | Sequence Number | Addressing fields | Auxiliary Security Header | Superframe Specification | GTS Info | Pending address | Beacon Payload | FCS |
| MHR           |                 |                   |                           | MAC Payload              |          |                 |                | MFR |

Figure 7-5—Beacon frame format

The Enhanced Beacon frame is differentiated from the Beacon by the frame version being set to 0b10. The MAC frame for the Enhanced Beacon frame shall be formatted as illustrated in Figure 7-6.

| Octets: 2     | 0/1             | variable          | variable                  | variable   |             | variable       | 2/4 |
|---------------|-----------------|-------------------|---------------------------|------------|-------------|----------------|-----|
| Frame Control | Sequence Number | Addressing fields | Auxiliary Security Header | IEs        |             | Beacon Payload | FCS |
|               |                 |                   |                           | Header IEs | Payload IEs |                |     |
| MHR           |                 |                   |                           |            | MAC Payload |                | MFR |

Figure 7-6—Enhanced Beacon frame format

#### 7.3.1.1 Beacon frame MHR field

The Frame Type field shall contain the value that indicates a Beacon frame, as shown in Table 7-1. The Security Enabled field shall be set to one if security is enabled and the Frame Version field is not zero. If a broadcast frame is pending, the Frame Pending field shall be set to one. If the Frame Version field is not 0b10, all other fields in the Frame Control field shall be set to zero and ignored on reception.

The Sequence Number field shall contain the current value of *macBsn* if it is a Beacon frame or *macEbsn* if it is an Enhanced Beacon frame. As a device may be sending both Beacon frames and Enhanced Beacon frames, separate sequence numbers shall be maintained. If sending an Enhanced Beacon frame, the sequence number may be suppressed by setting the Sequence Number Suppression field in the Frame Control field. When the frame version field is 0b00–0b01 the Sequence Number field shall be present.

When the Frame Version field is 0b00–0b01:

- The Source Addressing mode field shall be set to indicate that the Source Address and Source PAN ID fields are present.
- The Destination Addressing mode field shall be set to indicated that the Destination Address and Destination PAN ID fields are not present.
- The Source PAN ID field and Source Address field shall contain *macPanId* and the 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 7.2.7.

When the Frame Version field is 0b10:

- The MHR may contain a sequence number.
- The MHR may contain any addressing fields supported by the general frame format.
- The Source Address field shall contain the address of the device transmitting the beacon.

If the Enhanced Beacon frame is sent in response to an Enhanced Beacon Request command:

- If a PAN ID is required, then the Destination PAN ID field is set to the value of *macPanId* and the Source PAN ID field is omitted.
- The Destination Address field shall contain the source address contained in the received Enhanced Beacon Request command.
- The PAN ID Compression field is set as defined in Table 7-2.

### 7.3.1.2 IEs field

The IEs field may contain IEs.

### 7.3.1.3 Superframe Specification field

The Superframe Specification field shall be formatted as illustrated in Figure 7-7.

| 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 7-7—Format of the Superframe Specification field**

The Beacon Order field shall specify the transmission interval of the beacon. The relationship between the beacon order and the beacon interval is explained in 6.2.1.

The Superframe Order field shall specify the length of time during which the superframe is active (i.e., receiver enabled), including the Beacon frame transmission time. The relationship between the superframe order and the superframe duration is explained in 6.2.1.

The Final CAP Slot field specifies the final superframe slot utilized by the CAP. The duration of the CAP, as implied by this field, 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, as described in 7.3.1.4.

The Battery Life Extension (BLE) field 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 field shall be set to zero.

The PAN Coordinator field shall be set to one if the Beacon frame is being transmitted by the PAN coordinator. Otherwise, the PAN Coordinator field shall be set to zero.

The Association Permit field 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.

**7.3.1.4 GTS Info field**

The GTS Info field shall be formatted as illustrated in Figure 7-8.

| Octets: 1         | 0/1            | variable |
|-------------------|----------------|----------|
| GTS Specification | GTS Directions | GTS List |

**Figure 7-8—Format of the GTS information fields**

The GTS Specification field shall be formatted as illustrated in Figure 7-9.

| Bits: 0–2            | 3–6      | 7          |
|----------------------|----------|------------|
| GTS Descriptor Count | Reserved | GTS Permit |

**Figure 7-9—Format of the GTS Specification field**

The GTS Descriptor Count field specifies the number of 3-octet GTS descriptors contained in the GTS List field of the Beacon frame. If the value of this field 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 field. If the value of this field is zero, the GTS Directions field and GTS List field of the Beacon frame are not present.

The GTS Permit field 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.

The GTS Directions field shall be formatted as illustrated in Figure 7-10.

| Bits: 0–6           | 7        |
|---------------------|----------|
| GTS Directions Mask | Reserved |

**Figure 7-10—Format of the GTS Directions field**

The GTS Directions Mask field is 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.

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 shall be formatted as illustrated in Figure 7-11.

| Bits: 0–15           | 16–19             | 20–23      |
|----------------------|-------------------|------------|
| Device Short Address | GTS Starting Slot | GTS Length |

**Figure 7-11—Format of the GTS descriptor**

The Device Short Address field shall contain the short address of the device for which the GTS descriptor is intended.

The GTS Starting Slot field contains the superframe slot at which the GTS is to begin.

The GTS Length field contains the number of contiguous superframe slots over which the GTS is active.

### 7.3.1.5 Pending Address field

The Pending Address field shall be formatted as illustrated in Figure 7-12.

| Octets: 1                     | variable     |
|-------------------------------|--------------|
| Pending Address Specification | Address List |

**Figure 7-12—Pending Address field format**

The Pending Address Specification field shall be formatted as illustrated in Figure 7-13.

| Bits: 0–2                         | 3        | 4–6                                  | 7        |
|-----------------------------------|----------|--------------------------------------|----------|
| Number of Short Addresses Pending | Reserved | Number of Extended Addresses Pending | Reserved |

**Figure 7-13—Format of the Pending Address Specification field**

The Number of Short Addresses Pending field indicates the number of short addresses contained in the Address List field of the Beacon frame.

The Number of Extended Addresses Pending field indicates the number of extended addresses contained in the Address List field of the Beacon frame.

The size of the Address List field is determined by the values specified in the Pending Address Specification field. The Address List field contains the set of addresses of the devices that currently have messages pending with the coordinator. The address list shall not contain the broadcast short address.

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.

### 7.3.1.6 Beacon Payload field

The Beacon Payload field is an optional sequence octets to be transmitted, containing the value from *macBeaconPayload* set by the next higher layer.

### 7.3.2 Data frame format

The Data frame shall be formatted as illustrated in Figure 7-14.

| Octets: 2     | 0/1             | variable          | variable                  | variable   |             | variable     | 2/4 |
|---------------|-----------------|-------------------|---------------------------|------------|-------------|--------------|-----|
| Frame Control | Sequence Number | Addressing fields | Auxiliary Security Header | IEs        |             | Data Payload | FCS |
|               |                 |                   |                           | Header IEs | Payload IEs |              |     |
| MHR           |                 |                   |                           |            | MAC Payload |              | MFR |

Figure 7-14—Data frame format

#### 7.3.2.1 Data frame MHR field

The Frame Type field shall contain the value that indicates a Data frame, as shown in Table 7-1. The Security Enabled field shall be set to one if security is enabled and the Frame Version field is not zero. All other fields in the Frame Control field shall be set appropriately according to the intended use of the Data frame.

The Sequence Number field, if present, shall contain the current value of *macDsn*. If the frame version is 0b10, the sequence number may be suppressed by setting the Sequence Number Suppression field in the Frame Control field.

The Addressing fields comprise the destination address fields and/or the source address fields, depending 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 7.2.7.

If the frame version is 0b10, the IE Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 7.2.1.7, 7.2.8, and 7.4.

#### 7.3.2.2 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.

### 7.3.3 Ack frame format

The Ack frame format is used for both an Imm-Ack frame or Enh-Ack frame. The type of Ack frame is determined by the value of the Frame Version field. A Frame Version field values 0b00–0b01 indicates an Imm-Ack frame while a Frame Version field value of 0b10 indicates an Enh-Ack frame.

The Imm-Ack frame shall be formatted as illustrated in Figure 7-15.

| Octets: 2     | 1               | 2/4 |
|---------------|-----------------|-----|
| Frame Control | Sequence Number | FCS |
| MHR           |                 | MFR |

Figure 7-15—Imm-Ack frame format

The Enh-Ack frame shall be formatted as illustrated in Figure 7-16.

| Octets:2      | 0/1             | 0/2                | 0/2/8               | 0/2           | 0/2/8          | variable                  | variable    |             | variable      | 2/4 |
|---------------|-----------------|--------------------|---------------------|---------------|----------------|---------------------------|-------------|-------------|---------------|-----|
| Frame Control | Sequence Number | Destination PAN ID | Destination Address | Source PAN ID | Source Address | Auxiliary Security Header | IE          |             | Frame Payload | FCS |
|               |                 | Addressing fields  |                     |               |                |                           | Header IEs  | Payload IEs |               |     |
| MHR           |                 |                    |                     |               |                |                           | MAC Payload |             | MFR           |     |

**Figure 7-16—Enh-Ack frame format**

If the Ack frame is being sent in response to a received Data Request command the Frame Pending field shall be set as indicated in 6.7.3.

If the Ack frame is being sent in response to either a Data frame or another type of MAC command, the device shall set the Frame Pending field to zero.

The AR field in an Ack frame shall be set to zero.

In an Imm-Ack frame, all other fields in the Frame Control field shall be set to zero.

In an Enh-Ack frame, the other fields in the Frame Control fields shall be set as follows:

- If *macSecurityEnabled* is TRUE, the Security Enabled field shall be set as defined in 6.7.2 and shall be set to zero otherwise.
- The PAN ID compression field and Sequence Number Compression field shall be set to be the same value as the corresponding fields in the Frame Control field of the frame that is being acknowledged.
- The IEs present field shall be set to one if IEs are included in the Enh-Ack frame.
- The Destination Addressing Mode field shall contain the value of the Source Addressing Mode field in the Frame Control field of the frame that is being acknowledged.
- The Source Addressing mode field shall be set as appropriate for the address of the device transmitting the Enh-Ack frame, as described in Table 7-3.

The Source Address field, when present, contains the address of the device originating the Enh-Ack frame. The Source PAN ID field, when present, contains the PAN ID corresponding to the device originating the Enh-Ack frame.

The Destination Address field, when present, shall contain the value of the Source Address field from the frame that is being acknowledged. The Destination PAN ID, when present, contains the value from the Source PAN ID field from the frame that is being acknowledged.

If the Security Enabled field in the Frame Control field is set to one, the Enh-Ack frame is protected using the procedure described in 9.2.1. The SecurityLevel, KeyIdMode, KeySource and KeyIndex shall be set to match the corresponding fields of the frame that is being acknowledged.

The Sequence Number field, when present shall contain the value of the sequence number received in the frame for which the acknowledgment is to be sent.

### 7.3.4 MAC command frame format

The MAC command shall be formatted as illustrated in Figure 7-17.

| Octets: 2     | 0/1             | variable          | variable                  | variable   |             | 1          | variable | 2/4 |
|---------------|-----------------|-------------------|---------------------------|------------|-------------|------------|----------|-----|
| Frame Control | Sequence Number | Addressing fields | Auxiliary Security Header | IE         |             | Command ID | Content  | FCS |
|               |                 |                   |                           | Header IEs | Payload IEs |            |          |     |
| MHR           |                 |                   |                           |            | MAC Payload |            |          |     |

Figure 7-17—MAC command frame format

#### 7.3.4.1 MHR field

The Sequence Number field, if present, shall contain the current value of *macDsn*. If the Frame Version field is 0b10, the sequence number may be suppressed by setting the Sequence Number Suppression field in the Frame Control field. If the Frame Version field is 0b00 or 0b01 the Sequence Number field shall be present.

The addressing fields shall comprise the destination address fields and/or the source address fields, depending on the settings in the Frame Control field. If the Frame Version field is 0b10, then either address may be suppressed.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the MAC command, as specified in 7.2.7.

If the frame version is 0b10, the IE Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 7.2.1.7, 7.2.8, and 7.4.

#### 7.3.4.2 Command ID field

The Command ID field identifies the MAC command being used. Valid values of the Command ID field are defined in Table 7-49.

#### 7.3.4.3 Payload field

The Payload field contains the MAC command itself. The formats of the individual commands are described in 7.5.

### 7.3.5 Multipurpose frame format

The Multipurpose frame type provides a flexible format that may be used for a variety of purposes. The format supports a short and long form of the Frame Control field, and allows for all the fields in the MHR to be present or omitted as specified by the generating service. Multipurpose frames are treated in the same manner as Data frames, and their content is passed to/from the next higher layer using the MCPS DATA primitives. Some suggested uses are described in “Applications of IEEE Std 802.15.4” [B3].



The multipurpose frame shall be formatted as illustrated in Figure 7-18.

| Octets:<br>1/2      | 0/1                | 0/2                   | 0/2/8                  | 0/2/8             | variable                        | variable    |             | variable         | 2/4 |
|---------------------|--------------------|-----------------------|------------------------|-------------------|---------------------------------|-------------|-------------|------------------|-----|
| MP Frame<br>Control | Sequence<br>Number | Destination<br>PAN ID | Destination<br>Address | Source<br>Address | Auxiliary<br>Security<br>Header | IE          |             | Frame<br>Payload | FCS |
|                     |                    | Addressing fields     |                        |                   |                                 | Header IEs  | Payload IEs |                  |     |
| MHR                 |                    |                       |                        |                   |                                 | MAC Payload |             | MFR              |     |

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

The MP Frame Control field shall be formatted as illustrated in Figure 7-19.

| Bits: 0–2  | 3                  | 4–5                         | 6–7                    | 8              | 9                | 10                          | 11            | 12–13         | 14          | 15         |
|------------|--------------------|-----------------------------|------------------------|----------------|------------------|-----------------------------|---------------|---------------|-------------|------------|
| Frame Type | Long Frame Control | Destination Addressing Mode | Source Addressing Mode | PAN ID Present | Security Enabled | Sequence Number Suppression | Frame Pending | Frame Version | Ack Request | IE Present |

**Figure 7-19—Format of the MP Frame Control field**

#### 7.3.5.1 Frame Type field

The Frame Type field shall contain the value that indicates a Multipurpose frame, as shown in Table 7-1.

#### 7.3.5.2 Long Frame Control field

The Long Frame Control field shall be set to zero to indicate an MP Short Frame Control field (only bits 0 to 7 in Figure 7-19 make up the field), and to one to indicate an MP Long Frame Control field (bits 0 to 15 in Figure 7-19 make up the field).

#### 7.3.5.3 Destination Addressing Mode field

The Destination Addressing Mode field shall be set to one of the values listed in Table 7-3. The length of the Destination Address field in the MHR shall correspond to the value of the Destination Addressing Mode field.

#### 7.3.5.4 Source Addressing Mode field

The Source Addressing Mode field shall be set to one of the values listed in Table 7-3. The length of the Source Address field in the MHR shall correspond to the value of the Source Addressing Mode field.

#### 7.3.5.5 PAN ID Present field

The PAN ID Present field is present only if the Long Frame Control field is set to one. It shall be set to one if the destination PAN ID is present in the MHR, otherwise it is set to zero and the PAN ID is not present in the MHR.

The Source PAN ID, not present in the frame, is the same as the destination PAN ID.

**7.3.5.6 Security Enabled field**

The Security Enabled field shall be set to one if the frame is protected by the MAC sublayer and shall be set to zero otherwise.

**7.3.5.7 Sequence Number Suppression field**

The Sequence Number Suppression field shall be set to one if the Sequence Number field is not to be included in the MHR. Otherwise it shall be set to zero.

**7.3.5.8 Frame Pending field**

The Frame Pending field shall be set to one if the device sending the frame has more data for the recipient. This field shall be set to zero otherwise.

**7.3.5.9 Frame Version field**

The Frame Version field is an unsigned integer that specifies the version number of the frame. This field shall be set to zero.

**7.3.5.10 Ack Request field**

The Ack Request field shall be set to one if an acknowledgment is required from the recipient device on receipt of a valid frame. Otherwise it shall be set to zero.

**7.3.5.11 IEs Present field**

The IEs Present field shall be set to one if IEs are to be included in the frame. Otherwise it shall be set to zero.

**7.3.5.12 Sequence Number field**

The Sequence Number field specifies the sequence identifier for the frame, as described in 7.2.2. It is normally present, but can be suppressed using the Sequence Number Suppression field as described in 7.3.5.7.

**7.3.5.13 Destination PAN ID field**

The Destination PAN ID field specifies the unique PAN ID of the intended recipient of the frame, as described in 7.2.3.

**7.3.5.14 Destination Address field**

The Destination Address field, when present, specifies the address of the intended recipient of the frame.

**7.3.5.15 Source Address field**

The Source Address field, when present, specifies the address of the originator of the frame.

**7.3.5.16 Auxiliary Security Header field**

The Auxiliary Security Header field 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, as defined in 9.5. This field shall be present only if the Security Enabled field is set to one. For details on formatting, refer to 9.4.

### 7.3.5.17 IEs field

The IEs field is present if the IEs Present field is set to one. The format of IEs is specified in 7.4. Each type of IEs is terminated as required per 7.4.1.

### 7.3.5.18 Payload field

The Payload field has a variable length and contains unformatted (not in an IE) payload.

### 7.3.6 Extended frame format

The Extended frame shall be formatted as illustrated in Figure 7-20.

| Bits: 0–2  | 3–5                 | variable               |
|------------|---------------------|------------------------|
| Frame Type | Extended Frame Type | Extended Frame Payload |

**Figure 7-20—Extended frame format**

The Frame Type field is defined in 7.2.1.1.

The Extended Frame Type field is used to specify additional frame types by adding additional bits to the Frame Type field. Valid values of the Extended Frame Type field are give in Table 7-5.

**Table 7-5—Extended Frame Type values**

| Extended Frame Type<br>b5 b4 b3 | Description                                               |
|---------------------------------|-----------------------------------------------------------|
| 000–011                         | Reserved                                                  |
| 111                             | Assigned to Telecommunications Industry Association (TIA) |

The Extended Frame Payload field is defined by the frame type indicated by the Extended Frame Type field or by the organization to which the Extended Frame Type has been assigned. Currently, this standard does not have any frame formats defined for the reserved values Extended Frame Type fields.

## 7.4 IEs

### 7.4.1 IE list termination

When determining if a termination of the Header IEs or Payload IEs is required, the authentication or encryption procedures are not taken into account. Table 7-6 lists the rules for the inclusion of terminations IEs. In the table, HT1 refers to Header Termination 1 IE, as defined in 7.4.2.17, while HT2 refers to Header Termination 2 IE, as defined in 7.4.2.18.

NOTE—A header termination IE is required prior to the payload IEs because the address space of payload IEs and header IEs is not distinct. Hence it is not possible to detect when the header IEs stop and payload IEs begin.

**Table 7-6—Termination IE inclusion rules**

| Header IEs present | Payload IEs present | Data payload present | Header Termination IE included | Payload Termination IE included | Notes                                                                                                           |
|--------------------|---------------------|----------------------|--------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------|
| No                 | No                  | No                   | None                           | None                            | No IE lists present, hence no termination. Included here for completeness                                       |
| Yes                | No                  | No                   | None                           | None                            | No termination is required because the end of the frame can be determined from the frame length and FCS length. |
| No                 | Yes                 | No                   | HT1 required                   | Optional                        | Header Termination IE 1 is required to signal end of the MHR and beginning of the Payload IE list.              |
| Yes                | Yes                 | No                   | HT1 required                   | Optional                        | Header IE Termination 1 is required while the Payload IE Termination is not required, but is allowed.           |
| No                 | No                  | Yes                  | None                           | None                            | No IE lists present, hence no termination. Included here for completeness                                       |
| Yes                | No                  | Yes                  | HT2 required                   | None                            | Header Termination IE 2 is used in to signal end of the MHR and beginning of the MAC Payload.                   |
| No                 | Yes                 | Yes                  | HT1 required                   | Required                        | —                                                                                                               |
| Yes                | Yes                 | Yes                  | HT1 required                   | Required                        | —                                                                                                               |

**7.4.2 Header IEs****7.4.2.1 Header IE format**

The Header IE shall be formatted as illustrated in Figure 7-21.

| Bits: 0–6 | 7–14       | 15       | Octets: 0–127 |
|-----------|------------|----------|---------------|
| Length    | Element ID | Type = 0 | Content       |

**Figure 7-21—Format of Header IEs**

The Length field specifies the number of octets in the Content field.

The Element IDs are defined in Table 7-7 for each of the Header IEs. In the table, an X in the columns for frame indicates that the IE may be used in that frame. If there is not an X in the column, then the IE shall not be used in that frame. In the table, UL refers to the layers above the MAC.

Table 7-7—Element IDs for Header IEs

| Element ID | Name                                            | Enhanced Beacon | Enhanced ACK | Data | Multipurpose | MAC command | Format subclause | Use description  | Used by | Created by |
|------------|-------------------------------------------------|-----------------|--------------|------|--------------|-------------|------------------|------------------|---------|------------|
| 0x00       | Vendor Specific Header IE                       | X               | X            | X    | X            | X           | 7.4.2.2          | —                | UL      | UL         |
| 0x01–0x19  | Reserved                                        |                 |              |      |              |             |                  |                  |         |            |
| 0x1a       | CSL IE                                          | X               | X            | X    | X            |             | 7.4.2.3          | 6.12.2           | MAC     | MAC        |
| 0x1b       | RIT IE                                          | X               |              | X    |              | X           | 7.4.2.4          | 6.12.3           | MAC     | MAC        |
| 0x1c       | DSME PAN descriptor IE                          | X               |              |      |              |             | 7.4.2.5          | 6.11.2           | UL, MAC | UL         |
| 0x1d       | Rendezvous Time IE                              |                 | X            |      | X            |             | 7.4.2.6          | 6.12.2           | MAC     | MAC        |
| 0x1e       | Time Correction IE                              |                 | X            |      |              |             | 7.4.2.7          | 6.5.4.1, 6.7.4.2 | MAC     | MAC        |
| 0x1f–0x20  | Reserved                                        |                 |              |      |              |             |                  |                  |         |            |
| 0x21       | Extended DSME PAN descriptor IE                 | X               |              |      |              |             | 7.4.2.8          | 6.11.2           | UL, MAC | UL         |
| 0x22       | Fragment Sequence Context Description (FSCD) IE |                 |              |      | X            | X           | 7.4.2.9          | 23.3.1           | MAC     | MAC        |
| 0x23       | Simplified Superframe Specification IE          | X               |              |      |              |             | 7.4.2.10         | 6.2.3, [B3]      | MAC     | MAC        |
| 0x24       | Simplified GTS Specification IE                 | X               |              |      |              |             | 7.4.2.11         | 6.2.3, [B3]      | MAC     | MAC        |
| 0x25       | LECIM Capabilities IE                           | X               |              | X    | X            | X           | 7.4.2.12         | 10.1.2.10        | UL      | UL         |
| 0x26       | TRLE Descriptor IE                              | X               | X            | X    | X            | X           | F.5.1.1          | F.4.2, F.4.3     | MAC     | MAC        |
| 0x27       | RCC Capabilities IE                             | X               |              | X    | X            |             | 7.4.2.13         | 6.2.9, [B3]      | UL      | UL         |
| 0x28       | RCCN Descriptor IE                              | X               |              |      |              |             | 7.4.2.14         | 6.2.9, [B3]      | UL, MAC | UL         |
| 0x29       | Global Time IE                                  | X               |              |      |              |             | 7.4.2.15         |                  | UL      | UL         |
| 0x2a       | Assigned to external organization [B1]          |                 |              |      |              |             |                  |                  |         |            |
| 0x2b       | DA IE                                           | X               |              |      |              |             | 7.4.2.16         | 6.7.9            | UL      | UL         |
| 0x2b–0x7d  | Reserved                                        |                 |              |      |              |             |                  |                  |         |            |
| 0x7e       | Header Termination 1 IE                         | X               | X            | X    | X            | X           | 7.4.2.17         | 7.4.1            | MAC     | MAC        |
| 0x7f       | Header Termination 2 IE                         | X               | X            | X    | X            | X           | 7.4.2.18         | 7.4.1            | MAC     | MAC        |
| 0x80–0xff  | Reserved                                        |                 |              |      |              |             |                  |                  |         |            |

#### 7.4.2.2 Vendor Specific Header IE

The Vendor Specific Header IE is reserved for the use of other protocols and/or data relevant only to certain implementations. The Vendor Specific Header IE Content field shall be formatted as illustrated in Figure 7-22.

|                  |                             |
|------------------|-----------------------------|
| <b>Octets: 3</b> | <b>Variable</b>             |
| Vendor OUI       | Vendor Specific Information |

**Figure 7-22—Vendor Specific Header IE Content field format**

The Vendor OUI field is an organizationally unique identifier (OUI) or company identifier (CID) assigned by the IEEE registration authority, which shall be the sole registration authority. A value of the Vendor OUI field not understood by a receiving device causes the remainder of this IE to be ignored.

The Vendor Specific Information field is defined by the vendor identified in the Vendor OUI field. Its use is outside of the scope of this standard.

#### 7.4.2.3 CSL IE

The CSL IE shall be used in all Enh-Ack frames if *macLeEnabled* is TRUE.

The CSL IE Content field shall be formatted as illustrated in Figure 7-23.

|                  |            |                 |
|------------------|------------|-----------------|
| <b>Octets: 2</b> | <b>2</b>   | <b>0/2</b>      |
| CSL Phase        | CSL Period | Rendezvous Time |

**Figure 7-23—CSL IE Content field format**

The CSL Phase field shall be set to CSL phase in units of 10 symbols.

The CSL Period field shall be set to CSL period in units of 10 symbols.

The Rendezvous Time field is only present in the transmitted IE when *macCsInterval* is nonzero. The Rendezvous Time field contains the expected time, in units of 10 symbols, between the end of the transmission of the wake-up frame and the beginning of the transmission of the payload frame.

#### 7.4.2.4 RIT IE

The RIT IE Content field shall be formatted as illustrated in Figure 7-24. The use of the RIT IE is described in 6.3.4.

|                      |                         |                        |
|----------------------|-------------------------|------------------------|
| <b>Octets: 1</b>     | <b>1</b>                | <b>2</b>               |
| Time To First Listen | Number of Repeat Listen | Repeat Listen Interval |

**Figure 7-24—RIT IE Content field format**

The Time to First Listen field shall be set to the time interval in units of *aBaseSuperframeDuration*, until the first time that the device will be receiving, as defined in 6.12.3.2.

The Number of Repeat Listen field shall be set to the number of times that the listening interval will be repeated, as defined in 6.12.3.2.

Repeat Listen Interval shall be set to the time interval, in units of *aBaseSuperframeDuration*, between subsequent intervals during which the device will be in receiving, as defined in 6.12.3.2.

#### 7.4.2.5 DSME PAN descriptor IE

The DSME PAN Descriptor IE shall be included in enhanced beacons that are sent every beacon interval in a DSME-enabled PAN.

The DSME PAN Descriptor IE Content field shall be formatted as illustrated in Figure 7-25.

| Octets: 2                | variable        | 1                             | 8                                  | variable      | variable                      |
|--------------------------|-----------------|-------------------------------|------------------------------------|---------------|-------------------------------|
| Superframe Specification | Pending Address | DSME Superframe Specification | Time Synchronization Specification | Beacon Bitmap | Channel Hopping Specification |

**Figure 7-25—DSME PAN Descriptor IE Content field format**

The Superframe Specification field is described in 7.3.1.3.

The Pending Address field is described in 7.3.1.5.

The DSME Superframe Specification field shall be formatted as illustrated in Figure 7-26.

| Bits: 0–3              | 4                      | 5        | 6             | 7               |
|------------------------|------------------------|----------|---------------|-----------------|
| Multi-superframe Order | Channel Diversity Mode | Reserved | CAP Reduction | Deferred Beacon |

**Figure 7-26—DSME Superframe Specification field format**

The Multi-superframe Order field shall specify the length of time during which a group of superframes that is considered as one multi-superframe (i.e., receiver enabled). See 6.11.2 for an explanation of the relationship between the multi-superframe order and the multi-superframe duration.

The Channel Diversity Mode field shall indicate the type of channel diversity. If this value is zero, the DSME-enabled PAN operates on channel adaptation mode. If this value is one, the DSME-enabled PAN operates on channel hopping mode.

The CAP Reduction field shall be set to one if the CAP reduction is enabled, otherwise the CAP Reduction field shall be set to zero.

The Deferred Beacon shall be set to one if the device uses CCA before transmitting Beacon frame, otherwise the bit shall be set to zero if the device shall not use CCA before transmitting beacon.

The Time Synchronization Specification field shall be formatted as illustrated in Figure 7-27.

| Octets: 6        | 2                       |
|------------------|-------------------------|
| Beacon Timestamp | Beacon Offset Timestamp |

**Figure 7-27—Format of the Time Synchronization Specification field**

The Beacon Timestamp field shall specify the start time of the beacon slot in which the beacon was transmitted in units of microseconds.

The Beacon Offset Timestamp field specifies that the time difference between the start time of a superframe and the actual time of transmitting a Beacon frame. It reflects the delay due to CCA or processing time of the PHY to transmit a frame.

The Beacon Bitmap field shall be formatted as illustrated in Figure 7-28.

|                  |                  |                 |
|------------------|------------------|-----------------|
| <b>Octets: 2</b> | <b>2</b>         | <b>variable</b> |
| SD Index         | SD Bitmap Length | SD Bitmap       |

**Figure 7-28—Format of the Beacon Bitmap field**

The SD Index field specifies the index of current SD in a beacon interval. The superframe in which the PAN coordinator sends its beacons serves as the reference point (SD Index 0). When this field is contained in an enhanced beacon, this field specifies the index of the superframe that is allocated to the Source device of the beacon for beacon transmission.

The SD Bitmap Length field shall be set to the length of SD Bitmap field in octets.

The SD Bitmap field contains a bitmap indicating the Beacon frame allocation information of all  $2^{(BO-SO)}$  superframes within one beacon interval. Each corresponding bit in the bitmap shall be set to one if a beacon of a neighbor device is allocated in that SD, otherwise it is set to zero. It should be noted that the length of the Beacon frame allocation information bitmap,  $2^{(BO-SO)}$ , may be different than the value of the SD Bitmap Length field since the SD Bitmap Length field is represented in octets. In this case, the value of the SD Bitmap Length field shall be chosen to be the smallest integer that is greater than the length of this bitmap, and those bits after the first  $2^{(BO-SO)}$  bits in the SD Bitmap field shall be set to zero.

The Channel Hopping Specification field is present only if the value of Channel Diversity Mode field in DSME Superframe Specification is set to indicate channel hopping. The Channel Hopping Specification field shall be formatted as illustrated in Figure 7-29.

|                     |                     |                |                              |                       |
|---------------------|---------------------|----------------|------------------------------|-----------------------|
| <b>Octets: 1</b>    | <b>1</b>            | <b>2</b>       | <b>1</b>                     | <b>variable</b>       |
| Hopping Sequence ID | PAN Coordinator BSN | Channel Offset | Channel Offset Bitmap Length | Channel Offset Bitmap |

**Figure 7-29—Format of the Channel Hopping Specification field**

The Hopping Sequence ID field contains the ID of the channel hopping sequence in use. The Hopping Sequence ID field set to zero indicates that a default hopping sequence shall be used. A Hopping Sequence ID field set to one indicates that a hopping sequence generated by the PAN coordinator shall be used. The other values of Hopping Sequence ID denote the sequence is set by a higher layer. If the Hopping Sequence ID is one, a device shall request a channel hopping sequence from its coordinator when it associates to a PAN.

The PAN Coordinator BSN field shall contain the BSN of the PAN coordinator.

The Channel Offset field shall contain the channel hopping offset value of the device.

The Channel Offset Bitmap Length field shall be set to the length of ChannelOffsetBitmap field in octets.



The Channel Offset Bitmap field shall indicate the occupancy of the channel hopping offset values among neighbor devices and be represented in that bitmap. Each bit shall be set to one if the corresponding channel hopping offset value is already occupied by the neighbor devices, otherwise it shall be set to zero if the corresponding channel hopping value is not occupied. For instance, a Channel Offset Bitmap of 0b1100100..0 indicates that channel hopping offset values of 0, 1, and 4 are being used by neighbor devices. Note that the  $(i)^{\text{th}}$  bit in the Channel Offset Bitmap corresponds to  $(i-1)^{\text{th}}$  channel offset value.

#### 7.4.2.6 Rendezvous Time IE

The Rendezvous Time IE Content field shall be formatted as illustrated in Figure 7-30.

|                 |                  |
|-----------------|------------------|
| Octets: 2       | 2                |
| Rendezvous Time | Wake-up Interval |

Figure 7-30—Rendezvous Time IE Content field format

The Rendezvous Time field is the expected time in units of 10 symbols between the end of the transmission of the wake-up frame and the beginning of the transmission of the payload frame. The last wake-up frame in a wake-up sequence shall have Rendezvous Time field set to the value zero.

The Wake-up Interval field is only present in the transmitted IE when *macCslInterval* is nonzero. The Wake-up Interval field is the length of the interval between two successive LE wake-up frames in the wake-up sequence, in units of 10 symbols. The Wake-up Interval field shall be set to *macCslInterval*.

#### 7.4.2.7 Time Correction IE

The Time Correction IE Content field shall be formatted as illustrated in Figure 7-31.

|                |
|----------------|
| Octets: 2      |
| Time Sync Info |

Figure 7-31—Time Correction IE Content field format

The Time Sync Info field shall specify time synchronization information and acknowledgment status. This field is constructed by taking a signed 16-bit twos-complement time correction in the range of  $-2048 \mu\text{s}$  to  $2047 \mu\text{s}$ , AND'ing it with 0xfff, and OR'ing again with 0x8000 to indicate a negative acknowledgment. This field shall be set as indicated in Table 7-8.

Table 7-8—Values of the Time Sync Info field for ACK with timing Information

| Range         | Description                                                           |
|---------------|-----------------------------------------------------------------------|
| 0x000–0x7ff   | Acknowledge with positive time correction in microseconds             |
| 0x800–0xffff  | Acknowledge with negative time correction in microseconds             |
| 0x8000–0x87ff | Negative acknowledgment with positive time correction in microseconds |
| 0x8800–0x8fff | Negative acknowledgment with negative time correction in microseconds |

**7.4.2.8 Extended DSME PAN descriptor IE**

The format of the Extended DSME PAN Descriptor IE Content field shall be as illustrated in Figure 7-32.

| Octets: 2                | variable        | 2                                      | 8                                  | variable      | variable                      | 0/1                     | variable         |
|--------------------------|-----------------|----------------------------------------|------------------------------------|---------------|-------------------------------|-------------------------|------------------|
| Superframe Specification | Pending Address | Extended DSME Superframe Specification | Time Synchronization Specification | Beacon Bitmap | Channel Hopping Specification | Hopping Sequence Length | Hopping Sequence |

**Figure 7-32—Extended DSME PAN Descriptor IE Content field format**

The Superframe Specification field is described in 7.3.1.3.

The Pending Address field is described in 7.3.1.5.

The Extended DSME Superframe Specification field shall be formatted as illustrated in Figure 7-33.

| Bits: 0–7              | 8                      | 9        | 10            | 11              | 12                       | 13–15    |
|------------------------|------------------------|----------|---------------|-----------------|--------------------------|----------|
| Multi-superframe Order | Channel Diversity Mode | Reserved | CAP Reduction | Deferred Beacon | Hopping Sequence Present | Reserved |

**Figure 7-33—Format of the Extended DSME Superframe Specification field**

The Multi-superframe Order field is defined in 7.4.2.5.

The Channel Diversity Mode field is defined in 7.4.2.5.

The CAP Reduction field is defined in 7.4.2.5.

The Deferred Beacon field is defined in 7.4.2.5.

The Hopping Sequence Present field shall be set to one if an Association Request command is received before the Enhanced Beacon frame transmission and the Hopping Sequence ID of one is used in the DSME-enabled PAN.

The Time Synchronization Specification field is described in 7.4.2.5.

The Beacon Bitmap field is described in 7.4.2.5.

The Channel Hopping Specification field is described in 7.4.2.5 and is present only if the Channel Diversity Mode field in the Extended DSME Superframe Specification is set to indicate channel hopping.

The Hopping Sequence Length field is described in 7.4.2.5 and is present only if the Hopping Sequence Present field of the Extended DSME Superframe Specification field is one.

The Hopping Sequence field is described in 7.4.2.5 and is present only if the Hopping Sequence Present field of the Extended DSME Superframe Specification field is one.

**7.4.2.9 Fragment Sequence Context Description (FSCD) IE**

The FSCD IE Content field shall be formatted as illustrated in Figure 7-34.

| Octets: 2 |                 |          |      |             |             | 2         |                        |     |
|-----------|-----------------|----------|------|-------------|-------------|-----------|------------------------|-----|
| Bits: 0   | 1               | 2–6      | 7–12 | 13–14       | 15          | 16–25     | 26–31                  |     |
| Reserved  | Secure Fragment | Reserved | TID  | Frak Policy | FICS Length | PSDU Size | Addressing Information | ... |

|     | variable   | 0/4          |          |
|-----|------------|--------------|----------|
|     |            | Bits: 0–25   | 26–31    |
| ... | Addressing | PSDU Counter | Reserved |

**Figure 7-34—FSCD IE Content field format**

The Secure Fragment field is used to indicate whether the fragments in this transaction will be sent with authentication. When set, the PSDU Counter field shall be present in the FSCD IE, and the fragment validation field shall be set to the MIC, as described in 23.3.4. The field shall be set to one when *phyPSDUFragSecure* is set to TRUE.

A transaction identifier (TID) field value of zero indicates that the TID field will be a reserved field in the fragments that follow. When the TID field value is nonzero, the value identifies the fragment sequence. It associates the context information with each fragment in the transaction. The MAC sets the TID value and should assure that the current value is different from any other active transactions.

The fragment acknowledgment (Frak) Policy field shall be set to one of the values given in Table 23-4.

If Security Fragment field is set to one, then the FICS length field specifies the length of the Frak Validation field in the Frak frames. The FICS Length field shall be set to zero if a 2-octet FICS is used and shall be set to one if the 4-octet FICS is used. If Security Fragment field is set to zero, the PSDU Counter field and the Reserved field of the FSCD Content field shall not be present.

The PSDU Size field contains the number of octets in the PSDU.

The Addressing Information field shall be formatted as illustrated in Figure 7-35.

| Bit: 0                | 1                          | 2–3                 | 4–5                      |
|-----------------------|----------------------------|---------------------|--------------------------|
| Source PAN ID Present | Destination PAN ID Present | Source Address Mode | Destination Address Mode |

**Figure 7-35—Addressing Information field format**

The setting of the Addressing Information field shall be determined by the PAN ID and addressing mode fields of the PSDU being fragmented.

The Source PAN ID Present field shall be set to one if the source PAN ID is included in the Addressing field and shall be set to zero otherwise.

The Destination PAN ID Present field shall be set to one if the destination PAN ID is included in the Addressing field and shall be set to zero otherwise.

The Source Address Mode field shall be set to one of the values given in Table 7-3.

The Destination Address Mode field shall be set to one of the values given in Table 7-3.

The Addressing field contains source and/or destination addressing information associated with the MPDU being fragmented and shall be formatted as illustrated in Figure 7-36.

| Octets: 0/2   | 0/2                | 0/2/8          | 0/2/8               |
|---------------|--------------------|----------------|---------------------|
| Source PAN ID | Destination PAN ID | Source Address | Destination Address |

**Figure 7-36—Addressing field format**

The content of this field shall be set according to the addresses contained in the MHR of the PSDU being fragmented. Addresses may be omitted to fit into the PSDU size of the PHY in use; algorithms for address suppression are implementation dependent.

The PSDU Counter field shall be present when the Secure Fragment field is set to one. The MAC shall maintain a counter that is incremented with each fragmentation transaction, initiated such that the counter value is not repeated, as described in 9.3.2.3.

#### 7.4.2.10 Simplified Superframe Specification IE

The Simplified Superframe Specification IE Content field shall be formatted as illustrated in Figure 7-37.

| Octets: 2 | 2                        | 2                 |
|-----------|--------------------------|-------------------|
| Timestamp | Superframe Specification | CFP Specification |

**Figure 7-37—Simplified Superframe Specification IE Content field format**

The Timestamp field shall be updated between transmissions; the initial value, resolution (LSB value), and accuracy are implementation dependent.

The Superframe Specification field is as defined in 7.3.1.3.

The CFP Specification field shall be encoded as illustrated in Figure 7-38.

| Bits: 0–3      | 4–7                          | 8–11                        | 12         | 13–15    |
|----------------|------------------------------|-----------------------------|------------|----------|
| Number of GTSS | First CFP Slot in Superframe | Last CFP Slot in Superframe | GTS Permit | Reserved |

**Figure 7-38—CFP Specification field format**

The Number of GTSS field shall be set to the number of GTSS allocated by the coordinator.

The First CFP Slot in Superframe field shall be set to the slot number in which the CFP begins

The Last CFP Slot in Superframe field shall be set to the slot number in which the CFP ends.

The GTS Permit field shall be set to one if *macGtsPermit* is equal to TRUE, indicating that the coordinator is accepting GTS requests. Otherwise, the field shall be set to zero.

#### 7.4.2.11 Simplified GTS Specification IE

The Simplified GTS Specification IE Content field shall be formatted as illustrated in Figure 7-39.

| Octets: 1      | Variable                |
|----------------|-------------------------|
| GTS Directions | GTS Device Address List |

**Figure 7-39—Simplified GTS Specification IE Content field format**

The GTS Directions field is defined in 7.3.1.4.

The GTS Device Address List field is defined in 7.3.1.4.

#### 7.4.2.12 LECIM Capabilities IE

The LECIM Capabilities IE Content field shall be formatted as illustrated in Figure 7-40.

| Octets: 2                    | 2                            | 0/2                    | 0/2                     |
|------------------------------|------------------------------|------------------------|-------------------------|
| PHY Type and Bands Supported | LECIM PHY Features Supported | Lowest 2.4 GHz Channel | Highest 2.4 GHz Channel |

**Figure 7-40—Format of the LECIM Capabilities IE**

In the PHY Type and Bands Supported field, bit 0 indicates the PHY type supported, which is the PHY type being described by the IE. A value of one indicates that LECIM FSK is described; a value of zero indicates that LECIM DSSS is described. Bits 1–11 indicate support for different bands. A value of one indicates that a band is supported; a value of zero indicates that a band is not supported. The device shall indicate as supported only those bands that are implemented and defined for the indicated PHY type. The encoding for the field is shown in Table 7-9.

**Table 7-9—LECIM PHY Type and Bands Supported field encoding**

| Bit number | Description                                            |
|------------|--------------------------------------------------------|
| 0          | PHY type described:<br>0 = LECIM DSSS<br>1 = LECIM FSK |
| 1          | Band 169 supported                                     |
| 2          | Band 433 supported                                     |
| 3          | Band 470 supported                                     |
| 4          | Band 780 supported                                     |
| 5          | Band 863 supported                                     |
| 6          | Band 915 supported                                     |
| 7          | Band 917 supported                                     |

**Table 7-9—LECIM PHY Type and Bands Supported field encoding (continued)**

| Bit number | Description         |
|------------|---------------------|
| 8          | Band 920 supported  |
| 9          | Band 921 supported  |
| 10         | Band 922 supported  |
| 11         | Band 2450 supported |
| 12–15      | Reserved            |

When the PHY Type and Bands Supported field indicates a LECIM DSSS PHY, the LECIM PHY Features Supported field shall be encoded as shown in Table 7-10. A value of one indicates that the feature is supported; a value of zero indicates that the feature is not supported.

**Table 7-10—LECIM PHY Features Supported field encoding for DSSS**

| Bit number | Description                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------|
| 0          | BPSK modulation supported                                                                                         |
| 1          | O-QPSK modulation supported                                                                                       |
| 2–5        | Maximum spreading factor supported; see Table 11-2 for the definition of <i>phyLecimDsssPdsuSpreadingFactor</i> . |
| 6          | 16 octet PSDU supported                                                                                           |
| 7          | 24 octet PSDU supported                                                                                           |
| 8          | 32 octet PSDU supported                                                                                           |
| 9          | OVSF supported                                                                                                    |
| 10–15      | Reserved                                                                                                          |

When the PHY Type and Bands Supported field indicates a LECIM FSK PHY, the LECIM PHY Features Supported field shall be encoded as shown in Table 7-11. A value of one indicates that the feature is supported; a value of zero indicates that the feature is not supported.

**Table 7-11—LECIM PHY Features Supported field encoding for FSK**

| Bit number | Description                                                                |
|------------|----------------------------------------------------------------------------|
| 0          | Positional modulation supported                                            |
| 1          | Symbol rate 37.5 ksymbol/s supported, 200 kHz channel spacing <sup>a</sup> |
| 2          | Symbol rate 25 ksymbol/s supported, 200 kHz channel spacing <sup>a</sup>   |
| 3          | Symbol rate 12.5 ksymbol/s supported, 200 kHz channel spacing <sup>a</sup> |
| 4          | Symbol rate 37.5 ksymbol/s supported, 100 kHz channel spacing <sup>a</sup> |

**Table 7-11—LECIM PHY Features Supported field encoding for FSK (continued)**

| Bit number | Description                                                                |
|------------|----------------------------------------------------------------------------|
| 5          | Symbol rate 25 ksymbol/s supported, 100 kHz channel spacing <sup>a</sup>   |
| 6          | Symbol rate 12.5 ksymbol/s supported, 100 kHz channel spacing <sup>a</sup> |
| 7          | Forward error correction (FEC) supported                                   |
| 8          | Interleaving supported                                                     |
| 9          | Scrambling supported                                                       |
| 10         | SF 2 supported                                                             |
| 11         | SF 4 supported                                                             |
| 12         | SF 8 supported                                                             |
| 13         | SF 16 supported                                                            |
| 14         | Alternating SF pattern supported                                           |
| 15         | Non-alternating SF pattern supported                                       |

<sup>a</sup>Defined in 24.3.

The Lowest 2.4 GHz Channel and the Highest 2.4 GHz Channel fields specify the range of channels within the 2450 MHz band that are supported. These fields shall only be present when the PHY Type and Bands Supported field indicates support for the 2450 MHz band. The range of channels supported includes all of the channels starting from the channel number specified in the Lowest 2.4 GHz Channel field to the channel number specified in the Highest 2.4 GHz Channel field, inclusive.

#### 7.4.2.13 RCC Capabilities IE

The presence of this IE in a transmitted frame indicates that the coordinator supports an RCC PHY. The RCC Capabilities IE Content field shall be formatted as shown in Figure 7-41.

|                               |                                  |                                    |
|-------------------------------|----------------------------------|------------------------------------|
| Octets: 2                     | 2                                | 2                                  |
| RCC Frequency Bands Supported | RCC PHY and Modulation Supported | RCC DSSS DPSK Modulation Supported |

**Figure 7-41—RCC Capabilities IE Content field format**

The RCC Frequency Bands Supported field shall be formatted as given in Table 7-12.

**Table 7-12—RCC Frequency Bands Supported field encoding**

| Bit number | Description            |
|------------|------------------------|
| 0          | 161 MHz band supported |
| 1          | 216 MHz band supported |
| 2          | 217 MHz band supported |
| 3          | 220 MHz band supported |

**Table 7-12—RCC Frequency Bands Supported field encoding (continued)**

| Bit number | Description             |
|------------|-------------------------|
| 4          | 450 MHz band supported  |
| 5          | 770 MHz band supported  |
| 6          | 800 MHz band supported  |
| 7          | 806 MHz band supported  |
| 8          | 896 MHz band supported  |
| 9          | 915 MHz band supported  |
| 10         | 928 MHz band supported  |
| 11         | 2450 MHz band supported |
| 12         | 4965 MHz band supported |
| 13         | 5800 MHz band supported |
| 14–15      | Reserved                |

The RCC PHY and Modulation Supported field shall be formatted as given in Table 7-13

**Table 7-13—RCC PHY and Modulation Supported field encoding**

| Bit number | Description                         |
|------------|-------------------------------------|
| 0          | LMR GMSK 9.6 kb/s supported         |
| 1          | LMR GMSK 19.2 kb/s supported        |
| 2          | LMR 4-FSK 9.6 kb/s supported        |
| 3          | LMR 4-FSK 19.2 kb/s supported       |
| 4          | LMR 4-FSK 38.4 kb/s supported       |
| 5          | LMR QPSK 16 kb/s supported          |
| 6          | LMR QPSK 32 kb/s supported          |
| 7          | LMR $\pi/4$ DQPSK 16 kb/s supported |
| 8          | LMR $\pi/4$ DQPSK 32 kb/s supported |
| 9          | LMR $\pi/4$ DQPSK 36 kb/s supported |
| 10         | LMR DSSS DPSK supported             |
| 11         | LMR DSSS BPSK supported             |
| 12–15      | Reserved                            |



The RCC DSSS DPSK Modulation Supported field shall be formatted as given in Table 7-14.

**Table 7-14—RCC DSSS DPSK Modulation Supported field encoding**

| Bit number | Description                          |
|------------|--------------------------------------|
| 0          | 300 kcps chip rate supported         |
| 1          | 600 kcps chip rate supported         |
| 2          | 800 kcps chip rate supported         |
| 3          | 1 Mcps chip rate supported           |
| 4          | 1.6 Mcps chip rate supported         |
| 5          | 2 Mcps chip rate supported           |
| 6          | 3 Mcps chip rate supported           |
| 7          | 4 Mcps chip rate supported           |
| 8          | 11-chip spreading sequence supported |
| 9          | 15-chip spreading sequence supported |
| 10         | 20-chip spreading sequence supported |
| 11         | 40-chip spreading sequence supported |
| 12         | DSSS DBPSK supported                 |
| 13         | DSSS DQPSK supported                 |
| 14–15      | Reserved                             |

#### 7.4.2.14 RCCN Descriptor IE

The RCCN Descriptor IE Content field shall be formatted as shown in Figure 7-42.

| Octets: 1 | 2                    | 4          | 1                 | 1                 | variable   |
|-----------|----------------------|------------|-------------------|-------------------|------------|
| Version   | Slot Size Multiplier | RCCN Slots | Inactive Duration | Network ID Length | Network ID |

**Figure 7-42—RCCN Descriptor IE Content field format**

The Version field indicates the version of the RCCN descriptor IE and shall be set to zero for this version of the standard.

The Slot Size Multiplier field is used to calculate the superframe slot duration, as described in 6.2.9.

The RCCN Slots field shall be formatted as shown in Figure 7-43.

| Octets: 1                 | 1                       | 1         | 1         |
|---------------------------|-------------------------|-----------|-----------|
| Management Downlink Slots | Management Uplink Slots | CAP Slots | CFP Slots |

**Figure 7-43—RCCN Slots field format**

The Management Downlink Slots field indicates the number of superframe slots allocated to the PAN coordinator for the purpose of sending frames to RCCN endpoints.

The Management Uplink Slots field indicates the number of superframe slots allocated to RCCN endpoints for the purpose of sending frames to the RCCN PAN coordinator.

The CAP Slots field indicates the number of superframe slots allocated to the CAP.

The CFP Slots field indicates the number of superframe slots allocated to the CFP.

The Inactive Duration field specifies the length, in units of superframe slot duration, of the inactive portion of the superframe structure.

The Network ID Length field indicates the length of the Network ID field.

The Network ID field contains a set of octets which contains a network-specific identification. The value of the field shall be set to *macRccnNetId*.

#### 7.4.2.15 Global Time IE

The presence of this IE indicates that the transmitting device has a reference to global time, e.g., GPS reference time. The Global time IE Content field shall be formatted as shown in Figure 7-44.

|             |
|-------------|
| Octets: 4   |
| Global Time |

**Figure 7-44—Global Time IE Content field format**

The Global Time field is an unsigned integer that contains the number of seconds elapsed since 00:00:00 UTC, January 1, 1970.

#### 7.4.2.16 DA IE

The DA IE Content field shall be formatted as illustrated in Figure 7-45.

| Bits: 0      | 1                 | 2–5      | 6–15                | 16–20           | 21–23       | variable     |
|--------------|-------------------|----------|---------------------|-----------------|-------------|--------------|
| Address Mode | Addresses Pending | Reserved | Number of Addresses | Sequence Number | Page Number | Address List |

**Figure 7-45—DA IE Content field format**

When Address Mode field is set to zero, each address included in Address List is a short address. When Address Mode field is set to one, each address included in Address List is an extended address.

The Number of Addresses field is the number of neighbor addresses included in the Address List field of this DA IE.

The Address Pending field shall be set to one when the set of neighbor addresses is to be announced in multiple Beacon frames. An Addresses Pending field set to one indicates that this IE contains a subset of the set of neighbor addresses known by the device and more neighbor addresses are to be sent in following Beacon frames with a DA IE.

The Sequence Number field shall be set to a value identifying the set of addresses to be announced. An increment in the Sequence Number indicates that a new set of neighbor addresses is being announced. The Sequence Number shall be incremented when any address in the set of neighbor addresses has been changed.

The Page Number field shall be set to one for the first subset of address and incremented by one for each subsequent subset of addresses.

The Address Pending field shall be set to zero when the set of addresses to be announced is contained in a single DA IE. When Addresses Pending field is set to zero, the Sequence Number and Page Number fields shall be set to zero.

The Address List field contains the addresses of the device's neighbors.

#### 7.4.2.17 Header Termination 1 IE

The Header Termination 1 IE shall have a zero-length Content field. The use of the Header Termination 1 IE is described in 7.4.1.

#### 7.4.2.18 Header Termination 2 IE

The Header Termination 2 IE shall have a zero-length Content field. The use of the Header Termination 2 IE is described in 7.4.1.

### 7.4.3 Payload IEs

The general format of the Payload IE is shown in Figure 7-46.

| Bits: 0–10 | 11–14    | 15       | Octets: 0–2047 |
|------------|----------|----------|----------------|
| Length     | Group ID | Type = 1 | Content        |

**Figure 7-46—Format of Payload IEs**

The Length field specifies the number of octets in the Content field.

The Group ID values are defined in Table 7-15 for each of the Payload IEs. In the table, an X in the columns for frame indicates that the IE may be used in that frame. If there is not an X in the column, then the IE shall not be used in that frame. In the table, UL refers to the layers above the MAC.

#### 7.4.3.1 Encapsulated Service Data Unit (ESDU) IE

An ESDU IE encapsulates a higher layer payload that is carried in the Content field. Content is sent as received, i.e., no octet ordering changes shall be made.

**Table 7-15—Payload IE Group ID**

| Group ID value | Name                                     | Enhanced Beacon | Enhanced ACK | Data | Multipurpose | MAC command | Format subclause | Use description | Used by | Created by |
|----------------|------------------------------------------|-----------------|--------------|------|--------------|-------------|------------------|-----------------|---------|------------|
| 0x0            | Encapsulated Service Data Unit (ESDU) IE | X               |              | X    | X            | X           | 7.4.3.1          | 7.4.3.1         | UL      | UL         |
| 0x1            | MLME IE                                  | X               | X            | X    | X            | X           | 7.4.3.2          | 7.4.3.2         | MAC     | MAC        |
| 0x2            | Vendor Specific Nested IE                | X               | X            | X    | X            | X           | 7.4.4.30         | —               | UL      | UL         |
| 0x3–0xe        | Reserved                                 |                 |              |      |              |             |                  |                 |         |            |
| 0xf            | Payload Termination IE                   | X               | X            | X    | X            | X           | 7.4.3.3          | 7.4.1           | MAC     | MAC        |

**7.4.3.2 MLME IE**

The Content field of the MLME IE shall be formatted as illustrated in Figure 7-47.

| Octets: variable | variable  | ... | variable  |
|------------------|-----------|-----|-----------|
| Nested IE        | Nested IE | ... | Nested IE |

**Figure 7-47—MLME IE Content field format**

The Nested IEs formats are defined in 7.4.4.1.

**7.4.3.3 Payload Termination IE**

The Payload Termination IE shall have a zero-length Content field.

**7.4.4 Nested IE****7.4.4.1 Format of Nested IE**

The Nested IE of type short shall be formatted as illustrated in Figure 7-48.

| Bits: 0–7 | 8–14   | 15       | Octets: 0–255 |
|-----------|--------|----------|---------------|
| Length    | Sub-ID | Type = 0 | Content       |

**Figure 7-48—Nested IE of type short format**

The Nested IE of type long shall be formatted as illustrated in Figure 7-49.

| Bits: 0–10 | 11–14  | 15       | Octets: 0–2047 |
|------------|--------|----------|----------------|
| Length     | Sub-ID | Type = 1 | Content        |

**Figure 7-49—Nested IE of type long format**

The Length field specifies the number of octets in the Content field.

The Sub-ID field values for Nested IEs of type short are defined in Table 7-16. In the table, an X in the columns for frame indicates that the IE may be used in that frame. If there is not an X in the column, then the IE shall not be used in that frame. In the table, UL refers to the layers above the MAC.

**Table 7-16—Sub-ID allocation for short format**

| Sub-ID value | Name                          | Enhanced Beacon | Enhanced ACK | Data | Multipurpose | MAC command | Format subclause | Use description                        | Used by | Created by |
|--------------|-------------------------------|-----------------|--------------|------|--------------|-------------|------------------|----------------------------------------|---------|------------|
| 0x00–0x0f    | Reserved for type long format |                 |              |      |              |             |                  |                                        |         |            |
| 0x10–0x19    | Reserved                      |                 |              |      |              |             |                  |                                        |         |            |
| 0x1a         | TSCH Synchronization IE       | X               |              |      |              |             | 7.4.4.2          | 6.3.6                                  | MAC     | MAC        |
| 0x1b         | TSCH Slotframe and Link IE    | X               |              |      |              |             | 7.4.4.3          | 6.3.6                                  | UL      | UL         |
| 0x1c         | TSCH Timeslot IE              | X               |              |      |              |             | 7.4.4.4          | 6.3.6, 6.5.4                           | UL      | MAC        |
| 0x1d         | Hopping timing IE             | X               |              |      |              |             | 7.4.4.5          | 6.2.10                                 | MAC     | MAC        |
| 0x1e         | Enhanced Beacon Filter IE     |                 |              |      |              | X           | 7.4.4.6          | 6.3.1.2, 6.3.4                         | MAC     | UL         |
| 0x1f         | MAC Metrics IE                | X               |              | X    | X            |             | 7.4.4.7          | 8.4.2.6                                | UL      | MAC        |
| 0x20         | All MAC Metrics IE            | X               |              | X    | X            |             | 7.4.4.8          | 8.4.2.6                                | UL      | MAC        |
| 0x21         | Coexistence Specification IE  | X               |              |      |              |             | 7.4.4.9          | 6.2.3, 6.3.3.1, 6.14                   | UL      | MAC        |
| 0x22         | SUN Device Capabilities IE    |                 |              | X    | X            |             | 7.4.4.10         | 10.1.2.8, 20.2.1.2, 20.3.4, 20.5, [B3] | UL, MAC | MAC        |
| 0x23         | SUN FSK Generic PHY IE        | X               |              | X    | X            | X           | 7.4.4.11         | 10.1.2.8, 20.3                         | UL, MAC | MAC        |
| 0x24         | Mode Switch Parameter IE      | X               |              | X    | X            | X           | 7.4.4.12         | 20.2.3, 20.5                           | MAC     | UL         |
| 0x25         | PHY Parameter Change IE       | X               |              |      | X            |             | 7.4.4.13         | 6.10                                   | MAC     | UL         |

**Table 7-16—Sub-ID allocation for short format (continued)**

| Sub-ID value | Name                                   | Enhanced Beacon | Enhanced ACK | Data | Multipurpose | MAC command | Format subclause | Use description | Used by | Created by |
|--------------|----------------------------------------|-----------------|--------------|------|--------------|-------------|------------------|-----------------|---------|------------|
| 0x26         | O-QPSK PHY Mode IE                     |                 |              | X    | X            |             | 7.4.4.14         | 6.10            | MAC     | UL         |
| 0x27         | PCA Allocation IE                      | X               |              |      |              |             | 7.4.4.15         | 6.2.5.4         | MAC     | UL         |
| 0x28         | LECIM DSSS Operating Mode IE           |                 |              | X    | X            |             | 7.4.4.16         | 6.10            | MAC     | UL         |
| 0x29         | LECIM FSK Operating Mode IE            |                 |              | X    | X            |             | 7.4.4.17         | 6.10            | MAC     | UL         |
| 0x2a         | Reserved                               |                 |              |      |              |             |                  |                 |         |            |
| 0x2b         | TVWS PHY Operating Mode Description IE |                 |              |      | X            |             | 7.4.4.18         | 6.15            | MAC     | UL         |
| 0x2c         | TVWS Device Capabilities IE            | X               |              | X    | X            |             | 7.4.4.19         | 6.15            | UL, MAC | UL         |
| 0x2d         | TVWS Device Category IE                | X               |              |      |              |             | 7.4.4.20         | 6.15            | UL      | UL         |
| 0x2e         | TVWS Device Identification IE          | X               |              |      |              |             | 7.4.4.21         | 6.15            | UL      | UL         |
| 0x2f         | TVWS Device Location IE                | X               |              |      |              |             | 7.4.4.22         | 6.15            | UL      | UL         |
| 0x30         | TVWS Channel Information Query IE      | X               |              |      |              |             | 7.4.4.23         | 6.15            | UL      | UL         |
| 0x31         | TVWS Channel Information Source IE     | X               |              |      |              |             | 7.4.4.24         | 6.15            | UL      | UL         |
| 0x32         | CTM IE                                 | X               |              |      |              |             | 7.4.4.25         | 6.16            | UL      | UL         |
| 0x33         | Timestamp IE                           | X               |              |      |              |             | 7.4.4.26         | 6.9.5           | MAC     | MAC        |
| 0x34         | Timestamp Difference IE                | X               |              |      |              |             | 7.4.4.27         | 6.9.5, 6.7.4.2  | MAC     | MAC        |
| 0x35         | TMCIP Specification IE                 | X               |              |      |              |             | 7.4.4.28         | 6.13            | UL      | UL         |
| 0x36         | RCC PHY Operating Mode IE              |                 |              | X    | X            |             | 7.4.4.29         | 6.10            | MAC     | UL         |
| 0x37–0x7f    | Reserved                               |                 |              |      |              |             |                  |                 |         |            |

The Sub-ID field values for Nested IEs of type long are given in Table 7-17. In the table, an X in the columns for frame indicates that the IE may be used in that frame. If there is not an X in the column, then the IE shall not be used in that frame. In the table, UL refers to the layers above the MAC

**Table 7-17—Sub-ID allocation for long format**

| Sub-ID value | Name                      | Enhanced Beacon | Enhanced ACK | Data | Multipurpose | MAC command | Format subclause | Use description | Used by | Created by |
|--------------|---------------------------|-----------------|--------------|------|--------------|-------------|------------------|-----------------|---------|------------|
| 0x0–0x7      | Reserved                  |                 |              |      |              |             |                  |                 |         |            |
| 0x8          | Vendor Specific Nested IE | X               | X            | X    | X            | X           | 7.4.4.30         | —               | UL      | UL         |
| 0x9          | Channel hopping IE        | X               |              |      |              |             | 7.4.4.31         | 6.2.10, 6.3.6   | MAC     | MAC        |
| 0xa–0xf      | Reserved                  |                 |              |      |              |             |                  |                 |         |            |

**7.4.4.2 TSCH Synchronization IE**

The TSCH Synchronization IE Content field shall be formatted as illustrated in Figure 7-50.

|                  |             |
|------------------|-------------|
| <b>Octets: 5</b> | <b>1</b>    |
| ASN              | Join Metric |

**Figure 7-50—TSCH Synchronization IE Content field format**

The ASN field contains the ASN corresponding to the timeslot in which the enhanced beacon is sent. The ASN is used as the Frame Counter for security operations if enabled.

The Join Metric field is an unsigned integer and shall be set to *macJoinMetric*.

**7.4.4.3 TSCH Slotframe and Link IE**

The TSCH Slotframe and Link IE is used in Enhanced Beacon frames to allow new devices to synchronize to a TSCH PAN. The TSCH Slotframe and Link IE Content field shall be formatted as illustrated in Figure 7-51.

|                      |                        |            |                               |
|----------------------|------------------------|------------|-------------------------------|
| <b>Octets: 1</b>     | <b>variable</b>        | <b>...</b> | <b>variable</b>               |
| Number of Slotframes | Slotframe Descriptor 1 | ...        | Slotframe Descriptor <i>n</i> |

**Figure 7-51—TSCH Slotframe and Link IE Content field format**

The Number of Slotframes field is set to the total number of Slotframe Descriptor fields IE.

The Slotframe Descriptor shall be formatted as illustrated in Figure 7-52.

|                  |                |                 |                    |            |                           |
|------------------|----------------|-----------------|--------------------|------------|---------------------------|
| <b>Octets: 1</b> | <b>2</b>       | <b>1</b>        | <b>5</b>           | <b>...</b> | <b>5</b>                  |
| Slotframe handle | Slotframe size | Number of Links | Link Information 1 | ...        | Link Information <i>m</i> |

**Figure 7-52—Slotframe Descriptor field format**

The Slotframe Handle field shall be set to the *macSlotframeHandle* from *macSlotframeTable* for this slotframe.

The Slotframe Size field is the size of the slotframe in number of timeslots and shall be set to the corresponding *macSlotframeSize* from *macSlotframeTable*.

The Number of Links field shall be set to the number of links that belong to the slotframe identified by the Slotframe Handle field.

The Link Information field shall be formatted as illustrated in Figure 7-53.

|                  |                |              |
|------------------|----------------|--------------|
| <b>Octets: 2</b> | <b>2</b>       | <b>1</b>     |
| Timeslot         | Channel Offset | Link Options |

**Figure 7-53—Link Information field format**

The Timeslot field shall be set to *macTimeslot* of the *macLinkTable*.

The Channel Offset field shall be set to *macChannelOffset* of the *macLinkTable*.

The Link Options field shall be formatted as illustrated in Figure 7-54.

|                |          |             |             |          |            |
|----------------|----------|-------------|-------------|----------|------------|
| <b>Bits: 0</b> | <b>1</b> | <b>2</b>    | <b>3</b>    | <b>4</b> | <b>5–7</b> |
| TX Link        | RX Link  | Shared Link | Timekeeping | Priority | Reserved   |

**Figure 7-54—Link Options field format**

The TX Link field shall be set to one if it is a TX link and shall be set to zero otherwise.

TX Shared links, indicated by the TX link field and Shared Link field both set to one, may be used by a joining device to send an Association Request command or higher layer message to the advertising device.

The RX Link field shall be set to one if the link is an RX link and shall be set to zero otherwise. RX links are used by a joining device to receive an Association Response command or higher layer message from an advertising device.

The Shared Link field shall be set to one if the link is a shared link and shall be set to zero otherwise. A shared link is one that uses contention to access the medium.

A link may be used as both a TX shared link and RX link.



The Timekeeping field shall be set to one if the link is to be used for clock synchronization and shall be set to zero otherwise. RX links shall have the Timekeeping field set to one.

The Priority field shall be set to one if the link is a priority channel access, as defined in 6.2.5.2, and shall be set to zero otherwise.

#### 7.4.4.4 TSCH Timeslot IE

The TSCH Timeslot IE Content field shall be formatted as illustrated in Figure 7-55. The TSCH Timeslot IE may be sent with only the Timeslot ID field to reduce the size of the beacon. Otherwise, all the fields are included. All slotframes are referred to by an ID, *macTimeslotTemplateId*, with ID = 0x00 denoting the default values for the band in use, with values given in Table 8-86.

|             |            |     |           |           |              |              |         |          |       |         |        |                  |
|-------------|------------|-----|-----------|-----------|--------------|--------------|---------|----------|-------|---------|--------|------------------|
| Octets: 1   | 0/2        | 0/2 | 0/2       | 0/2       | 0/2          | 02           | 0/2     | 0/2      | 0/2   | 0/2     | 0/2/3  | 0/2/3            |
| Timeslot ID | CCA Offset | CCA | TX Offset | RX Offset | RX Ack Delay | TX Ack Delay | RX Wait | Ack Wait | RX TX | Max Ack | Max TX | Time-slot Length |

Figure 7-55—TSCH Timeslot IE Content field format

The Timeslot ID field is set to the value of *macTimeslotTemplateId* encoded as an unsigned integer.

The CCA Offset field shall be set to the value of *macTsCcaOffset* encoded as an unsigned integer.

The CCA field shall be set to *macTsCca* encoded as an unsigned integer.

The TX Offset field shall be set to *macTsTxOffset* encoded as an unsigned integer.

The RX Offset field shall be set to *macTsRxOffset* encoded as an unsigned integer.

The RX Ack Delay field shall be set to *macTsRxAckDelay* encoded as an unsigned integer.

The TX Ack Delay field shall be set to *macTsTxAckDelay* encoded as an unsigned integer.

The RX Wait field shall be set to *macTsRxWait* encoded as an unsigned integer.

The Ack Wait field shall be set to *macTsAckWait* encoded as an unsigned integer.

The RX TX field shall be set to *macTsRxTx* encoded as an unsigned integer.

The Max Ack field shall be set to *macTsMaxAck* encoded as an unsigned integer.

The length of the Max TX field and Timeslot Length field shall be the same and can be determined from the length of the IE.

The Max TX field shall be set to *macTsMaxTx* encoded as an unsigned integer.

The Timeslot Length field shall be set to *macTsTimeslotLength* encoded as an unsigned integer.

The PIB values used for this IE are defined in Table 8-85.

#### 7.4.4.5 Hopping timing IE

The Hopping Timing IE Content field shall be formatted as illustrated in Figure 7-56.

| Octets: 3               | 2              |
|-------------------------|----------------|
| Present Hop Time Offset | Hop Dwell Time |

Figure 7-56—Hopping Timing IE Content field format

The Present Hop Time Offset field contains the amount of time in microseconds that has passed at the time of frame transmission since the transmitting device hopped to the present channel.

The Hop Dwell Time field shall be set to *macHopDwellTime* encoded as an unsigned integer.

#### 7.4.4.6 Enhanced Beacon Filter IE

The Enhanced Beacon Filter IE Content field shall be formatted as illustrated in Figure 7-57. No more than one Enhanced Beacon Filter IE shall be conveyed per Enhanced Beacon Request frame.

| Bits: 0           | 1                           | 2                      | 3–4                  | 5–7      | Octets: 0/1  | 0/1            | 0/1/2/3       |
|-------------------|-----------------------------|------------------------|----------------------|----------|--------------|----------------|---------------|
| Permit Joining On | Include Link Quality Filter | Include Percent Filter | Attribute IDs Length | Reserved | Link Quality | Percent Filter | Attribute IDs |

Figure 7-57—Enhanced Beacon Filter IE Content field format

If the Permit Joining On field is set, only devices supporting Enhanced Beacon Request command/Enhanced Beacon frame with permit joining on shall respond to the Enhanced Beacon Request command.

The Include Link Quality Filter field shall be set to one if a Link Quality field is included and shall be set to zero otherwise.

The Link Quality filter is an unsigned integer that indicates the minimum link quality that should be met in order for a device to respond to the request.

The Include Percent filter bit shall be set to one if a Percent Filter field is included and shall be set to zero otherwise.

The Percent Filter field contains a scaled value from 0x00 to 0x64 representing zero to 100% probability for a given device to respond to the enhanced beacon request. The receiving device shall then randomly determine if it is to respond to the Enhanced Beacon Request command (if supporting Enhanced Beacon Request command/Enhanced Beacon frame) based on meeting this probability. For example, if the probability is set to 10%, then a device would respond, on average, to one out of every ten requests.

The Attribute IDs Length field is an unsigned integer that shall be set to the length in octets of the Attribute IDs field.

The Attribute IDs field is a bitmap identifying which of the Attribute Request IDs, defined in Table 6-3, are being requested. If attribute ID *n* is being requested, then bit *n* of the Attribute IDs field would be set to one. All other bits shall be set to zero.

**7.4.4.7 MAC Metrics IE**

The MAC Metrics IE Content field shall be formatted as illustrated in Figure 7-58.

|           |       |
|-----------|-------|
| Octets: 1 | 4     |
| Metric ID | Count |

**Figure 7-58—MAC Metrics IE Content field format**

The Metric ID field shall be set to one of the values in Table 7-18.

**Table 7-18—Metric Count IDs**

| Attribute name                | Metric ID |
|-------------------------------|-----------|
| <i>macCounterOctets</i>       | 0x01      |
| <i>macRetryCount</i>          | 0x02      |
| <i>macMultipleRetryCount</i>  | 0x03      |
| <i>macTxFailCount</i>         | 0x04      |
| <i>macTxSuccessCount</i>      | 0x05      |
| <i>macFcsErrorCount</i>       | 0x06      |
| <i>macSecurityFailure</i>     | 0x07      |
| <i>macDuplicateFrameCount</i> | 0x08      |
| <i>macRxSuccessCount</i>      | 0x09      |
| <i>macNackCount</i>           | 0x0a      |

**7.4.4.8 All MAC Metrics IE**

The All MAC Metrics IE Content shall be formatted as illustrated in Figure 7-59.

|              |              |   |                     |
|--------------|--------------|---|---------------------|
| Octets: 4    | 4            | – | 4                   |
| MAC Metric 1 | MAC Metric 2 | – | MAC Metric <i>n</i> |

**Figure 7-59—All MAC metrics IE Content field format**

The MAC Metric field contains the PIB attribute value in the order given in Table 7-18.

**7.4.4.9 Coexistence Specification IE**

The Coexistence Specification IE Content field shall be formatted as illustrated in Figure 7-60.

| Bits: 0–3    | 4–7              | 8–11           | 12–15                 | 16–19            | 20–23              | 24–39                 |
|--------------|------------------|----------------|-----------------------|------------------|--------------------|-----------------------|
| Beacon Order | Superframe Order | Final CAP Slot | Enhanced Beacon Order | Offset Time Slot | CAP Backoff Offset | Enhanced Beacon Order |

**Figure 7-60—Coexistence Specification IE Content field format**

The Beacon Order field, Superframe Order field, and Final CAP Slot field are as specified in 7.3.1.3. If the Beacon Order field is set to 15, the Superframe Order, Final CAP Slot, and Offset Time Slot fields shall be set to zero upon transmission and ignored upon reception.

The Enhanced Beacon Order field specifies the transmission interval of the Enhanced Beacon frames, as specified in 6.2.3.

The Offset Time Slot field specifies the time offset between the periodic beacon and the following Enhanced Beacon frame, as described in 6.2.3.

The CAP Backoff Offset field specifies the actual slot position in which the Enhanced Beacon frame is transmitted due to the backoff procedure in the CAP.

The Enhanced Beacon Order field specifies the transmission interval between consecutive Enhanced Beacon frames in the nonbeacon-enabled mode. The valid range for this field shall be 0–16383.

When generated in response to an Enhanced Beacon Request command that contained a set of requested IEs, the content of the Enhanced Beacon frame shall include the IEs corresponding to the requested IE IDs, as shown in Table 6-3.

**7.4.4.10 SUN Device Capabilities IE**

The SUN Device Capabilities IE Content field shall be formatted as illustrated in Figure 7-61.

| Octets: 1    | 2                         | Variable               |
|--------------|---------------------------|------------------------|
| SUN Features | Frequency Bands Supported | PHY Type Descriptor(s) |

**Figure 7-61—SUN Device Capabilities IE Content field format**

The SUN Features field shall be formatted as illustrated in Figure 7-62.

| Bits: 0 | 1              | 2            | 3      | 4         | 5       | 6           | 7        |
|---------|----------------|--------------|--------|-----------|---------|-------------|----------|
| Enh-Ack | Data Whitening | Interleaving | SFD G1 | NRNSC FEC | RSC FEC | Mode Switch | Reserved |

**Figure 7-62—SUN Features field format**

The Enh-Ack field shall be set to one if the Enh-Ack frame, as described in 6.7.4.2, is supported and shall be set to zero otherwise.

The Data Whitening field shall be set to one if data whitening, as described in 20.4, is supported and shall be set to zero otherwise.

The Interleaving field shall be set to one if interleaving, as described in 20.3.5, is supported and shall be set to zero otherwise.

The SFD G1 field shall be set to one if the start-of-frame delimiter (SFD) group 1, as described in 20.2.1.2, is supported and shall be set to zero otherwise.

The NRNSC FEC field shall be set to one if the nonrecursive and nonsystematic code (NRNSC) FEC, as described in 20.3.4, is supported and shall be set to zero otherwise.

The RSC FEC field shall be set to one if the recursive and systematic code (RSC) FEC, as described in 20.3.4, is supported and shall be set to zero otherwise.

The Mode Switch field shall be set to one if the mode switch mechanism, as described in 20.5, is supported and shall be set to zero otherwise.

The Frequency Bands Supported field is a bitmap indexed by the frequency band identifier values given in Table 7-19. The least significant bit of the bitmap corresponds to frequency band identifier zero. A bit set to one indicates that the device supports operation in that frequency band; otherwise, it does not.

**Table 7-19—Frequency band identifier values**

| Frequency band identifier | Band Designation |
|---------------------------|------------------|
| 0                         | 169 MHz band     |
| 1                         | 450 MHz band     |
| 2                         | 470 MHz band     |
| 3                         | 780 MHz band     |
| 4                         | 863 MHz band     |
| 5                         | 896 MHz band     |
| 6                         | 901 MHz band     |
| 7                         | 915 MHz band     |
| 8                         | 917 MHz band     |
| 9                         | 920 MHz band     |
| 10                        | 928 MHz band     |
| 11                        | 920 MHz band     |
| 12                        | 1427 MHz band    |
| 13                        | 2450 MHz band    |
| 14–15                     | Reserved         |

The PHY Type Descriptor field shall be formatted as illustrated in Figure 7-63.

| Bits: 0–3 | 4                   | 5–15                                                     | 16–31                                                                 |
|-----------|---------------------|----------------------------------------------------------|-----------------------------------------------------------------------|
| PHY Type  | All Frequency Bands | PHY Modes Supported<br>(PHY Mode ID bitmap:<br>b0...b10) | Specific Frequency Bands<br>(only present if All Frequency Bands = 0) |

**Figure 7-63—PHY Type Descriptor field format**

The PHY Type field contains an unsigned integer whose value identifies a PHY Type defined in Table 7-20.

The All Frequency Bands field indicates whether the optional Specific Frequency Bands field is present. If the All Frequency Bands field is set to one, the optional Specific Frequency Bands field is absent and the PHY Type is supported in all frequency bands declared in the Frequency Bands Supported field of the SUN device capabilities IE. If the All Frequency Bands field is set to zero, the optional Specific Frequency Bands field is present and the PHY Type is only supported in the frequency bands declared in the Specific Frequency Bands field of this PHY Type Descriptor.

The PHY Modes Supported field is a bitmap indicating which PHY modes are supported for the PHY Type. The PHY modes for each possible PHY Type are defined in Table 7-21, Table 7-22, Table 7-23, Table 7-24, Table 7-25, and Table 7-26. A bit set to one in bit  $b_n$  of the PHY Mode ID bitmap indicates that the PHY Mode with ID  $n$  in the table of PHY Modes corresponding to the PHY Type is supported; otherwise, it is not supported.

The optional Specific Frequency Bands field is encoded in the same manner as the Frequency Bands Supported field of the SUN device capabilities IE.

**Table 7-20—Modulation scheme encoding**

| PHY type | Modulation scheme |
|----------|-------------------|
| 0        | FSK-A             |
| 1        | FSK-B             |
| 2        | O-QPSK-A          |
| 3        | O-QPSK-B          |
| 4        | O-QPSK-C          |
| 5        | OFDM Option 1     |
| 6        | OFDM Option 2     |
| 7        | OFDM Option 3     |
| 8        | OFDM Option 4     |
| 9–15     | Reserved          |

The FSK-A PHY mode encodings are defined in Table 7-21.

**Table 7-21—FSK-A PHY mode encoding**

| PHY Mode ID | Narrowband FSK PHY mode                                       |
|-------------|---------------------------------------------------------------|
| 0           | 4.8 kb/s; 2-FSK; mod index = 1.0; channel spacing = 12.5 kHz  |
| 1           | 9.6 kb/s; 4-FSK; mod index = 0.33; channel spacing = 12.5 kHz |
| 2           | 10 kb/s; 2-FSK; mod index = 0.5; channel spacing = 12.5 kHz   |
| 3           | 20 kb/s; 2-FSK; mod index = 0.5; channel spacing = 12.5 kHz   |
| 4           | 40 kb/s; 2-FSK; mod index = 0.5; channel spacing = 12.5 kHz   |
| 5           | 4.8 kb/s; 2-FSK; mod index = 0.5; channel spacing = 12.5 kHz  |
| 6           | 2.4 kb/s; 2-FSK; mod index = 2.0; channel spacing = 12.5 kHz  |
| 7           | 9.6 kb/s; 4-FSK; mod index = 0.33; channel spacing = 12.5 kHz |
| 8–10        | Reserved                                                      |

The FSK-B PHY mode encodings are defined in Table 7-22.

**Table 7-22—FSK-B PHY mode encoding**

| PHY Mode ID | FSK PHY mode                                                 |
|-------------|--------------------------------------------------------------|
| 0           | 50 kb/s; 2-FSK; mod index = 1.0; channel spacing = 200 kHz   |
| 1           | 100 kb/s; 2-FSK; mod index = 1.0; channel spacing = 400 kHz  |
| 2           | 150 kb/s; 2-FSK; mod index = 0.5; channel spacing = 400 kHz  |
| 3           | 200 kb/s; 2-FSK; mod index = 0.5; channel spacing = 400 kHz  |
| 4           | 200 kb/s; 4-FSK; mod index = 0.33; channel spacing = 400 kHz |
| 5           | 200 kb/s; 2-FSK; mod index = 1.0; channel spacing = 600 kHz  |
| 6           | 400 kb/s; 4-FSK; mod index = 0.33; channel spacing = 600 kHz |
| 7–10        | Reserved                                                     |

The O-QPSK-A PHY mode encodings are defined in Table 7-23.

**Table 7-23—O-QPSK-A PHY mode encoding**

| PHY Mode ID | Narrowband O-QPSK PHY mode                                                         |
|-------------|------------------------------------------------------------------------------------|
| 0           | chip rate = 100 kchip/s; SpreadingMode = DSSS; RateMode = 0; data rate = 6.25 kb/s |
| 1           | chip rate = 100 kchip/s; SpreadingMode = DSSS; RateMode = 1; data rate = 12.5 kb/s |
| 2           | chip rate = 100 kchip/s; SpreadingMode = DSSS; RateMode = 2; data rate = 25 kb/s   |
| 3           | chip rate = 100 kchip/s; SpreadingMode = DSSS; RateMode = 3; data rate = 50 kb/s   |
| 4–10        | Reserved                                                                           |

The O-QPSK-B PHY mode encodings are defined in Table 7-24.

**Table 7-24—O-QPSK-B PHY mode encoding**

| PHY Mode ID | O-QPSK PHY mode                                                                      |
|-------------|--------------------------------------------------------------------------------------|
| 0           | chip rate = 1000 kchip/s; SpreadingMode = DSSS; RateMode = 0; data rate = 31.25 kb/s |
| 1           | chip rate = 1000 kchip/s; SpreadingMode = DSSS; RateMode = 1; data rate = 125 kb/s   |
| 2           | chip rate = 1000 kchip/s; SpreadingMode = DSSS; RateMode = 2; data rate = 250 kb/s   |
| 3           | chip rate = 1000 kchip/s; SpreadingMode = DSSS; RateMode = 3; data rate = 500 kb/s   |
| 4           | chip rate = 1000 kchip/s; SpreadingMode = MDSSS; RateMode = 0; data rate = 62.5 kb/s |
| 5           | chip rate = 1000 kchip/s; SpreadingMode = MDSSS; RateMode = 1; data rate = 125 kb/s  |
| 6           | chip rate = 1000 kchip/s; SpreadingMode = MDSSS; RateMode = 2; data rate = 250 kb/s  |
| 7           | chip rate = 1000 kchip/s; SpreadingMode = MDSSS; RateMode = 3; data rate = 500 kb/s  |
| 8–10        | Reserved                                                                             |

The O-QPSK-C PHY mode encodings are defined in Table 7-25.

**Table 7-25—O-QPSK-C PHY mode encoding**

| PHY Mode ID | O-QPSK PHY mode                                                                      |
|-------------|--------------------------------------------------------------------------------------|
| 0           | chip rate = 2000 kchip/s; SpreadingMode = DSSS; RateMode = 0; data rate = 31.25 kb/s |
| 1           | chip rate = 2000 kchip/s; SpreadingMode = DSSS; RateMode = 1; data rate = 125 kb/s   |
| 2           | chip rate = 2000 kchip/s; SpreadingMode = DSSS; RateMode = 2; data rate = 250 kb/s   |
| 3           | chip rate = 2000 kchip/s; SpreadingMode = DSSS; RateMode = 3; data rate = 500 kb/s   |
| 4           | chip rate = 2000 kchip/s; SpreadingMode = MDSSS; RateMode = 0; data rate = 62.5 kb/s |



**Table 7-25—O-QPSK-C PHY mode encoding (continued)**

| PHY Mode ID | O-QPSK PHY mode                                                                     |
|-------------|-------------------------------------------------------------------------------------|
| 5           | chip rate = 2000 kchip/s; SpreadingMode = MDSSS; RateMode = 1; data rate = 125 kb/s |
| 6           | chip rate = 2000 kchip/s; SpreadingMode = MDSSS; RateMode = 2; data rate = 250 kb/s |
| 7           | chip rate = 2000 kchip/s; SpreadingMode = MDSSS; RateMode = 3; data rate = 500 kb/s |
| 8–10        | Reserved                                                                            |

For each OFDM option, the supported MCSs are defined in Table 7-26.

**Table 7-26—OFDM PHY mode encoding**

| PHY Mode ID | OFDM PHY modes (Option 1,2,3,4) |
|-------------|---------------------------------|
| 0           | MCS0 supported                  |
| 1           | MCS1 supported                  |
| 2           | MCS2 supported                  |
| 3           | MCS3 supported                  |
| 4           | MCS4 supported                  |
| 5           | MCS5 supported                  |
| 6           | MCS6 supported                  |
| 7–10        | Reserved                        |

In each encoding of PHY modes in the PHY Type Descriptor field, a bit set to one in a bit position corresponding to a given modulation scheme in Table 7-21, Table 7-22, Table 7-23, Table 7-24, Table 7-25, and Table 7-26 indicates the PHY mode for that modulation scheme is supported. A value of zero similarly indicates that the PHY mode is not supported.

The value of the SUN PHY Capabilities IE length field is computed as follows:

$$Length = 1 + 2 + 2 \times NumPHYTypeAllFreq1 + 4 \times NumPHYTypeAllFreq0$$

where

$NumPHYTypeAllFreq1$  is the number of PHY Type Descriptor fields that have the All Frequency Bands field set to one

$NumPHYTypeAllFreq0$  is the number of PHY Type Descriptor fields that have the All Frequency Bands field set to zero

**7.4.4.11 SUN FSK Generic PHY IE**

The SUN FSK Generic PHY IE Content field shall be formatted as illustrated in Figure 7-64.

| Octets: 1 |                      |                              | 1                   |     | 4                                    | 2                     | 4                  | 4              |
|-----------|----------------------|------------------------------|---------------------|-----|--------------------------------------|-----------------------|--------------------|----------------|
| Bits: 0–1 | 2–3                  | 4–7                          | Bit #: 0–5          | 6–7 | First Channel<br>Center<br>Frequency | Number of<br>Channels | Channel<br>Spacing | Symbol<br>Rate |
| Reserved  | Modulation<br>Scheme | SUN FSK<br>Generic PHY<br>ID | Modulation<br>Index | BT  |                                      |                       |                    |                |

**Figure 7-64—SUN FSK Generic PHY IE Content field format**

The Modulation Scheme is an unsigned integer that shall be zero to indicate 2-FSK and shall be set to one to indicate 4-FSK. All other values are reserved.

SUN FSK Generic PHY ID is unsigned integers that is used to identify the PHY configuration contained in this IE.

The Modulation Index field is encoded as an unsigned integer with permissible values in the range 0–45. Values in the range 46–63 are undefined. The value of the modulation index for 2-FSK is computed using the value of the Modulation Index field in the following way:

$$\text{modulation index} = 0.25 + (\text{Modulation Index field value}) \times 0.05$$

For example, a modulation index of 0.5 would be encoded as 0b00101.

The BT field is an unsigned integer. The BT field shall be set to zero if the BT is 0.5 and shall be set to one if the BT is 1.0. All other values are reserved.

The First Channel Center Frequency field contains an unsigned integer whose value is the center frequency of the first channel in hertz.

The Number of Channels field contains an unsigned integer whose value is the number of channels for the generic PHY.

The Channel Spacing field contains an unsigned integer whose value is the difference between adjacent channel center frequencies in hertz. For example, 300 kHz channel spacing would be encoded as 0x493E0.

The Symbol Rate field contains an unsigned integer whose value is the number of symbols transmitted per second. For example, 100 ksymbol/s would be encoded as 0x186A0.

**7.4.4.12 Mode Switch Parameter IE**

The Mode Switch Parameter IE Content field shall be formatted as illustrated in Figure 7-65.

| Octets: 1         |             |             |                             | 1              | 1                             |
|-------------------|-------------|-------------|-----------------------------|----------------|-------------------------------|
| Bits: 0           | 1–4         | 5           | 6–7                         | Settling Delay | Secondary FSK Preamble Length |
| Secondary FSK SFD | Target Mode | Source Mode | Mode Switch Parameter Index |                |                               |

**Figure 7-65—Mode Switch Parameter IE Content field format**

The Secondary FSK SFD field value is set to one if a secondary SFD is present; otherwise, it is set to zero, as described in Table 11-3.

The Target Mode field contains an unsigned integer indicating the PHY Mode of the new mode PPDU. The values are as defined in Table 7-20.

The Source Mode field contains the mode that is used to transmit the mode switch PPDU. It is set to zero for 2-FSK or to one for 4-FSK.

The Mode Switch Parameter Index field contains an unsigned integer whose value is the index into the *phyModeSwitchParameterEntries* PIB attribute array containing the mode switch parameter.

The Settling Delay field contains an unsigned integer whose value times two is the settling delay in microseconds (0–510  $\mu$ sec), as described in Table 11-3.

The Secondary FSK Preamble Length field contains an unsigned integer whose value is the number of preamble repetitions for the secondary preamble when the new mode is SUN FSK, as described in Table 11-3.

**7.4.4.13 PHY Parameter Change IE**

The PHY Parameter Change IE Content field shall be formatted as illustrated in Figure 7-66.

| Octets: 4                | 4                 |
|--------------------------|-------------------|
| Effective Time of Change | Notification Time |

**Figure 7-66—PHY Parameter Change IE**

The Effective Time of Change field shall contain a time in the future, in microseconds, when the change is scheduled to occur.

The Notification Time field shall contain the local time value in the generating device at the time the frame containing the PHY Parameter Change IE is generated.

The PHY Parameter Change IE shall always be followed in the frame by a valid Operating Mode Description IE.

**7.4.4.14 O-QPSK PHY Mode IE**

The O-QPSK PHY Mode IE Content field shall be formatted as illustrated in Figure 7-67.

|                  |               |
|------------------|---------------|
| <b>Octets: 1</b> | <b>1</b>      |
| Band Selector    | Channel Index |

**Figure 7-67—O-QPSK PHY Mode IE Content field format**

The Band Selector field identifies the band for the operating mode and shall take one of the values in Table 7-27.

**Table 7-27—Band Selector frequency bands**

| Band Selector value | Corresponding band |
|---------------------|--------------------|
| 0                   | Reserved           |
| 1                   | 780 MHz band       |
| 2                   | 868 MHz band       |
| 3                   | 915 MHz band       |
| 4                   | 2380 MHz band      |
| 5                   | 2450 MHz band      |
| 6–7                 | Reserved           |

The Channel Index field shall contain a valid channel number as defined in 10.1.2.1, 10.1.2.2, or 10.1.2.9 as appropriate to the band selector value.

**7.4.4.15 PCA Allocation IE**

The PCA Allocation IE Content field shall be formatted as illustrated in Figure 7-68.

|               |            |                 |                 |
|---------------|------------|-----------------|-----------------|
| <b>Bit: 0</b> | <b>1</b>   | <b>2–15</b>     | <b>16–23</b>    |
| PCA Used      | Super-rate | Delay Tolerance | Allocation Rate |

**Figure 7-68—PCA Allocation IE Content field format**

The PCA Used field is set according to the MAC PIB attribute *macPriorityChannelAccess*, with a value of zero indicating *macPriorityChannelAccess* is FALSE and a value of one indicating *macPriorityChannelAccess* is TRUE.

The Super-rate field is set according to the MAC PIB attribute *macPcaAllocationSuperRate* with a value of zero indicating *macPcaAllocationSuperRate* is FALSE and a value of one indicating *macPcaAllocationSuperRate* is TRUE.

The Delay Tolerance field describes the delay tolerance of a critical event message in units of 4 ms. The relation of *macCritMsgDelayTol* to the Delay Tolerance field value is as follows:

$$\text{macCritMsgDelayTol} = 4 \times \text{Delay Tolerance (ms)}$$

The Allocation Rate field is set according to the MAC PIB attribute *macPcaAllocationRate*, and in conjunction with *macPcaAllocationSuperRate*, it provides the rate at which PCA allocations are made, as described in 6.2.5.4.

#### 7.4.4.16 LECIM DSSS Operating Mode IE

The LECIM DSSS Operating Mode IE Content field shall be formatted as illustrated in Figure 7-69.

| Bits: 0–3      | 4–12           | 13                   | 14–16     | 17              | 18–21     | 20               | ... |
|----------------|----------------|----------------------|-----------|-----------------|-----------|------------------|-----|
| Operating Band | Channel Number | Modulation Selection | Chip Rate | Channel Spacing | PSDU Size | Preamble Present | ... |

| ... | 21          | 22–25            | 26–29                 | 30–53                    | 54–63    |
|-----|-------------|------------------|-----------------------|--------------------------|----------|
| ... | SFD Present | Spreading Factor | OVSF Spreading Factor | Gold Code Initialization | Reserved |

**Figure 7-69—LECIM DSSS Operating Mode IE Content field**

The Operating Band field contains an unsigned integer corresponding to a bit number given in Table 7-9. Values not corresponding to an assigned frequency band in Table 7-9 are reserved.

The Channel Number field contains the channel number, as defined in 10.1.2.

The Modulation Selection field shall be set to zero for BPSK and shall be set to one for O-QPSK.

The Chip Rate field is an unsigned integer that corresponds the chip rate, as defined in Table 7-28:

**Table 7-28—Chip Rate field valid values**

| Field value | Chip rate (kchips/s) |
|-------------|----------------------|
| 0           | Reserved             |
| 1           | 100                  |
| 2           | 200                  |
| 3           | 400                  |
| 4           | 600                  |
| 5           | 800                  |
| 6           | 1000                 |
| 7           | 2000                 |

The Channel Spacing shall be set to zero for 0.1 MHz channel spacing and shall be set to one for 0.2 MHz channel spacing.

The PSDU Size field shall be set to one of the values in Table 7-29.

**Table 7-29—PSDU Size field valid values**

| Field value<br>(b18 b19) | PSDU size |
|--------------------------|-----------|
| 00                       | invalid   |
| 01                       | 16 octets |
| 10                       | 24 octets |
| 11                       | 32 octets |

The Preamble Present shall be set to one if the preamble is present and shall be set to zero if the preamble is not present.

The SFD Present shall be set to one if the SFD is present and shall be set to zero if the SFD is not present.

The Spreading Factor field contains *phyLecimDsssPsdusSpreadingFactor*, as defined in Table 11-2.

The OVFSF Spreading Factor field shall be set to one of the valid value defined in Table 7-30. All other values are reserved.

**Table 7-30—OVFSF Spreading Factor field values**

| Field value<br>(b26 b27 b28 b29) | OVFSF spreading factor |
|----------------------------------|------------------------|
| 0000                             | OVFSF not applied      |
| 0001                             | SF of 2                |
| 0010                             | SF of 4                |
| 0011                             | SF of 8                |
| 0100                             | SF of 16               |
| 0101                             | SF of 32               |
| 0110                             | SF of 64               |
| 0111                             | SF of 128              |
| 1000                             | SF of 256              |

The Gold Code Initialization field shall be set to the Gold code LFSR2 initialization value, as specified in 23.2.6.1.

**7.4.4.17 LECIM FSK Operating Mode IE**

The LECIM FSK Operating Mode IE Content field shall be formatted as illustrated in Figure 7-70.

| Bits: 0–3      | 4–12           | 13                  | 14–15       | 16              | 17          | 18                   | ... |
|----------------|----------------|---------------------|-------------|-----------------|-------------|----------------------|-----|
| Operating Band | Channel Number | Position Modulation | Symbol Rate | Channel Spacing | FEC Enabled | Interleaving Enabled | ... |

| ... | 19                | 20                | 21–22            | 23                | 23–31    |
|-----|-------------------|-------------------|------------------|-------------------|----------|
| ... | Scrambler Enabled | Spreading Enabled | Spreading Factor | Spreading Pattern | Reserved |

**Figure 7-70—LECIM FSK Operating Mode IE Content field**

The Operating Band field shall be set to an unsigned integer that corresponds to a bit number given in Table 7-9. Values not corresponding to a valid frequency band in Table 7-9 are reserved.

The Channel Number field is defined in 10.1.2 and 10.2.

The Position Modulation shall be set to one if the position modulation is used and shall be set to zero otherwise.

The Symbol Rate field is an unsigned integer that corresponds to a symbol rate in Table 7-31.

**Table 7-31—Symbol Rate field valid values**

| Field value | Symbol rate    |
|-------------|----------------|
| 0           | 37.5 ksymbol/s |
| 1           | 25 ksymbol/s   |
| 2           | 12.5 ksymbol/s |
| 3           | Reserved       |

The Channel Spacing field indicates the channel spacing in the operating band specified in this IE. It shall be set to zero to indicate a 200 kHz channel spacing and shall be set to one to indicate a 100 kHz channel spacing.

The FEC Enabled field shall be set to one if FEC is enabled and shall be set to zero otherwise.

The Interleaving Enabled field shall be set to one if interleaving is enabled and shall be set to zero otherwise.

The Scrambler Enabled field shall be set to one if the scrambler is enabled and shall be set to zero otherwise.

The Spreading Enabled field shall be set to one if spreading is enabled and shall be set to zero otherwise.

The Spreading Factor field is an unsigned integer that corresponds to a spreading factor in Table 7-32.

**Table 7-32—Spreading Factor field valid values**

| Field value | Spreading factor |
|-------------|------------------|
| 0           | 2                |
| 1           | 4                |
| 2           | 8                |
| 3           | 16               |

The Spreading Pattern field shall be set to zero if a non-alternating spreading pattern is used and shall be set to one if an alternating spreading pattern is used.

#### 7.4.4.18 TVWS PHY Operating Mode Description IE

The TVWS PHY Operating Mode Description IE Content field shall be formatted as shown in Figure 7-71. The values in the TVWS PHY Operating Mode Description IE correspond to those values that will take effect after the PHY change operation, as described in 6.10.

| Bits: 0–7 | 8–15           | 16–17             | 18–31                | 32–71                     |
|-----------|----------------|-------------------|----------------------|---------------------------|
| Band ID   | PHY Channel ID | PHY Type Selector | Operating Parameters | Available Frequency Range |

**Figure 7-71—TVWS PHY Operating Mode Description IE Content field format**

The Band ID field contains the bit number of the matching Band ID in Table 7-39.

The PHY Channel ID field contains channel number, *NumChan*, for the TVWS PHY channel as defined in 10.1.2.8.

The PHY Type Selector field indicates which TVWS PHY is described in the Operating Parameters field. Valid values for the PHY Type Selector field are given in Table 7-33.

**Table 7-33—PHY Type Selector field values**

| Field value | PHY type         |
|-------------|------------------|
| 0           | TVWS FSK PHY     |
| 1           | TVWS OFDM PHY    |
| 2           | TVWS NB OFDM PHY |
| 3           | Reserved         |



The Operating Parameters field for a TVWS FSK PHY shall be formatted as shown in Figure 7-72.

| Bits: 0 | 1–2        | 3            | 4         | 5              | 6–8  | 9                | 10         | 11–13    |
|---------|------------|--------------|-----------|----------------|------|------------------|------------|----------|
| FEC     | FEC scheme | Interleaving | Spreading | Data Whitening | Mode | Modulation Index | SFD Length | Reserved |

**Figure 7-72—Operating Parameters field format for TVWS FSK PHY**

The FEC field shall be set to one if FEC is enabled and shall be set to zero otherwise.

The FEC Scheme field indicates the type of FEC that will be used. When the FEC field is set to zero, the FEC Scheme field is reserved. Valid values of the FEC Scheme field are given in Table 7-34.

**Table 7-34—FEC Scheme field values**

| Field value | FEC scheme used (as defined in 25.2.2) |
|-------------|----------------------------------------|
| 0           | First FEC scheme                       |
| 1           | Second FEC scheme                      |
| 2           | Third FEC scheme                       |
| 3           | Reserved                               |

The Interleaving field shall be set to one if interleaving is enabled and shall be set to zero otherwise.

NOTE—The Interleaving field is reserved if FEC is not enabled or the FEC Scheme field value is not one.

The Spreading field shall be set to one if spreading is enabled and shall be set to zero otherwise.

The Data Whitening field shall be set to one if data whitening is enabled and shall be set to zero otherwise.

The Mode field shall be set to the TVWS FSK mode, as defined in Table 25-3, that is used.

For Mode field values of one, two or three, the Modulation Index field shall be set to zero for a modulation index of 0.5 and shall be set to one for a modulation index of 1.0. For all other values of the Mode field, the Modulation Index field is reserved.

The SFD Length field shall be set to zero for a 16-bit SFD and shall be set to one for a 24-bit SFD.

The Operating Parameters field for a TVWS OFDM PHY shall be formatted as illustrated in Figure 7-73.

| Bits: 0–1  | 2–4      | 5–13     |
|------------|----------|----------|
| Modulation | MCS Mode | Reserved |

**Figure 7-73—Operating Parameters field format for TVWS OFDM PHY**

The Modulation field specifies the modulation that is used. Valid values for the Modulation field are given in Table 7-35.

**Table 7-35—Modulation field values**

| Field value | Modulation type |
|-------------|-----------------|
| 0           | BPSK            |
| 1           | QPSK            |
| 2           | 16-QAM          |
| 3           | Reserved        |

The MCS Mode field shall be set to one of the values in Table 7-36.

**Table 7-36— MCS Mode field values for TVWS OFDM**

| MCS Mode field | MCS      |
|----------------|----------|
| 0              | Reserved |
| 1              | MCS 1    |
| 2              | MCS 2    |
| 3              | MCS 3    |
| 4              | MCS 4    |
| 5              | MCS 5    |
| 6–7            | Reserved |

The Operating Parameters field for a TVWS NB OFDM PHY shall be formatted as shown in Figure 7-74.

| Bits: 0–3 | 4                   | 5–13     |
|-----------|---------------------|----------|
| MCS Mode  | Channel Aggregation | Reserved |

**Figure 7-74—Operating Parameters field format for TVWS NB OFDM PHY**

The MCS Mode field shall be set to one of the values in Table 7-37. All other values are reserved.

**Table 7-37—MCS Mode field values for TVWS NB OFDM**

| MCS Mode field | MCS      |
|----------------|----------|
| 0              | Reserved |
| 1              | MCS 1    |
| 2              | MCS 2    |
| 3              | MCS 3    |
| 4              | MCS 4    |
| 5              | MCS 5    |
| 6              | MCS 6    |
| 7              | MCS 7    |
| 8              | MCS 8    |
| 9–15           | Reserved |

The Channel Aggregation shall be set to one if channel aggregation is enabled and shall be set to zero otherwise.

The Available Frequency Range field shall be only be present when the Band ID field value is less than or equal to five. The Available Frequency Range field shall be formatted as illustrated in Figure 7-91 with field values described in 7.4.4.23.

#### 7.4.4.19 TVWS Device Capabilities IE

The presence of this IE in a transmitted frame indicates that the device supports operation of a TVWS PHY. The TVWS Device Capabilities IE Content field shall be formatted as shown in Figure 7-75.

| Octets: 1     | 3               | 3                           | variable                |
|---------------|-----------------|-----------------------------|-------------------------|
| TVWS PHY Type | Supported Bands | TVWS Supported PHY Features | TVWS Supported Channels |

**Figure 7-75—TVWS Device Capabilities IE**

The TVWS PHY Type field indicates the PHY type being described in the IE. Valid values of the TVWS PHY Type field shall be set to one of the non-reserved values shown in Table 7-38.

**Table 7-38—TVWS PHY Type field values**

| Value | Description  |
|-------|--------------|
| 0     | TVWS-FSK     |
| 1     | TVWS-OFDM    |
| 2     | TVWS-NB-OFDM |
| 3–255 | Reserved     |

The Supported Bands field is a bitmap indicating the bands, as defined in Table 10-1, supported by the device. A value of one indicates that the band is supported, and zero indicates the band is not supported. The supported bands shall be encoded as shown in Table 7-39. The device shall indicate only those bands that are implemented and defined for the indicated PHY type.

**Table 7-39—Supported Bands field values**

| Bit number | Band ID          |
|------------|------------------|
| 0          | TVWS Band USA    |
| 1          | TVWS Band UK     |
| 2          | TVWS Band Japan  |
| 3          | TVWS Band Canada |
| 4          | TVWS Band Korea  |
| 5          | TVWS Band EU     |
| 6          | 450 MHz          |
| 7          | 470 MHz          |
| 8          | 780 MHz          |
| 9          | 863 MHz          |
| 10         | 896 MHz          |
| 11         | 901 MHz          |
| 12         | 915 MHz          |
| 13         | 917 MHz          |
| 14         | 928 MHz          |
| 15         | 920 MHz          |
| 16         | Reserved         |
| 17         | 2450 MHz         |
| 18–23      | Reserved         |

The TVWS Supported PHY Features field indicates the supported features of a TVWS PHY. The content depends on the value of the TVWS PHY Type field.

The Supported Features field for TVWS FSK PHY shall be formatted as shown in Figure 7-76.

| Bits: 0           | 1–4        | 5–8              | 9              | 10                            | 11                                | 12      | 13–20          | 21–23    |
|-------------------|------------|------------------|----------------|-------------------------------|-----------------------------------|---------|----------------|----------|
| 24-bit SFD Length | FEC Scheme | Spreading Factor | Data Whitening | Alternating Spreading Pattern | Non-alternating Spreading Pattern | Ranging | Operating Mode | Reserved |

**Figure 7-76—TVWS FSK PHY Supported Features field format**

The 24-bit SFD Length field shall be set to one if the 24-bit SFD length is supported and shall be set to zero otherwise.

The FEC Scheme field is a bitmap that indicates if an FEC scheme is supported. Valid values are given in Table 7-40.

**Table 7-40—FEC Scheme field valid values**

| Bit number | Description                                                |
|------------|------------------------------------------------------------|
| 1          | First FEC scheme as defined in 20.2.2                      |
| 2          | Second FEC scheme as defined in 20.2.2                     |
| 3          | Interleaver for the second FEC scheme as defined in 20.2.2 |
| 4          | Third FEC scheme as defined in 20.2.2                      |

The Spreading Factor field is a bitmap that indicates if a spreading factor is supported. Valid values are given in Table 7-41.

**Table 7-41—Spreading Factor field valid values**

| Bit number | Description                   |
|------------|-------------------------------|
| 5          | Spreading factor 2 supported  |
| 6          | Spreading factor 4 supported  |
| 7          | Spreading factor 8 supported  |
| 8          | Spreading factor 16 supported |

The Data Whitening field shall be set to one if data whitening is supported and shall be set to zero otherwise.

The Alternating Spreading Pattern field shall be set to one if alternating spreading patterns are supported and shall be set to zero otherwise.

The Non-alternating Spreading Pattern field shall be set to one if non-alternating spreading patterns are supported and shall be set to zero otherwise.

The Ranging field shall be set to one if ranging is supported and shall be set to zero otherwise.

The Operating Mode field is a bit map that indicates if a particular operating mode is supported. Valid values of the operating mode field are given in Table 7-42.

**Table 7-42—Operating Mode field valid values**

| Bit number | Description                        |
|------------|------------------------------------|
| 13         | Mode#1 with modulation index = 0.5 |
| 14         | Mode#1 with modulation index = 1   |
| 15         | Mode#2 with modulation index = 0.5 |
| 16         | Mode#2 with modulation index = 1   |
| 17         | Mode#3 with modulation index = 0.5 |
| 18         | Mode#3 with modulation index = 1   |
| 19         | Mode#4                             |
| 20         | Mode#5                             |

The Supported Features field for TVWS OFDM PHY shall be formatted as shown in Figure 7-77.

| Bits: 0 | 1    | 2     | 3–6                   | 7       | 8–23     |
|---------|------|-------|-----------------------|---------|----------|
| MCS3    | MCS4 | MCS 5 | Number of STF Symbols | Ranging | Reserved |

**Figure 7-77—TVWS-OFDM PHY Supported Features field format**

The MCS3 field shall be set to one if MCS3, as defined in 26.3, is supported and shall be set to zero otherwise.

The MCS4 field shall be set to one if MCS4, as defined in 26.3, is supported and shall be set to zero otherwise.

The MCS5 field shall be set to one if MCS5, as defined in 26.3, is supported and shall be set to zero otherwise.

The Number of STF Symbols field shall be set to the maximum number of STF symbols, as defined in 26.2.1, supported by the device during reception.

The Ranging field shall be set to one if ranging is supported and shall be set to zero otherwise.

The Supported Features field for TVWS NB OFDM PHY shall be formatted as shown in Figure 7-78.

| Bits: 0             | 1                  | 2                 | 3                   | 4                  | 5                    | 6                   | 7       | 8–23     |
|---------------------|--------------------|-------------------|---------------------|--------------------|----------------------|---------------------|---------|----------|
| Channel Aggregation | Cyclic prefix 1/16 | Cyclic prefix 1/8 | Guard Interval 1/16 | Guard Interval 1/8 | Symbol Interval 1/16 | Symbol Interval 1/8 | Ranging | Reserved |

**Figure 7-78—TVWS NB OFDM PHY Supported Features field format**

The Channel Aggregation field shall be set to one if channel aggregation is supported and shall be set to zero otherwise.

The Cyclic Prefix 1/16 field shall be set to one if a 1/16 cyclic prefix is supported and shall be set to zero otherwise.

The Cyclic Prefix 1/8 field shall be set to one if a 1/8 cyclic prefix is supported and shall be set to zero otherwise.

The Guard Interval 1/16 field shall be set to one if a 1/16 guard interval is supported and shall be set to zero otherwise.

The Guard Interval 1/8 field shall be set to one if a 1/8 guard interval is supported and shall be set to zero otherwise.

The Symbol Interval 1/16 field shall be set to one if a 1/16 symbol interval is supported and shall be set to zero otherwise.

The Symbol Interval 1/8 field shall be set to one if a 1/8 symbol interval is supported and shall be set to zero otherwise.

The Ranging field shall be set to one if ranging is supported and shall be set to zero otherwise.

The Channels Supported field is a set of channel lists that shall be formatted as described in Figure 7-79.

| Octets: 1/variable      | 1/variable              | ... | 1/variable                     |
|-------------------------|-------------------------|-----|--------------------------------|
| Channel List for Band 1 | Channel List for Band 2 | ... | Channel List for Band <i>n</i> |

**Figure 7-79—TVWS Supported Channels field format**

The TVWS Supported Channels field content depends on the value of the Supported Bands field. For each defined band, the channel numbering is given in Table 7-39. For each band indicated as supported, a corresponding Channel List for Band field map shall be included. Each Channel List for Band field shall be encoded as shown in Figure 7-80. The first bit field of each map indicates whether all channels in that band are supported. If this field is set to one, then all channels defined for the band in Table 7-39 are supported, and the channel list is 1 octet, with bits 1 to 7 set to zero. If the first bit field is set to zero (i.e., not all channels in that band are supported), then the subsequent bit fields indicate which individual channels are supported. The bit field corresponding to a channel number shall be set to one to indicate that the channel is supported and set to zero to indicate the channel is not supported. Bit maps are allocated on octet boundaries; unused bits are reserved. When multiple bands are supported, as indicated in the Supported Bands field, the corresponding channel lists are concatenated in order, such that the channel lists occur in the order of the bands given in Table 7-39, i.e., channel list corresponding to the band indicated by bit 0 of the Supported Bands field is first if bit 0 is set to one. The term channel here refers to a WPAN channel, as defined in 10.1.2.8. The mapping between a TVWS channel and a PHY channel is not within the scope of this standard.

| 0                      | 1                   | 2                   | ... | <i>n</i>                   |
|------------------------|---------------------|---------------------|-----|----------------------------|
| All channels supported | Channel 1 supported | Channel 2 supported | ... | Channel <i>n</i> supported |

**Figure 7-80—Channel List field format**

**7.4.4.20 TVWS Device Category IE**

The TVWS Device Category IE Content field shall be formatted as in Figure 7-81.

|                  |
|------------------|
| <b>Octets: 1</b> |
| Device Category  |

**Figure 7-81—TVWS Device Category IE Content field format**

Valid values of the Device Category field are given in Table 7-43.

**Table 7-43—Device Category field values**

| Value | Description                           |
|-------|---------------------------------------|
| 0     | Stationary and independent device     |
| 1     | Stationary and dependent device       |
| 2     | Non-stationary and independent device |
| 3     | Non-stationary and dependent device   |
| 4–255 | Reserved                              |

**7.4.4.21 TVWS Device Identification IE**

The TVWS Device Identification IE Content field shall be formatted as shown Figure 7-82.

|                  |                 |
|------------------|-----------------|
| <b>Octets: 1</b> | <b>variable</b> |
| ID Type          | Device ID       |

**Figure 7-82—TVWS Device Identification IE Content field format**

Valid values of the ID Type field are given in Table 7-44.

**Table 7-44—ID Type field values**

| Value | Description                           |
|-------|---------------------------------------|
| 0     | US specific regulator assigned ID     |
| 1     | UK specific regulator assigned ID     |
| 2     | Canada specific regulator assigned ID |
| 3     | Japan specific regulator assigned ID  |
| 4     | Korea specific regulator assigned ID  |
| 5     | EU specific regulator assigned ID     |



**Table 7-44—ID Type field values (continued)**

| Value | Description                |
|-------|----------------------------|
| 6     | Manufacturer serial number |
| 7     | Vendor specific            |
| 8–255 | Reserved                   |

For an ID Type field indicated as regulator assigned, the Device ID field shall be formatted as shown in Figure 7-83. For other non-reserved values in Table 7-44, the Device ID field contains only the ID String field.

|                 |           |
|-----------------|-----------|
| Octets: 1       | variable  |
| Device Category | ID String |

**Figure 7-83—Device ID field format**

The Device Category field is defined in 7.4.4.20.

The ID String field shall be formatted as shown in Figure 7-84.

|           |          |
|-----------|----------|
| Octets: 1 | variable |
| Length    | ID Value |

**Figure 7-84—ID String field format**

The Length field specifies the number of octets that follows in the ID Value field.

The encoding of characters into the ID Value field is defined by regulators or can be vendor specific and is outside the scope of this standard.

#### 7.4.4.22 TVWS Device Location IE

The TVWS Device Location IE Content field shall be formatted as shown Figure 7-85.

|                     |                   |                   |     |                          |
|---------------------|-------------------|-------------------|-----|--------------------------|
| Octets: 1           | 17                | 17                | ... | 17                       |
| Number of Locations | Device Location 1 | Device Location 2 | ... | Device Location <i>n</i> |

**Figure 7-85—TVWS Device Location IE Content field format**

The Number of Locations field indicates the number of Device Location fields in the IE.

The Device Location field shall be formatted as shown in Figure 7-86.

|                  |                         |
|------------------|-------------------------|
| <b>Octets: 1</b> | <b>16</b>               |
| Location ID      | Device Location Element |

**Figure 7-86—Device Location field format**

The Location ID field provides an ID of a particular location for which a device is requesting channel list info. This field is used as part of the channel list response to allocate available channel info to a particular location ID.

The Device Locations Element field shall be encoded as described in IETF RFC 6225, section 2.1. The first two fields, Option Code and OptLen are omitted for the Device Location Element field.

#### 7.4.4.23 TVWS Channel Information Query IE

The TVWS Channel Information Query IE Content field shall be formatted as shown in Figure 7-87.

|                  |                     |                   |
|------------------|---------------------|-------------------|
| <b>Octets: 1</b> | <b>1</b>            | <b>0/variable</b> |
| Channel List ID  | Channel Info Status | Channel List Info |

**Figure 7-87—TVWS Channel Information Query IE Content field format**

The Channel List ID field is incremented when the channel data is updated. When the Channel Info Status field indicates that this is a channel data request, the Channel List ID field is set to the ID value provided when channel data was last received. If channel data has not been received the Channel List ID field is set to zero in the request.

The Channel Info Status field shall be formatted as shown in Figure 7-88.

|                               |                     |            |
|-------------------------------|---------------------|------------|
| <b>Bits: 0</b>                | <b>1–3</b>          | <b>4–7</b> |
| Channel List Request/Response | Number of Locations | Reserved   |

**Figure 7-88—Channel Info Status Content field format**

The Channel List Request/Response field shall be set to zero if this is a request for information and shall be set to one if it is a response to a request. When the Channel Info Status field indicates a request, TVWS Device Identification IEs and a TVWS Device Location IE may be included in the request frame.

When the Channel List Request/Response field is set to indicate a response, the Number of Locations field indicates the number of locations for which the channel info is available. Otherwise, the Number of Locations field is reserved.

The Channel List info field shall be present when Channel List Request/Response is set to indicate a response.

Each entry in the Channel List Info field contains the specific information on available channels as shown in Figure 7-89.

| Octets: 1   | 1                   | 0/1                     | 0/variable        |
|-------------|---------------------|-------------------------|-------------------|
| Location ID | Channel List Status | Number of TVWS Channels | Available Channel |

**Figure 7-89—Channel List Info field format**

The Channel List Status field shall be present when Channel list Request/Response is set to indicate a response. Valid values of the Channel List Status field are given in Table 7-45.

**Table 7-45—Channel List Status field values**

| Status | Description                                                                          |
|--------|--------------------------------------------------------------------------------------|
| 0      | Available channel list verified for a device location                                |
| 1      | Request not successful due to device ID not being verified                           |
| 2      | Request not successful due to device location being out of the geographic coordinate |
| 3      | Request not successful due to one or more parameters having invalid values           |
| 4–255  | Reserved                                                                             |

When the Channel Info Status field indicates that this is a response with available channel list for verified device locations, the Number of TVWS Channels field and Available Channel field are included in the IE. For other status values these fields are not present.

The Available Channel field shall be formatted as shown in Figure 7-90.

| Octets: 5                 | 1                | 2          |
|---------------------------|------------------|------------|
| Available Frequency Range | Maximum TX Power | Valid Time |

**Figure 7-90—Available Channel field format**

Available Frequency Range field shall be formatted as shown in Figure 7-91.

| Octets: 3          | 2         |
|--------------------|-----------|
| Starting Frequency | Bandwidth |

**Figure 7-91—Available Frequency Range field format**

The Starting Frequency field specifies the starting frequency, in units of 1 kHz, where the TVWS spectrum is available.

The Bandwidth field indicates the bandwidth, in units of 1 kHz, of the available TVWS spectrum.

The Maximum TX Power field contains the maximum allowed transmit EIRP authorized for TVWS channel, encoded as a two's-complement number in 0.5 dBm increments, in the range of –64 dBm to 63.5 dBm.

The Valid Time field contains the time, in minutes from the time of transmission, that the channel availability data is expected to remain valid; a valid time of zero indicates that the channel is available until further notice (e.g., as might be used for contact verification).

#### 7.4.4.24 TVWS Channel Information Source IE

The TVWS Channel Information Source IE shall be formatted as shown in Figure 7-92.

| Octets: 1   | 0/16                  | 0/8                  | 0/8                            |
|-------------|-----------------------|----------------------|--------------------------------|
| Source Info | Known Source Location | Known Source Address | Known Source Available Channel |

**Figure 7-92—TVWS Channel Information Source IE Content field format**

The Source Info field shall be formatted as shown in Figure 7-93.

| Bits: 0          | 1               | 2                         | 3–7      |
|------------------|-----------------|---------------------------|----------|
| Location Present | Address Present | Available Channel Present | Reserved |

**Figure 7-93—Source Info field format**

The Location Present field shall be set to one if the Known Source Location field is present and shall be set to zero otherwise.

The Address Present field shall be set to one if the Known Source Address field is present and shall be set to zero otherwise.

The Available Channel Present field shall be set to one if the Known Source Available Channel field is present and shall be set to zero otherwise.

The Known Source Location field contains the location of the device acting as the source of channel availability data and shall be encoded as described in IETF RFC 6225, section 2.1. First two fields, Option Code and OptLen are omitted for the Device Location Element field.

The Known Source Address field contains the extended address of the known source device.

The Known Source Available Channel field contains the TVWS channel being used by the known source and shall be formatted as shown in Figure 7-90.

**7.4.4.25 CTM IE**

The CTM IE Content field shall be formatted as shown in Figure 7-94.

|                  |                              |            |                                     |
|------------------|------------------------------|------------|-------------------------------------|
| <b>Octets: 1</b> | <b>15</b>                    | <b>...</b> | <b>15</b>                           |
| CTM Control      | Channel Timing Information 1 | ...        | Channel Timing Information <i>n</i> |

**Figure 7-94—CTM IE Content field format**

Valid values of the CTM Control field are given in Table 7-46.

**Table 7-46—CTM Control field values**

| Value | Description                                                                        |
|-------|------------------------------------------------------------------------------------|
| 0     | Request for channel timing information                                             |
| 1     | Success with full channel timing information on the available channels             |
| 2     | Success with subset of channel timing information on the available channels        |
| 3     | Success with no channel timing changes from previous query                         |
| 4     | Request declined due to unspecified reason                                         |
| 5     | Request declined because of no capability for providing channel timing information |
| 6     | Request declined, database access timeout                                          |
| 7–255 | Reserved                                                                           |

The Channel Timing Information fields shall only be present, when it is a response to a request and the CTM Control field value indicates that channel timing information is available. The Channel Timing Information field shall be formatted as shown Figure 7-95.

|                           |                                    |            |
|---------------------------|------------------------------------|------------|
| <b>Octets: 5</b>          | <b>8</b>                           | <b>2</b>   |
| Available Frequency Range | Channel Availability Starting Time | Valid Time |

**Figure 7-95—Channel Timing Information field format**

The Available Frequency Range field is defined in Figure 7-91.

The Channel Availability Starting Time field indicates the starting time in Coordinated Universal Time (UTC) from when the channel indicated in the Channel Number field is available for operation. UTC is defined by CCIR Recommendation 460-4.

The Valid Time field indicates the duration of frequency availability as described in 7.4.4.23.

The information in Figure 7-95 may be aggregated to show multiple durations of the channel time scheduling.

**7.4.4.26 Timestamp IE**

The Timestamp IE is shall be formatted as shown in Figure 7-96.

|                    |
|--------------------|
| <b>Octets: 4</b>   |
| Transmit Timestamp |

**Figure 7-96—Timestamp IE Content field format**

The Transmit Timestamp field shall be set to the time, in units of 10 picoseconds, of the transmitter time reference at the antenna, when the packet containing this IE is transmitted. The timing reference is the RMARKER of the frame, as defined in 6.9.1.

**7.4.4.27 Timestamp Difference IE**

The Timestamp Difference IE shall be formatted as illustrated in Figure 7-97.

|                       |
|-----------------------|
| <b>Octets: 4</b>      |
| RX TX Time Difference |

**Figure 7-97—Timestamp Difference IE Content field format**

The RX TX Time Difference field contains the difference in time, in units of 10 picoseconds, from the time the most recent RFRAME was received to the time the frame containing the IE is transmitted. The reference for these time values is the RMARKER.

**7.4.4.28 TMCTP Specification IE**

The TMCTP Specification IE shall be formatted as illustrated in Figure 7-98

| <b>Bits: 0–3</b> | <b>4</b>            | <b>5</b>                  | <b>6</b>                      | <b>7</b>                            | <b>8–15</b>      | <b>16–23</b>              | <b>variable</b> |
|------------------|---------------------|---------------------------|-------------------------------|-------------------------------------|------------------|---------------------------|-----------------|
| BOP Order        | TMCTP Frame Pending | DBS Allocation Capability | Channel Allocation Capability | Channel Allocation Relay Capability | Hop Count to SPC | Number of PAN IDs Pending | PAN ID List     |

**Figure 7-98—TMCTP Specification IE Content field format**

The BOP Order field contains the value *macTmctpExtendedOrder* as an unsigned integer.

The TMCTP Frame Pending field shall be set to one if the TMCTP-parent PAN coordinator has more frames for the TMCTP-child PAN coordinator. Otherwise, this field shall be set to zero.

The DBS Allocation Capability field shall be set to one if the device is capable of allocating a DBS to the TMCTP-child PAN coordinator; it shall be set to zero otherwise.

The Channel Allocation Capability field shall be set to one if the device is capable of allocating a dedicated channel to the TMCTP-child PAN coordinator; it shall be set to zero otherwise.

The Channel Allocation Relay Capability field shall be set to one if the device is capable of relaying a DBS request of the TMCTP-child PAN coordinator; it shall be set to zero otherwise.

The Hop Count to SPC field indicates the number of hops to reach the SPC.

The Number of PAN IDs Pending field indicates the number of PAN IDs contained in the PAN ID List field of the beacon frame.

The PAN ID List field contains the PAN IDs of the TMCTP-child PAN coordinators that currently have messages pending with the TMCTP-parent PAN coordinator.

#### 7.4.4.29 RCC PHY Operating Mode IE

The RCC PHY Operating Mode IE Content field shall be formatted as shown in Figure 7-99.

| Bits: 0–13     | 14–18          | 19–22          | 23–25     | 26–27              | 28–31    |
|----------------|----------------|----------------|-----------|--------------------|----------|
| Channel Number | Operating Band | PHY/Modulation | Chip Rate | Spreading Sequence | Reserved |

**Figure 7-99—RCC PHY Operating Mode IE Content field format**

The Channel Number field contains a valid channel number for an RCC PHY, as defined in 10.1.2.11.

The Operating Band field shall be set to the current operating band where the bands are defined as the integers greater than zero that correspond to the bit numbers given in Table 7-12.

The PHY/Modulation field shall be set to the current PHY and modulation where the value corresponds to the bit number given in Table 7-13.

The Chip Rate field is valid for LMR PHY and is reserved otherwise. Valid values for the Chip Rate field are given in Table 7-47.

**Table 7-47—Chip Rate field values**

| Field value | Definition   |
|-------------|--------------|
| 0           | 300 kchip/s  |
| 1           | 600 kchip/s  |
| 2           | 800 kchip/s  |
| 3           | 1 Mchips/s   |
| 4           | 1.6 Mchips/s |
| 5           | 2 Mchips/s   |
| 6           | 3 Mchips/s   |
| 7           | 4 Mchips/s   |

The Spreading Sequence field is valid for LMR PHY is reserved otherwise. Valid value for the Spreading Sequence field are given in Table 7-48.

**Table 7-48—Spreading Sequence field values**

| Field value | Definition |
|-------------|------------|
| 0           | 11-chip    |
| 1           | 15-chip    |
| 2           | 20-chip    |
| 3           | 40-chip    |

#### 7.4.4.30 Vendor Specific Nested IE

The Vendor Specific Nested IE is reserved for the use of other protocols and/or data relevant only to certain implementations. The Vendor Specific Nested IE Content field shall be formatted as illustrated in Figure 7-100.

| Octets: 3  | Variable                    |
|------------|-----------------------------|
| Vendor OUI | Vendor Specific Information |

**Figure 7-100—Vendor Specific Nested IE Content field format**

The Vendor OUI field is defined in 7.4.2.2. A value of the Vendor OUI field not understood by a receiving device causes the remainder of this IE to be ignored.

The Vendor Specific Information field is defined by the vendor identified in the Vendor OUI field. Its use is outside of the scope of this standard.

#### 7.4.4.31 Channel hopping IE

The Channel Hopping IE Content field shall be formatted as illustrated in Figure 7-101. The Channel Hopping IE may be sent with only the Hopping Sequence ID field to reduce the size of the beacon. Otherwise, all the fields are included.

| Octets: 1           | 0/1          | 0/2                | 0/4               | 0/variable      | 0/2                     | 0/variable       | 0/2         |
|---------------------|--------------|--------------------|-------------------|-----------------|-------------------------|------------------|-------------|
| Hopping Sequence ID | Channel Page | Number of Channels | PHY Configuration | Extended Bitmap | Hopping Sequence Length | Hopping Sequence | Current Hop |

**Figure 7-101—Channel Hopping Timing IE Content field format**

The Hopping Sequence ID field shall be set equal to the value of *macHoppingSequenceId*.

The Channel Page field is shall be set equal to the value of *macChannelPage*.

The Number of Channels field shall be set equal to the value of *macNumberOfChannels*.



The PHY Configuration shall be set to *macPhyConfiguration*.

The Extended Bitmap field is valid only for channel pages 9 and 10. The Extended Bitmap field contains *macExtendedBitmap* which is a bitmap in which a bit is set to one if a channel is to be used and is set to zero otherwise. For channel pages 9 and 10, the Extended Bitmap field is *macNumberOfChannels* bits long, with additional bits added to make it an integer number of octets. For all other values of the Channel Page field, it is zero length.

The Hopping Sequence Length field shall be set to the number of channels in the Hopping Sequence field.

The Hopping Sequence field is a set of unsigned integers, each two octets in length, that contains a channel number in the hopping sequence.

The Current Hop field shall be set to the index of the current position in the hopping sequence list.

## 7.5 MAC commands

### 7.5.1 Command ID field

The MAC commands are listed in Table 7-49 along with their associated command ID. All FFDs shall be capable of transmitting and receiving all MAC command with Command ID field of values 0x01–0x08. The requirements for an RFD are indicated by an “X” in the table. An FFD supporting one of TRLE, DSME, RIT or DBS options shall support the associated MAC commands as identified by the associated functional group prefix, e.g., “DSME” for the DSME option.

**Table 7-49—MAC commands**

| Command ID | Command name                         | RFD |    | Subclause |
|------------|--------------------------------------|-----|----|-----------|
|            |                                      | TX  | RX |           |
| 0x01       | Association Request command          | X   |    | 7.5.2     |
| 0x02       | Association Response command         |     | X  | 7.5.3     |
| 0x03       | Disassociation Notification command  | X   | X  | 7.5.4     |
| 0x04       | Data Request command                 | X   |    | 7.5.5     |
| 0x05       | PAN ID Conflict Notification command | X   |    | 7.5.6     |
| 0x06       | Orphan Notification command          | X   |    | 7.5.7     |
| 0x07       | Beacon Request command               |     |    | 7.5.8     |
| 0x08       | Coordinator realignment command      |     | X  | 7.5.10    |
| 0x09       | GTS request command                  |     |    | 7.5.11    |
| 0x0a       | TRLE Management Request command      |     |    | F.5.2.1   |
| 0x0b       | TRLE Management Response command     |     |    | F.5.2.2   |
| 0x0c–0x12  | Reserved                             |     |    |           |
| 0x13       | DSME Association Request command     |     |    | 7.5.12    |
| 0x14       | DSME Association Response command    |     |    | 7.5.13    |

**Table 7-49—MAC commands (continued)**

| Command ID | Command name                                | RFD |    | Subclause |
|------------|---------------------------------------------|-----|----|-----------|
|            |                                             | TX  | RX |           |
| 0x15       | DSME GTS Request command                    |     |    | 7.5.14    |
| 0x16       | DSME GTS Response command                   |     |    | 7.5.15    |
| 0x17       | DSME GTS Notify command                     |     |    | 7.5.16    |
| 0x18       | DSME Information Request command            |     |    | 7.5.17    |
| 0x19       | DSME Information Response command           |     |    | 7.5.18    |
| 0x1a       | DSME Beacon Allocation Notification command |     |    | 7.5.19    |
| 0x1b       | DSME Beacon Collision Notification command  |     |    | 7.5.20    |
| 0x1c       | DSME Link Report command                    |     |    | 7.5.21    |
| 0x1d–0x1f  | Reserved                                    |     |    |           |
| 0x20       | RIT Data Request command                    |     |    | 7.5.22    |
| 0x21       | DBS Request command                         |     |    | 7.5.23    |
| 0x22       | DBS Response command                        |     |    | 7.5.24    |
| 0x23       | RIT Data Response command                   |     |    | 7.5.25    |
| 0x24       | Vendor Specific command                     |     |    | 7.5.26    |
| 0x25–0xff  | Reserved                                    |     |    | —         |

**7.5.2 Association Request command**

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, except for RFD-RX and RFD-TX devices, shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The Source Addressing Mode field shall be set to indicate extended addressing. The Destination Addressing Mode field shall be set to the same mode as indicated in the Beacon frame to which the Association Request command refers.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID 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. If the Version field is set to 0b10, the Source PAN ID field is omitted, otherwise the Source PAN ID field shall contain the broadcast PAN ID. The Source Address field shall contain the value of *macExtendedAddress*.

The Association Request command Content field shall be formatted as illustrated in Figure 7-102.

|                        |
|------------------------|
| <b>Octets: 1</b>       |
| Capability Information |

**Figure 7-102—Association Request command Content field format**

The Capability Information field shall be formatted as illustrated in Figure 7-103.

| Bits: 0  | 1           | 2            | 3                     | 4                | 5        | 6                   | 7                |
|----------|-------------|--------------|-----------------------|------------------|----------|---------------------|------------------|
| Reserved | Device Type | Power Source | Receiver On When Idle | Association Type | Reserved | Security Capability | Allocate Address |

**Figure 7-103—Capability Information field format**

The Device Type field shall be set to one if the device is an FFD. Otherwise, the Device Type field shall be set to zero to indicate an RFD.

The Power Source field shall be set to one if the device is receiving power from the alternating current mains. Otherwise, the Power Source field shall be set to zero.

The Receiver On When Idle field 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 field shall be set to zero.

The Security Capability field shall be set to one if the device is capable of sending and receiving cryptographically protected MAC frames as specified in 9.2; it shall be set to zero otherwise.

The Allocate Address field shall be set to one if the device wishes the coordinator to allocate a short address as a result of the association procedure. Otherwise, it shall be set to zero.

The Association Type field shall be set to one if the device requests fast association and zero otherwise.

### 7.5.3 Association Response command

This command shall only be sent by the PAN coordinator or coordinator to a device that is currently trying to associate.

All devices, except for RFD-RX and RFD-TX devices, shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The Destination Addressing Mode and Source Addressing Mode fields shall each be set to indicate extended addressing.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The Destination Address field shall contain the extended address of the device requesting association. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The Association Response command Content field shall be formatted as illustrated in Figure 7-104.

|               |                    |
|---------------|--------------------|
| Octets: 2     | 1                  |
| Short Address | Association Status |

**Figure 7-104—Association Response command Content field format**

If the coordinator was not able to associate the 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 extended address.

Valid values of the Association Status field are defined in Table 7-50.

**Table 7-50—Valid values of the Association Status field**

| Association status | Description                         |
|--------------------|-------------------------------------|
| 0x00               | Association successful.             |
| 0x01               | PAN at capacity.                    |
| 0x02               | PAN access denied.                  |
| 0x03               | Hopping sequence offset duplication |
| 0x04–0x7f          | Reserved.                           |
| 0x80               | Fast association successful.        |
| 0x81–0xff          | Reserved.                           |

#### 7.5.4 Disassociation Notification command

The PAN coordinator, a coordinator, or an associated device may send the Disassociate Notification command.

All devices, except for RFD-RX and RFD-TX devices, shall implement this command.

The Destination Addressing Mode field shall be set according to the addressing mode specified by the corresponding primitive. The Source Addressing Mode field shall be set to indicate extended addressing.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID field shall contain the value of *macPanId*. The Source PAN ID field shall be omitted. If the coordinator is disassociating a device from the PAN, then the Destination Address field shall contain the address of the device being removed from the PAN. If an associated device is disassociating from the PAN, then the Destination Address field shall contain the value of either *macCoordShortAddress*, if the Destination Addressing Mode field is set to indicated short addressing, or *macCoordExtendedAddress*,

if the Destination Addressing Mode field is set to indicated extended addressing. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The Disassociation Notification command Content field shall be formatted as illustrated in Figure 7-105.

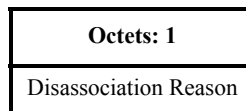


**Figure 7-105—Disassociation Notification command Content field format**

Valid values of the Disassociation Reason field are defined in Table 7-51.

**Table 7-51—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.                 |
| 0x03–0xff           | Reserved.                                           |

### 7.5.5 Data Request command

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* after the acknowledgment to an Association Request command.

All devices, except for RFD-TX devices, shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The Data Request command has a zero-length Content field.

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 field may be set to indicate that destination addressing information is not present if the Beacon frame indicated in its Superframe Specification field, as defined in 7.3.1.3, that it originated from the PAN coordinator, as defined in 7.2.1.8, 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 field shall be set according to the value of *macCoordShortAddress*. If *macCoordShortAddress* is equal to 0xffff, the Destination Addressing Mode field shall be set to indicate extended addressing, and the Destination Address field shall contain the value of *macCoordExtendedAddress*. Otherwise, the Destination Addressing Mode field shall be set to indicate short addressing 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 field shall be set according to the addressing mode used for the pending address. If the Source Addressing Mode field is set to indicate short addressing, the Source Address field shall contain the value of *macShortAddress*. Otherwise, the Source Addressing Mode field shall be set to indicate extended addressing and the Source Address field shall contain the value of *macExtendedAddress*.

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 field 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, the Destination Addressing Mode 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 field shall be set to use extended addressing.

If the Destination Addressing Mode field is set to indicate that destination addressing information is not present, the source PAN ID shall contain the value of *macPanId*. Otherwise, the Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

#### 7.5.6 PAN ID Conflict Notification command

All devices, except for RFD-RX and RFD-TX 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 has no Content field.

The Destination Addressing Mode and Source Addressing Mode fields shall both be set to indicate extended addressing.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The Destination Address field shall contain the value of *macCoordExtendedAddress*. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

#### 7.5.7 Orphan Notification command

All devices, except for RFD-RX and RFD-TX devices, shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The Orphan Notification command has no Content field.

The Source Addressing Mode field shall be set to indicate extended addressing. The Destination Addressing Mode field shall be set to indicate short addressing.

The Frame Pending field and AR field shall be set to zero and ignored upon reception.

The Destination PAN ID field shall contain the value of the broadcast PAN ID, while the Source PAN ID field shall be omitted. The Destination Address field shall contain the broadcast short address. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

#### 7.5.8 Beacon Request command

This command is optional for an RFD.

The Beacon Request command has no Content field.

The Destination Addressing Mode field shall be set to indicate short addressing, and the Source Addressing Mode field shall be set to indicate that the source addressing information is not present.

The Frame Pending field shall be set to zero and ignored upon reception. The AR field and Security Enabled field shall also be set to zero.

The Destination PAN ID field shall contain the broadcast PAN ID. The Destination Address field shall contain the broadcast short address.

#### 7.5.9 Enhanced Beacon Request command

The Enhanced Beacon Request command is a Beacon Request command with a Frame Version field set to 0b10.

The Enhanced Beacon Request command is optional for an FFD and an RFD.

The Enhanced Beacon Request command has no Content field.

The Enhanced Beacon Request command may contain an Enhanced Beacon Filter IE, as defined in 7.4.4.6, and/or a set of IE IDs. The Enhanced Beacon Filter allows a device to request Enhanced Beacon frames, as described in 6.3.4.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception.

The AR field shall be set to zero. If the Enhanced Beacon Request command is being sent on a particular PAN ID that is not the broadcast PAN ID, the Security Enable field may be set to one, otherwise it shall be set to zero.

The Destination PAN ID field, if present, shall contain either a specific PAN ID or a broadcast PAN ID if *macImplicitBroadcast* is FALSE.

Frames intended for all coordinators shall set the Destination Address field to the broadcast short address. Frames intended for a specific coordinator shall set the Destination Address field to the coordinator's short address.

#### 7.5.10 Coordinator realignment command

All devices, except for RFD-RX and RFD-TX devices, shall be capable of receiving this command, although an RFD is not required to be capable of transmitting it.

The Destination Addressing Mode field shall be set to indicate extended addressing if the command is directed to an orphaned device or set to indicate short addressing if it is to be broadcast to the PAN. The Source Addressing Mode field shall be set to indicate extended addressing.

The Frame Pending field shall be set to zero and ignored upon reception.

The AR 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 field shall be set to 0x01 if the Channel Page field is present.

The Destination PAN ID field when present shall contain the broadcast PAN ID. 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. The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macExtendedAddress*.

The Coordinator Realignment command Content field shall be formatted as illustrated in Figure 7-106.

| Octets: 2 | 2                         | 1              | 2             | Bits: 0–6                       | 7           |
|-----------|---------------------------|----------------|---------------|---------------------------------|-------------|
| PAN ID    | Coordinator Short Address | Channel Number | Short Address | Channel Page/<br>Channel Number | Page/Number |

**Figure 7-106—Coordinator Realignment command Content field format**

The PAN ID field shall contain the PAN ID that the coordinator intends to use for all future communications.

The Coordinator Short Address field shall contain the value of *macShortAddress*.

The Channel Page/Channel Number field and Page/Number field may be omitted from the Coordinator Realignment command. The Length field of the Coordinator Realignment command is used to determine if these fields are present.

The Page/Number field indicates the contents of the Channel Page/Channel Number field.

If the Page/Number field is set to zero, the Channel Page/Channel Number field 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. The Channel Number field shall contain the channel number that the coordinator intends to use for all future communications.

If the Page/Number field is set to one, the Channel Page/Channel Number field shall be used for the most significant bits of the channel number and the Channel Number field shall contain the least significant bits of the channel number.

For example, in the case that the Channel Page/Channel Number field contains 0x1E and the Channel Number field contains 0xad, then if the Page/Number field is set to zero, then the channel page is 0x1e and the channel number is 0xad. If the Page/Number field is set to one, then the channel page is unchanged and the channel number is 0x1ead.

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, the Short Address 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 extended address, this field shall contain the value 0xfffe.



**7.5.11 GTS request command**

Only devices that have been assigned a short address shall send GTS Request command. This command is optional.

The Destination Addressing Mode field shall be set to indicate that destination addressing information is not present, and the Source Addressing Mode field shall be set to indicate short addressing.

The Frame Pending field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The GTS Request command Content field shall be formatted as illustrated in Figure 7-107.

| Octets: 1           |
|---------------------|
| GTS Characteristics |

**Figure 7-107—GTS Request command Content field format**

The GTS Characteristics field shall be formatted as illustrated in Figure 7-108.

| Bits: 0–3  | 4             | 5                    | 6–7      |
|------------|---------------|----------------------|----------|
| GTS Length | GTS Direction | Characteristics Type | Reserved |

**Figure 7-108—GTS Characteristics field format**

The GTS Length field shall contain the number of superframe slots being requested for the GTS.

The GTS Direction field shall be set to one if the GTS is to be a receive-only GTS. Conversely, this field 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 field shall be set to one if the characteristics refer to a GTS allocation or zero if the characteristics refer to a GTS deallocation.

**7.5.12 DSME Association Request command**

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.

The Source Addressing Mode field of the Frame Control field shall be set to indicate extended addressing. The Destination Addressing Mode field shall be set to the same mode as indicated in the Beacon frame to which the DSME Association Request command refers.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID 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 DSME Association Request command is being sent. The Source PAN ID field shall contain the broadcast PAN ID. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The DSME Association Request command Content field shall be formatted as illustrated in Figure 7-109.

| Octets: 1              | 1                   | 2              | 0/1                          |
|------------------------|---------------------|----------------|------------------------------|
| Capability Information | Hopping Sequence ID | Channel Offset | Extended DSME GTS Allocation |

**Figure 7-109—DSME Association Request command Content field format**

The Capability Information field shall be formatted as illustrated in Figure 7-110.

| Bits: 0  | 1           | 2            | 3                     | 4                     | 5        | 6                   | 7                |
|----------|-------------|--------------|-----------------------|-----------------------|----------|---------------------|------------------|
| Reserved | Device Type | Power Source | Receiver On When Idle | DSME Association Type | Reserved | Security Capability | Allocate Address |

**Figure 7-110—Capability Information field format**

The Device Type field, Power Source field, Receiver On When Idle field, Security Capability field, and Allocate Address field are described in 5.3.1.2.

The DSME Association Type field shall be set to one if a device requests assignment of a DSME GTS by a coordinator during association. Otherwise, the DSME Association Type field shall be set to zero.

The Hopping Sequence ID field shall indicate the ID of channel hopping sequence in use. The Hopping Sequence ID of zero indicates that a default hopping sequence shall be used. The HoppingSequenceId of one indicates that a hopping sequence generated by PAN coordinator shall be used. The other value of the Hopping Sequence ID denotes the sequence set by a higher layer shall be used. A device is requesting a channel hopping sequence to its coordinator if the Hopping Sequence ID is one.

The Channel Offset field shall be set to the offset value of the unassociated device that wished to associate with a PAN, this value is specified by the next higher layer.

The Extended DSME GTS Allocation field shall be present if *macExtendedDsmeEnabled* is TRUE and the value of the DSME Association Type field is one. This field shall be formatted as illustrated in Figure 7-111.

| Bits: 0   | 1–4              | 5–7      |
|-----------|------------------|----------|
| Direction | Allocation Order | Reserved |

**Figure 7-111—Extended DSME GTS Allocation field format**

The Direction field specifies the direction of the DSME GTSs, which is relative to the data flow from the requesting device. The direction is specified as either transmit or receive. The value of this field shall be set to zero if the allocation is for transmission. The value shall be set to one if the allocation is for reception.

The Allocation Order field is described in 7.5.13.

**7.5.13 DSME Association Response command**

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.

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall each be set to indicate extended addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The Destination Address field shall contain the extended address of the device requesting association. The Source Address field shall contain the value of *macExtendedAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The DSME Association Response command Content field shall be formatted as illustrated in Figure 7-112.

| Octets: 2     | 1                  | 0/1                     | variable         | 0/1              | 0/1      | 0/2           | 0/1     | 0/2           |
|---------------|--------------------|-------------------------|------------------|------------------|----------|---------------|---------|---------------|
| Short Address | Association Status | Hopping Sequence Length | Hopping Sequence | Allocation Order | BI Index | Superframe ID | Slot ID | Channel Index |

**Figure 7-112—DSME Association Response command Content field format**

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 0xffff 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.

The Association Status field shall contain one of the nonreserved values listed in Table 7-50.

The Hopping Sequence Length field shall specify the length of the Hopping Sequence used in the PAN if the PAN runs in both beacon-enabled mode and channel hopping mode. When the value of HoppingSequenceId is other than one or DsmeAssociation is one, this field shall be set to zero. The Hopping Sequence field shall be present only if the value of the Hopping Sequence Length field is not zero.

The Hopping Sequence field shall contain the current value of *macHoppingSequenceList*. The size of the Hopping Sequence field is defined by the Hopping Sequence Length field and the Hopping Sequence field contains the current value of *macHoppingSequenceList*. This field shall be present only if the PAN runs in both beacon-enabled mode and channel hopping mode and the value of the Hopping Sequence Length field is not zero.

The Allocation Order field shall be present if *macExtendedDsmeEnabled* is TRUE and the DsmeAssociation parameter of the device requesting association is set to one. This field shall indicate the DSME GTS allocation interval and be set to the value of AllocationOrder, AO, of the device requesting association. The value of AO and the DSME GTS allocation interval are related as follows:

DSME-GTS allocation interval =  $BI \times 2^{(MO-BO)} / 2^{AO}$  for  $MO > BO$

If  $MO \leq BO$ , the DSME GTS allocation interval is the same as an MD.

The BI Index field shall be present if *macExtendedDsmeEnabled* is TRUE and the *DsmeAssociation* parameter of the device requesting association is set to one. This field shall contain the index of the beacon interval *macBiIndex*, BI, in which the DSME GTS needs to be allocated. The BI Index is the sequence number of the BI in a multi-superframe beginning from zero.

A device can locate the value of BI Index in which the DSME Associate response command is received by using the value of *macPanCoordinatorBsn* as follows:

BI Index =  $macPanCoordinatorBsn \% 2^{(MO-BO)}$

The Superframe ID field shall be present if *macExtendedDsmeEnabled* is TRUE and the *DsmeAssociation* parameter of the device requesting association is set to one. This field shall contain the index of the superframe in which the DSME GTS needs to be allocated. The Superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero. The superframe in which the PAN coordinator sends its beacons serves as the reference point (Superframe ID 0). An example of superframe IDs is illustrated in Figure 6-54.

The Slot ID field shall be present if *macExtendedDsmeEnabled* is TRUE and the *DsmeAssociation* parameter of the device requesting association is set to one. This field shall contain the index of the DSME GTS to be allocated. The slot ID is the sequence number of the DSME GTS in a superframe beginning from zero. An example of slot IDs is illustrated in Figure 6-54.

The Channel Index field shall be present if *macExtendedDsmeEnabled* is TRUE and the *DsmeAssociation* parameter of the device requesting association is set to one. This field shall contain the channel number of the DSME GTS to be allocated.

#### 7.5.14 DSME GTS Request command

Only devices that have been assigned a short address shall send this command in the CAP.

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to indicates short addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN ID field shall contain the value of *macPanId*, and the Destination Address field shall be set to the short address of the destination device.

The DSME GTS Request command Content field shall be formatted as illustrated in Figure 7-113.

| Octets: 1           | 0/1             | 0/2                  | 0/1               | Variable               | 0/1              |
|---------------------|-----------------|----------------------|-------------------|------------------------|------------------|
| DSME GTS Management | Number of Slots | Preferred Superframe | Preferred Slot ID | DSME SAB Specification | Allocation Order |

**Figure 7-113—DSME GTS Request command Content field format**

The DSME GTS Management field shall be formatted as illustrated in Figure 7-114.

| Bits: 0–2       | 3         | 4                          | 5–7      |
|-----------------|-----------|----------------------------|----------|
| Management Type | Direction | Prioritized Channel Access | Reserved |

**Figure 7-114—DSME GTS Management field format**

The Management Type field shall be set to one of the nonreserved values listed in Table 7-52.

**Table 7-52—Values of the Management Type field**

| Management Type value (b2 b1 b0) | Description                        |
|----------------------------------|------------------------------------|
| 000                              | Deallocation                       |
| 001                              | Allocation                         |
| 010                              | Duplicated allocation notification |
| 011                              | Reduce                             |
| 100                              | Restart                            |
| 101                              | DSME GTS expiration                |
| 110–111                          | Reserved                           |

The Direction field shall indicate the DSME GTSs are being allocated for TX or for RX of the requesting device. The value of this field is set to zero if the allocation is for TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not allocation.

The Prioritized Channel Access field shall be set to one if DSME GTS should be reserved as high priority, or set to zero if DSME GTS should be reserved as low priority. When the DSME GTS request command is used in the DSME GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME GTS.

The number of slots field shall contain the number DSME GTSs that this command is requesting. This field is valid only if the management type is allocation.

The Preferred Superframe ID field shall contain the index of the preferred superframe in which the DSME GTSs need be allocated. The superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero. The superframe in which the PAN coordinator sends its beacons serves as the reference point (Superframe ID 0). An example of superframe IDs are illustrated in Figure 6-50. This field is valid only if the management type is allocation.

The Preferred slot ID shall contain the index of the preferred slot from which the DSME GTSs need to be allocated. The slot ID is the sequence number of the DSME GTSs (not including beacon or CAP slots) in a superframe beginning from zero. An example of slot IDs are illustrated in Figure 6-50. This field is valid only if the management type is allocation.

The DSME SAB Specification field shall be formatted as illustrated in Figure 7-115.

| Octets: 1                 | 2                        | variable           |
|---------------------------|--------------------------|--------------------|
| DSME SAB Sub-block Length | DSME SAB Sub-block Index | DSME SAB Sub-block |

**Figure 7-115—DSME SAB Specification field format**

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB sub-block in the entire SAB as illustrated in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as described in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode. However, the meaning of zeros and ones in the DSME SAB sub-block has different meaning from the descriptions in those figures if the management type is not allocation. If the management type is allocation, the ones in the DSME SAB Sub-block field indicate readily allocated or unavailable slots and the zeros indicate vacant and available slots. If the management type is not allocation, the ones in the DSME SAB Sub-block field indicate the slots that are being requested for deallocation, duplicated allocation notification, reduce, or restart.

The Allocation Order field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall indicate the DSME GTS allocation interval and be set to the value of *AllocationOrder* of the device requesting a DSME GTS. The relationship between the value of this field and the DSME GTS allocation interval is described in 7.5.13.

#### 7.5.15 DSME GTS Response command

Only devices that have been assigned a short address shall send this command in the CAP.

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to indicate short addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN ID field shall contain the value of *macPanId*, and the Destination Address field shall be set to the broadcast PAN ID.

The DSME GTS Response command Content field shall be formatted as illustrated in Figure 7-116.

| Octets: 1           | 2                   | 0/2            | variable               | 0/1              | 0/1      | 0/2           | 0/1     | 0/2           |
|---------------------|---------------------|----------------|------------------------|------------------|----------|---------------|---------|---------------|
| DSME GTS Management | Destination Address | Channel Offset | DSME SAB Specification | Allocation Order | BI Index | Superframe ID | Slot ID | Channel Index |

**Figure 7-116—DSME GTS Response command Content field format**

The DSME GTS Management field shall be formatted as illustrated in Figure 7-117.

| Bits: 0–2       | 3         | 4                          | 5–7    |
|-----------------|-----------|----------------------------|--------|
| Management Type | Direction | Prioritized Channel Access | Status |

**Figure 7-117—DSME GTS Management field format**

The Management Type field shall be set to one of the nonreserved values listed in Table 7-52.

The Direction field shall indicate the DSME GTSs are being allocated for TX or for RX of the requesting device. The value of this field is set to 0 if the allocation is for TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not allocation.

The Prioritized Channel Access field shall be set to one if DSME GTS should be reserved as high priority, or set to zero if DSME GTS should be reserved as low priority. When the DSME GTS request command is used in the DSME GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME GTS.

The Status field indicates the result of a request. Valid values of the Status field are given in Table 7-53.

**Table 7-53—Status field values**

| Status (b0 b1 b2) | Description                       |
|-------------------|-----------------------------------|
| 000               | Approved                          |
| 001               | Disapproved: lack of availability |
| 010               | Disapproved, unknown GTS          |
| 011–111           | Reserved                          |

The DSME GTS Destination Address field shall contain the short address of the destination device.

The Channel Offset field shall contain the channel offset of the RX device of the allocated DSME GTSs. This field is not valid in the channel adaptation mode.

The DSME SAB Specification field is defined in 7.5.14.

The Allocation Order field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall indicate the DSME GTS allocation interval and be set to the value of AllocationOrder of the device requesting a DSME GTS. The relationship between the value of this field and the DSME GTS allocation interval is described in 7.5.13.



The BI Index field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall contain the index of the beacon interval *macBiIndex*, BI, in which the DSME GTS needs to be allocated. The BI Index is the sequence number of the BI in a multi-superframe beginning from zero.

The Superframe ID field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall contain the index of the superframe in which the DSME GTS is to be allocated. The Superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero. The superframe in which the PAN coordinator sends its beacons serves as the reference point (Superframe ID 0).

The Slot ID field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall contain the index of the DSME GTS to be allocated. The Slot ID is the sequence number of the DSME GTS in a superframe beginning from zero.

The Channel Index field shall be present if *macExtendedDsmeEnabled* is TRUE. This field shall contain the channel number of the DSME GTS to be allocated.

#### 7.5.16 DSME GTS Notify command

Only devices that have been assigned a short address shall send this command in the CAP.

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to indicate short addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN ID field shall contain the value of *macPanId*, and the Destination Address field shall be set to broadcast PAN ID.

The DSME GTS Notify command Content field shall be formatted as illustrated in Figure 7-118.

| Octets: 1           | 2                   | 0/2            | variable               |
|---------------------|---------------------|----------------|------------------------|
| DSME GTS Management | Destination Address | Channel Offset | DSME SAB Specification |

**Figure 7-118—DSME GTS Notify command Content field format**

The DSME GTS Management field shall be formatted as illustrated in Figure 7-117.

The Management Type field shall be set to one of the nonreserved values listed in Table 7-52.

The Direction field shall indicate the DSME GTSs are being allocated for TX or for RX of the requesting device. The value of this field is set to zero if the allocation is for TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not allocation.

The Prioritized Channel Access field shall be set to one if DSME GTS should be reserved as high priority, or set to zero if DSME GTS should be reserved as low priority. When the DSME GTS request command is used in the DSME GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME GTS.



The Status field indicates the result of a DSME GTS request. Valid values of the Status field are given in Table 7-53.

The DSME GTS Destination Address shall contain the short address of the Destination device.

The Channel Offset field shall contain the channel offset of the RX device of the allocated DSME GTSs. This field is not valid in the channel adaptation mode.

The DSME SAB Specification field is defined in 7.5.14.

#### 7.5.17 DSME Information Request command

The DSME Information Request command Content field shall be formatted as illustrated in Figure 7-119.

|           |                           |                          |
|-----------|---------------------------|--------------------------|
| Octets: 1 | 1                         | 2                        |
| Info type | DSME SAB sub-block length | DSME SAB sub-block index |

Figure 7-119—DSME Information Request command Content field format

The Info Type field indicates the type of DSME information that are requested by the device. Valid values of the Info Type field are given in Table 7-54.

Table 7-54—Info Type field values

| Info Type value | Information requested  |
|-----------------|------------------------|
| 0               | Timestamp              |
| 1               | DSME SAB Specification |
| 2               | DSME PAN descriptor    |
| 3–255           | Reserved               |

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode. This field is valid only if the value of the Info type field is set to request the DSME SAB specification.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB Sub-block in the entire SAB as illustrated in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode. This field is valid only if the value of the Info Type field is set to request the DSME SAB specification.

#### 7.5.18 DSME Information Response command

The DSME Information Response command Content field shall be formatted as illustrated in Figure 7-120.

|           |           |               |         |                        |                     |
|-----------|-----------|---------------|---------|------------------------|---------------------|
| Octets: 1 | 6         | 2             | 1       | variable               | variable            |
| Info Type | Timestamp | Superframe ID | Slot ID | DSME SAB Specification | DSME PAN Descriptor |

Figure 7-120—DSME Information Response command Content field format

The Info Type field indicates the type of DSME information that was requested. Valid value for the Info Type field are given in Table 7-55.

**Table 7-55—Info Type field**

| Info Type value | Fields present                    |
|-----------------|-----------------------------------|
| 0               | Timestamp, Superframe ID, Slot ID |
| 1               | DSME SAB Specification            |
| 2               | DSME PAN Descriptor               |
| 3–255           | Reserved                          |

The Timestamp field shall contain the time, in symbols, at which this command was transmitted. The time is measured from the first symbol of the most recent beacon transmitted by the coordinator.

The Superframe ID field shall contain the ID of the superframe in which this command will be transmitted. The superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero.

The Slot ID field shall contain the ID of the superframe slot in which this command will be transmitted. The slot ID is the sequence number of the DSME GTSs (not including beacon or CAP slots) in a superframe beginning from zero. This field is valid only if the value of the Info Type field is zero.

The DSME SAB Specification field shall be formatted as illustrated in Figure 7-115.

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 6-55.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB Sub-block in the entire SAB as illustrated in Figure 6-55.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as illustrated in Figure 6-55.

The DSME PAN Descriptor field shall be set as described in 7.4.2.5.

#### 7.5.19 DSME Beacon Allocation Notification command

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall both be set to indicate short addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to zero.

The Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The Destination Address field shall be set to 0xffff. The Source Address field shall contain the value of *macShortAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The DSME Beacon Allocation Notification command Content field shall be formatted as illustrated in Figure 7-121.

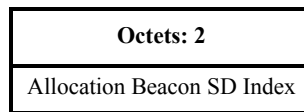


**Figure 7-121—DSME Beacon Allocation Notification command Content field format**

The Allocation Beacon SD Index field shall contain the allocating SD index number for Beacon frame that is allocated to the Source device.

#### 7.5.20 DSME Beacon Collision Notification command

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall both be set to indicate short addressing.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall be set to one.

The Destination PAN ID field shall contain the value of *macPanId*, while the Source PAN ID field shall be omitted. The Destination Address field shall be set to the node address that has requested later. The Source Address field shall contain the value of *macShortAddress*. The PAN ID Compression field shall be set as specified in 7.2.1.5.

The DSME Beacon Collision Notification command Content field shall be formatted as illustrated in Figure 7-122.

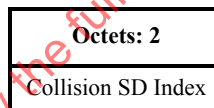


**Figure 7-122—DSME Beacon Collision Notification command Content field format**

The Collision SD Index field shall contain the SD index number of collision Beacon frame.

#### 7.5.21 DSME Link Report command

The Source Addressing Mode field of the Frame Control field shall be set to indicate extended addressing, and the Destination Addressing Mode field shall be set to the same mode as the destination device to which the command is sent.

The Frame Pending field of the Frame Control field shall be set to one, and the AR field shall be set to one.

The Destination PAN ID field shall contain the identifier of the PAN of the destination device. The Destination Address field shall contain the address of the destination device.

The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The DSME Link Report command Content field shall be formatted as illustrated in Figure 7-123.

|                  |
|------------------|
| Octets: variable |
| DSME Link Report |

**Figure 7-123—DSME Link Report command Content field format**

The DSME Link Report field shall be formatted as illustrated in Figure 7-124.

|                       |                   |     |                          |
|-----------------------|-------------------|-----|--------------------------|
| Octets: 1             | 4                 | ... | 4                        |
| Link Descriptor Count | Link Descriptor 1 | ... | Link Descriptor <i>n</i> |

**Figure 7-124—DSME Link Report field format**

The Link Descriptor Count field specifies the number of the Link Descriptors in the DSME Link Report field.

The Link Descriptor shall be formatted as illustrated in Figure 7-125.

|           |             |              |          |
|-----------|-------------|--------------|----------|
| Octets: 1 | 1           | 1            | 1        |
| Channel   | Average LQI | Average RSSI | Reserved |

**Figure 7-125—Link Descriptor field format**

The Channel field specifies the channel index reported by the source device.

The Average LQI field contains the average received LQI of the channel specified in Channel field within *macLinkStatisticPeriod* symbols.

The Average RSSI field contains the average received signal power by ED measurement during a period of *macLinkStatisticPeriod* symbols. The Average RSSI measurement shall be performed for each received packet, and the use of the Average RSSI result by the next higher layer is not specified in this standard.

#### 7.5.22 RIT Data Request command

The RIT Data Request command shall only be sent by a device supporting RIT mode. This command is optional and applicable for FFD only.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall also be set to zero. All other fields shall be set appropriately according to the intended use of the command.

The RIT Data Request command Content field shall be formatted as illustrated in Figure 7-126.

|                                |
|--------------------------------|
| Octets: variable               |
| Listen Information/RIT Payload |

**Figure 7-126—RIT Data Request command Content field format**

If the Content field is present and the first octet of the Content field is 0xff, then the Content field contains the RIT Payload field. Otherwise, if the Content field is present and the first octets is not 0xff, then Content field contains the List Information field.

The List Information field, when present, shall be formatted as illustrated in Figure 7-127.

| Octets: 1            | 1                       | 2                      |
|----------------------|-------------------------|------------------------|
| Time to First Listen | Number of Repeat Listen | Repeat Listen Interval |

**Figure 7-127—Format of the Listen Information field**

Time to First Listen field is defined in 7.4.2.4. The value 0xff shall not be used for the Time to First Listen field.

The Number of Repeat Listen field is defined in 7.4.2.4.

The Repeat Listen Interval field is defined in 7.4.2.4.

The RIT Payload field shall be included if and only if *macRitRequestPayload* is nonzero length. The RIT Payload field, when present, shall be formatted as illustrated in Figure 7-128.

| Octets: 1 | variable            |
|-----------|---------------------|
| 0xFF      | RIT Request Payload |

**Figure 7-128—Format of the RIT Payload field**

The RIT Request Payload field shall be set to the value of *macRitRequestPayload*.

### 7.5.23 DBS Request command

The Destination Addressing Mode field and Source Addressing Mode field shall be set to indicate short addressing.

The Frame Pending field shall be set to zero. The AR field shall be set to one. the Frame Version field shall be set to two.

The Destination PAN ID field shall contain the PAN ID of the SPC, and the Destination Address field shall contain the address of the SPC. The Source PAN ID field shall contain the value of *macPanId*. The network management entity should assure that each PAN coordinator has a unique PAN ID. The Source Address field shall contain the value of *macShortAddress*.

The DBS Request command Content field shall be formatted as shown in Figure 7-129.

| Octets: 4               |
|-------------------------|
| DBS Request Information |

**Figure 7-129—DBS Request command Content field format**

The DBS Request Information field shall be encoded as shown in Figure 7-130.

| Bits: 0:15              | 16:19      | 20:22    | 23                   | 24:31                    |
|-------------------------|------------|----------|----------------------|--------------------------|
| Requester Short Address | DBS Length | Reserved | Characteristics Type | Number of the Descendant |

**Figure 7-130—DBS Request Information field format**

The Requester Short Address field shall be set to the short address of the coordinator requesting a DBS.

The DBS Length field shall contain the number of *aBaseSlotDuration* being requested for a DBS.

The Characteristics Type field shall be set to one if the characteristics refer to a DBS allocation or zero if the characteristics refer to a DBS deallocation.

The Number of the Descendant field indicates the actual or expected number of TMCTP-child PAN coordinators. It may be set to zero if the PAN coordinator is not clear about how many descendants it will have.

#### 7.5.24 DBS Response command

The Destination Addressing Mode field and Source Addressing Mode field shall be set to indicate short addressing.

The Frame Pending field, the AR field, and the Frame Version field shall set to zero, one, and two, respectively.

The Destination PAN ID field shall contain the source PAN ID from the DBS request frame, and the Destination Address field shall contain the source address from the DBS request frame. The Source PAN ID field shall contain the value of *macPanId*, and the Source Address field shall contain the value of *macShortAddress*.

The DBS Response command Content field shall be formatted as shown in Figure 7-131.

| Octets: 10               |
|--------------------------|
| DBS Response Information |

**Figure 7-131—DBS Response command Content field format**

The DBS Response Information field shall be encoded as shown in Figure 7-132.

| Octets: 2               | 1                           | 1                    | 1                            | 3               | 1                       | 1                     |
|-------------------------|-----------------------------|----------------------|------------------------------|-----------------|-------------------------|-----------------------|
| Requester Short Address | Allocated DBS Starting Slot | Allocated DBS Length | Allocated PHY Channel Number | Start Band Edge | Starting PHY Channel ID | Ending PHY Channel ID |

**Figure 7-132—DBS Response Information field format**

The Requester Short Address field shall be set to the short address of the coordinator requesting a DBS.

The Allocated DBS Starting Slot field shall contain the time of first slot of the allocated DBS in the BOP, in units of *aBaseSlotDuration*.

The Allocated DBS Length field shall contain the length of the allocated DBS. If the Allocated DBS Length field is equal to zero, it indicates that the DBS slot and the dedicated channel are deallocated.

The Allocated PHY Channel Number field shall contain the channel number that the coordinator intends to use for all future communications.

The Start Band Edge field is the frequency in kilohertz indicating the lower edge of band that the coordinator will use for all future communications.

The Starting PHY Channel ID field shall contain the lowest channel number, which is assigned by the TMCTP-parent PAN coordinator, including the SPC.

The Ending PHY Channel ID field shall contain the highest channel number, which is assigned by the TMCTP-parent PAN coordinator, including the SPC.

#### 7.5.25 RIT Data Response command

The RIT Data Response command shall only be sent by a device supporting RIT mode.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the AR field shall also be set to zero. All other fields shall be set appropriately according to the intended use of the command.

The RIT Data Response command Content field shall be formatted as illustrated in Figure 7-133.

|                      |
|----------------------|
| Octets: variable     |
| RIT Response Payload |

**Figure 7-133—RIT Data Response command Content field format**

The RIT Response Payload field contains RitResponsePayload received in an MLME-RIT-RES.request primitive. The field is passed as received from the higher layer, no octet ordering changes shall be made.

#### 7.5.26 Vendor Specific command

The Vendor Specific command is reserved for the use of other protocols and/or data relevant only to certain implementations. The Vendor Specific command Content field shall be formatted as illustrated in Figure 7-134.

|            |                             |
|------------|-----------------------------|
| Octets: 3  | Variable                    |
| Vendor OUI | Vendor Specific Information |

**Figure 7-134—Vendor Specific command Content field format**

The Vendor OUI field is defined in 7.4.2.2. A value of the Vendor OUI field not understood by a receiving device causes the command to be ignored.

The Vendor Specific Information field is defined by the vendor identified in the Vendor OUI field. Its use is outside of the scope of this standard.

## 8. MAC services

### 8.1 Overview

The MAC sublayer provides an interface between the next higher layer and the PHY. The MAC sublayer conceptually includes a management entity called the MLME. This entity provides the service interfaces through which layer management 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 8-1 depicts the components and interfaces of the MAC sublayer.

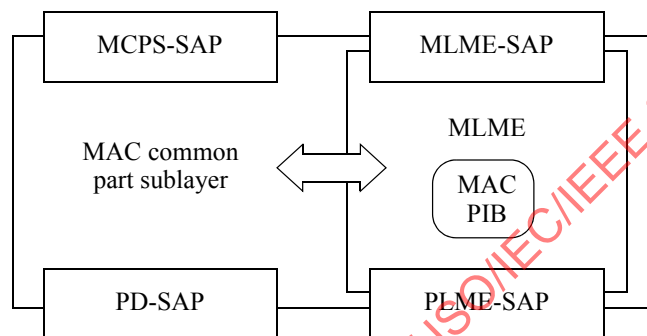


**Figure 8-1—MAC sublayer reference model**

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)
- The MAC management service, accessed through the MLME-SAP

These two services provide the interface between the next higher layer and the PHY. 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.

### 8.2 MAC management service

#### 8.2.1 Primitives supported by the MLME-SAP interface

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 8-1 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). Primitives marked with a circle (●) are optional for both RFD-RX and RFD-TX device types.



**Table 8-1—Summary of the primitives accessed through the MLME-SAP**

| Name                | Request    | Indication | Response  | Confirm    |
|---------------------|------------|------------|-----------|------------|
| MLME-ASSOCIATE      | 8.2.3.1●   | 8.2.3.2◆●  | 8.2.3.3◆● | 8.2.3.4●   |
| MLME-BEACON-NOTIFY  |            | 8.2.5.1●   |           |            |
| MLME-BEACON         | 8.2.18.1   |            |           | 8.2.18.2   |
| MLME-CALIBRATE      | 8.2.17.1*● |            |           | 8.2.17.2*● |
| MLME-COMM-STATUS    |            | 8.2.5.2    |           |            |
| MLME-DBS            | 8.2.23.1*  | 8.2.23.2*  | 8.2.23.3* | 8.2.23.4*  |
| MLME-DA             | 8.2.24.1*  | 8.2.24.2*  |           | 8.2.24.3*  |
| MLME-DISASSOCIATE   | 8.2.4.1●   | 8.2.4.2●   |           | 8.2.4.3●   |
| MLME-DPS            | 8.2.15.1*● | 8.2.15.3*● |           | 8.2.15.2*● |
| MLME-GET            | 8.2.6.1    |            |           | 8.2.6.2    |
| MLME-GTS            | 8.2.7.1*●  | 8.2.7.3*●  |           | 8.2.7.2*●  |
| MLME-IE-NOTIFY      |            | 8.2.5.3●   |           |            |
| MLME-ORPHAN         |            | 8.2.8.1◆●  | 8.2.8.2◆● |            |
| MLME-PHY-OP-SWITCH* | 8.2.20     | 8.2.21     |           | 8.2.22.3   |
| MLME-POLL           | 8.2.14.1●  |            |           | 8.2.14.2●  |
| MLME-RESET          | 8.2.9.1    |            |           | 8.2.9.2    |
| MLME-RIT-REQ        |            | 8.2.25.1*  |           |            |
| MLME-RIT-RES        | 8.2.25.2*  | 8.3*       |           | 8.2.25.4*  |
| MLME-RX-ENABLE      | 8.2.10.1*  |            |           | 8.2.10.2*  |
| MLME-SCAN           | 8.2.11.1●  |            |           | 8.2.11.2●  |
| MLME-SET            | 8.2.6.3    |            |           | 8.2.6.4    |
| MLME-START          | 8.2.12.1◆● |            |           | 8.2.12.2◆● |
| MLME-SYNC           | 8.2.13.1*● |            |           |            |
| MLME-SYNC-LOSS      |            | 8.2.13.2●  |           |            |
| MLME-SOUNDING       | 8.2.16.1*● |            |           | 8.2.16.1*● |

When the optional TSCH mode is implemented (i.e., *macTschEnabled* = TRUE), the primitives listed in Table 8-2 shall be implemented.

**Table 8-2—TSCH primitives**

| Name               | Request  | Indication | Response | Confirm  |
|--------------------|----------|------------|----------|----------|
| MLME-SET-SLOTFRAME | 8.2.19.1 | —          | —        | 8.2.19.2 |
| MLME-SET-LINK      | 8.2.19.3 | —          | —        | 8.2.19.4 |
| MLME-TSCH-MODE     | 8.2.19.5 | —          | —        | 8.2.19.6 |
| MLME-KEEP-ALIVE    | 8.2.19.7 | —          | —        | 8.2.19.8 |

In a DSME-enabled PAN, the primitives listed in in Table 8-3 shall be implemented.

**Table 8-3—DSME MLME-SAP primitives**

| Name                  | Request  | Indication | Response | Confirm  |
|-----------------------|----------|------------|----------|----------|
| MLME-DSME-GTS         | 8.2.20.1 | 8.2.20.2   | 8.2.20.3 | 8.2.20.4 |
| MLME-DSME-LINK-REPORT | 8.2.21.1 | 8.2.21.2   | —        | 8.2.21.3 |

### 8.2.2 Common requirements for MLME primitives

If any error occurs during the outgoing frame security procedure, as described in 9.2.1, for a request primitive, the MLME will discard the frame and issue the corresponding confirm primitive with the error returned by the outgoing frame security procedure.

If any error occurs during the outgoing frame security procedure, as described in 9.2.1, for a response primitive, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with the error returned by the outgoing frame security procedure.

If any parameter in request primitive is not supported or is out of range, the MAC sublayer will issue the corresponding confirm primitive with a Status of INVALID\_PARAMETER.

If any parameter in 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.

If the MLME is unable to send the frame required by a request primitive due to a CSMA-CA algorithm failure, the MLME will issue the corresponding confirm primitive with a Status parameter value of CHANNEL\_ACCESS\_FAILURE.

If the MLME successfully transmits the frame required by a request primitive, but the expected acknowledgment is not received, the MLME will issue the corresponding confirm primitive with a Status parameter value of NO\_ACK.

**8.2.3 Association primitives**

These primitives are used when a device becomes associated with a PAN.

**8.2.3.1 MLME-ASSOCIATE.request**

The MLME-ASSOCIATE.request primitive is used by a device to request an association with a coordinator.

The semantics of this primitive are as follows:

```
MLME-ASSOCIATE.request (
    ChannelNumber,
    ChannelPage,
    CoordAddrMode,
    CoordPanId,
    CoordAddress,
    CapabilityInformation,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex,
    ChannelOffset,
    HoppingSequenceId,
    DsmeAssociation,
    Direction,
    AllocationOrder,
    HoppingSequenceRequest
)
```

The primitive parameters are defined in Table 8-4.

**Table 8-4—MLME-ASSOCIATE.request parameters**

| Name                  | Type                              | Valid range                                 | Description                                                             |
|-----------------------|-----------------------------------|---------------------------------------------|-------------------------------------------------------------------------|
| ChannelNumber         | Integer                           | Any valid channel number                    | The channel number on which to attempt association.                     |
| ChannelPage           | Integer                           | Any valid channel page                      | The channel page on which to attempt association.                       |
| CoordAddrMode         | Enumeration                       | SHORT, EXTENDED                             | The coordinator addressing mode for this primitive and subsequent MPDU. |
| CoordPanId            | Integer                           | 0x0000–0xffff                               | The identifier of the PAN with which to associate.                      |
| CoordAddress          | Short address or extended address | As specified by the CoordAddrMode parameter | The address of the coordinator with which to associate.                 |
| CapabilityInformation | Bitmap                            | As defined in 7.5.2                         | Specifies the operational capabilities of the associating device.       |
| SecurityLevel         | Integer                           | As defined in Table 8-75                    | As defined in Table 8-75.                                               |

**Table 8-4—MLME-ASSOCIATE.request parameters (continued)**

| Name                   | Type          | Valid range              | Description                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------|---------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KeyIdMode              | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                              |
| KeySource              | Set of octets | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                              |
| KeyIndex               | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                              |
| ChannelOffset          | Integer       | 0x00–0xffff              | Specifies the offset value of Hopping Sequence.                                                                                                                                                                                                                                                                                                                                        |
| HoppingSequenceId      | Integer       | 0x00–0x0f                | Indicate the ID of channel hopping sequence in use:<br>0x00: a default hopping sequence<br>0x01: a hopping sequence generated by PAN coordinator<br>0x02–0x0f: a hopping sequence set by the next higher layer<br>If a coordinator receives an Association Request command with HoppingSequenceId of 1, it replies with a channel hopping sequence in an association response command. |
| DsmeAssociation        | Boolean       | TRUE, FALSE              | Set to TRUE if the device is requesting GTS allocation during association, FALSE otherwise.                                                                                                                                                                                                                                                                                            |
| Direction              | Integer       | 0x00–0x01                | As defined in Table 8-52.                                                                                                                                                                                                                                                                                                                                                              |
| AllocationOrder        | Integer       | 0x00–0x08                | As defined in 7.5.13.                                                                                                                                                                                                                                                                                                                                                                  |
| HoppingSequenceRequest | Boolean       | TRUE, FALSE              | Indicates whether a hopping sequence is requested. A value of FALSE indicates that a hopping sequence is not requested. A value of TRUE indicates that a hopping sequence is requested.                                                                                                                                                                                                |

On receipt of the MLME-ASSOCIATE.request primitive, the MLME of an unassociated device first updates the appropriate PHY and MAC PIB attributes, as described in 6.4.1, and then generates an Association Request command, as defined in 7.5.2.

The SecurityLevel parameter specifies the level of security to be applied to the Association Request command. 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.

### 8.2.3.2 MLME-ASSOCIATE.indication

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

The semantics of this primitive are as follows:

MLME-ASSOCIATE.indication (

- DeviceAddress,
- CapabilityInformation,
- SecurityLevel,
- KeyIdMode,
- KeySource,
- KeyIndex
- ChannelOffset,
- HoppingSequenceId,
- DsmeAssociation,
- Direction,
- AllocationOrder,
- HoppingSequenceRequest

)

The primitive parameters are defined in Table 8-5.

**Table 8-5—MLME-ASSOCIATE.indication parameters**

| Name                  | Type             | Valid range              | Description                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddress         | Extended address | An extended address      | The address of the device requesting association.                                                                                                                                                                                                                                                                                                                                            |
| CapabilityInformation | Bitmap           | As defined in 7.5.2      | The operational capabilities of the device requesting association.                                                                                                                                                                                                                                                                                                                           |
| SecurityLevel         | Integer          | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                                                                                                                                                                                                                                                                                    |
| KeyIdMode             | Integer          | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                                                                                                                                                                                                                                                                                    |
| KeySource             | Set of octets    | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                                                                                                                                                                                                                                                                                    |
| KeyIndex              | Integer          | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                                                                                                                                                                                                                                                                                    |
| ChannelOffset         | Integer          | 0x0000–0xffff            | Specifies the offset value of Hopping Sequence.                                                                                                                                                                                                                                                                                                                                              |
| HoppingSequenceId     | Integer          | 0x00–0x0f                | Indicates the ID of channel hopping sequence in use:<br>0x00: the default hopping sequence<br>0x01: a hopping sequence generated by PAN coordinator<br>0x02–0x0f: a hopping sequence set by the next higher layer<br>If a coordinator receives an Association Request command with HoppingSequenceId of 0x01, it replies with a channel hopping sequence in an association response command. |

**Table 8-5—MLME-ASSOCIATE.indication parameters (continued)**

| Name                   | Type    | Valid range | Description                                                                                                                                                                             |
|------------------------|---------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DsmeAssociation        | Boolean | TRUE, FALSE | Set to TRUE if the device is requesting GTS allocation during association, FALSE otherwise.                                                                                             |
| Direction              | Integer | 0x00–0x01   | As defined in Table 8-52.                                                                                                                                                               |
| AllocationOrder        | Integer | 0x00–0x08   | As defined in 7.5.13.                                                                                                                                                                   |
| HoppingSequenceRequest | Boolean | TRUE, FALSE | Indicates whether a hopping sequence is requested. A value of FALSE indicates that a hopping sequence is not requested. A value of TRUE indicates that a hopping sequence is requested. |

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.

### 8.2.3.3 MLME-ASSOCIATE.response

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

The semantics of this primitive are as follows:

```

MLME-ASSOCIATE.response (
    DeviceAddress,
    AssocShortAddress,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex,
    ChannelOffset,
    HoppingSequence,
    DsmeAssociation,
    AllocationOrder,
    BIndex,
    SuperframeId,
    SlotId,
    ChannelIndex,
    Status
)

```

The primitive parameters are defined in Table 8-6.

When the MLME of a coordinator receives the MLME-ASSOCIATE.response primitive, it generates an Association Response command, as described in 7.5.3, and attempts to send it to the device requesting association, as described in 6.4.1.

If the Status field of MLME-ASSOCIATE.response primitive is set to Fast association successful, the association response command shall be sent to the device requesting fast association directly.

**Table 8-6—MLME-ASSOCIATE.response parameters**

| Name              | Type             | Valid range                | Description                                                                                                                                    |
|-------------------|------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddress     | Extended address | Any valid extended address | The address of the device requesting association.                                                                                              |
| AssocShortAddress | Integer          | 0x0000–0xffff              | The short address allocated by the coordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful. |
| SecurityLevel     | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                      |
| KeyIdMode         | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                      |
| KeySource         | Set of octets    | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                      |
| KeyIndex          | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                      |
| ChannelOffset     | Integer          | 0x0000–0xffff              | Specifies the offset value of Hopping Sequence.                                                                                                |
| Hopping Sequence  | Set of octets    | —                          | For TSCH, specifies the sequence of channel numbers that is set by a higher layer as described in 6.2.10. For DSME, as defined in 7.5.13.      |
| DsmeAssociation   | Boolean          | TRUE, FALSE                | Set to TRUE if the device is requesting GTS allocation during association, FALSE otherwise.                                                    |
| AllocationOrder   | Integer          | 0x00–0x08                  | As defined in 7.5.13.                                                                                                                          |
| BiIndex           | Integer          | 0x00–0xff                  | As defined in 7.5.13.                                                                                                                          |
| SuperframeId      | Integer          | 0x0000–0xffff              | As defined in 7.5.13.                                                                                                                          |
| SlotId            | Integer          | 0x00–0x0e                  | As defined in 7.5.13.                                                                                                                          |
| ChannelIndex      | Integer          | 0x00–0x1f                  | As defined in 7.5.13.                                                                                                                          |
| Status            | Enumeration      | As defined in 7.5.3        | The status of the association attempt.                                                                                                         |

**8.2.3.4 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.

The semantics of this primitive are as follows:

MLME-ASSOCIATE.confirm  
(  
  AssocShortAddress,  
  SecurityLevel,  
  KeyIdMode,  
  KeySource,  
  KeyIndex  
  ChannelOffset,  
  HoppingSequence,  
  DsmeAssociation,

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

AllocationOrder,  
BilIndex,  
SuperframeId,  
SlotId,  
ChannelIndex,  
Status  
)

The primitive parameters are defined in Table 8-7.

**Table 8-7—MLME-ASSOCIATE.confirm parameters**

| Name              | Type          | Valid range                            | Description                                                                                                                                                                                                                                                             |
|-------------------|---------------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AssocShortAddress | Integer       | 0x0000–0xffff                          | The short address allocated by the coordinator on successful association. This parameter will be equal to 0xffff if the association attempt was unsuccessful.                                                                                                           |
| SecurityLevel     | Integer       | As defined in Table 8-75 or Table 8-77 | If the primitive were generated following failed outgoing processing of an Association Request command, then it is as defined in Table 8-75. If the primitive were generated following receipt of an association response command, then it is as defined in Table 8-77. |
| KeyIdMode         | Integer       | As defined in Table 8-75 or Table 8-77 | If the primitive were generated following failed outgoing processing of an Association Request command, then it is as defined in Table 8-75. If the primitive were generated following receipt of an association response command, then it is as defined in Table 8-77. |
| KeySource         | Set of octets | As defined in Table 8-75 or Table 8-77 | If the primitive were generated following failed outgoing processing of an Association Request command, then it is as defined in Table 8-75. If the primitive were generated following receipt of an association response command, then it is as defined in Table 8-77. |
| KeyIndex          | Integer       | As defined in Table 8-75 or Table 8-77 | If the primitive were generated following failed outgoing processing of an Association Request command, then it is as defined in Table 8-75. If the primitive were generated following receipt of an association response command, then it is as defined in Table 8-77. |
| ChannelOffset     | Integer       | 0x0000–0xffff                          | Specifies the offset value of Hopping Sequence.                                                                                                                                                                                                                         |
| Hopping Sequence  | Set of octets | —                                      | As defined in Table 8-6.                                                                                                                                                                                                                                                |



**Table 8-7—MLME-ASSOCIATE.confirm parameters (continued)**

| Name            | Type        | Valid range                                                                                                                                                                                                                                                                           | Description                                                                                 |
|-----------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| DsmeAssociation | Boolean     | TRUE, FALSE                                                                                                                                                                                                                                                                           | Set to TRUE if the device is requesting GTS allocation during association, FALSE otherwise. |
| AllocationOrder | Integer     | 0x00–0x08                                                                                                                                                                                                                                                                             | As defined in 7.5.13.                                                                       |
| BiIndex         | Integer     | 0x00–0xff                                                                                                                                                                                                                                                                             | As defined in 7.5.13.                                                                       |
| SuperframeId    | Integer     | 0x0000–0xffff                                                                                                                                                                                                                                                                         | As defined in 7.5.13.                                                                       |
| SlotId          | Integer     | 0x00–0x0e                                                                                                                                                                                                                                                                             | As defined in 7.5.13.                                                                       |
| ChannelIndex    | Integer     | 0x00–0x1f                                                                                                                                                                                                                                                                             | As defined in 7.5.13.                                                                       |
| Status          | Enumeration | SUCCESS,<br>CHANNEL_ACCESS_FAILURE,<br>NO_ACK,<br>NO_DATA,<br>COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>IMPROPER_KEY_TYPE,<br>IMPROPER_SECURITY_LEVEL,<br>SECURITY_ERROR,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_LEGACY,<br>UNSUPPORTED_SECURITY,<br>UNSUPPORTED_FEATURE,<br>INVALID_PARAMETER | The status of the association attempt.                                                      |

If the association request was successful, then the Status parameter will be set to SUCCESS. Otherwise, the Status parameter will be set to indicate the type of failure.

#### 8.2.4 Disassociation primitives

These primitives are used by a device to disassociate from a PAN or by the coordinator to disassociate a device from a PAN.

##### 8.2.4.1 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.

The semantics of this primitive are as follows:

```
MLME-DISASSOCIATE.request (
    DeviceAddrMode,
    DevicePanId,
    DeviceAddress,
    DisassociateReason,
    TxIndirect,
    SecurityLevel,
    KeyIdMode,
```

KeySource,  
KeyIndex  
)

The primitive parameters are defined in Table 8-8.

**Table 8-8—MLME-DISASSOCIATE.request parameters**

| Name               | Type                              | Valid range                                  | Description                                                                                 |
|--------------------|-----------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------|
| DeviceAddrMode     | Enumeration                       | SHORT, EXTENDED                              | The addressing mode of the device to which to send the Disassociation Notification command. |
| DevicePanId        | Integer                           | 0x0000–0xffff                                | The PAN ID of the device to which to send the Disassociation Notification command.          |
| DeviceAddress      | Short address or extended 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, as described in 7.5.4.                                   |
| TxIndirect         | Boolean                           | TRUE, FALSE                                  | TRUE if the Disassociation Notification command is to be sent indirectly.                   |
| SecurityLevel      | Integer                           | As defined in Table 8-75                     | As defined in Table 8-75.                                                                   |
| KeyIdMode          | Integer                           | As defined in Table 8-75                     | As defined in Table 8-75.                                                                   |
| KeySource          | Set of octets                     | As defined in Table 8-75                     | As defined in Table 8-75.                                                                   |
| KeyIndex           | Integer                           | As defined in Table 8-75                     | As defined in Table 8-75.                                                                   |

If the DeviceAddrMode parameter is equal to SHORT and the DeviceAddress parameter is equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to EXTENDED and the DeviceAddress parameter is equal to *macCoordExtendedAddress*, the TxIndirect parameter is ignored, and the MLME sends a Disassociation Notification command, as defined in 7.5.4, to its coordinator in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN.

If the DeviceAddrMode parameter is equal to SHORT and the DeviceAddress parameter is not equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to EXTENDED 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, as described in 6.6.

If the DeviceAddrMode parameter is equal to SHORT and the DeviceAddress parameter is not equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to EXTENDED 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.

**8.2.4.2 MLME-DISASSOCIATE.indication**

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

The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 8-9.

**Table 8-9—MLME-DISASSOCIATE.indication parameters**

| Name               | Type             | Valid range                | Description                                             |
|--------------------|------------------|----------------------------|---------------------------------------------------------|
| DeviceAddress      | Extended address | Any valid extended address | The address of the device requesting disassociation.    |
| DisassociateReason | Integer          | 0x00–0xff                  | The reason for the disassociation, as defined in 7.5.4. |
| SecurityLevel      | Integer          | As defined in Table 8-77   | As defined in Table 8-77.                               |
| KeyIdMode          | Integer          | As defined in Table 8-77   | As defined in Table 8-77.                               |
| KeySource          | Set of octets    | As defined in Table 8-77   | As defined in Table 8-77.                               |
| KeyIndex           | Integer          | As defined in Table 8-77   | As defined in Table 8-77.                               |

**8.2.4.3 MLME-DISASSOCIATE.confirm**

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

The semantics of this primitive are as follows:

```
MLME-DISASSOCIATE.confirm (
    DeviceAddrMode,
    DevicePanId,
    DeviceAddress,
    Status
)
```

The primitive parameters are defined in Table 8-10.

**Table 8-10—MLME-DISASSOCIATE.confirm parameters**

| Name           | Type                              | Valid range                                                                                                                                                                 | Description                                                                                                                       |
|----------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddrMode | Enumeration                       | SHORT, EXTENDED                                                                                                                                                             | 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 ID of the device that has either requested disassociation or been instructed to disassociate by its coordinator.          |
| DeviceAddress  | Short address or extended 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.         |
| Status         | Enumeration                       | SUCCESS, NO_ACK, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER | The status of the disassociation attempt.                                                                                         |

This primitive returns a Status of either SUCCESS, indicating that the Disassociation Notification was successful, or the appropriate Status parameter value indicating the reason for failure.

If the DevicePanId parameter is not equal to *macPanId* in the MLME-DISASSOCIATE.request primitive, the Status parameter shall be set to INVALID\_PARAMETER.

### 8.2.5 Communications notification primitives

The MLME-SAP beacon primitives define how a coordinator in a nonbeacon-enabled PAN may send a beacon or an enhanced beacon, or respond to beacon/enhanced beacon requests. Coordinators operating in a beaconing PAN use MLME-START to configure beacons and enhanced beacons. Only FFDs capable of acting as a coordinator are required to provide the MLME-BEACON SAPs.

The MLME-BEACON-NOTIFY.indication primitive is used to notify the next higher layer when a beacon or enhanced beacon is received during normal operating conditions. The MLME-COMM-STATUS.indication primitive is used to notify the next higher layer that an error has occurred during the processing of a frame that was instigated by a response primitive.

#### 8.2.5.1 MLME-BEACON-NOTIFY.indication

The MLME-BEACON-NOTIFY.indication primitive is used to send parameters contained within a beacon frame or an enhanced beacon frame received by the MAC sublayer to the next higher layer when either *macAutoRequest* is set to FALSE or when the beacon frame contains one or more octets of payload. The primitive also sends a measure of the LQI and the time the beacon frame was received. When an enhanced beacon is received, the SDU contains a list of IEs, and Superframe Specification, GTS fields, PendingADD, and beacon Payload parameters are not present.

The semantics of this primitive are as follows:

MLME-BEACON-NOTIFY.indication (

- Bsn,
- PanDescriptor,
- PendAddrSpec,
- AddrList,
- Ebsn,
- BeaconType,
- HeaderIeList,
- PayloadIeList,
- BeaconPayload

)

The primitive parameters are defined in Table 8-11.

**Table 8-11—MLME-BEACON-NOTIFY.indication parameters**

| Name          | Type                                     | Valid range                | Description                                                                                                                      |
|---------------|------------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Bsn           | Integer                                  | 0x00–0xff                  | The BSN of the Beacon frame.                                                                                                     |
| PanDescriptor | PANDescriptor value                      | As defined in Table 8-12   | The PANDescriptor for the received Beacon frame.                                                                                 |
| PendAddrSpec  | Bitmap                                   | As defined in 7.3.1        | The beacon pending address specification.                                                                                        |
| AddrList      | Set of extended addresses                | —                          | The addresses of the devices for which the beacon source has data.                                                               |
| Ebsn          | Integer                                  | 0x00–0xff                  | BSN used for Enhanced Beacon frames.                                                                                             |
| BeaconType    | Enumeration                              | BEACON,<br>ENHANCED_BEACON | Indicates a beacon or enhanced beacon was received.                                                                              |
| HeaderIeList  | Set of header IEs as described in 7.4.2  | —                          | The header IEs, excluding Termination IEs, that were included in the Beacon frame. If empty, then no header IEs were included.   |
| PayloadIeList | Set of payload IEs as described in 7.4.3 | —                          | The payload IEs, excluding Termination IEs, that were included in the Beacon frame. If empty, then no payload IEs were included. |
| BeaconPayload | Set of octets                            | —                          | The set of octets comprising the Beacon Payload field, if present.                                                               |

The elements of the PANDescriptor type are defined in Table 8-12.

**Table 8-12—Elements of PANDescriptor**

| Name           | Type                              | Valid range                                                                                                                                   | Description                                                                                                                                                                                                                                                                       |
|----------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CoordAddrMode  | Enumeration                       | SHORT, EXTENDED                                                                                                                               | The coordinator addressing mode corresponding to the received beacon frame.                                                                                                                                                                                                       |
| CoordPanId     | Integer                           | 0x0000–0xffff                                                                                                                                 | The PAN ID of the coordinator as specified in the received beacon frame.                                                                                                                                                                                                          |
| CoordAddress   | Short address or extended address | As specified by the CoordAddrMode parameter                                                                                                   | The address of the coordinator as specified in the received beacon frame.                                                                                                                                                                                                         |
| ChannelNumber  | Integer                           | Any valid channel number                                                                                                                      | The current channel number occupied by the network.                                                                                                                                                                                                                               |
| ChannelPage    | Integer                           | Any valid channel page                                                                                                                        | The current channel page occupied by the network.                                                                                                                                                                                                                                 |
| SuperframeSpec | Bitmap                            | As defined in 7.3.1                                                                                                                           | The superframe specification as specified in the received beacon frame.                                                                                                                                                                                                           |
| GtsPermit      | Boolean                           | TRUE, 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, as defined in 10.2.6.                                                                                                                                                                         |
| TimeStamp      | Integer                           | 0x000000–0xffffffff                                                                                                                           | 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 6.5.2. The precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant. |
| SecurityStatus | Enumeration                       | 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, as described in 9.2.3.                                                                                             |
| SecurityLevel  | Integer                           | As defined in Table 8-77                                                                                                                      | As defined in Table 8-77.                                                                                                                                                                                                                                                         |
| KeyIdMode      | Integer                           | As defined in Table 8-77                                                                                                                      | As defined in Table 8-77.                                                                                                                                                                                                                                                         |
| KeySource      | Set of octets                     | As defined in Table 8-77                                                                                                                      | As defined in Table 8-77.                                                                                                                                                                                                                                                         |
| KeyIndex       | Integer                           | As defined in Table 8-77                                                                                                                      | As defined in Table 8-77.                                                                                                                                                                                                                                                         |

**Table 8-12—Elements of PANDescriptor (continued)**

| Name                             | Type                  | Valid range           | Description                                                                                                                                                                                                                                                  |
|----------------------------------|-----------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CodeList                         | Set of integers       | —                     | The HRP UWB preamble codes, as described in 16.2.5.1, or CSS subchirp codes, as described in 15.3, in use when the channel was detected. For all other PHY types, this is empty.                                                                             |
| DsmeSuperframeSpecification      | Bitmap                | Defined in 7.4.2.5    | The DSME superframe specification as specified in the received enhanced frame. This element is present when <i>macDsmeEnabled</i> is TRUE.                                                                                                                   |
| TimeSynchronizationSpecification | Bitmap                | Defined in 7.4.2.5    | The time synchronization specification as specified in the received enhanced beacon frame. This element is present when <i>macDsmeEnabled</i> is TRUE.                                                                                                       |
| BeaconBitmap                     | Bitmap                | Defined in 7.4.2.5    | The beacon bitmap as specified in the received enhanced beacon frame. This element is present when <i>macDsmeEnabled</i> is TRUE.                                                                                                                            |
| ChannelHoppingSpecification      | Bitmap                | Defined in 7.4.2.5    | The channel hopping specification specified in the received enhanced beacon frame. This element is present when <i>macDsmeEnabled</i> is TRUE and the value of Channel Diversity Mode field in the received enhanced beacon frame indicates channel hopping. |
| TrleDescriptor                   | TRLE Descriptor value | As defined in F.5.1.1 | The TRLE Descriptor for the received beacon.                                                                                                                                                                                                                 |

**8.2.5.2 MLME-COMM-STATUS.indication**

The MLME-COMM-STATUS indication primitive allows the MLME to indicate a communications status.

The semantics of this primitive are as follows:

```

MLME-COMM-STATUS.indication (
    PanId,
    SrcAddrMode,
    SrcAddr,
    DstAddrMode,
    DstAddr,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex,
    Status
)

```

The primitive parameters are defined in Table 8-13.

**Table 8-13—MLME-COMM-STATUS.indication parameters**

| Name          | Type          | Valid range                               | Description                                                                                                                                                                                                                                                                           |
|---------------|---------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PanId         | Integer       | 0x0000–0xffff                             | The PAN ID of the device from which the frame was received or to which the frame was being sent.                                                                                                                                                                                      |
| SrcAddrMode   | Enumeration   | NONE, SHORT, EXTENDED                     | The source addressing mode for this primitive.                                                                                                                                                                                                                                        |
| SrcAddr       | —             | As specified by the SrcAddrMode parameter | The address of the entity from which the frame causing the error originated.                                                                                                                                                                                                          |
| DstAddrMode   | Enumeration   | NONE, SHORT, EXTENDED                     | The destination addressing mode for this primitive.                                                                                                                                                                                                                                   |
| DstAddr       | —             | As specified by the DstAddrMode parameter | The address of the device for which the frame was intended.                                                                                                                                                                                                                           |
| SecurityLevel | Integer       | As defined in Table 8-75 or Table 8-77    | If the primitive were generated following a transmission instigated through a response primitive, then it is as defined in Table 8-75. If the primitive were generated on receipt of a frame that generates an error in its security processing, then it is as defined in Table 8-77. |
| KeyIdMode     | Integer       | As defined in Table 8-75 or Table 8-77    | If the primitive were generated following a transmission instigated through a response primitive, then it is as defined in Table 8-75. If the primitive were generated on receipt of a frame that generates an error in its security processing, then it is as defined in Table 8-77. |
| KeySource     | Set of octets | As defined in Table 8-75 or Table 8-77    | If the primitive were generated following a transmission instigated through a response primitive, then it is as defined in Table 8-75. If the primitive were generated on receipt of a frame that generates an error in its security processing, then it is as defined in Table 8-77. |
| KeyIndex      | Integer       | As defined in Table 8-75 or Table 8-77    | If the primitive were generated following a transmission instigated through a response primitive, then it is as defined in Table 8-75. If the primitive were generated on receipt of a frame that generates an error in its security processing, then it is as defined in Table 8-77. |



**Table 8-13—MLME-COMM-STATUS.indication parameters (continued)**

| Name   | Type        | Valid range                                                                                                                                                                                                                                                                                     | Description                |
|--------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| Status | Enumeration | SUCCESS,<br>TRANSACTION_OVERFLOW,<br>TRANSACTION_EXPIRED,<br>CHANNEL_ACCESS_FAILURE,<br>NO_ACK, COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>IMPROPER_KEY_TYPE,<br>IMPROPER_SECURITY_LEVEL,<br>SECURITY_ERROR,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_LEGACY,<br>UNSUPPORTED_SECURITY,<br>INVALID_PARAMETER | The communications status. |

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, as described in 9.2.3. If the request to transmit was successful, the Status parameter shall be set to SUCCESS. Otherwise, the Status parameter shall be set to indicate the error with one of the following values:

- TRANSACTION\_OVERFLOW – The MLME does not have the capacity to store the frame that was to be sent using indirect transmission.
- TRANSACTION\_EXPIRED – The transaction was not handled within *macTransactionPersistenceTime*.
- CHANNEL\_ACCESS\_FAILURE – There was a failure in the CSMA-CA algorithm while attempting to send the frame.
- NO\_ACK – An acknowledgment was expected but not received.
- INVALID\_PARAMETER – One or more of the parameters in the response primitive were in error.
- A security error, as defined in 9.2.

#### 8.2.5.3 MLME-IE-NOTIFY.indication

The MLME-IE-NOTIFY.indication primitive is used to send IEs contained in a MAC Command frame or an Enh-Ack frame.

The semantics of this primitive are as follows:

MLME-IE-NOTIFY.indication (

- SrcAddrMode,
- SrcPanId,
- SrcAddr,
- DstAddrMode,
- DstPanId,
- DstAddr,
- FrameType,
- CommandIdentifier,
- HeaderIeList,
- PayloadIeList,
- SecurityLevel,

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

KeyIdMode,  
KeySource,  
KeyIndex  
)

The primitive parameters are defined in Table 8-14.

**Table 8-14—MLME-IE-NOTIFY.indication parameters**

| Name              | Type                                     | Valid range                               | Description                                                                                                                           |
|-------------------|------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode       | Enumeration                              | NONE, SHORT, EXTENDED                     | The source addressing mode of the source of the frame that was received.                                                              |
| SrcPanId          | Integer                                  | 0x0000–0xffff                             | The PAN ID of the source of the frame that was received. Valid only when a source PAN ID is included in the received frame.           |
| SrcAddr           | —                                        | As specified by the SrcAddrMode parameter | The source address of the frame that was received. Valid only when the source address is included in the received frame.              |
| DstAddrMode       | Enumeration                              | NONE, SHORT, EXTENDED                     | The destination addressing mode of the source of the frame that was received.                                                         |
| DstPanId          | Integer                                  | 0x0000–0xffff                             | The PAN ID of the destination of the frame that was received. Valid only when a destination PAN ID is included in the received frame. |
| DstAddr           | —                                        | As specified by the DstAddrMode parameter | The destination address of the frame that was received. Valid only when the destination address is included in the received frame.    |
| FrameType         | Enumeration                              | MAC_COMMAND, ENH_ACK, MULTIPURPOSE        | The type of frame in which the IEs were received.                                                                                     |
| CommandIdentifier | Integer                                  | As defined in Table 7-49                  | If the frame is a MAC Command, the command type of the frame, otherwise empty.                                                        |
| HeaderIeList      | Set of header IEs as described in 7.4.2  | —                                         | The header IEs, excluding Termination IEs, that were included in the frame. If empty, then no header IEs were included.               |
| PayloadIeList     | Set of payload IEs as described in 7.4.3 | —                                         | The payload IEs, excluding Termination IEs, that were included in the frame. If empty, then no payload IEs were included.             |
| SecurityLevel     | Integer                                  | As defined in Table 8-77                  | As defined in Table 8-77.                                                                                                             |
| KeyIdMode         | Integer                                  | As defined in Table 8-77                  | As defined in Table 8-77.                                                                                                             |
| KeySource         | Set of octets                            | As defined in Table 8-77                  | As defined in Table 8-77.                                                                                                             |
| KeyIndex          | Integer                                  | As defined in Table 8-77                  | As defined in Table 8-77.                                                                                                             |

**8.2.6 Primitives for reading and writing PIB attributes**

These primitives are used to read values from the PIB and write values to the PIB.

**8.2.6.1 MLME-GET.request**

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

The semantics of this primitive are as follows:

```
MLME-GET.request      (
                        PibAttribute
                        )
```

The primitive parameters are defined in Table 8-15.

**Table 8-15—MLME-GET.request parameters**

| Name         | Type         | Valid range                                                                      | Description                            |
|--------------|--------------|----------------------------------------------------------------------------------|----------------------------------------|
| PibAttribute | Octet string | Any of the PIB attributes defined in 8.4.2 and those in Table 9-8, or Table 11-2 | The name of the PIB attribute to read. |

On receipt of the MLME-GET.request primitive, the MLME checks to see whether 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 requested attribute is a PHY PIB attribute, the MLME attempts to retrieve the value from the PHY.

**8.2.6.2 MLME-GET.confirm**

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

The semantics of this primitive are as follows:

```
MLME-GET.confirm      (
                        PibAttribute,
                        PibAttributeValue,
                        Status
                        )
```

The primitive parameters are defined in Table 8-16.

If the request to read a PIB attribute was successful, the primitive returns with a Status of SUCCESS. If the identifier of the PIB attribute is not found, the primitive returns with a Status of UNSUPPORTED\_ATTRIBUTE. When an error code of UNSUPPORTED\_ATTRIBUTE is returned, the PibAttribute value parameter will be set to length zero.

**Table 8-16—MLME-GET.confirm parameters**

| Name              | Type         | Valid range                                                                          | Description                                                                                                                                       |
|-------------------|--------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| PibAttribute      | Octet string | Any of the PIB attributes defined in 8.4.2 and those in Table 9-8, or Table 11-2     | The name of the PIB attribute that was read.                                                                                                      |
| PibAttributeValue | Various      | Attribute specific; as defined in As defined in Table 8-81, Table 9-8, or Table 11-2 | The value of the indicated PIB attribute that was read. This parameter has zero length when the Status parameter is set to UNSUPPORTED_ATTRIBUTE. |
| Status            | Enumeration  | SUCCESS, UNSUPPORTED_ATTRIBUTE                                                       | The result of the request for PIB attribute information.                                                                                          |

**8.2.6.3 MLME-SET.request**

The MLME-SET.request primitive attempts to write the given value to the indicated PIB attribute.

The semantics of this primitive are as follows:

```
MLME-SET.request
(
  PibAttribute,
  PibAttributeValue
)
```

The primitive parameters are defined in Table 8-17.

**Table 8-17—MLME-SET.request parameters**

| Name              | Type         | Valid range                                                            | Description                                        |
|-------------------|--------------|------------------------------------------------------------------------|----------------------------------------------------|
| PibAttribute      | Octet string | As defined in Table 8-81, Table 9-8, or Table 11-2                     | The name of the PIB attribute to write.            |
| PibAttributeValue | Various      | Attribute specific; as defined in Table 8-81, Table 9-8, or Table 11-2 | The value to write to the indicated PIB attribute. |

On receipt of the MLME-SET.request primitive, the MLME checks to see whether the PIB attribute is a MAC PIB attribute or a 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. If the requested attribute is a PHY attribute, the MLME attempts to write the given value to the indicated PHY PIB attribute.

**8.2.6.4 MLME-SET.confirm**

The MLME-SET.confirm primitive reports the results of an attempt to write a value to a PIB attribute.

The semantics of this primitive are as follows:

```
MLME-SET.confirm
(
  PibAttribute,
  Status
)
```

The primitive parameters are defined in Table 8-18.

**Table 8-18—MLME-SET.confirm parameters**

| Name         | Type         | Valid range                                                                 | Description                                           |
|--------------|--------------|-----------------------------------------------------------------------------|-------------------------------------------------------|
| PibAttribute | Octet string | As defined in Table 8-81, Table 9-8, or Table 11-2                          | The name of the PIB attribute that was written.       |
| Status       | Enumeration  | SUCCESS, READ_ONLY, UNSUPPORTED_ATTRIBUTE, INVALID_INDEX, INVALID_PARAMETER | The result of the request to write the PIB attribute. |

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 with the Status parameter set as follows:

- READ\_ONLY – The PIBAttribute parameter specifies an attribute that is a read-only attribute.
- UNSUPPORTED\_ATTRIBUTE – The PIBAttribute parameter specifies an attribute that was not found in the database.
- INVALID\_PARAMETER – The PIBAttributeValue parameter specifies a value that is out of the valid range for the given attribute.

If the PIBAttribute parameter indicates that *macBeaconPayload* 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 *macBeaconPayload* and will issue the MLME-SET.confirm primitive with a Status of INVALID\_PARAMETER.

## 8.2.7 GTS management primitives

These primitives are used to request and maintain GTSs.

### 8.2.7.1 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.

The semantics of this primitive are as follows:

```
MLME-GTS.request      (
                        GtsCharacteristics,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

The primitive parameters are defined in Table 8-19.

**Table 8-19—MLME-GTS.request parameters**

| Name               | Type                | Valid range              | Description                                                                                                                                      |
|--------------------|---------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| GtsCharacteristics | GTS characteristics | As defined in 7.5.11     | 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             | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                        |
| KeyIdMode          | Integer             | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                        |
| KeySource          | Set of octets       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                        |
| KeyIndex           | Integer             | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                        |

On receipt of the MLME-GTS.request primitive by a device, the MLME of a device performs either the GTS request procedure, as described in 6.8.3, or the GTS deallocation procedure, as described in 6.8.5, depending on the value of the GTSCharacteristics field.

#### 8.2.7.2 MLME-GTS.confirm

The MLME-GTS.confirm primitive reports the results of a request to allocate a new GTS or to deallocate an existing GTS.

The semantics of this primitive are as follows:

```

MLME-GTS.confirm
(
    GtsCharacteristics,
    Status
)

```

The primitive parameters are defined in Table 8-20.

**Table 8-20—MLME-GTS.confirm parameters**

| Name               | Type                | Valid range                                                                                                                                                  | Description                     |
|--------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| GtsCharacteristics | GTS characteristics | As defined in 7.5.11                                                                                                                                         | The characteristics of the GTS. |
| Status             | Enumeration         | SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER. | The status of the GTS request.  |

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, as defined in 6.8.3 or 6.8.5.

If *macShortAddress* is equal to 0xffffe or 0xffff, the device is not permitted to request a GTS and the Status parameter will be set to NO\_SHORT\_ADDRESS.

### 8.2.7.3 MLME-GTS.indication

The MLME-GTS.indication primitive indicates that a GTS has been allocated or that a previously allocated GTS has been deallocated.

The semantics of this primitive are as follows:

```
MLME-GTS.indication
(
  DeviceAddress,
  GtsCharacteristics,
  SecurityLevel,
  KeyIdMode,
  KeySource,
  KeyIndex
)
```

The primitive parameters are defined in Table 8-21.

**Table 8-21—MLME-GTS.indication parameters**

| Name               | Type                | Valid range              | Description                                                                                                                                                                                                                                                                                                         |
|--------------------|---------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddress      | Short address       | 0x0000–0xffffd           | The short address of the device that has been allocated or deallocated a GTS.                                                                                                                                                                                                                                       |
| GtsCharacteristics | GTS characteristics | As defined in 7.5.11     | The characteristics of the GTS.                                                                                                                                                                                                                                                                                     |
| SecurityLevel      | Integer             | As defined in Table 8-77 | If the primitive were 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 were generated whenever a GTS is allocated or deallocated following the reception of a GTS request command, then it is as defined in Table 8-77. |
| KeyIdMode          | Integer             | As defined in Table 8-77 | If the primitive were generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive were generated whenever a GTS is allocated or deallocated following the reception of a GTS request command, then it is as defined in Table 8-77.                    |

**Table 8-21—MLME-GTS.indication parameters (continued)**

| Name      | Type          | Valid range              | Description                                                                                                                                                                                                                                                                                      |
|-----------|---------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KeySource | Set of octets | As defined in Table 8-77 | If the primitive were generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive were generated whenever a GTS is allocated or deallocated following the reception of a GTS request command, then it is as defined in Table 8-77. |
| KeyIndex  | Integer       | As defined in Table 8-77 | If the primitive were generated when a GTS deallocation is initiated by the PAN coordinator itself, this parameter is ignored. If the primitive were generated whenever a GTS is allocated or deallocated following the reception of a GTS request command, then it is as defined in Table 8-77. |

The value of the Characteristics Type field, as defined in 7.5.11, in the GtsCharacteristics parameter indicates if the GTS has been allocated or if a GTS has been deallocated.

### 8.2.8 Primitives for orphan notification

These primitives are used by a coordinator to issue a notification of an orphaned device.

#### 8.2.8.1 MLME-ORPHAN.indication

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, as defined in 7.5.7.

The semantics of this primitive are as follows:

```
MLME-ORPHAN.indication (
    OrphanAddress,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex
)
```

The primitive parameters are defined in Table 8-22.

**Table 8-22—MLME-ORPHAN.indication parameters**

| Name          | Type             | Valid range                | Description                         |
|---------------|------------------|----------------------------|-------------------------------------|
| OrphanAddress | Extended address | Any valid extended address | The address of the orphaned device. |
| SecurityLevel | Integer          | As defined in Table 8-77   | As defined in Table 8-77.           |
| KeyIdMode     | Integer          | As defined in Table 8-77   | As defined in Table 8-77.           |
| KeySource     | Set of octets    | As defined in Table 8-77   | As defined in Table 8-77.           |
| KeyIndex      | Integer          | As defined in Table 8-77   | As defined in Table 8-77.           |



**8.2.8.2 MLME-ORPHAN.response**

The MLME-ORPHAN.response primitive allows the next higher layer of a coordinator to respond to the MLME-ORPHAN.indication primitive.

The semantics of this primitive are as follows:

```
MLME-ORPHAN.response (
    OrphanAddress,
    ShortAddress,
    AssociatedMember,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex
)
```

The primitive parameters are defined in Table 8-23.

**Table 8-23—MLME-ORPHAN.response parameters**

| Name             | Type             | Valid range                | Description                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OrphanAddress    | Extended address | Any valid extended address | The address of the orphaned device.                                                                                                                                                                                                                                                                                                                                           |
| ShortAddress     | Integer          | 0x0000–0xffff              | The 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 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, FALSE                | TRUE if the orphaned device is associated with this coordinator or FALSE otherwise.                                                                                                                                                                                                                                                                                           |
| SecurityLevel    | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                     |
| KeyIdMode        | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                     |
| KeySource        | Set of octets    | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                     |
| KeyIndex         | Integer          | As defined in Table 8-75   | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                     |

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, as defined in 7.5.10, 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*, it will assume it is not associated to any coordinator in range.

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.

### 8.2.9 Primitives for resetting the MAC sublayer

These primitives are used to reset the MAC sublayer.

#### 8.2.9.1 MLME-RESET.request

The MLME-RESET.request primitive is used by the next higher layer to request that the MLME performs a reset operation.

The semantics of this primitive are as follows:

```
MLME-RESET.request      (
                          SetDefaultPib
                          )
```

The primitive parameters are defined in Table 8-24.

**Table 8-24—MLME-RESET.request parameter**

| Name          | Type    | Valid range | Description                                                                                                                                                                                                                                      |
|---------------|---------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SetDefaultPib | Boolean | TRUE, 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. |

On receipt of the MLME-RESET.request primitive, the MLME resets the PHY in an implementation-dependent manner.

#### 8.2.9.2 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

The semantics of this primitive are as follows:

```
MLME-RESET.confirm      (
                          Status
                          )
```

The primitive parameter is defined in Table 8-25.

**Table 8-25—MLME-RESET.confirm parameter**

| Name   | Type        | Valid range | Description                        |
|--------|-------------|-------------|------------------------------------|
| Status | Enumeration | SUCCESS     | The result of the reset operation. |

The Status parameter is set to SUCCESS on completion of the reset procedure.

### 8.2.10 Primitives for specifying the receiver enable time

These primitives are used to enable or disable a device's receiver at a given time.

#### 8.2.10.1 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.

The semantics of this primitive are as follows:

```
MLME-RX-ENABLE.request (
    DeferPermit,
    RxOnTime,
    RxOnDuration,
    RangingRxControl
)
```

The primitive parameters are defined in Table 8-26.

**Table 8-26—MLME-RX-ENABLE.request parameters**

| Name             | Type        | Valid range             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeferPermit      | Boolean     | TRUE, 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 <i>superframe</i> refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated.                                                      |
| RxOnTime         | Integer     | 0x000000–0xfffff        | 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 <i>superframe</i> refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated. |
| RxOnDuration     | Integer     | 0x000000–0xfffff        | 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.                                                                                                                                                                                                                                                                                                                                                                                                    |
| RangingRxControl | Enumeration | RANGING_OFF, RANGING_ON | Configure the transceiver to Rx with ranging for a value of RANGING_ON or to not enable ranging for RANGING_OFF.                                                                                                                                                                                                                                                                                                                                                                                                                          |

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.

The MLME will treat the request to enable or disable the receiver as secondary to other responsibilities of the device (e.g., GTSSs, coordinator beacon tracking, or 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 has 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. If the current time measured from the start of the superframe is less than  $(RxOnTime - macSifsPeriod)$ , the MLME attempts to enable the receiver in the current superframe. If the current time measured from the start of the superframe is greater than or equal to  $(RxOnTime - macSifsPeriod)$  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.

#### 8.2.10.2 MLME-RX-ENABLE.confirm

The MLME-RX-ENABLE.confirm primitive reports the results of the attempt to enable or disable the receiver.

The semantics of this primitive are as follows:

```
MLME-RX-ENABLE.confirm      (
                               Status
                              )
```

The primitive parameters are defined in Table 8-27.

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. 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 8.2.10.1.

**Table 8-27—MLME-RX-ENABLE.confirm parameter**

| Name   | Type        | Valid range                                                                             | Description                                                  |
|--------|-------------|-----------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Status | Enumeration | SUCCESS, PAST_TIME,<br>ON_TIME_TOO_LONG,<br>INVALID_PARAMETER,<br>RANGING_NOT_SUPPORTED | The result of the request to enable or disable the receiver. |

**8.2.11 Primitives for channel scanning**

These primitives are used to either find PANs in a channel or measure the energy in the channel.

**8.2.11.1 MLME-SCAN.request**

The MLME-SCAN.request primitive is used to initiate a channel scan over a given set of channels.

The semantics of this primitive are as follows:

```

MLME-SCAN.request
(
  ScanType,
  ScanChannels,
  ScanDuration,
  ChannelPage,
  SecurityLevel,
  KeyIdMode,
  KeySource,
  KeyIndex,
  LinkQualityScan,
  PanIdSuppressed,
  SeqNumSuppressed,
  HeaderIdList,
  PayloadIdList,
  HeaderIdList,
  NestedIdSubIdList,
  MpmScanDurationBPan,
  MpmScanDurationNbPan,
  MpmScan,
  MpmScanType
)

```

The primitive parameters are defined in Table 8-28.

**Table 8-28—MLME-SCAN.request parameters**

| Name     | Type        | Valid range                                                             | Description                                                                                                                                             |
|----------|-------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| ScanType | Enumeration | ED, ACTIVE,<br>PASSIVE, ORPHAN,<br>ENHANCED_ACTIVE_SCAN,<br>RIT_PASSIVE | Indicates the type of scan performed, as described in 6.3.1. For an RFD, ED, ACTIVE, ENHANCED_ACTIVE_SCAN, and RIT_PASSIVE are optional. SetDefaultPib. |

**Table 8-28—MLME-SCAN.request parameters (continued)**

| Name              | Type                                                                | Valid range                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|---------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ScanChannels      | Set of integers                                                     | Any set of valid channel numbers | The channel numbers to be scanned. For pages 0 to 6, the 27 bits (b0, b1, ..., b26) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 channels supported by the ChannelPage parameter. For pages 7 and 8, a bitmap corresponding to the number of channels, $n$ , where (b0, b1, ..., bn) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the $n$ channels supported by the ChannelPage and PHY configuration. |
| 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 $[aBaseSuperframeDuration \times (2^n + 1)]$ , where $n$ is the value of the ScanDuration parameter.                                                                                                                                                                                 |
| ChannelPage       | Integer                                                             | Any valid channel page           | The channel page on which to perform the scan.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| SecurityLevel     | Integer                                                             | As defined in Table 8-75         | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| KeyIdMode         | Integer                                                             | As defined in Table 8-75         | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| KeySource         | Set of octets                                                       | As defined in Table 8-75         | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| KeyIndex          | Integer                                                             | As defined in Table 8-75         | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| LinkQualityScan   | Boolean                                                             | TRUE, FALSE                      | If TRUE, Link Quality Scan should be enabled, otherwise FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| PanIdSuppressed   | Boolean                                                             | TRUE, FALSE                      | Set to TRUE if the PAN ID is suppressed in the frame, FALSE otherwise.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| SeqNumSuppressed  | Boolean                                                             | TRUE, FALSE                      | Set to TRUE if the sequence number is suppressed in the frame, FALSE otherwise.                                                                                                                                                                                                                                                                                                                                                                                                         |
| HeaderIeList      | List of IEs as described in 7.4.2                                   | —                                | The header IEs, excluding Termination IEs, to be added to the frame.                                                                                                                                                                                                                                                                                                                                                                                                                    |
| PayloadIeList     | List of IEs as described in 7.4.3                                   | —                                | The payload IEs, excluding Termination IEs, to be added to the frame.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| HeaderIeIdList    | Set of header IE IDs, as described in Table 7-7                     | —                                | The IDs of header IEs, excluding Termination IEs, to be added to the frame. The MAC will provide the content for the IE.                                                                                                                                                                                                                                                                                                                                                                |
| NestedIeSubIdList | Set of Nested IE sub-IDs, as described in Table 7-16 and Table 7-17 | —                                | The sub-IDs of nested IEs, excluding Termination IEs, to be added to the frame. The MAC will provide the content for the IE.                                                                                                                                                                                                                                                                                                                                                            |

Table 8-28—MLME-SCAN.request parameters (continued)

| Name                 | Type        | Valid range                       | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|-------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MpmScanDurationBPan  | Integer     | 0–14                              | The maximum time spent scanning for an Enhanced Beacon frame in a beacon-enabled PAN on the channel is $[aBaseSuperframeDuration * 2^n]$ symbols, where symbol refers to the symbol time in the current PHY, and $n$ is equal to MpmScanDurationBPan. See 6.2.3.                                                                                                                                                    |
| MpmScanDurationNbPan | Integer     | 1–16 383                          | The maximum time spent scanning for an Enhanced Beacon frame in a nonbeacon-enabled PAN on the channel is $[aBaseSlotDuration * n]$ symbols, where symbol refers to the symbol time in the current PHY, and $n$ is equal to MpmScanDurationNbPan. See 6.2.3.                                                                                                                                                        |
| MpmScan              | Boolean     | TRUE, FALSE                       | This parameter is only valid for SUN devices. When set to TRUE, an MPM scan is invoked and the ScanDuration parameter shall be ignored. When set to FALSE, the scan duration shall be set according to the ScanDuration parameter; the MpmScanType, MpmScanDurationBPan, and MpmScanDurationNbPan parameters shall be ignored.                                                                                      |
| MpmScanType          | Enumeration | BEACON_ENABLED, NONBEACON_ENABLED | This parameter is only valid for SUN devices. When the MpmScan parameter is set to TRUE and the MpmScanType parameter is set to BEACON_ENABLED, the scan duration shall be set according to the MpmScanDurationBPan parameter. When the MpmScan parameter is set to TRUE and the MpmScanType parameter is set to NONBEACON_ENABLED, the scan duration shall be set according to the MpmScanDurationNbPan parameter. |

When the MLME receives this primitive, it begins the appropriate scan procedure, as defined in 6.3.

The SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters are used only in an orphan scan.

When the ScanType is set to ENHANCED\_ACTIVE\_SCAN, this primitive is used to initiate an enhanced active scan.

The active scan, as described in 6.3.1.2, is performed on each channel by the MLME first sending a Beacon Request command as described in 7.5.8 or an Enhanced Beacon Request command as described in 7.5.9 if using an enhanced active scan. The MLME then enables the receiver and records the information contained in each received beacon in a PAN descriptor structure as described in Table 8-12. The active scan on a particular channel terminates when the number of PAN descriptors stored equals an implementation-specified maximum or when  $[aBaseSuperframeDuration \times (2^n + 1)]$  symbols, where  $n$  is the value of the ScanDuration parameter, have elapsed, whichever comes first.

**8.2.11.2 MLME-SCAN.confirm**

The MLME-SCAN.confirm primitive reports the result of the channel scan request.

The semantics of this primitive are as follows:

```

MLME-SCAN.confirm
(
    ScanType,
    ChannelPage,
    UnscannedChannels,
    ResultListSize,
    EnergyDetectList,
    PanDescriptorList,
    DetectedCategory,
    HrpUwbEnergyDetectList,
    Status
)

```

The primitive parameters are defined in Table 8-29.

**Table 8-29—MLME-SCAN.confirm parameters**

| Name               | Type                          | Valid range                                                    | Description                                                                                                                                                                                                                                                     |
|--------------------|-------------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ScanType           | Integer                       | ED, ACTIVE, PASSIVE, ORPHAN, RIT_PASSIVE, ENHANCED_ACTIVE_SCAN | As defined in Table 8-29.                                                                                                                                                                                                                                       |
| ChannelPage        | Integer                       | Any valid channel page                                         | The channel page on which the scan was performed, as defined in 10.1.2.                                                                                                                                                                                         |
| Unscanned Channels | List of integers              | Any list of valid channels                                     | A list of the channels given in the request which were not scanned. 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 | As defined in Table 8-12                                       | The list of PAN descriptors, one for each beacon found during an active or passive scan if <i>macAutoRequest</i> is set to TRUE. This parameter is null for ED and orphan scans or when <i>macAutoRequest</i> is set to FALSE during an active or passive scan. |



**Table 8-29—MLME-SCAN.confirm parameters (continued)**

| Name                   | Type             | Valid range                                                                                                                                               | Description                                                                                                                                                                                                                   |
|------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DetectedCategory       | Integer          | 0x00–0xff                                                                                                                                                 | Categorization of energy detected in channel with the following values:<br>0: Category detection is not supported<br>1: HRP UWB PHY detected<br>2: Non-HRP UWB PHY signal source detected<br>3–25: Reserved for future use.   |
| HrpUwbEnergyDetectList | List of Integers | 0x00–0xff for each element of the list                                                                                                                    | For HRP UWB PHYs, the list of energy measurements taken. The total number of measurements is indicated by ResultListSize. This parameter is null for active, passive, and orphan scans. It is also null for non-HRP UWB PHYs. |
| Status                 | Enumeration      | SUCCESS, LIMIT_REACHED, NO_BEACON, SCAN_IN_PROGRESS, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, BAD_CHANNEL, INVALID_PARAMETER | The status of the scan request.                                                                                                                                                                                               |

If the requested scan was successful, the Status parameter will be set to SUCCESS.

If the MLME receives the MLME-SCAN.request primitive while performing a previously initiated scan operation, the MLME will not perform the scan and the Status parameter will be set to SCAN\_IN\_PROGRESS.

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 set 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.

If the MLME-SCAN.request primitive requested an orphan scan, the ResultListSize parameter will be set to zero. 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 set 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.

If the MLME-SCAN.request primitive requested an active, passive, or orphan scan, the EnergyDetectList and HrpUwbEnergyDetectList parameters will be null. If the MLME-SCAN.request primitive requested an ED or orphan scan, the PANDescriptorList parameter will be null.

If, during an ED, active, or passive scan, the implementation-specified maximum of PAN descriptors is reached thus terminating the scan procedure, the MAC sublayer will issue the MLME-SCAN.confirm primitive with a Status of LIMIT\_REACHED.

If the MLME-SCAN.request primitive requested an ED and the PHY type is HRP UWB, as indicated by the *phyChannelPage*, then the *HrpUwbEnergyDetectList* contains the results for the HRP UWB channels scanned, and the *EnergyDetectList* and *PANDescriptorList* are null. The HRP UWB scan is fully described in 6.3.1.

### 8.2.12 Primitives for updating the superframe configuration

These primitives are used by an FFD to initiate a PAN, to begin using a new superframe configuration, or to stop transmitting beacons. In addition, a device uses these primitives to begin using a new superframe configuration.

#### 8.2.12.1 MLME-START.request

The MLME-START.request primitive is used by the PAN coordinator to initiate a new PAN or to begin using a new superframe configuration. This primitive is also used by a device already associated with an existing PAN to begin using a new superframe configuration.

The semantics of this primitive are as follows:

```
MLME-START.request
(
    PanId,
    ChannelNumber,
    ChannelPage,
    StartTime,
    BeaconOrder,
    SuperframeOrder,
    PanCoordinator,
    BatteryLifeExtension,
    CoordRealignement,
    CoordRealignSecurityLevel,
    CoordRealignKeyIdMode,
    CoordRealignKeySource,
    CoordRealignKeyIndex,
    BeaconSecurityLevel,
    BeaconKeyIdMode,
    BeaconKeySource,
    BeaconKeyIndex,
    HeaderIdList,
    PayloadIdList,
    HeaderIdList,
    NestedIdSubIdList,
)
```

The primitive parameters are defined in Table 8-30.

**Table 8-30—MLME-START.request parameters**

| Name          | Type    | Valid range              | Description                          |
|---------------|---------|--------------------------|--------------------------------------|
| PanId         | Integer | 0x0000–0xffff            | The PAN ID to be used by the device. |
| ChannelNumber | Integer | Any valid channel number | The channel number to use.           |

**Table 8-30—MLME-START.request parameters (continued)**

| Name                      | Type    | Valid range            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------|---------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ChannelPage               | Integer | Any valid channel page | The channel page to use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| StartTime                 | Integer | 0x000000–0xfffff       | 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 frame 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. This parameter is ignored in a DSME-enabled PAN. The time is specified in symbols and is rounded to a backoff period boundary. 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                   | Indicates the frequency with which the beacon is transmitted, as defined in 6.2.1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| SuperframeOrder           | Integer | 0–BO or 15             | The length of the active portion of the superframe, including the beacon frame, as defined in 6.2.1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| PanCoordinator            | Boolean | TRUE, 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, FALSE            | If this value is TRUE, the receiver of the beaconing device is disabled <i>macBattLifeExtPeriods</i> full backoff periods after the 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.                                                                                                                                                                                                                                                                                      |
| CoordRealign              | Boolean | TRUE, FALSE            | TRUE if a coordinator realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| CoordRealignSecurityLevel | Integer | 0x00–0x07              | The security level to be used for Coordinator Realign commands, as described in Table 9-6.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Table 8-30—MLME-START.request parameters (continued)**

| Name                  | Type                                                                | Valid range                                         | Description                                                                                                                                                           |
|-----------------------|---------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CoordRealignKeyIdMode | Integer                                                             | 0x00–0x03                                           | The mode used to identify the key to be used, as described in 9.4.1.2. This parameter is ignored if the CoordRealignSecurityLevel parameter is set to 0x00.           |
| CoordRealignKeySource | Set of octets                                                       | As specified by the CoordRealignKeyIdMode parameter | The originator of the key to be used, as described in 9.4.3.1. This parameter is ignored if the CoordRealignKeyIdMode parameter is ignored or is set to 0x00 or 0x01. |
| CoordRealignKeyIndex  | Integer                                                             | 0x01–0xff                                           | The index of the key to be used, as described in 9.4.3.2. This parameter is ignored if the CoordRealignKeyIdMode parameter is ignored or is set to 0x00.              |
| BeaconSecurityLevel   | Integer                                                             | 0x00–0x07                                           | The security level to be used for beacon frames, as described in Table 9-6.                                                                                           |
| BeaconKeyIdMode       | Integer                                                             | 0x00–0x03                                           | The mode used to identify the key to be used, as described in Table 9-7. This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.               |
| BeaconKeySource       | Set of octets                                                       | As specified by the BeaconKeyIdMode parameter       | The originator of the key to be used, as described in 9.4.3.1. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00 or 0x01.          |
| BeaconKeyIndex        | Integer                                                             | 0x01–0xff                                           | The index of the key to be used, as described in 9.4.3.2. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.                       |
| HeaderIeList          | Set of IEs as described in 7.4.2                                    | —                                                   | The header IEs, excluding Termination IEs, that are to be included in the Beacon frame, in addition to any header IEs added by the MAC.                               |
| PayloadIeList         | Set of IEs as described in 7.4.3                                    | —                                                   | The payload IEs, excluding Termination IEs, that are to be included in the Beacon frame, in addition to any payload IEs added by the MAC.                             |
| HeaderIeIdList        | Set of header IE IDs, as described in Table 7-7                     | —                                                   | The IDs of header IEs, excluding Termination IEs, to be added to the Beacon frame. The MAC will provide the content for the IE.                                       |
| NestedIeSubIdList     | Set of Nested IE sub-IDs, as described in Table 7-16 and Table 7-17 | —                                                   | The sub-IDs of nested IEs, excluding Termination IEs, to be added to the Beacon frame. The MAC will provide the content for the IE.                                   |

When the `CoordRealignment` parameter is set to `TRUE`, the coordinator attempts to transmit a Coordinator Realignment command as described in 7.5.10. If the transmission of the coordinator realignment 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). If the coordinator realignment command is successfully transmitted, the MLME updates the PIB attributes `BeaconOrder`, `SuperframeOrder`, `PanId`, `ChannelPage`, and `ChannelNumber` parameters.

When the `CoordRealignment` parameter is set to `FALSE`, the MLME updates the appropriate PIB attributes with the values of the `BeaconOrder`, `SuperframeOrder`, `PanId`, `ChannelPage`, and `ChannelNumber` parameters, as described in 6.3.3.4.

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 `CoordRealignment` parameter is set to `TRUE`, the `CoordRealignSecurityLevel`, `CoordRealignKeyIdMode`, `CoordRealignKeySource`, and `CoordRealignKeyIndex` parameters will be used to process the MAC command. If the `BeaconOrder` parameter indicates a beacon-enabled network, the `BeaconSecurityLevel`, `BeaconKeyIdMode`, `BeaconKeySource`, and `BeaconKeyIndex` parameters will be used to process the beacon frame.

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. 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` 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. Otherwise, the MLME then begins beacon transmissions when the current time, obtained from the local clock, equals the calculated time.

If the `SdIndex` parameter is nonzero and the MLME is not currently tracking the beacon of the coordinator through which it is associated, the MLME shall issue the `MLME-START.confirm` primitive with a Status of `TRACKING_OFF`.

If `macUseEnhancedBeacon` is `TRUE`, the MLME shall use enhanced beacons rather than standard beacons containing the IEs provided. The enhanced beacon shall be secured according to the value of the `BeaconSecurityLevel` parameter.

#### 8.2.12.2 MLME-START.confirm

The `MLME-START.confirm` primitive reports the results of the attempt to start using a new superframe configuration.

The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 8-31.

**Table 8-31—MLME-START.confirm parameters**

| Name   | Type        | Valid range                                                                                                                                                                                             | Description                                                                   |
|--------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Status | Enumeration | SUCCESS,<br>NO_SHORT_ADDRESS,<br>SUPERFRAME_OVERLAP,<br>TRACKING_OFF,<br>INVALID_PARAMETER,<br>COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_SECURITY,<br>CHANNEL_ACCESS_FAILURE | The result of the attempt to start using an updated superframe configuration. |

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 as follows:

- NO\_SHORT\_ADDRESS – The *macShortAddress* is set to 0xffff.
- CHANNEL\_ACCESS\_FAILURE – The transmission of the coordinator realignment frame failed.
- FRAME\_TOO\_LONG – The length of the beacon frame exceeds *aMaxPhyPacketSize*.
- SUPERFRAME\_OVERLAP – The outgoing superframe overlaps the incoming superframe.
- TRACKING\_OFF – The *StartTime* parameter is nonzero, and the MLME is not currently tracking the beacon of the coordinator through which it is associated.
- A security error code, as defined in 9.2.

### 8.2.13 Primitives for synchronizing with a coordinator

These primitives are used to synchronize with a coordinator and to communicate loss of synchronization to the next higher layer.

#### 8.2.13.1 MLME-SYNC.request

The MLME-SYNC.request primitive requests to synchronize with the coordinator by acquiring and, if specified, tracking its beacons.

The semantics of this primitive are as follows:

```
MLME-SYNC.request
(
    ChannelNumber,
    ChannelPage,
    TrackBeacon
)
```

The primitive parameters are defined in Table 8-32.

**Table 8-32—MLME-SYNC.request parameters**

| Name          | Type    | Valid range              | Description                                                                                                                                                      |
|---------------|---------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ChannelNumber | Integer | Any valid channel number | The channel number on which to attempt coordinator synchronization.                                                                                              |
| ChannelPage   | Integer | Any valid channel page   | The channel page on which to attempt coordinator synchronization.                                                                                                |
| TrackBeacon   | Boolean | TRUE, FALSE              | TRUE if the MLME is to synchronize with the next beacon and attempts to track all future beacons. FALSE if the MLME is to synchronize with only the next beacon. |

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 ChannelNumber parameters, respectively. 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 will treat it as a new synchronization request.

#### 8.2.13.2 MLME-SYNC-LOSS.indication

The MLME-SYNC-LOSS.indication primitive indicates the loss of synchronization with a coordinator.

The semantics of this primitive are as follows:

```

MLME-SYNC-LOSS.indication (
    LossReason,
    PanId,
    ChannelNumber,
    ChannelPage,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex
)

```

The primitive parameters are defined in Table 8-33.

**Table 8-33—MLME-SYNC-LOSS.indication parameters**

| Name       | Type        | Valid range                                                            | Description                               |
|------------|-------------|------------------------------------------------------------------------|-------------------------------------------|
| LossReason | Enumeration | PAN_ID_CONFLICT,<br>REALIGNMENT,<br>BEACON_LOST,<br>SUPERFRAME_OVERLAP | The reason that synchronization was lost. |

**Table 8-33—MLME-SYNC-LOSS.indication parameters (continued)**

| Name          | Type          | Valid range              | Description                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|---------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PanId         | Integer       | 0x0000–0xffff            | The PAN ID with which the device lost synchronization or to which it was realigned.                                                                                                                                                                                                                                                                                                           |
| ChannelNumber | Integer       | Any valid channel        | The channel number on which the device lost synchronization or to which it was realigned.                                                                                                                                                                                                                                                                                                     |
| ChannelPage   | Integer       | Any valid channel page   | The channel page on which the device lost synchronization or to which it was realigned.                                                                                                                                                                                                                                                                                                       |
| SecurityLevel | Integer       | As defined in Table 8-77 | If the primitive were 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 were generated following the reception of either a coordinator realignment command or a PAN ID Conflict Notification command, then it is as defined in Table 8-77. |
| KeyIdMode     | Integer       | As defined in Table 8-77 | If the primitive were 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 were generated following the reception of either a coordinator realignment command or a PAN ID Conflict Notification command, then it is as defined in Table 8-77.         |
| KeySource     | Set of octets | As defined in Table 8-77 | If the primitive were 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 were generated following the reception of either a coordinator realignment command or a PAN ID Conflict Notification command, then it is as defined in Table 8-77.         |
| KeyIndex      | Integer       | As defined in Table 8-77 | If the primitive were 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 were generated following the reception of either a coordinator realignment command or a PAN ID Conflict Notification command, then it is as defined in Table 8-77.         |

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 either a PAN ID conflict or an overlap between the outgoing superframe and the incoming superframe, as described in 6.2.2.



The LossReason parameter indicates the reason why the primitive was issued. Values for the LossReason parameter are as follows:

- PAN\_ID\_CONFLICT – The device has detected a PAN ID conflict and has communicated it to the PAN coordinator or the PAN coordinator has received a PAN ID Conflict Notification command regarding a device that is associated with it, as described in 6.3.2.
- REALIGNMENT – The device has received the coordinator realignment command from the coordinator through which it is associated and the MLME was not carrying out an orphan scan, as described in 6.3.3.3.
- BEACON\_LOST – The device has missed too many beacons, as described in 6.5.2.
- SUPERFRAME\_OVERLAP – The device has received a coordinator realignment comment from its coordinator that would cause the incoming and outgoing superframes to overlap, as described in 6.2.2.

#### 8.2.14 Primitives for requesting data from a coordinator

These primitives are used to request data from a coordinator.

##### 8.2.14.1 MLME-POLL.request

The MLME-POLL.request primitive prompts the device to request data from the coordinator.

The semantics of this primitive are as follows:

```
MLME-POLL.request
(
  CoordAddrMode,
  CoordPanId,
  CoordAddress,
  SecurityLevel,
  KeyIdMode,
  KeySource,
  KeyIndex
)
```

The primitive parameters are defined in Table 8-34.

**Table 8-34—MLME-POLL.request parameters**

| Name          | Type          | Valid range                                 | Description                                                           |
|---------------|---------------|---------------------------------------------|-----------------------------------------------------------------------|
| CoordAddrMode | Enumeration   | SHORT, EXTENDED                             | The addressing mode of the coordinator to which the poll is intended. |
| CoordPanId    | Integer       | 0x0000–0xffff                               | The PAN ID of the coordinator to which the poll is intended.          |
| CoordAddress  | —             | As specified by the CoordAddrMode parameter | The address of the coordinator to which the poll is intended.         |
| SecurityLevel | Integer       | As defined in Table 8-75                    | As defined in Table 8-75.                                             |
| KeyIdMode     | Integer       | As defined in Table 8-75                    | As defined in Table 8-75.                                             |
| KeySource     | Set of octets | As defined in Table 8-75                    | As defined in Table 8-75.                                             |
| KeyIndex      | Integer       | As defined in Table 8-75                    | As defined in Table 8-75.                                             |

On receipt of the MLME-POLL.request primitive, the MLME requests data from the coordinator, as described in 6.7.3. 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.

#### 8.2.14.2 MLME-POLL.confirm

The MLME-POLL.confirm primitive reports the results of a request to poll the coordinator for data.

The semantics of this primitive are as follows:

MLME-POLL.confirm ( Status )

The primitive parameters are defined in Table 8-35.

**Table 8-35—MLME-POLL.confirm parameters**

| Name   | Type    | Valid range                                                                                                                                                    | Description                     |
|--------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| Status | Integer | SUCCESS,<br>CHANNEL_ACCESS_FAILURE,<br>NO_ACK, NO_DATA,<br>COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_SECURITY,<br>INVALID_PARAMETER | The status of the data request. |

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 8.2.14.1.

If the Frame Pending field of the Ack 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, 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, as described in 8.3.3.

If a frame is not received at the expected time even though the acknowledgment to the Data Request command has its Frame Pending field set to one, the MLME will issue the MLME-POLL.confirm primitive with a Status of NO\_DATA.

#### 8.2.15 Primitives for specifying dynamic preamble

These primitives are used by a device to enable or disable DPS as well as to define the value of dynamic preamble for transmission and reception for a given time. DPS is only supported by the HRP UWB PHY.

**8.2.15.1 MLME-DPS.request**

The MLME-DPS.request primitive allows the next higher layer to request that the PHY utilize a given pair of preamble codes for a single use pending expiration of the DpsIndexDuration.

The semantics of this primitive are as follows:

```
MLME-DPS.request      (
                        TxDpsIndex
                        RxDpsIndex
                        DpsIndexDuration
                        )
```

The primitive parameters are defined in Table 8-36.

**Table 8-36—MLME-DPS.request parameters**

| Name             | Type    | Valid range         | Description                                                                                                                                                                                                                       |
|------------------|---------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TxDpsIndex       | Integer | 0, 13–16, 21–24     | The index value for the transmitter. A value of 0 disables the index and indicates that the <i>phyCurrentCode</i> value is to be used, as defined in 16.2.5.1. Other values indicate the preamble code, as defined in Table 16-7. |
| RxDpsIndex       | Integer | 0, 13–16, 21–24     | The index value for the receiver. A value of 0 disables the index and indicates that the <i>phyCurrentCode</i> value is to be used, as defined in 16.2.5.1. Other values indicate the preamble code, as defined in Table 16-7.    |
| DpsIndexDuration | Integer | 0x000000–0xffffffff | The number of symbols for which the transmitter and receiver will utilize the respective DPS indices if a MCPS-DATA.request primitive is not issued.                                                                              |

This primitive may also be generated to cancel a previously generated request to enable the transmitter and receiver dynamic preambles. The use of the index for the transmitter and receiver is enabled or disabled exactly once per primitive request.

The MLME starts the timer that assures that the device returns to a normal operating state with default preambles if a following MCPS-DATA.request primitive does not occur. After starting the timer, the MLME responds with a MLME-DPS.confirm primitive with the appropriate Status parameter.

**8.2.15.2 MLME-DPS.confirm**

The MLME-DPS.confirm primitive reports the results of the attempt to enable or disable the DPS.

The semantics of this primitive are as follows:

```
MLME-DPS.confirm      (
                        Status
                        )
```

The primitive parameter is defined in Table 8-37.

| Name   | Type        | Valid range                   | Description                                                       |
|--------|-------------|-------------------------------|-------------------------------------------------------------------|
| Status | Enumeration | SUCCESS,<br>DPS_NOT_SUPPORTED | The result of the request to enable or disable dynamic preambles. |

If any parameter in the MLME-DPS.request primitive is not supported or is out of range, the Status of DPS\_NOT\_SUPPORTED is returned. If the request to enable or disable the DPS was successful, the MLME issues the MLME-DPS.confirm primitive with a Status of SUCCESS.

The semantics of this primitive are as follows:

If a MCPS-DATA.request primitive is not received before the timer expires, the MLME issues the MLME-DPS.indication primitive to the next higher layer.

These primitives are used to obtain the results of a channel sounding from an RDEV that supports the optional sounding capability.

The MLME-SOUNDING.request primitive is used by the next higher layer to request that the PHY respond with channel sounding information. The MLME-SOUNDING.request primitive shall be supported by all RDEVs; however, the underlying sounding capability is optional in all cases.

MLME-SOUNDING.request ()

### 8.2.16.2 MLME-SOUNDING.confirm

Copyright © 2016 IEEE. All rights reserved.

The semantics of this primitive are as follows:

MLME-SOUNDING.confirm  
(  
  SoundingList,  
  Status  
)

The primitive parameters are defined in Table 8-38.

**Table 8-38—MLME-SOUNDING.confirm parameters**

| Name         | Type                    | Valid range                              | Description                                        |
|--------------|-------------------------|------------------------------------------|----------------------------------------------------|
| SoundingList | List of sounding points | —                                        | Results of the sounding measurement.               |
| Status       | Enumeration             | SUCCESS, NO_DATA, SOUNDING_NOT_SUPPORTED | The status of the attempt to return sounding data. |

The elements of a SoundingList are defined in Table 8-39.

**Table 8-39—Elements of a SoundingList**

| Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Type           | Valid range | Description                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------------|---------------------------------------------------------|
| SoundingTime                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Signed integer | —           | The LSB represents a nominal 16 ps. (see NOTE).         |
| SoundingAmplitude                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Signed integer | —           | A relative measurement of the received signal strength. |
| NOTE—Each element of the <i>SoundingList</i> contains a <i>SoundingTime</i> and a <i>SoundingAmplitude</i> . The <i>SoundingTime</i> is a signed integer, and the LSB for the HRP UWB PHY represents a nominal 16 ps ( $2^{-7}$ of a chip time), and for the LRP UWB PHY 1 ps ( $2^{-20}$ of a chip time). The <i>SoundingAmplitude</i> is a signed integer representing a relative measurement. The <i>SoundingAmplitudes</i> have no absolute meaning, only a relative meaning. |                |             |                                                         |

If the channel sounding information is available, the Status parameter will be set to SUCCESS and the SoundingList will contain valid data.

If the MLME-SOUNDING.request primitive is received when there is no information present, e.g., when the PHY is in the process of performing a measurement, the Status parameter will be set to NO\_DATA.

If the channel sounding capability is not supported by the PHY, the Status parameters will be set to UNSUPPORTED\_ATTRIBUTE.

### 8.2.17 Primitives for ranging calibration

These primitives are used to obtain the results of a ranging calibration request from an RDEV.

**8.2.17.1 MLME-CALIBRATE.request**

The MLME-CALIBRATE.request primitive attempts to have the PHY respond with RMARKER offset information. The MLME-CALIBRATE.request primitive shall be implemented by RDEVs.

The semantics of this primitive are as follows:

MLME-CALIBRATE.request                   ()

The MLME issues the MLME-CALIBRATE.confirm primitive with the appropriate information.

**8.2.17.2 MLME-CALIBRATE.confirm**

The MLME-CALIBRATE.confirm primitive reports the result of a request to the PHY to provide internal propagation path information. The MLME-CALIBRATE.confirm primitive shall be implemented by RDEVs.

MLME-CALIBRATE.confirm                   (  
    CalTxRMarkerOffset,  
    CalRxRMarkerOffset,  
    Status  
)

The primitive parameters are defined in Table 8-40.

**Table 8-40—MLME-CALIBRATE.confirm parameters**

| Name               | Type             | Valid range                                                 | Description                                                                                                                                                                                                                                                                                                             |
|--------------------|------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CalTxRMarkerOffset | Unsigned Integer | 0x00000000–0xffffffff                                       | A count of the propagation time from the ranging counter to the transmit antenna. For the HRP UWB PHY the LSB of a time value represents $2^{-7}$ of a chip time at the mandatory chipping rate of 499.2 MHz. For the LRP UWB PHY the LSB of a time value represents $2^{-20}$ of the base mode chipping rate of 1 MHz. |
| CalRxRMarkerOffset | Unsigned Integer | 0x00000000–0xffffffff                                       | A count of the propagation time from the receive antenna to the ranging counter. For the HRP UWB PHY the LSB of a time value represents $2^{-7}$ of a chip time at the mandatory chipping rate of 499.2 MHz. For the LRP UWB PHY the LSB of a time value represents $2^{-20}$ of the base mode chipping rate of 1 MHz.  |
| Status             | Enumeration      | SUCCESS, NO_DATA, COMPUTATION_NEEDED, UNSUPPORTED_ATTRIBUTE | The status of the attempt to retrieve internal propagation path information from the PHY.                                                                                                                                                                                                                               |

The MLME-CALIBRATE.confirm primitive is generated by the MLME and issued to its next higher layer in response to a MLME-CALIBRATE.request primitive.

If the feature is supported, the MLME issues the MLME-CALIBRATE.confirm primitive with a Status of SUCCESS.

If the MLME-CALIBRATE.request primitive is received when there is no information present, e.g., when the PHY is in the process of performing a measurement, the Status parameter will be set to NO\_DATA.

If the PHY does not support autonomous self-calibration, the Status parameter will be set to a value of COMPUTATION\_NEEDED. This indicates to the next higher layer that it should use the sounding primitives to finish the calibration.

If the channel sounding capability is not present in the PHY, the Status parameter will be set to a value of UNSUPPORTED\_ATTRIBUTE.

## 8.2.18 Primitives for Beacon Generation

### 8.2.18.1 MLME-BEACON.request

The MLME-BEACON.request primitive requests the generation of a Beacon frame or Enhanced Beacon frame in a nonbeacon-enabled PAN, either in response to a Beacon Request command or Enhanced Beacon Request command when *macBeaconAutoRespond* is FALSE, or on demand, e.g., to send beacons to enable a TSCH passive scan.

The semantics of this primitive are as follows:

```
MLME-BEACON.request      (
                           BeaconType,
                           Channel,
                           ChannelPage,
                           SuperFrameOrder,
                           HeaderIdList,
                           PayloadIdList,
                           HeaderIdList,
                           NestedIdSubIdList,
                           BeaconSecurityLevel,
                           BeaconKeyIdMode,
                           BeaconKeySource,
                           BeaconKeyIndex,
                           DstAddrMode,
                           DstAddr,
                           BsnSuppression
                           )
```

The primitive parameters are defined in Table 8-41.

**Table 8-41—MLME-BEACON.request parameters**

| Name       | Type        | Valid range                | Description                                                                   |
|------------|-------------|----------------------------|-------------------------------------------------------------------------------|
| BeaconType | Enumeration | BEACON,<br>ENHANCED BEACON | Indicates if the beacon request to be sent is a beacon or an enhanced beacon. |
| Channel    | Integer     | Any valid channel number   | The channel number to use.                                                    |

**Table 8-41—MLME-BEACON.request parameters (continued)**

| Name                | Type                                                                 | Valid range                                   | Description                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ChannelPage         | Integer                                                              | Any valid channel page                        | The channel page to use.                                                                                                                                                                 |
| SuperframeOrder     | Integer                                                              | 0–15                                          | The length of the active portion of the superframe, including the beacon frame.                                                                                                          |
| HeaderIeList        | List of header IEs as described in 7.4.2                             | —                                             | The header IEs, excluding Termination IEs, to be included in the Beacon frame. If empty, then no header IEs are included.                                                                |
| PayloadIeList       | List of payload IEs as described in 7.4.3                            | —                                             | The payload IEs, excluding Termination IEs, to be included in the Beacon frame. If empty, then no payload IEs are included.                                                              |
| HeaderIeIdList      | List of header IE IDs, as described in Table 7-7.                    | —                                             | The IDs of header IEs, excluding Termination IEs, to be added to the frame. The MAC will provide the content for the IE.                                                                 |
| NestedIeSubIdList   | List of Nested IE sub-IDs, as described in Table 7-16 and Table 7-17 | —                                             | The sub-IDs of nested IEs, excluding Termination IEs, to be added to the Data frame. The MAC will provide the content for the IE.                                                        |
| BeaconSecurityLevel | Integer                                                              | 0x00–0x07                                     | The security level to be used for beacon frames, as described in Table 9-6.                                                                                                              |
| BeaconKeyIdMode     | Integer                                                              | 0x00–0x03                                     | The mode used to identify the key to be used, as defined in Table 9-7. This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.                                    |
| BeaconKeySource     | Set of octets                                                        | As specified by the BeaconKeyIdMode parameter | The originator of the key to be used, as defined in 9.4.3.1. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00 or 0x01.                               |
| BeaconKeyIndex      | Integer                                                              | 0x01–0xff                                     | The index of the key to be used, as defined in 9.4.3.2. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.                                            |
| DstAddrMode         | Enumeration                                                          | NO_ADDRESS, SHORT, EXTENDED                   | The destination addressing mode, as described in 7.2.1.8, for this primitive and subsequent beacon.                                                                                      |
| DstAddr             | Short address or extended address                                    | As specified by the DstAddrMode parameter     | If sent in response to an MLME-BEACON-REQUEST.indication, the device who sent the beacon request, otherwise the short broadcast address.                                                 |
| BsnSuppression      | Boolean                                                              | TRUE, FALSE                                   | If BeaconType is ENHANCED_BEACON, then if BsnSuppression is TRUE, the EBSN is omitted from the frame and the Sequence Number Suppression field of the Frame Control field is set to one. |



The MLME-BEACON.request primitive may be generated by a higher layer when a beacon or enhanced beacon is to be sent in a nonbeacon-enabled PAN, either on demand, or in response to a MLME-BEACON-REQUEST.indication when *macBeaconAutoRespond* is FALSE.

On receipt of the MLME-BEACON.request primitive, the MAC sublayer entity constructs and transmits a beacon or enhanced beacon depending on the value of the BeaconType parameter.

The MAC sublayer builds an MPDU to transmit from the supplied arguments. The DstAddrMode parameters correspond to the Addressing fields in the Frame Control field, as described in 7.2.1, and are used to construct both the Frame Control and Addressing fields of the MHR.

The frame control and addressing mode options for beacon and enhanced beacons are described in 7.3.1.

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 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 field of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame, as described in 7.2.1.

If BeaconType is set to zero, the *macBsn* is used for the sequence number. If BeaconType is set to one, and BsnSuppression is FALSE, the *macEbsn* is used. If BeaconType is set to one, and BsnSuppression is TRUE, then the sequence number is omitted and the Sequence Number Suppression field of the Frame Control field is set to one.

If BeaconType is set to one, then an enhanced beacon is constructed using the list of IEs in *macEbleList*, and the IE Present field of the Frame Control field is set to one.

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-BEACON.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 MLME-BEACON.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 MLME-BEACON.confirm primitive with a Status of SUCCESS.

If any parameter in the MLME-BEACON.request primitive is not supported or is out of range, the MAC sublayer will issue the MLME-BEACON.confirm primitive with a Status of INVALID\_PARAMETER.

#### 8.2.18.2 MLME-BEACON.confirm

The semantics of this primitive are as follows:

```
MLME-BEACON.confirm      (
                           Status
                           )
```

The primitive parameter is defined in Table 8-42.

**Table 8-42—MLME-BEACON.confirm parameters**

| Name   | Type        | Valid range                                                                 | Description                                                      |
|--------|-------------|-----------------------------------------------------------------------------|------------------------------------------------------------------|
| Status | Enumeration | SUCCESS,<br>CHANNEL_ACCESS_FAILURE,<br>FRAME_TOO_LONG,<br>INVALID_PARAMETER | The result of the attempt to send the beacon or enhanced beacon. |

The MLME-BEACON.confirm primitive is generated by the MAC sublayer entity in response to an MLME-BEACON.request primitive. The MLME-BEACON.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 8.2.18.1.

On receipt of the MLME-BEACON.confirm primitive, the higher layer is notified of the result of its request to transmit a beacon or enhanced beacon. If the transmission attempt was successful, the Status parameter will be set to SUCCESS. Otherwise, the Status parameter will indicate the error. If an enhanced beacon was requested, and the device does not support enhanced beacons, INVALID\_PARAMETER will be returned.

### 8.2.18.3 MLME-BEACON-REQUEST.indication

The MLME-BEACON-REQUEST.indication primitive indicates the receipt of a Beacon Request command or an Enhanced Beacon Request command. It is only available when *macBeaconAutoRespond* is FALSE.

The semantics of this primitive are as follows:

```
MLME-BEACON-REQUEST.indication(
    BeaconType,
    SrcAddrMode,
    SrcAddr,
    DstPanId,
    HeaderIeList,
    PayloadIeList
)
```

The primitive parameters are defined in Table 8-43.

**Table 8-43—MLME-BEACON-REQUEST.indication parameters**

| Name         | Type                              | Valid range                               | Description                                                                                                 |
|--------------|-----------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| BeaconType   | Enumeration                       | BEACON,<br>ENHANCED BEACON                | Indicates if the beacon request was for a beacon or an enhanced beacon.                                     |
| SrcAddr Mode | Enumeration                       | NO_ADDRESS,<br>SHORT, EXTENDED            | The source addressing mode, as described in 7.2.1.10, for device from whom the beacon request was received. |
| SrcAddr      | Short address or extended address | As specified by the SrcAddrMode parameter | The device who sent the beacon request, if present, otherwise the short broadcast address.                  |

**Table 8-43—MLME-BEACON-REQUEST.indication parameters (continued)**

| Name          | Type                                     | Valid range   | Description                                                                                                                 |
|---------------|------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------|
| DstPanId      | Integer                                  | 0x0000–0xffff | The PAN ID contained in the beacon request, or the broadcast PAN ID if PAN ID not present.                                  |
| HeaderIeList  | Set of header IEs as described in 7.4.2  | —             | The header IEs, excluding Termination IEs, to be included in the Beacon frame. If empty, then no header IEs are included.   |
| PayloadIeList | Set of payload IEs as described in 7.4.3 | —             | The payload IEs, excluding Termination IEs, to be included in the Beacon frame. If empty, then no payload IEs are included. |

The MLME-BEACON-REQUEST.indication primitive is generated by the MAC sublayer and issued to a higher layer on receipt of a beacon request or enhanced beacon request. The MLME-BEACON-REQUEST.indication is only generated when *macBeaconAutoRespond* is FALSE.

A higher layer may use the information contained in the MLME-BEACON-REQUEST.indication to construct and send a beacon or enhanced beacon using the MLME-BEACON.request primitive.

## 8.2.19 Primitives for TSCH

### 8.2.19.1 MLME-SET-SLOTFRAME.request

The MLME-SET-SLOTFRAME.request primitive is used to add, delete, or modify a slotframe at the MAC sublayer. The SlotframeHandle is supplied by a higher layer.

The semantics of this primitive are as follows:

```
MLME-SET-SLOTFRAME.request (
    SlotframeHandle,
    Operation,
    Size
)
```

The primitive parameters are defined in Table 8-44.

**Table 8-44—MLME-SET-SLOTFRAME.request parameters**

| Name            | Type        | Valid range         | Description                               |
|-----------------|-------------|---------------------|-------------------------------------------|
| SlotframeHandle | Integer     | 0x00–0xff           | Unique identifier of the slotframe.       |
| Operation       | Enumeration | ADD, DELETE, MODIFY | Operation to perform on the slotframe.    |
| Size            | Integer     | 0x0000–0xffff       | Number of timeslots in the new slotframe. |

An MLME-SET-SLOTFRAME.request may be used by a higher layer to add, delete, or modify a slotframe at the MAC sublayer.

On receipt of an MLME-SET-SLOTFRAME.request, the MLME shall verify the parameters passed with the primitive. If the requested operation is set to ADD, the MLME shall attempt to add an entry into the *macSlotframeTable*. A *macSlotframeTable* entry shall be stored for each slotframe. If the operation is set to DELETE, all parameters except SlotframeHandle and operation shall be ignored, and the slotframe record shall be deleted from the *macSlotframeTable*. If there are links in the slotframe that are being deleted, the links shall be deleted from the MAC layer. If the device is in the middle of using a link in the slotframe that is being updated or deleted, the update shall be postponed until after the link operation completes either through a successful unacknowledged transmission, time-out for receipt of an expected acknowledgment, receipt of an invalid or unacknowledged frame, or transmission of an acknowledgment upon receipt of a valid frame. If the operation is set to MODIFY, it shall attempt to update an existing slotframe record in the table.

### 8.2.19.2 MLME-SET-SLOTFRAME.confirm

The MLME-SET-SLOTFRAME.confirm primitive reports the results of the MLME-SET-SLOTFRAME.request command. The SlotframeHandle is that which was supplied by a higher layer in the prior call to the MLME-SET-SLOTFRAME.request.

The semantics of this primitive are as follows:

```
MLME-SET-SLOTFRAME.confirm (
    SlotframeHandle,
    Status
)
```

The primitive parameters are defined in Table 8-45.

**Table 8-45—MLME-SET-SLOTFRAME.confirm parameters**

| Name             | Type        | Valid range                                                              | Description                                                           |
|------------------|-------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Slotframe Handle | Integer     | 0x00–0xff                                                                | Unique identifier of the slotframe to be added, deleted, or modified. |
| Status           | Enumeration | SUCCESS, INVALID_PARAMETER, SLOTFRAME_NOT_FOUND, MAX_SLOTFRAMES_EXCEEDED | Indicates results of the MLME-SET-SLOTFRAME.request.                  |

The MLME-SET-SLOTFRAME.confirm primitive is generated by the MLME when the MLME-SET-SLOTFRAME.request is completed.

If any of the arguments fail a range check, the Status shall be INVALID\_PARAMETER. If a new slotframe is being added and the *macSlotframeTable* is already full, the Status shall be MAX\_SLOTFRAMES\_EXCEEDED. If an update or deletion is being requested and the corresponding slotframe cannot be found, the Status shall be SLOTFRAME\_NOT\_FOUND. If an add is being requested with a SlotframeHandle corresponding to an existing slotframe, the Status shall be INVALID\_PARAMETER. Otherwise the Status code shall be set to SUCCESS.

### 8.2.19.3 MLME-SET-LINK.request

The MLME-SET-LINK.request primitive requests to add a new link, or delete or modify an existing link at the MAC sublayer.

The semantics of this primitive are as follows:

```
MLME-SET-LINK.request (
    Operation,
    LinkHandle,
    SlotframeHandle,
    Timeslot,
    ChannelOffset,
    TxLink,
    RxLink,
    SharedLink,
    TimekeepingLink,
    PriorityLink,
    LinkType,
    NodeAddrMode,
    NodeAddr
)
```

The primitive parameters are defined in Table 8-46.

**Table 8-46—MLME-SET-LINK.request parameters**

| Name            | Type        | Valid range                              | Description                                                                                                         |
|-----------------|-------------|------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Operation       | Enumeration | ADD_LINK,<br>DELETE_LINK,<br>MODIFY_LINK | Type of link management operation to be performed.                                                                  |
| LinkHandle      | Integer     | 0x0000–0xffff                            | Unique identifier, local to specified slotframe, for the link, as described in Table 8-85.                          |
| SlotframeHandle | Integer     | 0x00–0xff                                | The slotframe handle of the slotframe to which the link is associated.                                              |
| Timeslot        | Integer     | 0x0000–0xffff                            | Timeslot of the link to be added, as described in 6.2.6.                                                            |
| ChannelOffset   | Integer     | As defined in 6.2.6.3                    | The Channel offset of the link.                                                                                     |
| TxLink          | Boolean     | TRUE, FALSE                              | Set to TRUE if the link is a TX link, otherwise set to FALSE.                                                       |
| RxLink          | Boolean     | TRUE, FALSE                              | Set to TRUE if the link is an RX link, otherwise set to FALSE.                                                      |
| SharedLink      | Boolean     | TRUE, FALSE                              | Set to TRUE if the link is a shared link, otherwise set to FALSE.                                                   |
| TimekeepingLink | Boolean     | TRUE, FALSE                              | Set to TRUE if the link is to be used for clock synchronization, as described 6.5.4, otherwise set to FALSE.        |
| PriorityLink    | Boolean     | TRUE, FALSE                              | Set to TRUE if the link is to be used only for high priority traffic, as described 6.2.5.2, otherwise set to FALSE. |

**Table 8-46—MLME-SET-LINK.request parameters**

| Name         | Type                                       | Valid range                     | Description                                                                                     |
|--------------|--------------------------------------------|---------------------------------|-------------------------------------------------------------------------------------------------|
| LinkType     | Enumeration                                | ADVERTISING,<br>NORMAL          | Set to ADVERTISING if the link is to be used to advertise the network, otherwise set to NORMAL. |
| NodeAddrMode | Enumeration                                | SHORT,<br>EXTENDED              | The addressing mode of the neighbor device connected to the link.                               |
| NodeAddr     | Short<br>address or<br>extended<br>address | As specified by<br>NodeAddrMode | Address of neighbor device connected by the link.                                               |

MLME-SET-LINK.request primitive may be used by the device management layer to add, delete, or modify a link in a slotframe.

When Operation is set to ADD\_LINK, the MAC layer shall attempt to add the link to a new *macLinkTable* associated with the indicated slotframe. When Operation is set to DELETE\_LINK, all parameters except LinkHandle and SlotframeHandle shall be ignored, and the indicated link shall be deleted from the associated *macLinkTable*. When Operation is set to MODIFY\_LINK, the MAC layer shall attempt to update the indicated link. If the link is currently in use, the delete or modify operation shall be postponed until the link operation completes, either through a successful unacknowledged transmission, time-out for receipt of an expected acknowledgment, receipt of an invalid or unacknowledged frame, or transmission of an acknowledgment upon receipt of a valid frame. Upon completion, the result of the operation shall be reported through the corresponding MLME-SET-LINK.confirm primitive.

If the TxLink is TRUE and SharedLink is TRUE, then the device shall back off according to the method described in 6.2.5.2.

If TimekeepingLink is TRUE and RxLink is TRUE, then a neighbor is to be used for timing synchronization.

If LinkType is set to ADVERTISE, the links may be used to send Enhanced Beacon frames as the result of the MAC receiving a MLME-BEACON.request.

#### 8.2.19.4 MLME-SET-LINK.confirm

The MLME-SET-LINK.confirm primitive indicates the result of add, delete, or modify link operation. The LinkHandle and SlotframeHandle are those that were supplied by a higher layer in the prior call to MLME-SET-LINK.request

The semantics of this primitive are as follows:

```
MLME-SET-LINK.confirm
(
    LinkHandle,
    SlotframeHandle,
    Status
)
```

The primitive parameters are defined in Table 8-47.

**Table 8-47—MLME-SET-LINK.confirm parameters**

| Name            | Type        | Valid range                                                           | Description                                                                                |
|-----------------|-------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| LinkHandle      | Integer     | 0x0000–0xffff                                                         | Unique identifier, local to specified slotframe, for the link, as described in Table 8-85. |
| SlotframeHandle | Integer     | 0x00–0xff                                                             | The slotframe handle of the slotframe to which the link is associated.                     |
| Status          | Enumeration | SUCCESS,<br>INVALID_PARAMETER,<br>UNKNOWN_LINK,<br>MAX_LINKS_EXCEEDED | Result of the request operation.                                                           |

The Status of the primitive shall indicate SUCCESS if the operation completed successfully. If any of the arguments fail a range check, the Status shall be INVALID\_PARAMETER. If a new link is being added and the *macLinkTable* is already full, the Status shall be MAX\_LINKS\_EXCEEDED. A *macLinkTable* shall be stored for each link in a slotframe. If an update or deletion is being requested and the corresponding link cannot be found, the Status shall be UNKNOWN\_LINK. If an add is being requested with a LinkHandle corresponding to an existing link, the Status shall be INVALID\_PARAMETER.

#### 8.2.19.5 MLME-TSCH-MODE.request

The MLME-TSCH-MODE.request requests to put the MAC into or out of the TSCH mode.

The semantics of this primitive are as follows:

MLME-TSCH-MODE.request ( TschMode,  
TschCca  
)

The primitive parameters are defined in Table 8-48.

**Table 8-48—MLME-TSCH-MODE.request parameters**

| Name     | Type        | Valid range | Description                                                    |
|----------|-------------|-------------|----------------------------------------------------------------|
| TschMode | Enumeration | ON, OFF     | Used to indicate if the TSCH mode is to be started or stopped. |
| TschCca  | Enumeration | ON, OFF     | Used to indicate that CCA is to be used for transmission.      |

The MLME-TSCH-MODE.request may be generated by a higher layer after the device has received advertisements from the network and is synchronized to a network, i.e., in response to an MLME-BEACON-NOTIFY.indication.

Upon receipt of the request with TschMode set to ON, the MAC shall start operating its TSCH state machine using slotframes and links already contained in its *macSlotframeTable* and *macLinkTable* MAC PIB attributes. To successfully complete this request the device shall already be synchronized to a network. The MAC shall stop using slotframes and links upon receipt of the request with TschMode set to off.

**8.2.19.6 MLME-TSCH-MODE.confirm**

The MLME-TSCH-MODE.confirm primitive reports the result of the MLME-TSCH-MODE.request primitive.

The semantics of this primitive are as follows:

```
MLME-TSCH-MODE.confirm    (
                           TschMode,
                           Status
                           )
```

The primitive parameters are defined in Table 8-49.

**Table 8-49—MLME-TSCH-MODE.confirm parameters**

| Name     | Type        | Valid range      | Description                                               |
|----------|-------------|------------------|-----------------------------------------------------------|
| TschMode | Enumeration | ON, OFF          | Used to indicate if the TSCH mode was started or stopped. |
| Status   | Enumeration | SUCCESS, NO_SYNC | Indicates results of the MLME-TSCH-MODE.request.          |

The MLME-TSCH-MODE.confirm is generated by the MAC layer to indicate completion of the corresponding request. If the corresponding request was to turn on its TSCH state machine, but the MAC layer has not been synchronized to a network, the Status shall be NO\_SYNC. Otherwise, the Status shall be SUCCESS.

If the corresponding request was to turn off its TSCH state machine, the Status shall be SUCCESS, and the MAC layer shall stop its TSCH state machine if in use.

**8.2.19.7 MLME-KEEP-ALIVE.request**

The MLME-KEEP-ALIVE.request primitive requests that frames be sent to a device with a minimum period.

The semantics of this primitive are as follows:

```
MLME-KEEP-ALIVE.request  (
                           DstAddr,
                           DstAddrMode,
                           KeepAlivePeriod
                           )
```

The primitive parameters are defined in Table 8-50.

Upon receipt of the request, the MAC layer shall monitor for frames sent to the destination node specified in the DstAddr parameter. If no frame is sent to the destination node in KeepAlivePeriod, the MAC shall send a Data frame with no payload and the AR field set to request acknowledgment to its time source neighbor and use the resulting Enh-Ack frame to perform acknowledgment based synchronization.



**Table 8-50—MLME-KEEP-ALIVE.request parameters**

| Name            | Type                              | Valid range                               | Description                                                                             |
|-----------------|-----------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------|
| DstAddr         | Short address or extended address | As specified by the DstAddrMode parameter | Address of the neighbor device with which to maintain timing.                           |
| DstAddrMode     | Enumeration                       | SHORT, EXTENDED                           | The destination addressing mode of the device to monitor.                               |
| KeepAlivePeriod | Integer                           | 0x0001–0xffff                             | Period in timeslots after which a frame is sent if no frames have been sent to DstAddr. |

**8.2.19.8 MLME-KEEP-ALIVE.confirm**

The MLME-KEEP-ALIVE.confirm primitive reports the results of a request that frames be sent to a device with a minimum period.

The semantics of this primitive are as follows:

```
MLME-KEEP-ALIVE.confirm      (
                                Status
                                )
```

The primitive parameter is defined in Table 8-51.

**Table 8-51—MLME-KEEP-ALIVE.confirm parameters**

| Name   | Type        | Valid range                | Description                                       |
|--------|-------------|----------------------------|---------------------------------------------------|
| Status | Enumeration | SUCCESS, INVALID_PARAMETER | Indicates results of the MLME-KEEP-ALIVE.request. |

The MAC layer shall generate MLME-KEEP-ALIVE.confirm to acknowledge that it received MLME-KEEP-ALIVE request. If the DstAddr of the MLME-KEEP-ALIVE.request does not exist in the devices neighbor table and is not 0xffff, the Status of the primitive shall indicate INVALID\_PARAMETER. Otherwise it shall return SUCCESS.

**8.2.20 Primitives for DSME GTS management**

The MLME-SAP DSME GTS management primitives define how DSME GTSs are requested and maintained. A device wishing to use these primitives and DSME GTSs in general will already be tracking the beacons of its coordinator.

**8.2.20.1 MLME-DSME-GTS.request**

This primitive allows a DSME-enabled device to perform one of the following actions:

- Allocation of new DSME GTS
- Deallocation of an existing DSME GTS
- Notification of a duplicated allocation

- Reduction of an existing DSME GTSS
- Restart of an existing DSME GTSS

The semantics of this primitive are as follows:

MLME-DSME-GTS.request (

DeviceAddress,  
ManagementType,  
Direction,  
PrioritizedChannelAccess,  
NumSlot,  
PreferredSuperframeId,  
PreferredSlotId,  
DsmeSubSpecification,  
SecurityLevel,  
KeyIdMode,  
KeySource,  
KeyIndex,  
AllocationOrder

)

The primitive parameters are defined in Table 8-52.

**Table 8-52—MLME-DSME-GTS.request parameters**

| Name                     | Type        | Valid range                                                                                    | Description                                                                                                                     |
|--------------------------|-------------|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddress            | Integer     | 0x0000–0xffffd                                                                                 | The address of the neighboring device to request the management of DSME GTSS.                                                   |
| ManagementType           | Enumeration | DEALLOCATION,<br>ALLOCATION,<br>DUPLICATE,<br>ALLOCATION,<br>REDUCE,<br>RESTART,<br>EXPIRATION | The type of the management request.                                                                                             |
| Direction                | Enumeration | TX, RX                                                                                         | The direction of DSME GTSS.                                                                                                     |
| PrioritizedChannelAccess | Enumeration | LOW, HIGH                                                                                      | The priority level.                                                                                                             |
| NumSlot                  | Integer     | 0x00–0xff                                                                                      | The number of slots to be requested for allocation. This parameter is ignored if the ManagementType is not ALLOCATION.          |
| PreferredSuperframeId    | Integer     | 0x0000–0xffff                                                                                  | The index of the preferred superframe in a multi-superframe. This parameter is ignored if the ManagementType is not ALLOCATION. |
| PreferredSlotId          | Integer     | 0x00–0x0e                                                                                      | The index of the preferred slot in the preferred superframe. This parameter is ignored if the ManagementType is not ALLOCATION. |

**Table 8-52—MLME-DSME-GTS.request parameters (continued)**

| Name                 | Type          | Valid range              | Description                                                                                                                                                                                                                                                                                                                         |
|----------------------|---------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DsmeSabSpecification | —             | As defined in Table 8-53 | If the ManagementType is ALLOCATION, this parameter contains the information of the current DSME GTS allocation and slot availability in one hop neighborhood of the requesting device. If the ManagementType is not 0b001, this parameter indicates the DSME GTSs to deallocate, notify duplicated allocation, reduce, or restart. |
| SecurityLevel        | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                           |
| KeyIdMode            | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                           |
| KeySource            | Set of octets | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                           |
| KeyIndex             | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                                                                                                                                                                                                           |
| AllocationOrder      | Integer       | 0x00–0x08                | As defined in 7.5.13.                                                                                                                                                                                                                                                                                                               |

The elements of the DsmeSabSpecification are given in Table 8-53.

**Table 8-53—Elements of a DsmeSabSpecification**

| Name                  | Type    | Valid range   | Description                                                                                                                                                     |
|-----------------------|---------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DsmeSabSubBlockLength | Integer | 0x00–0xff     | The length of the DSME SAB sub-block in units in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode.                            |
| DsmeSabSubBlockIndex  | Integer | 0x0000–0xffff | The beginning of the DSME SAB sub-block in the entire SAB as illustrated in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode. |
| DsmeSabSubBlock       | Integer | —             | The sub-block of the DSME SAB as described in Figure 6-53 for channel adaptation mode or in Figure 6-55 for channel hopping mode.                               |

The MLME-DSME-GTS.request primitive is generated by the higher layer of a device and issued to its MLME to request the allocation of new DSME GTSs or to request the deallocation, duplicated allocation notification, reduce, or change of existing DSME GTSs.

On receipt of the MLME-DSME-GTS.request primitive, the MLME of the device shall send a DSME GTS Request command, as described in 7.5.14, to the DeviceAddress. The ManagementType, Direction, PrioritizedChannelAccess, NumSlot, PreferredSuperframeId, PreferredSlotId, DsmeSabSubBlockIndex, and DsmeSabSubBlockLength parameters shall be contained in the corresponding fields of the command.

The DSME SAB Sub-block field shall contain a bitmap of a sub-block of *macDsmeSab*. The *DsmeSabSubBlockIndex* and *DsmeSabSubBlockLength* indicate the bitmap of which sub-block shall be contained in the command.

If *macShortAddress* is equal to 0xffff or 0xffff, the Source device is not permitted to request a DSME GTS allocation. In this case, the MLME issues the MLME-DSME-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 shall set the Security Enabled field of the Frame Control field to one. The MAC sublayer shall perform outgoing processing on the frame based on *SecurityLevel*, *KeyIdMode*, *KeySource*, and *KeyIndex* parameters, as described in 9.2.1. If any error occurs during outgoing frame processing, the MLME shall discard the frame and issue the MLME-DSME-GTS.confirm primitive with the Status returned by outgoing frame processing.

If the DSME GTS Request command cannot be sent due to the channel condition, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a Status of CHANNEL\_ACCESS\_FAILURE.

If the MLME successfully transmits a DSME GTS request command, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a Status of NO\_ACK.

If the DSME GTS Response command from the destination device is not received within the expected time, the MLME of the source device shall notify the next higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a Status of NO\_DATA.

#### 8.2.20.2 MLME-DSME-GTS.indication

This primitive reports the reception of a DSME GTS request command.

The semantics of this primitive are as follows:

```
MLME-DSME-GTS.indication (
    DeviceAddress,
    ManagementType,
    Direction,
    PrioritizedChannelAccess,
    NumSlot,
    PreferredSuperframeId,
    PreferredSlotId,
    DsmeSabSpecification,
    AllocationOrder
)
```

The primitive parameters are defined in Table 8-54.

This primitive is generated by the MLME of a device and issued to its next higher layer upon the reception of a DSME GTS Request command.

On receipt of the MLME-DSME-GTS.indication primitive, the higher layer is notified of the reception of a DSME GTS request.

**Table 8-54—MLME-DSME-GTS.indication parameters**

| Name                     | Type          | Valid range                                                                                     | Description                                                                                                                                                                                                                                                                                                                              |
|--------------------------|---------------|-------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddress            | Short address | 0x0000–0xffffd                                                                                  | The address of the device that has transmitted the received DSME GTS request command.                                                                                                                                                                                                                                                    |
| ManagementType           | Enumeration   | DEALLOCATION,<br>ALLOCATION,<br>DUPLICATED_<br>ALLOCATION,<br>REDUCE,<br>RESTART,<br>EXPIRATION | The type of the management request.                                                                                                                                                                                                                                                                                                      |
| Direction                | Enumeration   | TX, RX                                                                                          | The direction of DSME GTSs.                                                                                                                                                                                                                                                                                                              |
| PrioritizedChannelAccess | Enumeration   | LOW, HIGH                                                                                       | The priority level.                                                                                                                                                                                                                                                                                                                      |
| NumSlot                  | Integer       | 0x00–0xff                                                                                       | The number of slots to be requested for allocation. This parameter is ignored if the ManagementType is not ALLOCATION.                                                                                                                                                                                                                   |
| PreferredSuperframeId    | Integer       | 0x0000–0xffff                                                                                   | The index of the preferred superframe in a multi-superframe. This parameter is ignored if the ManagementType is not ALLOCATION.                                                                                                                                                                                                          |
| PreferredSlotId          | Integer       | 0x00–0x0e                                                                                       | The index of the preferred slot in the preferred superframe. This parameter is ignored if the ManagementType is not ALLOCATION.                                                                                                                                                                                                          |
| DsmeSabSpecification     | —             | As defined in Table 8-53                                                                        | If the ManagementType is ALLOCATION, this parameter contains the information of the current DSME GTS allocation and slot availability in one hop neighborhood of the requesting device. If the ManagementType is not ALLOCATION, this parameter indicates the DSME GTSs to deallocate, notify duplicated allocation, reduce, or restart. |
| AllocationOrder          | Integer       | 0x00–0x08                                                                                       | As defined in 7.5.13.                                                                                                                                                                                                                                                                                                                    |

**8.2.20.3 MLME-DSME-GTS.response**

This primitive allows the next higher layer of a device to respond to the MLME-DSME-GTS.indication primitive.

The semantics of this primitive are as follows:

MLME-DSME-GTS.response (

- DeviceAddress,
- ManagementType,
- Direction,
- PrioritizedChannelAccess,
- ChannelOffset,
- DsmeSabSpecification,
- AllocationOrder,
- BitIndex,
- SuperframeId,

SlotId  
ChannelIndex,  
Status  
)

The primitive parameters are defined in Table 8-55.

**Table 8-55—MLME-DSME-GTS.response parameters**

| Name                      | Type          | Valid range                                                                                     | Description                                                                                                       |
|---------------------------|---------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| DeviceAddress             | Short address | 0x0000–0xffffd                                                                                  | The address of the device that has transmitted the received DSME GTS request command.                             |
| ManagementType            | Enumeration   | DEALLOCATION,<br>ALLOCATION,<br>DUPLICATED_<br>ALLOCATION,<br>REDUCE,<br>RESTART,<br>EXPIRATION | The type of the management request.                                                                               |
| Direction                 | Enumeration   | TX, RX                                                                                          | The direction of DSME GTSs.                                                                                       |
| PrioritizedChannel Access | Enumeration   | LOW, HIGH                                                                                       | The priority level.                                                                                               |
| ChannelOffset             | Integer       | 0x0000–0xffff                                                                                   | This parameter specifies the offset value of Hopping Sequence.                                                    |
| DsmeSabSpecification      | —             | As defined in Table 8-53                                                                        | This parameter indicates the DSME GTSs to allocate, deallocate, notify duplicated allocation, reduce, or restart. |
| AllocationOrder           | Integer       | 0x00–0x08                                                                                       | As defined in 7.5.13.                                                                                             |
| BiIndex                   | Integer       | 0x00–0xff                                                                                       | As defined in 7.5.13.                                                                                             |
| SuperframeId              | Integer       | 0x0000–0xffff                                                                                   | As defined in 7.5.13.                                                                                             |
| SlotId                    | Integer       | 0x00–0x0e                                                                                       | As defined in 7.5.13.                                                                                             |
| ChannelIndex              | Integer       | 0x0000–0xffff                                                                                   | As defined in 7.5.13.                                                                                             |
| Status                    | Enumeration   | SUCCESS, DENIED,<br>INVALID_PARAMETER                                                           | The status of the DSME-GTS request.                                                                               |

The MLME-DSME-GTS.response primitive can be generated by the next higher layer and issued to its MLME to respond to the allocation, deallocation, duplicated allocation notification, reduce, or restart of DSME GTS.

On receipt of the MLME-DSME-GTS.response primitive, the MLME of the device shall generate a DSME GTS Response command, as described in 7.5.15. The ManagementType, Direction, PrioritizedChannelAccess, ChannelOffset, and DsmeSabSpecification parameters shall be contained in the corresponding fields of the command. The DSME GTS Destination Address field shall be set to the value of DeviceAddress parameter. The Status field of the command shall be set to zero if the Status parameter value is SUCCESS. The Status field shall be set to one if the Status parameter value is DENIED. The Status field shall be set to two if the Status parameter value is INVALID\_PARAMETER. Then the device shall send the

DSME GTS Response command to its one-hop neighbors with the destination address set to the short broadcast address.

#### 8.2.20.4 MLME-DSME-GTS.confirm

This primitive reports the results of a request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME GTSs to the higher layer of the device.

The semantics of this primitive are as follows:

```
MLME-DSME-GTS.confirm (
    DeviceAddress,
    ManagementType,
    Direction,
    PrioritizedChannelAccess,
    ChannelOffset,
    DsmeSabSpecification,
    Status,
    AllocationOrder
    BiIndex,
    SuperframeId,
    SlotId,
    ChannelIndex
)
```

The primitive parameters are defined in Table 8-56.

**Table 8-56—MLME-DSME-GTS.confirm parameters**

| Name                      | Type          | Valid range                                                                                    | Description                                                                                                       |
|---------------------------|---------------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| DeviceAddress             | Short address | 0x0000–0xffffd                                                                                 | The address of the device that has transmitted the received DSME-GTS Response command.                            |
| ManagementType            | Enumeration   | DEALLOCATION,<br>ALLOCATION,<br>DUPLICATED<br>ALLOCATION,<br>REDUCE,<br>RESTART,<br>EXPIRATION | The type of the management request.                                                                               |
| Direction                 | Enumeration   | TX, RX                                                                                         | The direction of DSME GTSs.                                                                                       |
| PrioritizedChannel Access | Enumeration   | LOW, HIGH                                                                                      | The priority level.                                                                                               |
| ChannelOffset             | Integer       | 0x0000–0xffff                                                                                  | This parameter specifies the offset value of Hopping Sequence.                                                    |
| DsmeSabSpecification      | —             | As defined in Table 8-53                                                                       | This parameter indicates the DSME GTSs to allocate, deallocate, notify duplicated allocation, reduce, or restart. |
| AllocationOrder           | Integer       | 0x00–0x08                                                                                      | As defined in 7.5.13.                                                                                             |
| BiIndex                   | Integer       | 0x00–0xff                                                                                      | As defined in 7.5.13.                                                                                             |

**Table 8-56—MLME-DSME-GTS.confirm parameters (continued)**

| Name         | Type        | Valid range                                                                             | Description                         |
|--------------|-------------|-----------------------------------------------------------------------------------------|-------------------------------------|
| SuperframeId | Integer     | 0x0000–0xffff                                                                           | As defined in 7.5.13.               |
| SlotId       | Integer     | 0x00–0x0e                                                                               | As defined in 7.5.13.               |
| ChannelIndex | Integer     | 0x00–0x1f                                                                               | As defined in 7.5.13.               |
| Status       | Enumeration | SUCCESS, DENIED,<br>INVALID_PARAMETER,<br>NO_ACK_NO_DATA,<br>CHANNEL_ACCESS_<br>FAILURE | The status of the DSME-GTS request. |

On receipt of a DSME GTS Response command, the device shall check the DSME GTS Destination Address field of the command.

If the DSME GTS Destination Address in this command is the same as *macShortAddress*, the device shall check the Status field of the command. If the value of the Status field in the command is zero (SUCCESS), the device shall generate a DSME GTS Notify command, as described in The ManagementType, Direction, PrioritizedChannelAccess, ChannelOffset, and DsmeSabSpecification parameters shall be contained in the corresponding fields of the command. The DSME GTS Destination Address field shall be set to the value of DeviceAddress parameter. The Status field of the command shall be set to zero if the Status parameter value is SUCCESS. The Status field shall be set to one if the Status parameter value is DENIED. The Status field shall be set to two if the Status parameter value is INVALID\_PARAMETER. Then the device shall broadcast the DSME GTS notify command to its one-hop neighbor.

Also, the MLME of the device shall notify the higher layer with the result of the request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME GTSs. If the value of the Status field in the command is zero, the Status in this primitive shall be set to SUCCESS. If the value of the Status field is one, the Status parameter shall be set to DENIED. If the value of the Status field is two, the Status parameter shall be set to INVALID\_PARAMETER.

If the DSME GTS Destination Address field is different from *macShortAddress*, the device shall update its DSME SAB according to the DSME GTS SAB Specification of the received command.

On receipt of the MLME-DSME-GTS.confirm primitive, the higher layer is notified of the result of its request to notify duplicated allocation, reduce, or restart a DSME GTS.

### 8.2.21 Primitives for reporting the link status

#### 8.2.21.1 MLME-DSME-LINK-REPORT.request

The MLME-DSME-LINK-REPORT.request primitive is used to request that a device start monitoring link quality statistic and periodically report the statistic results to the destination device.

The semantics of this primitive are as follows:

```
MLME-DSME-LINK-REPORT.request(
    DstAddr,
    ReportPeriod
)
```



The primitive parameters are defined in Table 8-57.

**Table 8-57—MLME-DSME-LINK-REPORT.request parameters**

| Name         | Type    | Valid range       | Description                                                                                                                                                                                                                                              |
|--------------|---------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DstAddr      | Integer | 0x0000–0xffff     | Short address of the Destination device to which the DSME-link report request is intended.                                                                                                                                                               |
| ReportPeriod | Integer | 0x000000–0xffffff | The time interval between two DSME Link Status Report commands is defined as $\text{ReportPeriod} \times aBaseSuperframeDuration \times 2^{MO}$ symbols. If the parameter equals to 0x000000, DSME Link Status Report command is not allowed to be sent. |

On receipt of MLME-DSME-LINK-REPORT.request primitive by a device, the MLME of the device attempts to generate a DSME Link Status Report command as described in 8.2.21 with the information contained in this primitive, and if successful, sends it to the destination device according to the DstAddr parameter.

If the DSME Link Status Report command cannot be sent due to a CSMA-CA algorithm failure, the MLME shall issue the MLME-DSME-LINK-REPORT.confirm primitive with a Status of CHANNEL\_ACCESS\_FAILURE.

If the MLME successfully transmits a DSME Link Status Report command, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-LINK-REPORT.confirm primitive with a Status of NO\_ACK.

If the DSME Link Status Report command has been acknowledged, the device shall send another DSME Link Report command again in the interval defined in the parameter ReportPeriod.

#### 8.2.21.2 MLME-DSME-LINK-REPORT.indication

The MLME-DSME-LINK-REPORT.indication primitive is generated by the MAC sublayer and issued to the next higher layer on receipt of a DSME Link Status Report command.

The semantics of this primitive are as follows:

```
MLME-DSME-LINK-REPORT.indication(
    DstAddr,
    LinkReportSpecification
)
```

Table 8-58 specifies the parameters for the MLME-DSME-LINK-REPORT.indication request primitive.

The usage of the MLME-DSME-LINK-REPORT.indication primitive by the next higher layer is beyond the scope of this document.

**Table 8-58—MLME-DSME-LINK-REPORT.indication parameters**

| Name                    | Type                      | Valid range          | Description                                                                                |
|-------------------------|---------------------------|----------------------|--------------------------------------------------------------------------------------------|
| DstAddr                 | Integer                   | 0x0000–0xffff        | Short address of the Destination device to which the DSME link report request is intended. |
| LinkReportSpecification | Link Report Specification | As defined in 7.5.21 | As defined in 7.5.21.                                                                      |

**8.2.21.3 MLME-DSME-LINK-REPORT.confirm**

The MLME-DSME-LINK-REPORT.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-DSME-LINK-REPORT.request primitive.

The semantics of this primitive are as follows:

```
MLME-DSME-LINK-REPORT.confirm (
    Status
)
```

Table 8-59 specifies the parameters for the MLME-DSME-LINK-REPORT.confirm primitive.

**Table 8-59—MLME-DSME-LINK-REPORT.confirm parameters**

| Name   | Type        | Valid range                                | Description                              |
|--------|-------------|--------------------------------------------|------------------------------------------|
| Status | Enumeration | CHANNEL_ACCESS_FAILURE,<br>NO_ACK, SUCCESS | The status of starting DSME link report. |

The MLME-DSME-LINK-REPORT.confirm primitive returns a Status of either SUCCESS, indicating the MAC sublayer has started reporting its statistic results periodically, or the appropriate error code.

On receipt of the MLME-DSME-LINK-REPORT.confirm primitive by a device, the next higher layer is notified of the result of its request to start reporting link status in the PAN. If the request was successful, the Status parameter shall be set to SUCCESS. Otherwise, the Status parameter shall indicate the error.

**8.2.22 Operating parameter change primitives**

These primitives support the coordination of a change in PHY operating parameters among peer devices.

**8.2.22.1 MLME-PHY-OP-SWITCH.request**

The MLME-PHY-OP-SWITCH.request primitive is used by a device to instruct a second device to switch PHY operating parameters, including channel, band, PHY type, or other parameters specific to a PHY.

The semantics of this primitive are as follows:

```
MLME-PHY-OP-SWITCH.request (
    DeviceAddrMode,
    DeviceAddr,
    PhyParameterList,
    TxIndirect,
    TargetTime,
    SignalMethod,
    RepeatCount,
    RepeatInterval,
    SecurityLevel,
    KeyIdMode,
    KeySource,
    KeyIndex
)
```

The primitive parameters are defined in Table 8-60.

**Table 8-60—MLME-PHY-OP-SWITCH.request parameters**

| Name             | Type                                  | Valid range                                  | Description                                                                                                                                                                                                         |
|------------------|---------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeviceAddrMode   | Enumeration                           | SHORT, EXTENDED                              | The addressing mode of the device being instructed to change its operating parameters.                                                                                                                              |
| DeviceAddr       | Short address or extended address     | As specified by the DeviceAddrMode parameter | The address of the device being instructed to change its operating parameters.                                                                                                                                      |
| PhyParameterList | List of PHY PIB attributes and values | As defined in 11.3                           | A list of the PHY PIB attribute names and values representing the PHY operating parameters to be changed.                                                                                                           |
| TxIndirect       | Boolean                               | TRUE, FALSE                                  | When the TxIndirect parameter is set to TRUE, the multipurpose frame shall be sent using indirect transmission. When the parameter is set to FALSE, the multipurpose frame shall be sent using direct transmission. |
| TargetTime       | Integer                               | $0-(2^{32}-1)$                               | The time, in microseconds, from the current time that the PHY operating parameter switch is to be carried out.                                                                                                      |
| SignalMethod     | Enumeration                           | USE_MP, USE_BEACON                           | The method to be used to signal intended switch.                                                                                                                                                                    |
| RepeatCount      | Integer                               | 0–127                                        | Number of times that the notification containing the PHY Parameter Change IE is repeated.                                                                                                                           |
| RepeatInterval   | Integer                               | 0–0xffff                                     | The time, in microseconds, to delay between repeated transmissions.                                                                                                                                                 |

**Table 8-60—MLME-PHY-OP-SWITCH.request parameters (continued)**

| Name          | Type          | Valid range              | Description               |
|---------------|---------------|--------------------------|---------------------------|
| SecurityLevel | Integer       | As defined in Table 8-75 | As defined in Table 8-75. |
| KeyIdMode     | Integer       | As defined in Table 8-75 | As defined in Table 8-75. |
| KeySource     | Set of octets | As defined in Table 8-75 | As defined in Table 8-75. |
| KeyIndex      | Integer       | As defined in Table 8-75 | As defined in Table 8-75. |

On receipt of the MLME-PHY-OP-SWITCH.request primitive, the MLME initiates the PHY parameter change notification procedure, as defined in 6.10.

If the device is the PAN coordinator of a beacon-enabled PAN that is using enhanced beacons, and the SignalMethod parameter value is USE\_BEACON, the method described in 6.10.1 shall be initiated.

If the SignalMethod parameter value is USE\_MP indicating the use of a multipurpose frame, the method described in 6.10.2 shall be initiated. The RepeatInterval parameter value should be greater than the time required to complete a transmission, acknowledgment, and possible retransmissions.

This primitive returns a Status of either SUCCESS, if the PHY parameter change notification procedure has been completed, or the appropriate Status parameter value indicating the reason for the request failure.

If the SignalMethod parameter in the request primitive is USE\_BEACON and the device is a PAN coordinator in a beacon-enabled PAN that is not using enhanced beacon frames, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of UNSUPPORTED\_FEATURE.

If the SignalMethod parameter in the request primitive is USE\_BEACON and the device is not a PAN coordinator in a beacon-enabled PAN, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of INVALID\_PARAMETER.

If the SignalMethod parameter value is USE\_MP and the device does not support the use of multipurpose frames, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of UNSUPPORTED\_FEATURE.

If the SignalMethod parameter value is USE\_MP, the RepeatCount parameter value in the request primitive is greater than zero, and the RepeatInterval value is not greater than zero, the MLME-PHY-OP-SWITCH.confirm primitive shall return with a Status of INVALID\_PARAMETER.

#### 8.2.22.2 MLME-PHY-OP-SWITCH.indication

The MLME-PHY-OP-SWITCH.indication primitive is used to indicate the reception of a multipurpose frame with a PHY Parameter Change IE and a Operating Mode Description IE.

The semantics of this primitive are as follows:

```
MLME-PHY-OP-SWITCH.indication (
    DeviceAddrMode,
    DeviceAddress,
    PhyParameterList,
    TargetTime,
    NotificationTime,
```

LocalTime,  
SecurityLevel,  
KeyIdMode,  
KeySource,  
KeyIndex  
)

The primitive parameters are defined in Table 8-61.

**Table 8-61—MLME-PHY-OP-SWITCH.indication parameters**

| Name             | Type                                  | Valid range                                  | Description                                                                                                    |
|------------------|---------------------------------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| DeviceAddrMode   | Enumeration                           | SHORT, EXTENDED                              | The addressing mode of the device that transmitted the channel switch notification command.                    |
| DeviceAddress    | —                                     | As specified by the DeviceAddrMode parameter | The address of the device that transmitted the channel switch notification command.                            |
| PhyParameterList | List of PHY PIB attributes and values | As described in 11.3                         | A list of the PHY PIB attribute names and values representing the PHY operating parameters to be changed.      |
| TargetTime       | Integer                               | $0-(2^{32}-1)$                               | The time, in microseconds, from the current time that the PHY operating parameter switch is to be carried out. |
| NotificationTime | Integer                               | 0–0xffff                                     | Value of the Notification Time field of the received PHY Parameter Change IE.                                  |
| LocalTime        | Integer                               | Implementation-dependent                     | The time of reception of the multipurpose frame containing the IEs in the local device time reference.         |
| SecurityLevel    | Integer                               | As defined in Table 8-77                     | As defined in Table 8-77.                                                                                      |
| KeyIdMode        | Integer                               | As defined in Table 8-77                     | As defined in Table 8-77.                                                                                      |
| KeySource        | Set of octets                         | As defined in Table 8-77                     | As defined in Table 8-77.                                                                                      |
| KeyIndex         | Integer                               | As defined in Table 8-77                     | As defined in Table 8-77.                                                                                      |

On receipt of the MLME-PHY-OP-SWITCH.request primitive, the MLME initiates the PHY parameter change notification procedure, as defined in 6.10.

If the device is the PAN coordinator of a beacon-enabled PAN that is using enhanced beacons, and the SignalMethod parameter value is USE\_BEACON, the method described in 6.10.1 shall be initiated.

If the SignalMethod parameter value is USE\_MP indicating the use of a multipurpose frame, the method described in 6.10.2 shall be initiated. The RepeatInterval parameter value should be greater than the time required to complete a transmission, acknowledgment, and possible retransmissions.

**8.2.22.3 MLME-PHY-OP-SWITCH.confirm**

The MLME-PHY-OP-SWITCH.confirm primitive is used to inform the next higher layer of the initiating device whether the channel switching notification has completed successfully.

The semantics of this primitive are as follows:

```
MLME-PHY-OP-SWITCH.confirm (
    DeviceAddrMode,
    DeviceAddress,
    Status
)
```

The primitive parameters are defined in Table 8-62.

**Table 8-62—MLME-PHY-OP-SWITCH.confirm parameters**

| Name           | Type                              | Valid range                                                                                                                                                                                                                    | Description                                                                       |
|----------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| DeviceAddrMode | Enumeration                       | SHORT, EXTENDED                                                                                                                                                                                                                | The addressing mode given in the request primitive.                               |
| DeviceAddress  | Short address or extended address | As specified by the DeviceAddrMode parameter                                                                                                                                                                                   | The address of the device given in the request primitive.                         |
| Status         | Enumeration                       | SUCCESS,<br>TRANSACTION_OVERFLOW,<br>TRANSACTION_EXPIRED,<br>NO_ACK,<br>CHANNEL_ACCESS_FAILURE,<br>COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_SECURITY,<br>INVALID_PARAMETER,<br>UNSUPPORTED_FEATURE | The status of the attempt to transmit the channel switching notification command. |

This primitive returns a Status of either SUCCESS, if the PHY parameter change notification procedure has been completed, or the appropriate Status parameter value indicating the reason for the request failure.

If the SignalMethod parameter in the request primitive is USE\_BEACON and the device is a PAN coordinator in a beacon-enabled PAN that is not using enhanced beacon frames, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of UNSUPPORTED\_FEATURE.

If the SignalMethod parameter in the request primitive is USE\_BEACON and the device is not a PAN coordinator in a beacon-enabled PAN, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of INVALID\_PARAMETER.

If the SignalMethod parameter value is USE\_MP and the device does not support the use of multipurpose frames, the MLME-PHY-OP-SWITCH.confirm primitive shall return a Status of UNSUPPORTED\_FEATURE.

If the SignalMethod parameter value is USE\_MP, the RepeatCount parameter value in the request primitive is greater than zero, and the RepeatInterval value is not greater than zero, the MLME-PHY-OP-SWITCH.confirm primitive shall return with a Status of INVALID\_PARAMETER.

### 8.2.23 TMCTP DBS allocation primitives

#### 8.2.23.1 MLME-DBS.request

The MLME-DBS.request primitive is used when a TMCTP-child PAN coordinator requests the allocation or deallocation of a DBS and a channel to its TMCTP-parent PAN coordinator including the SPC.

The semantics of this primitive are as follows:

```
MLME-DBS.request
(
  RequesterCoordAddr,
  RequestType,
  DbsLength,
  NumberOfDescendents,
  SecurityLevel,
  KeyIdMode,
  KeySource,
  KeyIndex
)
```

The primitive parameters are defined in Table 8-63.

**Table 8-63—MLME-DBS.request parameters**

| Name                | Type          | Valid range              | Description                                                                                                                                            |
|---------------------|---------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| RequesterCoordAddr  | Short address | 0x0000–0xffff            | The address of the source requester PAN coordinator.                                                                                                   |
| RequestType         | Enumeration   | ALLOCATION, DEALLOCATION | Indicates if the request is for allocation or deallocation of a TMCTP DBS.                                                                             |
| DbsLength           | Integer       | 0x00–0xff                | Number of BOP slots being requested for the DBS.                                                                                                       |
| NumberOfDescendents | Integer       | 0x00–0xff                | The actual or expected number of descendant PAN coordinators. Set as zero if the PAN coordinator is not clear about how many descendants it will have. |
| SecurityLevel       | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                              |
| KeyIdMode           | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                              |
| KeySource           | Set of octets | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                              |
| KeyIndex            | Integer       | As defined in Table 8-75 | As defined in Table 8-75.                                                                                                                              |

On receipt of an MLME-DBS.request primitive, the MLME generates a DBS request command, as described in 7.5.23.

The SecurityLevel parameter specifies the level of security to be applied to the DBS request command frame. Typically, the DBS request command should not be implemented using security. However, if the TMCTP-child PAN coordinator requesting DBS allocation shares a key with the TMCTP-parent PAN coordinator, then security may be specified.

### 8.2.23.2 MLME-DBS.indication

The MLME-DBS.indication primitive is generated to indicate the reception of a DBS Request command.

The semantics of this primitive are as follows:

```
MLME-DBS.indication      (
                           CoordAddress,
                           RequesterCoordAddr,
                           RequestType,
                           DbsLength,
                           NumberOfDescendents,
                           SecurityLevel,
                           KeyIdMode,
                           KeySource,
                           KeyIndex
                           )
```

The primitive parameters are defined in Table 8-64.

**Table 8-64—MLME-DBS.indication parameters**

| Name                | Type          | Valid range              | Description                                                                                                                          |
|---------------------|---------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| CoordAddress        | Short address | 0x0000–0xffff            | The address of the Coordinator that sent the DBS Request command.                                                                    |
| RequesterCoord Addr | Short address | 0x0000–0xffff            | The address of the source requester PAN coordinator.                                                                                 |
| RequestType         | Enumeration   | ALLOCATION, DEALLOCATION | Indicates if the received request is for an allocation or deallocation of TMCTP DBS.                                                 |
| DbsLength           | Integer       | 0x00–0xff                | The value of the DBS Length field of the received DBS Request command.                                                               |
| NumberOfDescendents | Integer       | 0x00–0xff                | The number of TMCTP-child PAN coordinators. Set as zero if the PAN coordinator is not clear about how many descendants it will have. |
| SecurityLevel       | Integer       | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                            |
| KeyIdMode           | Integer       | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                            |
| KeySource           | Set of octets | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                            |
| KeyIndex            | Integer       | As defined in Table 8-77 | As defined in Table 8-77.                                                                                                            |



Upon receipt of this primitive, the next higher layer of a TMCTP-parent PAN coordinator is notified of a request for a DBS allocation or deallocation.

### 8.2.23.3 MLME-DBS.response

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

The semantics of this primitive are as follows:

```
MLME-DBS.response      (
                        CoordAddress,
                        RequesterCoordAddr,
                        DbsStartingSlot,
                        DbsLength,
                        ChannelNumber,
                        StartBandEdge,
                        StartingChNum,
                        EndingChNum,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

The primitive parameters are defined in Table 8-65.

**Table 8-65—MLME-DBS.response parameters**

| Name               | Type           | Valid range              | Description                                                                             |
|--------------------|----------------|--------------------------|-----------------------------------------------------------------------------------------|
| CoordAddress       | Short address  | 0x0000–0xffff            | The address of the Coordinator that sent DBS Request command.                           |
| RequesterCoordAddr | Short address  | 0x0000–0xffff            | The address of the source requester PAN coordinator.                                    |
| DbsStartingSlot    | Integer        | 0x0000–0xffff            | The first slot of the allocated DBS in the BOP.                                         |
| DbsLength          | Integer        | 0x00–0xff                | The size, in BOP slots, of the allocated DBS.                                           |
| ChannelNumber      | PHY Channel ID | As defined in Table 7-39 | The channel number that the coordinator intends to use for all future communications.   |
| StartBandEdge      | Set of octets  | As defined in 7.5.24     | The frequency in kilohertz indicating the lower edge of the band, as defined in 7.5.24. |
| StartingChNum      | PHY Channel ID | As defined in Table 7-39 | The lowest channel number, which is assigned by the TMCTP-parent PAN coordinator.       |
| EndingChNum        | PHY Channel ID | As defined in Table 7-39 | The highest channel number, which is assigned by the TMCTP-parent PAN coordinator.      |
| SecurityLevel      | Integer        | As defined in Table 8-75 | As defined in Table 8-75.                                                               |
| KeyIdMode          | Integer        | As defined in Table 8-75 | As defined in Table 8-75.                                                               |

**Table 8-65—MLME-DBS.response parameters (continued)**

| Name      | Type          | Valid range              | Description               |
|-----------|---------------|--------------------------|---------------------------|
| KeySource | Set of octets | As defined in Table 8-75 | As defined in Table 8-75. |
| KeyIndex  | Integer       | As defined in Table 8-75 | As defined in Table 8-75. |

When the MLME of a TMCTP-parent PAN coordinator receives an MLME-DBS.response primitive, it generates a DBS response command, as described in 7.5.24, and attempts to send it to the TMCTP-child PAN coordinator requesting the allocation of a DBS and a channel.

#### 8.2.23.4 MLME-DBS.confirm

The MLME-DBS.confirm primitive is used to inform the next higher layer of the initiating device of the result of its request for the allocation of a DBS and a channel.

The semantics of this primitive are as follows:

MLME-DBS.confirm

(  
RequesterCoordAddr,  
DbsStartingSlot,  
DbsLength,  
ChannelNumber,  
StartBandEdge,  
StartingChNum,  
EndingChNum,  
Status  
)

The primitive parameters are defined in Table 8-66.

**Table 8-66—MLME-DBS.confirm parameters**

| Name               | Type           | Valid range              | Description                                                                           |
|--------------------|----------------|--------------------------|---------------------------------------------------------------------------------------|
| RequesterCoordAddr | Short address  | 0x0000–0xffff            | The address of the source requester PAN coordinator.                                  |
| DbsStartingSlot    | Integer        | 0x0000–0xffff            | The first slot of the allocated DBS in the BOP.                                       |
| DbsLength          | Integer        | 0x00–0xff                | Number of slots allocated for the DBS.                                                |
| ChannelNumber      | PHY Channel ID | As defined in Table 7-39 | The channel number that the coordinator intends to use for all future communications. |
| StartBandEdge      | Set of octets  | As in defined in 7.5.24  | The frequency in kilohertz indicating the lower edge of the band.                     |
| StartingChNum      | PHY Channel ID | As defined in Table 7-39 | The lowest channel number, which is assigned by the TMCTP-parent PAN coordinator.     |

**Table 8-66—MLME-DBS.confirm parameters (continued)**

| Name        | Type           | Valid range                                                                       | Description                                                                        |
|-------------|----------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| EndingChNum | PHY Channel ID | As defined in Table 7-39                                                          | The highest channel number, which is assigned by the TMCTP-parent PAN coordinator. |
| Status      | Enumeration    | SUCCESS, NO_ACK, DENIED, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER | The status of the attempt of the allocation of a DBS and a channel.                |

If the DBS allocation request was successful, then the status parameter will be set to SUCCESS. Otherwise, the status parameter will be set to indicate the type of failure.

## 8.2.24 Primitives for device announcement

### 8.2.24.1 MLME-DA.request primitive

The MLME-DA.request primitive requests that the device to announce its address to neighbor devices.

The semantics of this primitive are as follows:

MLME-DA.request  
(  
CoordAddrMode,  
CoordPanId,  
CoordAddress,  
DaAddrMode,  
DaAddrList  
)

The primitive parameters are defined in Table 8-67.

**Table 8-67—MLME-DA.request parameters**

| Name          | Type           | Valid range                                 | Description                                                                     |
|---------------|----------------|---------------------------------------------|---------------------------------------------------------------------------------|
| CoordAddrMode | Enumeration    | SHORT, EXTENDED                             | The addressing mode of the coordinator to which this device is associated with. |
| CoordPanId    | Integer        | 0x0000–0xffff                               | The identifier of the PAN to which this device is associated with.              |
| CoordAddress  | —              | As specified by the CoordAddrMode parameter | The address of the coordinator to which this device is associated with.         |
| DaAddrMode    | Enumeration    | SHORT, EXTENDED                             | The addressing mode of the neighbors' addresses to be transmitted.              |
| DaAddrList    | Addresses list | As specified by the DaAddrMode parameter    | The neighbors' addresses list to be sent.                                       |

**8.2.24.2 MLME-DA.indication primitive**

The MLME-DA.indication primitive indicates reception addresses with a DA IE.

The semantics of this primitive are as follows:

```
MLME-DA.indication
(
  CoordAddrMode,
  CoordPanId,
  CoordAddress,
  AddrMode,
  Address,
  DaAddrMode,
  DaAddrList
)
```

The primitive parameters are defined in Table 8-68.

**Table 8-68—MLME-DA.indication parameters**

| Name          | Type             | Valid range                                 | Description                                                                              |
|---------------|------------------|---------------------------------------------|------------------------------------------------------------------------------------------|
| CoordAddrMode | Enumeration      | SHORT, EXTENDED                             | The addressing mode of the coordinator with which the transmitting device is associated. |
| CoordPanId    | Integer          | 0x0000–0xffff                               | The identifier of the PAN with which the transmitting device is associated.              |
| CoordAddress  | —                | As specified by the CoordAddrMode parameter | The address of the coordinator with which the transmitting device is associated.         |
| AddrMode      | Enumeration      | SHORT, EXTENDED                             | The addressing mode of the transmitting device.                                          |
| Address       | —                | As specified by the AddrMode parameter      | The address of the transmitting device.                                                  |
| DaAddrMode    | Enumeration      | SHORT, EXTENDED                             | The addressing mode of the addresses included in the Address List field of a DA IE.      |
| DaAddrList    | Set of addresses | As specified by the DaAddrMode parameter    | The addresses included in a DA IE.                                                       |

When the MAC has received all the DA IEs sent by the transmitting device, the MAC passes the addresses to the next higher layer using the MLME-DA.indication primitive.

**8.2.24.3 MLME-DA.confirm primitive**

The MLME-DA.confirm primitive reports results of broadcasting the beacon frame with DA IE.

The semantics of this primitive are as follows:

```
MLME-DA.confirm
(
  Status
)
```

The primitive parameter is defined in Table 8-69.

**Table 8-69—MLME-DA.confirm parameter**

| Name   | Type        | Valid range      | Description                                            |
|--------|-------------|------------------|--------------------------------------------------------|
| Status | Enumeration | SUCCESS, FAILURE | The results of broadcasting a beacon frame with DA IE. |

## 8.2.25 RIT data commands

### 8.2.25.1 MLME-RIT-REQ.indication

This primitive reports the reception of an RIT Data Request command with RIT Request Payload field.

The semantics of this primitive are as follows:

MLME-RIT-Data-Req.indication (

- SrcAddrMode,
- SrcPanId,
- SrcAddr,
- DstAddrMode,
- DstPanId,
- DstAddr,
- RitRequestPayload,
- HeaderIdList,
- PayloadIdList,
- LinkQuality,
- Dsn,
- Timestamp,
- SecurityLevel,
- KeyIdMode,
- KeySource,
- KeyIndex

)

The primitive parameters are defined in Table 8-70.

**Table 8-70—MLME-RIT-REQ.indication parameters**

| Name        | Type        | Valid range                               | Description                                                                                                            |
|-------------|-------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode | Enumeration | NONE, SHORT, EXTENDED                     | The source addressing mode of the frame that was received.                                                             |
| SrcPanId    | Integer     | 0x0000–0xffff                             | The Source PAN ID from the frame that was received. Valid only when a source PAN ID is included in the received frame. |
| SrcAddr     | —           | As specified by the SrcAddrMode parameter | The source address of the frame that was received.                                                                     |
| DstAddrMode | Enumeration | NONE, SHORT, EXTENDED                     | The destination addressing mode of the frame that was received.                                                        |

**Table 8-70—MLME-RIT-REQ.indication parameters (continued)**

| Name              | Type                                      | Valid range                               | Description                                                                                                                                                                                                                                                                                           |
|-------------------|-------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DstPanId          | Integer                                   | 0x0000–0xffff                             | The Destination PAN ID from the frame that was received. Set to the receiver's PAN ID if the PAN ID is not carried in the received frame.                                                                                                                                                             |
| DstAddr           | —                                         | As specified by the DstAddrMode parameter | The destination address of the frame that was received.                                                                                                                                                                                                                                               |
| RitRequestPayload | Set of octets                             | —                                         | The set of octets from the RIT Request Payload field in the frame that was received.                                                                                                                                                                                                                  |
| HeaderIeList      | List of header IEs as described in 7.4.2  | —                                         | The header IEs, excluding Termination IEs, that were included in the frame. If empty, then no header IEs were in the frame.                                                                                                                                                                           |
| PayloadIeList     | List of payload IEs as described in 7.4.3 | —                                         | The payload IEs, excluding Termination IEs, that were included in the frame. If empty, then no payload IEs were in the frame.                                                                                                                                                                         |
| LinkQuality       | Integer                                   | 0x00–0xff                                 | LQI value measured during reception of the RIT Data Request command. Lower values represent lower LQI, as described in 10.2.6.                                                                                                                                                                        |
| Dsn               | Integer                                   | 0x00–0xff                                 | The DSN of the received RIT Data Request command, if one was present.                                                                                                                                                                                                                                 |
| Timestamp         | Integer                                   | 0x000000–0xffffffff                       | Optional. The time, in symbols, at which the command was received, as described in 6.5.2. The symbol boundary is described by <i>macSyncSymbolOffset</i> , as described in Table 8-81. 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 frame, as defined in Table 9-6.                                                                                                                                                                                                                   |
| KeyIdMode         | Integer                                   | 0x00–0x03                                 | The mode used to identify the key purportedly used by the originator of the received frame, as defined in Table 9-7. This parameter is invalid if the SecurityLevel parameter is set to 0x00.                                                                                                         |
| KeySource         | Set of octets                             | As specified by the KeyIdMode parameter   | The originator of the key purportedly used by the originator of the received frame, as described in 9.4.3.1. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00 or 0x01.                                                                                                  |
| KeyIndex          | Integer                                   | 0x01–0xff                                 | The index of the key purportedly used by the originator of the received frame, as described in 9.4.3.2. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.                                                                                                               |

This primitive is generated by the MLME of a device and issued to its next higher layer upon the reception of an RIT Data Request command containing an RIT Request Payload field.

**8.2.25.2 MLME-RIT-RES.request**

This primitive allows the next higher layer of a device to respond to the MLME-RIT-RES.indication primitive.

The semantics of this primitive are as follows:

```
MLME-RIT-RES.request      (
                            SrcAddrMode,
                            DstAddrMode,
                            DstPanId,
                            DstAddr,
                            RitResponsePayload,
                            HeaderIeList,
                            PayloadIeList,
                            HeaderIeldList,
                            NestedIeldList,
                            AckTx,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

The primitive parameters are defined in Table 8-71.

**Table 8-71—MLME-RIT-RES.response parameters**

| Name               | Type                                  | Valid range                               | Description                                                                                                                                                                                 |
|--------------------|---------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode        | Enumeration                           | NONE, SHORT, EXTENDED                     | The source addressing mode for this frame.                                                                                                                                                  |
| DstAddrMode        | Enumeration                           | NONE, SHORT, EXTENDED                     | The destination addressing mode for this frame.                                                                                                                                             |
| DstPanId           | Integer                               | 0x0000–0xffff                             | The PAN ID of the entity to which the frame is being transferred.                                                                                                                           |
| DstAddr            | —                                     | As specified by the DstAddrMode parameter | The address of the entity to which the frame is being transferred.                                                                                                                          |
| RitResponsePayload | Set of octets                         | —                                         | The set of octets forming the RIT Response Payload field in the frame.                                                                                                                      |
| HeaderIeList       | Set of IEs as described in Table 7-7  | As defined in Table 7-7                   | The header IEs, excluding Termination IEs, that are to be included with the frame in addition to any header IEs added by the MAC. If empty, no additional header IEs are to be included.    |
| PayloadIeList      | Set of IEs as described in Table 7-15 | As defined in Table 7-15                  | The payload IEs, excluding Termination IEs, that are to be included with the frame in addition to any payload IEs added by the MAC. If empty, no additional payload IEs are to be included. |

**Table 8-71—MLME-RIT-RES.response parameters (continued)**

| Name            | Type                                                                 | Valid range                             | Description                                                                                                                                            |
|-----------------|----------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| HeaderIdList    | List of header IE IDs, as described in Table 7-7                     | —                                       | The IDs of header IEs, excluding Termination IEs, to be added to the frame. The MAC will provide the content for the IE.                               |
| NestedIdSubList | List of Nested IE sub-IDs, as described in Table 7-16 and Table 7-17 | —                                       | The sub-IDs of nested IEs, excluding Termination IEs, to be added to the Data frame. The MAC will provide the content for the IE.                      |
| AckTx           | Boolean                                                              | TRUE, FALSE                             | TRUE if acknowledged transmission is used, FALSE otherwise.                                                                                            |
| SecurityLevel   | Integer                                                              | As defined in Table 9-6                 | The security level to be used.                                                                                                                         |
| KeyIdMode       | Integer                                                              | As defined in Table 9-7                 | The mode used to identify the key to be used. This parameter is ignored if the SecurityLevel parameter is set to 0x00.                                 |
| KeySource       | Set of octets                                                        | As specified by the KeyIdMode parameter | The originator of the key to be used, as described in 9.4.3.1. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00 or 0x01. |
| KeyIndex        | Integer                                                              | 0x01–0xff                               | The index of the key to be used, as described in 9.4.3.2. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.              |

If the MLME-RIT-RES.request primitive has invalid parameters, the MLME will issues an MLME-RIT-RES.confirm primitive with Status set to INVALID\_PARAMETER and take no further action.

Otherwise, the MLME of the device shall generate an RIT Data Response command, as described in 7.5.25. If the acknowledgment is received for the command, the MLME shall issue an MLME-RIT-RES.confirm primitive with Status set to SUCCESS. If the number of retries is exceeded and an acknowledgment is not received, the MLME shall issue an MLME-RIT-RES.confirm primitive with Status set to NO\_ACK.

### 8.2.25.3 MLME-RIT-RES.indication

This primitive reports the reception of an RIT Data Response command.

The semantics of this primitive are as follows:

MLME-RIT-Data-Response.indication (

- SrcAddrMode,
- SrcPanId,
- SrcAddr,
- DstAddrMode,
- DstPanId,
- DstAddr,
- RitResponsePayload,
- HeaderIdList,



PayloadIeList,  
LinkQuality,  
Dsn,  
Timestamp,  
SecurityLevel,  
KeyIdMode,  
KeySource,  
KeyIndex  
)

The primitive parameters are defined in Table 8-72.

**Table 8-72—MLME-RIT-RES.indication parameters**

| Name               | Type                                      | Valid range                               | Description                                                                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode        | Enumeration                               | NONE, SHORT, EXTENDED                     | The source addressing mode of the frame that was received.                                                                                                                                                                                                                                            |
| SrcPanId           | Integer                                   | 0x0000–0xffff                             | The Source PAN ID from the frame that was received. Valid only when a source PAN ID is included in the received frame.                                                                                                                                                                                |
| SrcAddr            | —                                         | As specified by the SrcAddrMode parameter | The source address of the frame that was received.                                                                                                                                                                                                                                                    |
| DstAddrMode        | Enumeration                               | NONE, SHORT, EXTENDED                     | The destination addressing mode of the frame that was received.                                                                                                                                                                                                                                       |
| DstPanId           | Integer                                   | 0x0000–0xffff                             | The Destination PAN ID from the frame that was received. Set to the receiver's PAN ID if the PAN ID is not carried in the received frame.                                                                                                                                                             |
| DstAddr            | —                                         | As specified by the DstAddrMode parameter | The destination address of the frame that was received.                                                                                                                                                                                                                                               |
| RitResponsePayload | Set of octets                             | —                                         | The set of octets from the RIT Response Payload field in the frame that was received.                                                                                                                                                                                                                 |
| HeaderIeList       | List of header IEs as described in 7.4.2  | —                                         | The header IEs, excluding Termination IEs, that were included in the frame. If empty, then no header IEs were in the frame.                                                                                                                                                                           |
| PayloadIeList      | List of payload IEs as described in 7.4.3 | —                                         | The payload IEs, excluding Termination IEs, that were included in the frame. If empty, then no payload IEs were in the frame.                                                                                                                                                                         |
| LinkQuality        | Integer                                   | 0x00–0xff                                 | LQI value measured during reception of the RIT Data Request command. Lower values represent lower LQI, as described in 10.2.6.                                                                                                                                                                        |
| Dsn                | Integer                                   | 0x00–0xff                                 | The DSN of the received RIT Data Request command, if one was present.                                                                                                                                                                                                                                 |
| Timestamp          | Integer                                   | 0x000000–0xffffffff                       | Optional. The time, in symbols, at which the command was received, as described in 6.5.2. The symbol boundary is described by <i>macSyncSymbolOffset</i> , as described in Table 8-81. The precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant. |

**Table 8-72—MLME-RIT-RES.indication parameters (continued)**

| Name          | Type          | Valid range                             | Description                                                                                                                                                                                          |
|---------------|---------------|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SecurityLevel | Integer       | 0x00–0x07                               | The security level purportedly used by the received frame, as defined in Table 9-6.                                                                                                                  |
| KeyIdMode     | Integer       | 0x00–0x03                               | The mode used to identify the key purportedly used by the originator of the received frame, as defined in Table 9-7. This parameter is invalid if the SecurityLevel parameter is set to 0x00.        |
| KeySource     | Set of octets | As specified by the KeyIdMode parameter | The originator of the key purportedly used by the originator of the received frame, as described in 9.4.3.1. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00 or 0x01. |
| KeyIndex      | Integer       | 0x01–0xff                               | The index of the key purportedly used by the originator of the received frame, as described in 9.4.3.2. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.              |

This primitive is generated by the MLME of a device and issued to its next higher layer upon the reception of an RIT Data Response command.

#### 8.2.25.4 MLME-RIT-RES.confirm

This primitive reports the result of a request to send an RIT Data Response command.

The semantics of this primitive are as follows:

```
MLME-RIT-Data-Response.confirm (
    Status
)
```

The primitive parameter is defined in Table 8-73.

**Table 8-73—MLME-RIT-RES.confirm parameter**

| Name   | Type        | Valid Range                        | Description                                   |
|--------|-------------|------------------------------------|-----------------------------------------------|
| Status | Enumeration | SUCCESS, NO_ACK, INVALID_PARAMETER | The status of MLME-RIT-RES.request primitive. |

The value of the Status parameter is set as described in 8.2.25.2.

### 8.3 MAC data service

The MCPS-SAP supports the transport of data. Table 8-74 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 8-74—MCPS-SAP primitives**

| MCPS-SAP primitive | Request | Confirm | Indication |
|--------------------|---------|---------|------------|
| MCPS-DATA          | 8.3.1   | 8.3.2   | 8.3.3      |
| MCPS-PURGE         | 8.3.4♦  | 8.3.5♦  | —          |

#### 8.3.1 MCPS-DATA.request

The MCPS-DATA.request primitive requests the transfer of data to another device.

The semantics of this primitive are as follows:

MCPS-DATA.request

```
(
  SrcAddrMode,
  DstAddrMode,
  DstPanId,
  DstAddr,
  Msdu,
  MsduHandle,
  HeaderIdList,
  PayloadIdList,
  HeaderIdList,
  NestedIdSubIdList,
  AckTx,
  GtsTx,
  IndirectTx,
  SecurityLevel,
  KeyIdMode,
  KeySource,
  KeyIndex,
  UwbPrf,
  Ranging,
  UwbPreambleSymbolRepetitions,
  DataRate,
  LocationEnhancingInformationPostamble,
  LocationEnhancingInformationPostambleLength,
  PanIdSuppressed,
  SeqNumSuppressed,
  SendMultipurpose
  FrakPolicy,
  CriticalEventMessage
)
```

The primitive parameters are defined in Table 8-75.

**Table 8-75—MCPS-DATA.request parameters**

| Name              | Type                                                                | Valid range                               | Description                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode       | Enumeration                                                         | NONE, SHORT, EXTENDED                     | The source addressing mode for this MPDU.                                                                                                                                                   |
| DstAddrMode       | Enumeration                                                         | NONE, SHORT, EXTENDED                     | The destination addressing mode for this MPDU.                                                                                                                                              |
| DstPanId          | Integer                                                             | 0x0000–0xffff                             | The PAN ID of the entity to which the MSDU is being transferred.                                                                                                                            |
| DstAddr           | —                                                                   | As specified by the DstAddrMode parameter | The address of the entity to which the MSDU is being transferred.                                                                                                                           |
| 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.                                                                                                           |
| HeaderIeList      | Set of IEs as described in Table 7-7                                | As defined in Table 7-7                   | The header IEs, excluding Termination IEs, that are to be included with the frame in addition to any header IEs added by the MAC. If empty, no additional header IEs are to be included.    |
| PayloadIeList     | Set of IEs as described in Table 7-15                               | As defined in Table 7-15                  | The payload IEs, excluding Termination IEs, that are to be included with the frame in addition to any payload IEs added by the MAC. If empty, no additional payload IEs are to be included. |
| HeaderIeIdList    | Set of header IE IDs, as described in Table 7-7                     | —                                         | The IDs of header IEs, excluding Termination IEs, to be added to the Data frame. The MAC will provide the content for the IE.                                                               |
| NestedIeSubIdList | Set of Nested IE sub-IDs, as described in Table 7-16 and Table 7-17 | —                                         | The sub-IDs of nested IEs, excluding Termination IEs, to be added to the Data frame. The MAC will provide the content for the IE.                                                           |
| AckTx             | Boolean                                                             | TRUE, FALSE                               | TRUE if acknowledged transmission is used, FALSE otherwise.                                                                                                                                 |
| GtsTx             | Boolean                                                             | TRUE, FALSE                               | TRUE if a GTS is to be used for transmission. FALSE indicates that the CAP will be used.                                                                                                    |
| IndirectTx        | Boolean                                                             | TRUE, FALSE                               | TRUE if indirect transmission is to be used, FALSE otherwise.                                                                                                                               |
| SecurityLevel     | Integer                                                             | As defined in Table 9-6                   | The security level to be used.                                                                                                                                                              |
| KeyIdMode         | Integer                                                             | As defined in Table 9-7                   | The mode used to identify the key to be used. This parameter is ignored if the SecurityLevel parameter is set to 0x00.                                                                      |

**Table 8-75—MCPS-DATA.request parameters (continued)**

| Name                         | Type          | Valid range                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------|---------------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KeySource                    | Set of octets | As specified by the KeyIdMode parameter                   | The originator of the key to be used, as described in 9.4.3.1. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00 or 0x01.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| KeyIndex                     | Integer       | 0x01–0xff                                                 | The index of the key to be used, as described in 9.4.3.2. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| UwbPrf                       | Enumeration   | PRF_OFF,<br>NOMINAL_4_M,<br>NOMINAL_16_M,<br>NOMINAL_64_M | The pulse repetition value of the transmitted PPDU. Non-HRP UWB PHYs use a value of PRF_OFF.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Ranging                      | Enumeration   | NON_RANGING,<br>ALL_RANGING,<br>PHY_HEADER_ONLY           | A value of NON_RANGING indicates that ranging is not to be used. A value of ALL_RANGING indicates that ranging operations using both the ranging bit in the PHR and the counter operation are enabled. A value of PHY_HEADER_ONLY indicates that only the ranging bit in the PHR will be used. A value of NON_RANGING is PHYs that do not support ranging.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| UwbPreambleSymbolRepetitions | Integer       | 0, 16, 32, 64, 128, 256,<br>512, 1024, 4096, 8192         | The preamble symbol repetitions of the HRP UWB PHY or LRP UWB frame. A zero value is used for all other PHYs. (see NOTE)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| DataRate                     | Integer       | —                                                         | Indicates the data rate. For CSS PHYs, a value of one indicates 250 kb/s while a value of two indicates 1 Mb/s. For HRP UWB PHYs, values 1–4 are valid and are defined in 16.2.6. For LRP UWB PHYs, valid values are defined in Table 19-1. For the SUN OFDM PHY, values 1–7 are valid; each data rate value corresponds to the variable MCS{DataRate–1}, as described in Table 21-9. For the SUN O-QPSK PHY with DSSS spreading, values 1–4 are valid; each data rate value corresponds to the Rate Mode plus one, as described in Table 22-1. For the SUN O-QPSK PHY with MDSSS spreading, values 5–8 are valid; each data rate value corresponds to the Rate Mode plus five, as described in Table 22-1. For MSK PHYs, valid values are defined in Table 18-1. For all other PHYs, the parameter is set to zero. |

**Table 8-75—MCPS-DATA.request parameters (continued)**

| Name                                                                                                                                                      | Type        | Valid range                                                                                                       | Description                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LocationEnhancingInformationPostamble                                                                                                                     | Enumeration | LEIP_NONE,<br>LEIP_IMMEDIATE,<br>LEIP_DELAYED                                                                     | For the LRP UWB PHY this parameter specifies whether the Location enhancing information postamble sequence is to be sent or not and, if present, whether it directly follows the CRC or is delayed by the <i>aLeipDelayTime</i> . A value of LEIP_NONE is used for non-LRP UWB PHYs.                                                                        |
| LocationEnhancingInformationPostambleLength                                                                                                               | Enumeration | LEIP_LEN_16,<br>LEIP_LEN_64,<br>LEIP_LEN_128,<br>LEIP_LEN_192,<br>LEIP_LEN_256,<br>LEIP_LEN_512,<br>LEIP_LEN_1024 | For the LRP UWB PHY when the LocationEnhancingInformationPostamble parameter has a value of either LEIP_IMMEDIATE or LEIP_DELAYED, then this parameter specifies the length in pulses of the location enhancing information postamble to send. This parameter is ignored when the LocationEnhancingInformationPostamble parameter has a value of LEIP_NONE. |
| PanIdSuppressed                                                                                                                                           | Boolean     | TRUE, FALSE                                                                                                       | Set to TRUE if the PAN ID is suppressed in the frame, FALSE otherwise.                                                                                                                                                                                                                                                                                      |
| SeqNumSuppressed                                                                                                                                          | Boolean     | TRUE, FALSE                                                                                                       | Set to TRUE if the sequence number is suppressed in the frame, FALSE otherwise.                                                                                                                                                                                                                                                                             |
| SendMultipurpose                                                                                                                                          | Boolean     | TRUE, FALSE                                                                                                       | If TRUE, use a Multipurpose frame. If FALSE, use a Data frame.                                                                                                                                                                                                                                                                                              |
| FrakPolicy                                                                                                                                                | Integer     | 0b00–0b10                                                                                                         | Specifies the Frak policy to be employed, as described in 7.4.2.9. This parameter is only used when MPDU fragmentation is enabled.                                                                                                                                                                                                                          |
| CriticalEventMessage                                                                                                                                      | Boolean     | TRUE, FALSE                                                                                                       | A value of TRUE indicates that the message shall be processed as a critical event message, as described in 6.2.5.4. A value of FALSE indicates that the message is not a critical event message and shall be processed as described in 6.2.5.1.                                                                                                             |
| NOTE—Some values may be unsupported or invalid depending on the capabilities of the PHY or its current transmission mode as selected by other parameters. |             |                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                             |

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

The TxOptions parameter indicates the method used by the MAC sublayer data service to transmit the supplied MSDU. If the TxOptions parameter specifies that an acknowledged transmission is required, the AR field will be set appropriately, as described in 6.7.4.

If the TxOptions parameter specifies that a GTS transmission is required, the MAC sublayer will determine whether it has a valid GTS as described 6.8.4. If a valid GTS could not be found, the MAC sublayer will discard the MSDU. 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, as described in 6.6 and 6.7.3.

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, or at the next timeslot to the destination address if in TSCH mode.

If SendMultipurpose is TRUE, then the Msdu is to be sent using a Multipurpose frame.

If SendMultipurpose is FALSE, then the Msdu is to be sent using a Data frame.

### 8.3.2 MCPS-DATA.confirm

The MCPS-DATA.confirm primitive reports the results of a request to transfer data to another device.

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

MCPS-DATA.confirm

```
(
    MsduHandle,
    Timestamp,
    RangingReceived,
    RangingCounterStart,
    RangingCounterStop,
    RangingTrackingInterval,
    RangingOffset,
    RangingFom,
    NumBackoffs,
    AckPayload,
    Status
)
```

The primitive parameters are defined in Table 8-76.

**Table 8-76—MCPS-DATA.confirm parameters**

| Name       | Type    | Valid range | Description                                          |
|------------|---------|-------------|------------------------------------------------------|
| MsduHandle | Integer | 0x00–0xff   | The handle associated with the MSDU being confirmed. |

**Table 8-76—MCPS-DATA.confirm parameters (continued)**

| Name                    | Type                     | Valid range           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------|--------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Timestamp               | Integer                  | 0x000000–0xffffffff   | Optional. The time, in symbols, at which the data were transmitted, as described in 6.5.2. 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 <i>macSyncSymbolOffset</i> , as described in Table 8-81. The precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant. |
| RangingReceived         | Boolean                  | TRUE, FALSE           | A value of FALSE indicates that ranging is either not supported by the PHY or that it was not indicated by the received PSDU. A value of TRUE indicates ranging operations were indicated for this PSDU.                                                                                                                                                                                                                                                                                                                                         |
| RangingCounterStart     | Unsigned Integer         | 0x00000000–0xffffffff | A count of the time units corresponding to an RMARKER at the antenna at the beginning of a ranging exchange, as described in 16.7.1. A value of 0x00000000 is used if ranging is not supported, not enabled or if counter was not used for this PPDU. A value of 0x00000000 is also used when one-way ranging is being employed. One-way ranging is described in “Applications of IEEE Std 802.15.4” [B3].                                                                                                                                       |
| RangingCounterStop      | Unsigned Integer         | 0x00000000–0xffffffff | A count of the time units corresponding to an RMARKER at the antenna at the end of a ranging exchange, as described in 16.7.1. A value of 0x00000000 is used if ranging is not supported, not enabled, or if the counter is not used for this PPDU. For one-way ranging this parameter reports the arrival time of the RMARKER at the antenna.                                                                                                                                                                                                   |
| RangingTrackingInterval | Integer                  | 0x00000000–0xffffffff | A count of the time units in a message exchange over which the tracking offset was measured, as described in 16.7.2.2. If tracking-based crystal characterization is not supported or if ranging is not supported, a value of 0x00000000 is used.                                                                                                                                                                                                                                                                                                |
| RangingOffset           | Signed Magnitude Integer | 0x000000–0xffffffff   | A count of the time units slipped or advanced by the radio tracking system over the course of the entire tracking interval, as described in 16.7.2.1. The top 4 bits are reserved and set to zero. The most significant of the active bits is the sign bit.                                                                                                                                                                                                                                                                                      |



**Table 8-76—MCPS-DATA.confirm parameters (continued)**

| Name        | Type          | Valid range                                                                                                                                                                                                                                                                                                                                                                                           | Description                                                                                                                                                                                               |
|-------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RangingFom  | Integer       | 0x00–0x7f                                                                                                                                                                                                                                                                                                                                                                                             | The figure of merit (FoM) characterizing the ranging measurement, as described in 16.7.3.                                                                                                                 |
| NumBackoffs | Integer       | 0x00–0xff                                                                                                                                                                                                                                                                                                                                                                                             | The number of times the CSMA-CA algorithm was required to backoff as described in 6.2.3 while attempting the current transmission. If “Status” is anything other than “SUCCESS,” this value is undefined. |
| AckPayload  | Set of octets | —                                                                                                                                                                                                                                                                                                                                                                                                     | The set of octets received in the Frame Payload field of the Ack frame, if present.                                                                                                                       |
| Status      | Enumeration   | SUCCESS,<br>TRANSACTION_OVERFLOW,<br>TRANSACTION_EXPIRED,<br>CHANNEL_ACCESS_FAILURE,<br>INVALID_ADDRESS,<br>INVALID_GTS, NO_ACK,<br>COUNTER_ERROR,<br>FRAME_TOO_LONG,<br>UNAVAILABLE_KEY,<br>UNSUPPORTED_SECURITY,<br>UNSUPPORTED_FEATURE,<br>INVALID_PARAMETER,<br>UNSUPPORTED_PRF,<br>UNSUPPORTED_RANGING,<br>UNSUPPORTED_PSR,<br>UNSUPPORTED_DATARATE,<br>UNSUPPORTED_LEIP,<br>ACK_RCVD_NODSN_NOSA | The status of the last MSDU transmission.                                                                                                                                                                 |

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.

If the generated MAC frame is a Data frame with the Frame Version subfield set to 0b00 or 0b01, and both the SrcAddrMode and the DstAddrMode parameters are set to NO\_ADDRESS in the MCPS-DATA.request primitive, the Status shall be set to INVALID\_ADDRESS.

If a valid GTS could not be found, the Status shall be set to INVALID\_GTS.

If there is no capacity to store the transaction, the Status will be set to TRANSACTION\_OVERFLOW.

If the transaction is not handled within the required time, the transaction information will be discarded and the Status will be set to TRANSACTION\_EXPIRED.

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, as described in 6.7.4, it will discard the MSDU and issue the MCPS-DATA.confirm primitive with a Status of NO\_ACK.

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 the Status will be set to CHANNEL\_ACCESS\_FAILURE.

If an acknowledgment is received without a DSN or Source Address field, the Status will be set to ACK\_RCVD\_NODSN\_NOSA.

If an acknowledgment is received that contains user data encapsulated in payload IEs or included in the Payload field, the received AckPayload will contain the received data.

If the MAC PIB attributes *macPriorityChannelAccess* or *macPcaAllocationSuperRate* are set differently from their respective conditions in Table 6-1, or if the attribute *macPcaAllocationRate* does not satisfy the minimum value defined by its respective condition in Table 6-1, then the MAC sublayer may discard the PCA MSDU and set the Status of the MCPS-DATA.confirm primitive to INVALID\_PARAMETER.

If the MAC PIB attribute *macPriorityChannelAccess* is set to FALSE, the MAC sublayer will discard the PCA MSDU and the Status of the MCPS-DATA.confirm primitive will be set to INVALID\_PARAMETER.

### 8.3.3 MCPS-DATA.indication

The MCPS-DATA.indication primitive indicates the reception of data from another device.

The semantics of this primitive are as follows:

|                      |                          |
|----------------------|--------------------------|
| MCPS-DATA.indication | (                        |
|                      | SrcAddrMode,             |
|                      | SrcPanId,                |
|                      | SrcAddr,                 |
|                      | DstAddrMode,             |
|                      | DstPanId,                |
|                      | DstAddr,                 |
|                      | Msdu,                    |
|                      | HeaderIeList,            |
|                      | PayloadIeList,           |
|                      | MpduLinkQuality,         |
|                      | Dsn,                     |
|                      | Timestamp,               |
|                      | SecurityLevel,           |
|                      | KeyIdMode,               |
|                      | KeySource,               |
|                      | KeyIndex,                |
|                      | RangingReceived,         |
|                      | RangingCounterStart,     |
|                      | RangingCounterStop,      |
|                      | RangingTrackingInterval, |
|                      | RangingOffset,           |
|                      | RangingFom,              |
|                      | AngleOfArrivalAzimuth,   |
|                      | AngleOfArrivalElevation, |
|                      | AngleOfArrivalSupported, |
|                      | DataRate,                |
|                      | Rssi                     |
|                      | )                        |

The primitive parameters are defined in Table 8-77.

**Table 8-77—MCPS-DATA.indication parameters**

| Name            | Type                                     | Valid range                               | Description                                                                                                                                                                                                                                                                                         |
|-----------------|------------------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SrcAddrMode     | Enumeration                              | NONE, SHORT, EXTENDED                     | The source addressing mode for this primitive corresponding to the received MPDU.                                                                                                                                                                                                                   |
| SrcPanId        | Integer                                  | 0x0000–0xffff                             | The PAN ID of the entity from which the MSDU was received. Valid only when a source PAN ID is included in the received frame.                                                                                                                                                                       |
| SrcAddr         | —                                        | As specified by the SrcAddrMode parameter | The address of the entity from which the MSDU was received.                                                                                                                                                                                                                                         |
| DstAddrMode     | Enumeration                              | NONE, SHORT, EXTENDED                     | The destination addressing mode for this primitive corresponding to the received MPDU.                                                                                                                                                                                                              |
| DstPanId        | Integer                                  | 0x0000–0xffff                             | The PAN ID of the entity to which the MSDU is being transferred. Set to the receiver's PAN ID if the PAN ID is not carried in the received frame.                                                                                                                                                   |
| DstAddr         | —                                        | As specified by the DstAddrMode parameter | The address of the entity to which the MSDU is being transferred.                                                                                                                                                                                                                                   |
| Msdus           | Set of octets                            | —                                         | The set of octets forming the MSDU being indicated by the MAC sublayer entity including payload IEs if present.                                                                                                                                                                                     |
| HeaderIeList    | Set of header IEs as described in 7.4.2  | —                                         | The header IEs, excluding Termination IEs, that were included in the MAC frame. If empty, then no header IEs were in the MAC frame.                                                                                                                                                                 |
| PayloadIeList   | Set of payload IEs as described in 7.4.3 | —                                         | The payload IEs, excluding Termination IEs, that were included in the MAC frame. If empty, then no payload IEs were in the MAC frame.                                                                                                                                                               |
| MpduLinkQuality | Integer                                  | 0x00–0xff                                 | LQI value measured during reception of the MPDU. Lower values represent lower LQI, as described in 10.2.6.                                                                                                                                                                                          |
| Dsn             | Integer                                  | 0x00–0xff                                 | The DSN of the received Data frame if one was present.                                                                                                                                                                                                                                              |
| Timestamp       | Integer                                  | 0x000000–0xffffffff                       | Optional. The time, in symbols, at which the data were received, as described in 6.5.2. The symbol boundary is described by <i>macSyncSymbolOffset</i> , as described in Table 8-81. 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, as defined in Table 9-6.                                                                                                                                                                                                            |

**Table 8-77—MCPS-DATA.indication parameters (continued)**

| Name                    | Type                     | Valid range                                                               | Description                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------|--------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KeyIdMode               | Integer                  | 0x00–0x03                                                                 | The mode used to identify the key purportedly used by the originator of the received frame, as defined in Table 9-7. This parameter is invalid if the SecurityLevel parameter is set to 0x00.                                                                                                                                                                            |
| KeySource               | Set of octets            | As specified by the KeyIdMode parameter                                   | The originator of the key purportedly used by the originator of the received frame, as described in 9.4.3.1. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00 or 0x01.                                                                                                                                                                     |
| KeyIndex                | Integer                  | 0x01–0xff                                                                 | The index of the key purportedly used by the originator of the received frame, as described in 9.4.3.2. This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.                                                                                                                                                                                  |
| RangingReceived         | Enumeration              | NO_RANGING_REQUESTED, RANGING_ACTIVE, RANGING_REQUESTED_BUT_NOT_SUPPORTED | A value of RANGING_REQUESTED_BUT_NOT_SUPPORTED indicates that ranging is not supported but has been requested. A value of NO_RANGING_REQUESTED indicates that no ranging is requested for the PSDU received. A value of RANGING_ACTIVE denotes ranging operations requested for this PSDU. A value of NO_RANGING_REQUESTED is used for PHYs that do not support ranging. |
| RangingCounterStart     | Unsigned Integer         | As defined in Table 8-76                                                  | As defined in Table 8-76.                                                                                                                                                                                                                                                                                                                                                |
| RangingCounterStop      | Unsigned Integer         | As defined in Table 8-76                                                  | As defined in Table 8-76.                                                                                                                                                                                                                                                                                                                                                |
| RangingTrackingInterval | Integer                  | As defined in Table 8-76                                                  | As defined in Table 8-76.                                                                                                                                                                                                                                                                                                                                                |
| RangingOffset           | Signed Magnitude Integer | As defined in Table 8-76                                                  | As defined in Table 8-76.                                                                                                                                                                                                                                                                                                                                                |
| RangingFom              | Integer                  | As defined in Table 8-76                                                  | As defined in Table 8-76.                                                                                                                                                                                                                                                                                                                                                |
| AngleOfArrivalAzimuth   | Float                    | $-\pi$ to $+\pi$                                                          | Angle of arrival of signal in azimuth measured in radians. This parameter is valid only when AngleOfArrivalSupported is set to either AZIMUTH, or BOTH. The real world direction indicated (e.g., by 0 radians) is a system set-up parameter beyond the scope of this standard.                                                                                          |
| AngleOfArrivalElevation | Float                    | $-\pi/2$ to $+\pi/2$                                                      | Angle of arrival of signal in elevation measured in radians. This parameter is valid only when AngleOfArrivalSupported is set to either ELEVATION, or BOTH.                                                                                                                                                                                                              |
| AngleOfArrivalSupported | Enumeration              | NONE, BOTH, AZIMUTH, ELEVATION                                            | Indicates validity of AngleOfArrivalAzimuth and AngleOfArrivalElevation. Where the underlying PHY does not support angle of arrival measurement, then this parameter shall be set to NONE.                                                                                                                                                                               |

**Table 8-77—MCPS-DATA.indication parameters (continued)**

| Name     | Type    | Valid range | Description                                                                                                                                                               |
|----------|---------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DataRate | Integer | —           | As defined in Table 8-75.                                                                                                                                                 |
| Rssi     | Integer | 0x00–0xff   | The Received Signal Strength Indicator is a measure of the RF power level at the input of the transceiver measured during the PHR and is valid after the SFD is detected. |

The MCPS-DATA.indication primitive is generated by the MAC sublayer and issued to the next higher layer on receipt of a Data frame or a Multipurpose frame at the local MAC sublayer entity that passes the appropriate message filtering operations as described in 6.7.2. If the primitive is received while the device is in promiscuous mode, the parameters will be set as specified in 6.7.7.

### 8.3.4 MCPS-PURGE.request

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

The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 8-78.

**Table 8-78—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. |

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.

### 8.3.5 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.

The semantics of this primitive are as follows:

```
MCPS-PURGE.confirm      (
                          MsduHandle,
                          Status
                          )
```

The primitive parameters are defined in Table 8-79.

**Table 8-79—MCPS-PURGE.confirm parameters**

| Name      | Type        | Valid range                | Description                                                               |
|-----------|-------------|----------------------------|---------------------------------------------------------------------------|
| MsdHandle | Integer     | 0x00–0xff                  | The handle of the MSDU requested to be purged from the transaction queue. |
| Status    | Enumeration | SUCCESS,<br>INVALID_HANDLE | The status of the request to purge an MSDU from the transaction queue.    |

The MCPS-PURGE.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-PURGE.request primitive. If an MSDU matching the given handle is found, the Status will be set to SUCCESS. If an MSDU matching the given handle is not found, the Status will be set to INVALID\_HANDLE.

## 8.4 MAC constants and PIB attributes

This subclause specifies the constants and attributes required by the MAC sublayer.

### 8.4.1 MAC constants

The constants that define the characteristics of the MAC sublayer are presented in Table 8-80.

**Table 8-80—MAC sublayer constants**

| Constant                       | Description                                                                                                                                                                                                                                                                                                                                                                                                  | Value                                          |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| <i>aBaseSlotDuration</i>       | The number of symbols forming a superframe slot when the <i>superframe order</i> is equal to zero, as described in 6.2.1.                                                                                                                                                                                                                                                                                    | 60                                             |
| <i>aBaseSuperframeDuration</i> | The number of symbols forming a superframe when the <i>superframe order</i> is equal to zero.                                                                                                                                                                                                                                                                                                                | $aBaseSlotDuration \times aNumSuperframeSlots$ |
| <i>aGtsDescPersistenceTime</i> | The number of superframes in which a GTS descriptor exists in the beacon frame of the PAN coordinator.                                                                                                                                                                                                                                                                                                       | 4                                              |
| <i>aMaxLostBeacons</i>         | The number of consecutive lost beacons that will cause the MAC sublayer of a receiving device to declare a loss of synchronization.                                                                                                                                                                                                                                                                          | 4                                              |
| <i>aMaxSifsFrameSize</i>       | The maximum size of an MPDU, in octets, that can be followed by a SIFS period.                                                                                                                                                                                                                                                                                                                               | 18                                             |
| <i>aMinCapLength</i>           | 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, as described in 7.3.1.4. Additional restrictions apply when PCA is enabled, as described in 6.2.5.4. | 440                                            |

**Table 8-80—MAC sublayer constants (continued)**

| Constant                     | Description                                                                        | Value                                                                                                                                                                       |
|------------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>aNumSuperframeSlots</i>   | The number of slots contained in any superframe.                                   | 16                                                                                                                                                                          |
| <i>aUnitBackoffPeriod</i>    | The number of symbols forming the basic time period used by the CSMA-CA algorithm. | For all PHYs except SUN PHYs operating in the 920 MHz band, $aTurnaroundTime + aCcaTime$ . For SUN PHYs operating in the 920 MHz band, $aTurnaroundTime + phyCCADuration$ . |
| <i>aRccnBaseSlotDuration</i> | The number of symbols forming an RCCN superframe slot.                             | 60                                                                                                                                                                          |

**8.4.2 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 8-81, Table 8-82, Table 8-83, Table 8-88, Table 8-91, Table 8-92, Table 8-93, and Table 8-94. 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).

**Table 8-81—MAC PIB attributes**

| Attribute                              | Type         | Range           | Description                                                                                                                                                                                                   | Default |
|----------------------------------------|--------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macExtendedAddress</i> <sup>†</sup> | IEEE address | Device specific | The extended address assigned to the device.                                                                                                                                                                  | —       |
| <i>macAssociatedPanCoord</i>           | Boolean      | TRUE, 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   |
| <i>macAssociation Permit</i> ◆         | Boolean      | TRUE, FALSE     | Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.                                                                               | FALSE   |

Table 8-81—MAC PIB attributes (continued)

| Attribute                    | Type          | Range               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Default                                                                 |
|------------------------------|---------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <i>macAutoRequest</i>        | Boolean       | TRUE, 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, as described in 8.2.5.1.                                                                                                                                                                                                                                                                                                         | TRUE                                                                    |
| <i>macBattLifeExt</i>        | Boolean       | TRUE, 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. The effect of this attribute on the backoff exponent in the CSMA-CA algorithm is described in 6.2.5.1.                                                                                                                                                                                                                                                                                                                                                           | FALSE                                                                   |
| <i>macBattLifeExtPeriods</i> | 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:<br>Term 1: The value $2^x - 1$ , where $x$ is the maximum value of <i>macMinBe</i> in BLE mode (equal to two). This term is thus equal to three backoff periods.<br>Term 2: The duration of the initial contention window length, as described in 6.2.5.1.<br>Term 3: The Preamble field length and the SFD field length of the supported PHY summed together and rounded up (if necessary) to an integer number of backoff periods. | Dependent on currently selected PHY, indicated by <i>phyCurrentPage</i> |
| <i>macBeaconPayload</i> ♦    | Set of octets | —                   | A sequence of zero or more octets to be transmitted in the Beacon Payload field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | NULL                                                                    |
| <i>macBeaconOrder</i> ♦      | Integer       | 0–15                | Indicates the frequency with which the beacon is transmitted, as defined in 6.2.1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 15                                                                      |
| <i>macBeaconTxTime</i> † ♦   | Integer       | 0x000000–0xffffffff | 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. The precision of this value shall be a minimum of 20 bits, with the lowest four bits being the least significant.                                                                                                                                                                                                                                                                               | 0x000000                                                                |



Table 8-81—MAC PIB attributes (*continued*)

| Attribute                      | Type         | Range                    | Description                                                                                                                                                                                                                        | Default                            |
|--------------------------------|--------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| <i>macBsn</i> ♦                | Integer      | 0x00–0xff                | The sequence number added to the transmitted beacon frame.                                                                                                                                                                         | Random value from within the range |
| <i>macCoordExtendedAddress</i> | IEEE address | An extended IEEE address | The address of the coordinator through which the device is associated.                                                                                                                                                             | —                                  |
| <i>macCoordShortAddress</i>    | Integer      | 0x0000–0xffff            | The short address assigned to the coordinator through which the device is associated. A value of 0xfffe indicates that the coordinator is only using its extended address. A value of 0xffff indicates that this value is unknown. | 0xffff                             |
| <i>macDsn</i>                  | Integer      | 0x00–0xff                | The sequence number added to the transmitted Data frame or MAC command.                                                                                                                                                            | Random value from within the range |
| <i>macGtsPermit</i> *          | Boolean      | TRUE, FALSE              | TRUE if the PAN coordinator is to accept GTS requests. FALSE otherwise.                                                                                                                                                            | TRUE                               |
| <i>macMaxBe</i>                | Integer      | 3–8                      | The maximum value of the backoff exponent, BE, in the CSMA-CA algorithm, as defined in 6.2.5.1.                                                                                                                                    | 5                                  |
| <i>macMaxCsmBackoffs</i>       | Integer      | 0–5                      | The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure.                                                                                                                       | 4                                  |
| <i>macMaxFrameRetries</i>      | Integer      | 0–7                      | The maximum number of retries allowed after a transmission failure.                                                                                                                                                                | 3                                  |
| <i>macMinBe</i>                | Integer      | 0– <i>macMaxBe</i>       | The minimum value of the backoff exponent (BE) in the CSMA-CA algorithm, as described in 6.2.5.1.                                                                                                                                  | 3                                  |
| <i>macLifsPeriod</i> †         | Integer      | As defined in 10.1.3     | The minimum time forming a LIFS period.                                                                                                                                                                                            | PHY dependent                      |
| <i>macSifsPeriod</i> †         | Integer      | As defined in 10.1.3     | The minimum time forming a SIFS period.                                                                                                                                                                                            | PHY dependent                      |
| <i>macPanId</i>                | Integer      | 0x0000–0xffff            | The identifier of the PAN on which the device is operating. If this value is 0xffff, the device is not associated.                                                                                                                 | 0xffff                             |
| <i>macPromiscuousMode</i> ♦    | Boolean      | TRUE, 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                              |

Table 8-81—MAC PIB attributes (continued)

| Attribute                                | Type    | Range                                                                                                                | Description                                                                                                                                                                                                                                                      | Default |
|------------------------------------------|---------|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macRangingSupported</i> <sup>†*</sup> | Boolean | TRUE, FALSE                                                                                                          | This indicates whether the MAC sublayer supports the optional ranging features.                                                                                                                                                                                  | —       |
| <i>macResponseWaitTime</i>               | Integer | 2–64                                                                                                                 | The maximum time, in multiples of <i>aBaseSuperframeDuration</i> , a device shall wait for a response command to be available following a request command.                                                                                                       | 32      |
| <i>macRxOnWhenIdle</i>                   | Boolean | TRUE, 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   |
| <i>macSecurityEnabled</i>                | Boolean | TRUE, 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   |
| <i>macShortAddress</i>                   | Integer | 0x0000–0xffff                                                                                                        | The 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 short address is allocated by a coordinator during association.                             | 0xffff  |
| <i>macSuperframeOrder</i> <sup>†♦</sup>  | Integer | 0–15                                                                                                                 | The length of the active portion of the outgoing superframe, including the beacon frame, as defined in 6.2.1                                                                                                                                                     | 15      |
| <i>macSyncSymbolOffset</i> <sup>†</sup>  | Integer | 0x000–0x100 for the 2.4 GHz band<br>0x000–0x400 for the 868 MHz and 915 MHz bands, and the SUN FSK and SUN OFDM PHYs | 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.                                                                     | —       |
| <i>macEnhancedBeaconOrder</i>            | Integer | 0–15                                                                                                                 | Specification of how often the coordinator transmits an Enhanced Beacon frame. If <i>macEnhancedBeaconOrder</i> = 15, no periodic Enhanced Beacon frame will be transmitted.                                                                                     | 0       |

Table 8-81—MAC PIB attributes (continued)

| Attribute                                         | Type    | Range                 | Description                                                                                                                                                                                                                                                                                                                                                        | Default |
|---------------------------------------------------|---------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macMpmIe</i>                                   | Boolean | TRUE, FALSE           | An indication of whether the Coexistence Specification IE, as defined in 7.4.4.9, is to be included in the Enhanced Beacon frame. If this value is TRUE, the Enhanced Beacon frame will include the Coexistence Specification IE. If this value is FALSE, the Enhanced Beacon frame will not include the Coexistence Specification IE.                             | FALSE   |
| <i>macNbPanEnhancedBeaconOrder</i>                | Integer | 0–16383               | Specification of how often the coordinator transmits an Enhanced Beacon frame in a nonbeacon-enabled PAN (i.e., <i>macBeaconOrder</i> = 15). If <i>macNbPanEnhancedBeaconOrder</i> = 16383, no Enhanced Beacon frame will be transmitted.                                                                                                                          | 16383   |
| <i>macOffsetTimeSlot</i>                          | Integer | 1–15                  | The offset between the start of the periodic beacon transmission and the start of the following Enhanced Beacon frame transmission expressed in superframe time slots.                                                                                                                                                                                             | 15      |
| <i>macFcsType</i>                                 | Integer | 0–1                   | The type of the FCS, as defined in 7.2.10. A value of zero indicates a 4-octet FCS. A value of one indicates a 2-octet FCS. This attribute is only valid for LECIM, TVWS, and SUN PHYs.                                                                                                                                                                            | 0       |
| <i>macTimestampSupported</i> <sup>†</sup>         | Boolean | TRUE, FALSE           | Indication of whether the MAC sublayer supports the optional timestamping feature for incoming and outgoing Data frames.                                                                                                                                                                                                                                           | —       |
| <i>macTransactionPersistenceTime</i> <sup>♦</sup> | Integer | 0x0000–0xffff         | The maximum time (in unit periods) that a transaction is stored by a coordinator and indicated in its beacon.<br>The unit period is governed by <i>macBeaconOrder</i> , <i>BO</i> , as follows:<br>For $0 \leq BO \leq 14$ , the unit period will be $aBaseSuperframeDuration \times 2^{BO}$ . For $BO = 15$ , the unit period will be $aBaseSuperframeDuration$ . | 0x01f4  |
| <i>macImplicitBroadcast</i>                       | Boolean | TRUE, FALSE           | Indicates whether frames without a destination PAN ID and a destination address are to be treated as though they are addressed to the broadcast PAN ID and broadcast short address.                                                                                                                                                                                | FALSE   |
| <i>macLecimAlohaUnitBackoffPeriod</i>             | Integer | As defined in 6.2.5.5 | The number of symbols for backoff when PCA backoff algorithm is in use, as defined in 6.2.5.5.                                                                                                                                                                                                                                                                     | —       |

Table 8-81—MAC PIB attributes (continued)

| Attribute                        | Type    | Range                                                     | Description                                                                                                                                                                                                                           | Default            |
|----------------------------------|---------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <i>macLecimAlohaBe</i>           | Integer | 0– <i>macMinBe</i>                                        | The value of the constant backoff exponent for priority messages using CCA Mode 4 (ALOHA), as described in 6.2.5.5.                                                                                                                   | <i>macMinBe</i> –1 |
| <i>macPriorityChannelAccess</i>  | Boolean | TRUE, FALSE                                               | Indicates whether PCA is enabled. A value of TRUE indicates that it is enabled, while a value of FALSE indicates that it is disabled.                                                                                                 | FALSE              |
| <i>macPcaAllocationSuperRate</i> | Boolean | TRUE, FALSE                                               | Indicates the PCA allocation rate per superframe. A value of TRUE indicates one or more allocations per superframe. A value of FALSE indicates less than one allocation per superframe.                                               | TRUE               |
| <i>macPcaAllocationRate</i>      | Integer | Minimum rate defined in 6.2.5.4; the maximum rate is 255. | The PCA allocation rate. If <i>macPcaAllocationSuperRate</i> is TRUE, the value is the number of allocations per superframe. If <i>macPcaAllocationSuperRate</i> is FALSE, the value is the number of superframes per PCA allocation. | 1                  |
| <i>macCritMsgDelayTol</i>        | Integer | 0–65 532                                                  | The maximum transaction delay, in milliseconds, for a critical event message before issuing an MCPS-DATA.confirm with Status set to TRANSACTION_EXPIRED, as defined in 7.4.4.15.                                                      | 15 000             |
| <i>macStartBandEdge</i>          | Integer | 0–16 777 215                                              | Frequency in kilohertz indicating the lower edge of the band.                                                                                                                                                                         | 608 000            |
| <i>macEndBandEdge</i>            | Integer | 0–16 777 215                                              | Frequency in kilohertz indicating the upper edge of the band.                                                                                                                                                                         | 614 000            |
| <i>macGroupRxMode</i>            | Boolean | TRUE, FALSE                                               | Enables the reception of EUI-64 group addresses.                                                                                                                                                                                      | FALSE              |
| <i>macTmctpExtendedOrder</i>     | Integer | 0–14                                                      | The extended length of the active portion of the superframe, as defined in 6.2.8.                                                                                                                                                     | 0                  |

#### 8.4.2.1 General MAC PIB attributes for functional organization

Table 8-82 provides the General MAC PIB attributes for functional organization. Attributes marked with a dagger (†) are read-only attributes (i.e., attribute may 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.

**Table 8-82—General MAC PIB attributes for functional organization**

| Attribute                                  | Type    | Range          | Description                                                                                                                                                                                          | Default |
|--------------------------------------------|---------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macTschCapable</i>                      | Boolean | TRUE,<br>FALSE | If TRUE, the device is capable of functionality specific to TSCH.                                                                                                                                    | —       |
| <i>macDsmeCapable</i>                      | Boolean | TRUE,<br>FALSE | If TRUE, the device is capable of functionality specific to DSME.                                                                                                                                    | —       |
| <i>macLeCapable</i>                        | Boolean | TRUE,<br>FALSE | If TRUE, the device is capable of functionality specific to low energy.                                                                                                                              | —       |
| <i>macHoppingCapable</i>                   | Boolean | TRUE,<br>FALSE | If TRUE, the device is capable of unslotted channel hopping.                                                                                                                                         | —       |
| <i>macMetricsCapable</i>                   | Boolean | TRUE,<br>FALSE | If TRUE, the device is capable of providing additional MAC metrics.                                                                                                                                  | —       |
| <i>macTschEnabled</i>                      | Boolean | TRUE,<br>FALSE | If TRUE, the device is using functionality specific to TSCH.                                                                                                                                         | —       |
| <i>macDsmeEnabled</i>                      | Boolean | TRUE,<br>FALSE | If TRUE, the device is using functionality specific to DSME.                                                                                                                                         | —       |
| <i>macLeEnabled</i>                        | Boolean | TRUE,<br>FALSE | If TRUE, the device is using functionality specific to low energy.                                                                                                                                   | —       |
| <i>macHoppingEnabled</i>                   | Boolean | TRUE,<br>FALSE | If TRUE, the device is using unslotted channel hopping.                                                                                                                                              | —       |
| <i>macMetricsEnabled</i>                   | Boolean | TRUE,<br>FALSE | If TRUE, the device is providing additional MAC metrics.                                                                                                                                             | —       |
| <i>macExtendedDsmeCapable</i> <sup>†</sup> | Boolean | TRUE,<br>FALSE | A value of TRUE indicates that the device is capable of functionality specific to ExtendedDSME. A value of FALSE indicates that the device is not capable of functionality specific to ExtendedDSME. | —       |
| <i>macExtendedDsmeEnabled</i>              | Boolean | TRUE,<br>FALSE | A value of TRUE indicates that the device is using functionality specific to ExtendedDSME. A value of FALSE indicates that the device is not using functionality specific to ExtendedDSME.           | FALSE   |

**Table 8-82—General MAC PIB attributes for functional organization (continued)**

| Attribute                  | Type           | Range                  | Description                                                                                                                                                                                        | Default |
|----------------------------|----------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macLeHsEnabled</i>      | Boolean        | TRUE, FALSE            | A value of TRUE indicates that the device is using functionality specific to low-energy handshake. A value of FALSE indicates that the low-energy handshake is not used.                           | —       |
| <i>macRelayingMode</i>     | Boolean        | TRUE, FALSE            | If TRUE, the device is using TRLE relaying mode, as defined in F.3. If FALSE, the device is not using TRLE relaying mode.                                                                          | —       |
| <i>macTrleEnabled</i>      | Boolean        | TRUE, FALSE            | If TRUE, the device is using functionality specific to TRLE. If FALSE, the device is not operating as a TRLE PAN.                                                                                  | —       |
| <i>macRccnEnabled</i>      | Boolean        | TRUE, FALSE            | An indication of whether the device is using functionality specific to an RCCN. If TRUE, the device is using this functionality. If FALSE, it is not.                                              | —       |
| <i>macRccnNumTimeSlots</i> | Integer        | 0–254                  | The number of timeslots within a superframe, excluding the timeslot for beacon frame and management timeslots.                                                                                     | 48      |
| <i>macRccnNumMgmtTs</i>    | Integer        | 0–254                  | The number of management timeslots.                                                                                                                                                                | 4       |
| <i>macRccnNumGtsTs</i>     | Integer        | 0–254                  | The number of GTS timeslots.                                                                                                                                                                       | 24      |
| <i>macRccnNetId</i>        | Set of octets  | —                      | A network specific identification.                                                                                                                                                                 | 0       |
| <i>macRccnDevType</i>      | Enumeration    | RCCPANC, MOBILE, FIXED | Indicates the type of RCCN device. It may be one of the following device types: a PAN coordinator, a mobile device that is not a PAN coordinator, or a fixed device that is not a PAN coordinator. | —       |
| <i>macRccnCap†</i>         | List of octets | —                      | The PHY capabilities supported.                                                                                                                                                                    | —       |

**8.4.2.2 TSCH-specific MAC PIB attributes**

Table 8-83 contains the PIB values for TSCH.

**Table 8-83—TSCH-specific MAC PIB attributes**

| Attribute                | Type    | Range                   | Description                                                                                                                                                                                                                                      | Default                |
|--------------------------|---------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <i>macMinBe</i>          | Integer | 0— <i>macMaxBe</i>      | The minimum value of the backoff exponent (BE) in the CSMA-CA algorithm, as defined in 6.2.5.1, or the TSCH-CA algorithm, as defined in 6.2.5.3.                                                                                                 | 3—CSMA-CA<br>1—TSCH-CA |
| <i>macMaxBe</i>          | Integer | 3—8                     | The maximum value of the BE in the CSMA-CA algorithm, as defined in 6.2.5.1, or the TSCH-CA algorithm, as defined in 6.2.5.3.                                                                                                                    | 5—CSMA-CA<br>7—TSCH-CA |
| <i>macDisconnectTime</i> | Integer | 0x0000—0xffff           | Time (in Timeslots) to send out Disassociate frames before disconnecting.                                                                                                                                                                        | 0xff                   |
| <i>macJoinMetric</i>     | Integer | 0x00—0xff               | The sum of one and the value of the Join Metric field from the TSCH Synchronization IE, 7.4.4.2, received in the Enhanced Beacon frame used by the device joining the network. If the device is the an endpoint, the value shall be set to zero. | 1                      |
| <i>macAsn</i>            | Integer | 0x0000000000—0xffffffff | The Absolute Slot Number, i.e., the number of slots that has elapsed since the start of the network. Used for nonce construction when security is enabled.                                                                                       | 0x0000000000           |

**8.4.2.2.1 TSCH MAC PIB attributes for *macSlotframeTable***

The attributes contained in the MAC PIB for *macSlotframeTable* are presented in Table 8-84. Each slotframe requires a *macSlotframeTable* to be stored.

**Table 8-84—TSCH MAC PIB attributes for *macSlotframeTable***

| Attribute                 | Type    | Range         | Description                                           | Default |
|---------------------------|---------|---------------|-------------------------------------------------------|---------|
| <i>macSlotframeHandle</i> | Integer | 0x00—0xff     | Identifier of the slotframe, as described in 6.2.6.1. | —       |
| <i>macSlotframeSize</i>   | Integer | 0x0000—0xffff | Number of timeslots in the slotframe.                 | —       |

**8.4.2.2.2 TSCH MAC PIB attributes for *macLinkTable***

The attributes contained in the MAC PIB for *macLinkTable* are presented in Table 8-85. Each link requires a *macLinkTable* to be stored.

**Table 8-85—TSCH MAC PIB attributes for *macLinkTable***

| Attribute                 | Type                              | Range                                     | Description                                                                                                       | Default |
|---------------------------|-----------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------|
| <i>macLinkHandle</i>      | Integer                           | 0x0000–0xffff                             | Identifier of Link.                                                                                               | —       |
| <i>macTxType</i>          | Boolean                           | TRUE, FALSE                               | Set to TRUE if the link is a TX link, otherwise set to FALSE.                                                     | —       |
| <i>macRxType</i>          | Boolean                           | TRUE, FALSE                               | Set to TRUE if the link is a RX link, otherwise set to FALSE.                                                     | —       |
| <i>macSharedType</i>      | Boolean                           | TRUE, FALSE                               | Set to TRUE if the link is a shared link, otherwise set to FALSE.                                                 | —       |
| <i>macLinkTimekeeping</i> | Boolean                           | TRUE, FALSE                               | Set to TRUE if the link is a timekeeping link, FALSE otherwise.                                                   | —       |
| <i>macPriorityType</i>    | Boolean                           | TRUE, FALSE                               | Set to TRUE if the link is to be used for high priority traffic, as described in 6.2.5.2, otherwise set to FALSE. | —       |
| <i>macLinkType</i>        | Enumeration                       | NORMAL, ADVERTISING                       | Type of link.                                                                                                     | NORMAL  |
| <i>macSlotframeHandle</i> | Integer                           | 0x00–0xff                                 | Identifier of Slotframe to which this link belongs.                                                               | —       |
| <i>macNodeAddressMode</i> | Enumeration                       | SHORT, EXTENDED                           | Addressing mode of the neighbor device connected to the link.                                                     | —       |
| <i>macNodeAddress</i>     | Short address or extended address | As specified by <i>macNodeAddressMode</i> | Address of the neighbor device connected to this link or the broadcast address.                                   | —       |
| <i>macTimeslot</i>        | Integer                           | 0x0000–0xffff                             | Timeslot for this link.                                                                                           | —       |
| <i>macChannelOffset</i>   | Integer                           | 0x0000–0xffff                             | Channel offset for this link.                                                                                     | —       |



**8.4.2.2.3 TSCH MAC PIB attributes for *macTimeslotTemplate***

The attributes contained in the MAC PIB for *macTimeslotTemplate* are presented in Table 8-86.

**Table 8-86—TSCH MAC PIB attributes for *macTimeslotTemplate***

| Attribute                    | Type    | Range           | Description                                                                                                         | Band defaults |         |
|------------------------------|---------|-----------------|---------------------------------------------------------------------------------------------------------------------|---------------|---------|
|                              |         |                 |                                                                                                                     | 2450 MHz      | 915 MHz |
| <i>macTimeslotTemplateId</i> | Integer | 0x0–0xf0        | Identifier of Timeslot Template.                                                                                    | 0x00          | 0x00    |
| <i>macTsCcaOffset</i>        | Integer | 0x0000–0xffff   | The time between the beginning of timeslot and start of CCA operation, in $\mu$ s.                                  | 1800          | 1800    |
| <i>macTsCca</i>              | Integer | 0x0000–0xffff   | Duration of CCA, in $\mu$ s.                                                                                        | 128           | 128     |
| <i>macTsTxOffset</i>         | Integer | 0x0000–0xffff   | The time between the beginning of the timeslot and the start of frame transmission, in $\mu$ s.                     | 2120          | 2800    |
| <i>macTsRxOffset</i>         | Integer | 0x0000–0xffff   | Beginning of the timeslot to when the receiver shall be listening, in $\mu$ s.                                      | 1020          | 1800    |
| <i>macTsRxAckDelay</i>       | Integer | 0x0000–0xffff   | End of frame to when the transmitter shall listen for acknowledgment, in $\mu$ s.                                   | 800           | 800     |
| <i>macTsTxAckDelay</i>       | Integer | 0x0000–0xffff   | End of frame to start of acknowledgment, in $\mu$ s.                                                                | 1000          | 1000    |
| <i>macTsRxWait</i>           | Integer | 0x0000–0xffff   | The time to wait for start of frame, in $\mu$ s.                                                                    | 2200          | 6000    |
| <i>macTsRxTx</i>             | Integer | 0x0000–0xffff   | Transmit to Receive turnaround, in $\mu$ s.                                                                         | 192           | 1000    |
| <i>macTsMaxAck</i>           | Integer | 0x0000–0xffff   | Transmission time to send an acknowledgment, in $\mu$ s.                                                            | 2400          | 6000    |
| <i>macTsMaxTx</i>            | Integer | 0x00000–0xfffff | Transmission time to send the maximum length frame, in $\mu$ s.                                                     | 4256          | 103040  |
| <i>macTsTimeslotLength</i>   | Integer | 0x00000–0xfffff | The total length of the timeslot including any unused time after frame transmission and acknowledgment, in $\mu$ s. | 10000         | 120000  |
| <i>macTsAckWait</i>          | Integer | 0x0000–0xffff   | The minimum time to wait for the start of an acknowledgment in $\mu$ s.                                             | 400           |         |

**8.4.2.3 MAC PIB attributes for hopping sequence**

The attributes contained in the MAC PIB for hopping sequence are presented in Table 8-87.

**Table 8-87—MAC PIB attributes for hopping sequence**

| Attribute                       | Type            | Range                          | Description                                                                                                                                                                                                                                                                                                                   | Default |
|---------------------------------|-----------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macHoppingSequenceId</i>     | Integer         | 0x00–0x0f                      | The unique ID of the hopping sequence.                                                                                                                                                                                                                                                                                        | 0       |
| <i>macChannelPage</i>           | Integer         | 0x00–0x1f                      | The channel page for the hopping sequence. Note this may not correspond to the current channelPage.                                                                                                                                                                                                                           | —       |
| <i>macNumberOfChannels</i>      | Integer         | 0x0000–0x01ff                  | Number of channels supported by the PHY on this channelPage.                                                                                                                                                                                                                                                                  | —       |
| <i>macPhyConfiguration</i>      | Integer         | 0x00000000–0x7fffffff          | For channel pages other than 9 and 10, the 27 LSBs (b0, b1, ..., b26) indicate the status (1 = to be used, 0 = not to be used) for each of the up to 27 channels available to the PHY. For pages 9 and 10, the 27 LSBs indicate the configuration of the PHY, and the channel list is contained in the Extended Bitmap field. | —       |
| <i>macExtendedBitmap</i>        | Bitmap          | varies                         | For channel pages 9 and 10, a bitmap of <i>macNumberOfChannels</i> bits, where $b_k$ shall indicate the status of channel $k$ for each of the up to <i>macNumberOfChannels</i> valid channels supported by that channel page and phyConfiguration. Otherwise it is a null string.                                             | —       |
| <i>macHoppingSequenceLength</i> | Integer         | 0x0000–0xffff                  | The number of channels in the Hopping Sequence. Does not necessarily equal <i>macNumberOfChannels</i> .                                                                                                                                                                                                                       | —       |
| <i>macHoppingSequenceList</i>   | Set of Integers | 0x0000–0x01ff for each channel | A <i>macHoppingSequenceLength</i> -element set of channels to be hopped over.                                                                                                                                                                                                                                                 | —       |
| <i>macHopDwellTime</i>          | Integer         | 0x0000–0xffff                  | For unslotted channel hopping modes, this field is the channel dwell time, in units of 10 $\mu$ s. For other modes, the field is empty.                                                                                                                                                                                       | —       |

## 8.4.2.4 DSME specific MAC PIB attributes

DSME specific attributes are described in Table 8-88 and Table 8-89.

Table 8-88—DSME-specific MAC PIB attributes

| Attribute                           | Type                                                        | Range                | Description                                                                                                                                                                                                                                              | Default    |
|-------------------------------------|-------------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <i>macChannelIndex</i>              | Integer                                                     | 0–31                 | Specifies the channel index of the channel's DSME link reported by the source device.                                                                                                                                                                    | —          |
| <i>macAvgLqi</i>                    | Integer                                                     | 0x00–0xff            | A characterization of the link quality between a source device and a destination device on the channel defined by Channel Index, the measurement shall be performed for each received packet during a period of <i>LinkStatusStatisticPeriod</i> .       | —          |
| <i>macAvgRssi</i>                   | Integer                                                     | 0–255                | Average RSSI.                                                                                                                                                                                                                                            | —          |
| <i>macLinkStatusStatisticPeriod</i> | Integer                                                     | 0x000000–0xfffff     | The time interval between two times of link status statistics, which is defined as $\text{LinkStatusStatisticPeriod} = aBaseSuperframeDuration \times 2^{\text{MO}}$ symbols. If the parameter equals to 0x000000, link status statistic is not allowed. | —          |
| <i>macCapReduction</i>              | Boolean                                                     | TRUE, FALSE          | Indication of whether the CAP reduction is enabled. A value of TRUE indicates that the CAP reduction is enabled.                                                                                                                                         | FALSE      |
| <i>macChannelDiversityMode</i>      | Enumeration                                                 | ADAPTATION, HOPPING  | Indicates the method of channel diversity in a beacon-enabled PAN, either channel adaptation or channel hopping.                                                                                                                                         | ADAPTATION |
| <i>macMultisuperframeOrder</i>      | Integer                                                     | 0–22                 | The length of a multi-superframe, which is a cycle of the repeated superframes.                                                                                                                                                                          | 15         |
| <i>macConnecDev</i>                 | Boolean                                                     | TRUE, FALSE          | Indication of whether the device is a Connection Device or not. If this attribute is TRUE, the device is a Connection Device. This attribute shall be set to FALSE if the device is not a Connection Device.                                             | FALSE      |
| <i>macDsmeSab</i>                   | Bitmap                                                      | As defined in 7.5.18 | The slot allocation bitmap of the DSME-GTS schedule.                                                                                                                                                                                                     | 0          |
| <i>macDsmeAct</i>                   | List of Allocation Counter Tables, as defined in Table 8-90 | —                    | A list of allocation counter tables of the DSME GTSS allocated to the device.                                                                                                                                                                            | 0          |
| <i>macSdIndex</i>                   | Integer                                                     | 0x0000–0xffff        | Specifies the allocating SD index number for beacon frame.                                                                                                                                                                                               | 0x0000     |

**Table 8-88—DSME-specific MAC PIB attributes (continued)**

| Attribute                           | Type                                                   | Range                 | Description                                                                                                                                                                                                                                                                                                                                       | Default |
|-------------------------------------|--------------------------------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macSdBitmap</i>                  | Bitmap                                                 | As defined in 7.4.2.5 | Indicates the beacon frame allocation information of neighbor nodes. This field is expressed in bitmap format that orderly represents the schedule of beacons, with corresponding bit shall be set to one if a beacon is allocated in that SD.                                                                                                    | —       |
| <i>macChannelOffset</i>             | Integer                                                | 0x0000–0xffff         | ChannelOffset is the offset value of Hopping Sequence.                                                                                                                                                                                                                                                                                            | 0       |
| <i>macDeferredBeaconUsed</i>        | Boolean                                                | TRUE, FALSE           | Indication of whether the device uses CCA before transmitting beacon frame. A value of TRUE indicates that the device uses CCA before transmitting beacon frame.                                                                                                                                                                                  | FALSE   |
| <i>macSyncParentExtendedAddress</i> | IEEE address                                           | —                     | The extended address of the coordinator through which the device is synchronized.                                                                                                                                                                                                                                                                 | —       |
| <i>macSyncParentShortAddress</i>    | Integer                                                | 0x0000–0xffff         | The short address assigned to the coordinator through which the device is synchronized. A value of 0xffff indicates that the coordinator is only using its extended address. A value of 0xffff indicates that this value is unknown.                                                                                                              | 0xffff  |
| <i>macSyncParentSdIndex</i>         | Integer                                                | 0x0000–0xffff         | Indication of SD index the synchronized parent used.                                                                                                                                                                                                                                                                                              | 0       |
| <i>macChannelStatus</i>             | List of Link-Report entries as described in Table 8-58 | Refer to Table 8-58   | Link report for each used channel.                                                                                                                                                                                                                                                                                                                | —       |
| <i>macBeaconSlotLength</i>          | Integer                                                | 0x0000–0xffff         | The number of symbols forming a beacon slot.                                                                                                                                                                                                                                                                                                      | 60      |
| <i>macDsmeGtsExpirationTime</i>     | Integer                                                | 0x00–0xff             | The number of idle incidents before expiring a DSME GTS.                                                                                                                                                                                                                                                                                          | 7       |
| <i>macChannelOffsetBitmapLength</i> | Integer                                                | 0x00–0xff             | Specifies the length of ChannelOffsetBitmap in octets.                                                                                                                                                                                                                                                                                            | —       |
| <i>macChannelOffsetBitmap</i>       | Set of octets                                          | —                     | A bitmap that represents whether the corresponding channel offset is used. If the corresponding channel offset is used, the bit value shall be set to one. Otherwise, it shall be set to zero. For instance, if the 1st, 2nd, 4th channels offset are used with ChannelOffsetBitmapLength of 16, ChannelOffsetBitmap shall be 0b0110100000000000. | —       |
| <i>macPanCoordinatorBsn</i>         | Integer                                                | 0x00–0xff             | The sequence number added to the transmitted beacon frame of a PAN coordinator.                                                                                                                                                                                                                                                                   | —       |

**Table 8-88—DSME-specific MAC PIB attributes (continued)**

| Attribute                          | Type                                                            | Range       | Description                                                                                                                                                                 | Default |
|------------------------------------|-----------------------------------------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macNeighborInformationTable</i> | List of Neighbor Information entries as described in Table 8-89 | —           | A table of the neighbor device's information entries.                                                                                                                       | Null    |
| <i>macAllocationOrder</i>          | Integer                                                         | 0–8         | As defined in 7.5.13. If MO < BO, the value shall be set to zero.                                                                                                           | 0       |
| <i>macBilIndex</i>                 | Integer                                                         | 0–255       | As defined in 7.5.13.                                                                                                                                                       | 0       |
| <i>macDsmeAssociation</i>          | Boolean                                                         | TRUE, FALSE | Indicates whether DSME GTSS are allocated during the association procedure. This attribute is set to TRUE if a device requests assignment of a DSME GTS during association. | TRUE    |

**Table 8-89—Elements of Neighbor Information**

| Attribute              | Type         | Range               | Description                                                                                                                                                                                     |
|------------------------|--------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ShortAddress</i>    | Integer      | 0x0000–0xffff       | The short address of the neighbor device.                                                                                                                                                       |
| <i>ExtendedAddress</i> | IEEE address | An extended address | The extended address of the neighbor device.                                                                                                                                                    |
| <i>SdIndex</i>         | Integer      | 0x0000–0xffff       | The allocating SD index number for beacon frame.                                                                                                                                                |
| <i>ChannelOffset</i>   | Integer      | 0x0000–0xffff       | The offset value of ChannelHoppingSequence.                                                                                                                                                     |
| <i>TrackBeacon</i>     | Boolean      | TRUE, FALSE         | TRUE if the MLME is to track all future beacons of the neighbor device. FALSE if the MLME is not to track beacon of the neighbor device.                                                        |
| <i>BeaconLostCount</i> | Integer      | 0x00–0xff           | The number of consecutive lost beacons.                                                                                                                                                         |
| <i>AllocationOrder</i> | Integer      | 0–8                 | As defined in 7.5.13.                                                                                                                                                                           |
| <i>AssociationType</i> | Integer      | 0–1                 | Indicates whether DSME GTSS are allocated during the association procedure. This element shall be set to one if DSME GTSS are allocated during association. Otherwise, it shall be set to zero. |

The Allocation Counter Table format is defined in Table 8-90.

**Table 8-90—Elements of allocation counter table**

| Field                             | Type          | Valid range    | Description                                                                                                                                          |
|-----------------------------------|---------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Superframe ID                     | Integer       | 0x0000–0xffffd | The superframe ID of the DSME GTS in a multi-superframe.                                                                                             |
| Slot ID                           | Integer       | 0–14           | The slot ID of the DSME GTS in the superframe.                                                                                                       |
| Channel ID                        | Integer       | 0x0000–0xffff  | In channel adaptation, this field shall contain the Channel number of the DSME GTS. In channel hopping, this field shall contain the Channel Offset. |
| Direction                         | Enumeration   | TX, RX         | The direction of the allocated DSME-GTS.                                                                                                             |
| Prioritized Channel Access        | Boolean       | TRUE, FALSE    | If set to TRUE, the DSME GTS is high priority, otherwise FALSE and the DSME GTS is low priority.                                                     |
| Source/<br>Destination<br>Address | Short address | 0x0000–0xffffd | The short address of the device that is the source (if RX) or the destination of the allocated DSME-GTS.                                             |
| Counter                           | Integer       | 0x0000–0xffff  | An idle counter, in other word, the number of idle multi-superframes since the allocated DSME-GTS was used.                                          |
| LinkQuality                       | Integer       | 0x0000–0xffff  | The link quality of the allocated DSME-GTS.                                                                                                          |
| Allocation Order                  | Integer       | 0x00–0x08      | As defined in 5.3.11.3.6. If $MO \leq BO$ , the value of <i>macAllocationOrder</i> shall be set to zero.                                             |
| BI Index                          | Integer       | 0x00–0xff      | As defined in 5.3.11.3.7.                                                                                                                            |

#### 8.4.2.5 LE specific MAC PIB attributes

LE specific attributes are described in Table 8-91.

**Table 8-91—LE-specific MAC PIB attributes**

| Attribute              | Type    | Range    | Description                                                                                                                                                                                                                                               | Default             |
|------------------------|---------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <i>macCslPeriod</i>    | Integer | 0–0xffff | CSL sampled listening period in units of 10 symbols. Zero means always listening, i.e., CSL off.                                                                                                                                                          | 0                   |
| <i>macCslMaxPeriod</i> | Integer | 0–0xffff | Maximum CSL sampled listening period in units of 10 symbols in the entire PAN. This determines the length of the wake-up sequence when communicating to a device whose CSL listen period is unknown. It is set to zero to stop sending wake-up sequences. | <i>macCslPeriod</i> |

Table 8-91—LE-specific MAC PIB attributes (continued)

| Attribute                                 | Type          | Range                                                        | Description                                                                                                                                                                                                                                                                                                                                                | Default    |
|-------------------------------------------|---------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <i>macCslChannelMask</i>                  | Bitmap        | —                                                            | 32-bit bitmap relative to <i>phyCurrentPage</i> of channels. It represents the list of channels CSL operates on. Zero means CSL operates on <i>phyCurrentChannel</i> of <i>phyCurrentPage</i> .                                                                                                                                                            | 0x00000000 |
| <i>macCslFramePendingWait</i>             | Integer       | <i>macLifsPeriod</i> + max number of symbols per PPDU–0xffff | Number of symbols to keep the receiver on after receiving a payload frame with Frame Control field frame pending bit set to one.                                                                                                                                                                                                                           | —          |
| <i>macRitPeriod</i>                       | Integer       | 0x000000–0xffff                                              | The interval (in unit periods) for periodic transmission of RIT Data Request command in RIT mode. The unit period is <i>aBaseSuperframeDuration</i> . A value of zero indicates that RIT is off.                                                                                                                                                           | 0          |
| <i>macRitDataWaitDuration</i>             | Integer       | 0x00–0xff                                                    | The maximum time (in unit period) to wait for Data frame after transmission of RIT Data Request command in RIT mode. The unit period is <i>aBaseSuperframeDuration</i> .                                                                                                                                                                                   | 0          |
| <i>macRitTxWaitDuration</i>               | Integer       | <i>Period</i> –0xffff                                        | The maximum time (in unit periods) that a transaction is stored by a device in RIT mode. The unit period is <i>aBaseSuperframeDuration</i> .                                                                                                                                                                                                               | 0          |
| <i>macRitRequestPayload</i>               | Set of octets | —                                                            | The payload to include an an RIT Data Request command.                                                                                                                                                                                                                                                                                                     |            |
| <i>macLowEnergySuperframeSupported</i>    | Boolean       | TRUE, FALSE                                                  | Indication of whether the low-energy superframe is operational or not. If this attribute is TRUE, the coordinator shall not transmit beacon frames regardless of BO value. This attribute shall be set to FALSE if the device is aware of the existence of allocated GTS.                                                                                  | —          |
| <i>macLowEnergySuperframeSyncInterval</i> | Integer       | 0x0000–0xffff                                                | Indication of beacon transmission interval when <i>macLowEnergySuperframeSupported</i> is TRUE. The beacon transmission interval shall be indicated by <i>macLowEnergySuperframeSyncInterval</i> times the beacon interval in the case this attribute is not zero. The beacon shall be transmitted only when requested in the case this attribute is zero. | —          |
| <i>macCslInterval</i>                     | Integer       | 0–0xffff                                                     | Specifies the interval between two successive CSL wake-up frames in the wake-up sequence, in units of 10 symbols.                                                                                                                                                                                                                                          | —          |

**Table 8-91—LE-specific MAC PIB attributes (continued)**

| Attribute                    | Type    | Range         | Description                                                                                                                                                                                        | Default |
|------------------------------|---------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macIRitOffsetInterval</i> | Integer | 0x0000–0xffff | A value of zero indicates that I-RIT is disabled. A nonzero value specifies the interval, in symbol periods, from the end of the transmitted frame to the beginning of the I-RIT listening period. | 0x00    |
| <i>macIRitListenDuration</i> | Integer | 0x00–0xff     | The duration of listening time, in symbol periods, for which the receiver is listening for the beginning of a frame to receive.                                                                    | 0x64    |
| <i>macIRitEnabled</i>        | Boolean | TRUE, FALSE   | If TRUE, the I-RIT mode of operation is enabled, as described in 6.12.4. If FALSE, the I-RIT mode of operation is disabled.                                                                        | FALSE   |

**8.4.2.6 MAC performance metrics specific MAC PIB attributes**

If *macMetricsEnabled* is TRUE, the MAC shall collect the metrics listed in Table 8-92.

The counters listed in Table 8-92 shall wrap to 0 when incremented beyond their maximum value. *macCounterOctets* is read only, other MAC PIB attributes in Table 8-92 are read/write and may be reset by higher layers by writing a 0 value.

The attributes *macRetryCount*, *macMultipleRetryCount*, *macTxFailCount*, and *macTxSuccessCount* relate to Data frame transmission. Each MSDU transferred into the MAC layer through the MCPS-DATA.request primitive shall increment exactly one of these four attribute counters depending on the final disposition of the frame as described in 6.7.2.

The attributes *macFcsErrorCount*, *macSecurityFailure*, *macDuplicateFrameCount*, and *macRxSuccessCount* relate to Data frame reception. Each frame received by the MAC layer shall increment at least one of these four attribute counters based on the status of the frame.

**Table 8-92—Metrics-specific MAC PIB attributes**

| Attribute                    | Type    | Range                                        | Description                                                                                                 | Default |
|------------------------------|---------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------|---------|
| <i>macCounterOctets</i>      | Integer | 1–4                                          | The size of the MAC metrics counters in octets.                                                             | —       |
| <i>macRetryCount</i>         | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of transmitted frames that required exactly one retry before acknowledgment.                     | 0       |
| <i>macMultipleRetryCount</i> | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of transmitted frames that required more than one retry before acknowledgment.                   | 0       |
| <i>macTxFailCount</i>        | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of transmitted frames that did not result in an acknowledgment after <i>macMaxFrameRetries</i> . | 0       |



**Table 8-92—Metrics-specific MAC PIB attributes (continued)**

| Attribute                     | Type    | Range                                        | Description                                                                                                             | Default |
|-------------------------------|---------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macTxSuccessCount</i>      | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of transmitted frames that were acknowledged successfully after the initial Data frame transmission.         | 0       |
| <i>macFcsErrorCount</i>       | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of frames that were discarded due to an incorrect FCS.                                                       | 0       |
| <i>macSecurityFailure</i>     | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of frames that were returned from the procedure described in 9.2.3 with any Status other than SUCCESS.       | 0       |
| <i>macDuplicateFrameCount</i> | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of frames that contained the same sequence number as a frame previously received.                            | 0       |
| <i>macRxSuccessCount</i>      | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of frames that were received correctly.                                                                      | 0       |
| <i>macNackCount</i>           | Integer | $0-2^{8 \times \text{macCounterOctets}} - 1$ | The number of transmitted frames that were acknowledged with a Time Correction IE indicating a negative acknowledgment. | 0       |

**8.4.2.7 Enhanced Beacon Request command specific MAC PIB attributes**

Enhanced Beacon Request command specific attributes are described in Table 8-93.

**Table 8-93—Enhanced Beacon Request command specific MAC PIB attributes**

| Attribute                   | Type                          | Range       | Description                                                                                                                                                                                                | Default |
|-----------------------------|-------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macEbrPermitJoining</i>  | Boolean                       | TRUE, FALSE | When TRUE, the Permit Joining request will be included in the Enhanced Beacon Request command.                                                                                                             | —       |
| <i>macEbrFilters</i>        | List of Booleans              |             | Contains which Enhanced Beacon Request command filter field bits should be set.                                                                                                                            | —       |
| <i>macEbrLinkQuality</i>    | Integer                       | 0x00–0xff   | Link quality level to be transmitted in the Enhanced Beacon Request command.                                                                                                                               | —       |
| <i>macEbrPercentFilter</i>  | Integer                       | 0–100       | Percent filter threshold value to be transmitted in the Enhanced Beacon Request command.                                                                                                                   | —       |
| <i>macEbrAttributeList</i>  | List of attribute identifiers | 0x00–0xff   | Contains attribute identifiers, as defined in Table 6-3.                                                                                                                                                   | —       |
| <i>macBeaconAutoRespond</i> | Boolean                       | TRUE, FALSE | When TRUE, device responds to beacon requests and enhanced beacon requests automatically. When FALSE, device passes beacon/enhanced beacon payload up to higher layer using MLME-BEACONREQUEST.indication. | TRUE    |

**8.4.2.8 Enhanced Beacon frame specific MAC PIB attributes**

Enhanced Beacon frame specific attributes are described in Table 8-94.

**Table 8-94—Enhanced Beacon frame specific MAC PIB attributes**

| Attribute                    | Type                | Range             | Description                                                                                                                 | Default |
|------------------------------|---------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------|---------|
| <i>macUseEnhancedBeacon</i>  | Boolean             | TRUE, FALSE       | When TRUE, in a beacon-enabled PAN the device should use Enhanced Beacons rather than standard beacons.                     | —       |
| <i>macEbHeaderIeList</i>     | List of Header IEs  | As defined in 7.4 | A list of Header IEs to include in the enhanced beacon.                                                                     | —       |
| <i>macEbPayloadIeList</i>    | List of Payload IEs | As defined in 7.4 | A list of Payload IEs to include in the enhanced beacon.                                                                    | —       |
| <i>macEbFilteringEnabled</i> | Boolean             | TRUE, FALSE       | Indicates if devices should perform filtering in response to Enhanced Beacon Request command.                               | —       |
| <i>macEbsn</i>               | Integer             | 0x00–0xff         | BSN used for Enhanced Beacon frames.                                                                                        | —       |
| <i>macEbAutoSa</i>           | Enumeration         | NONE, SHORT, FULL | Indicates if Enhanced Beacon frames generated by the MAC in response to Enhanced Beacon frame include Source Address field. | FULL    |

## 9. Security

### 9.1 Overview

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:

- Data confidentiality
- Data authenticity
- Replay protection (when not using TSCH mode)

When TSCH mode is enabled, the nonce is generated again for each retransmission, thus the receiving device has no way of distinguishing whether the frame is a retransmission or not from the ASN. Thus full replay protection in TSCH mode is not provided.

### 9.2 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 only if the *macSecurityEnabled* attribute is set to TRUE.

Security specific MAC PIB values are italicized and have a prefix of *sec*, i.e., *secKeyFrameCounter*.

Informative diagrams of the various security state machines can be found in Kivinen [B14].

#### 9.2.1 Outgoing frame security procedure

The inputs to this procedure are the frame to be secured and the *SecurityLevel*, *KeyIdMode*, *KeySource*, and *KeyIndex* parameters. If the frame was generated in response to an MLME or MCPS primitive, then the value of *SecurityLevel*, *KeyIdMode*, *KeySource* and *KeyIndex* are set to the corresponding values of the primitive parameters. Otherwise, the inputs are as follows:

- *SecurityLevel* shall be set to *secAutoRequestSecurityLevel*
- *KeyIdMode* shall be set to *secAutoRequestKeyIdMode*
- *KeySource* shall be set to *secAutoRequestKeySource*
- *KeyIndex* shall be set to *secAutoRequestKeyIndex*

The outputs from this procedure are the Status of the procedure and, if this Status is SUCCESS, the secured frame.

This procedure involves the following steps:

- a) **Is security needed?** If the *SecurityLevel* parameter is zero, the procedure shall set the secured frame to be the frame to be secured and return with a Status of SUCCESS.
- b) **Is security enabled?** If *macSecurityEnabled* is set to FALSE, the procedure shall return with a Status of UNSUPPORTED\_SECURITY.
- c) **Obtain KeyDescriptor.** The procedure shall obtain the KeyDescriptor using the KeyDescriptor lookup procedure as described in 9.2.2 using the *KeyIdMode*, *KeyIndex*, and *KeySource* with the

DeviceAddressingMode set to Destination Addressing Mode field, DevicePanId set to the Destination PAN ID field, and DeviceAddress set to the Destination Address field. If that procedure fails, the procedure shall return with a Status of UNAVAILABLE\_KEY.

d) **Check frame counter value.**

- 1) If TSCH mode is not being used and the *secFrameCounterPerKey* in the KeyDescriptor is set to FALSE and *secFrameCounter* has the value 0xffffffff, the procedure shall return with a Status of COUNTER\_ERROR.
- 2) If TSCH mode is not being used and the *secFrameCounterPerKey* in the KeyDescriptor is set to TRUE and *secKeyFrameCounter* element of the KeyDescriptor has the value 0xffffffff, the procedure shall return with a Status of COUNTER\_ERROR.

e) **Insert Auxiliary Security Header field.** The procedure shall insert the Auxiliary Security Header field in the frame to be secured, with the fields set as follows:

- 1) The Security Level field of the Security Control field shall be set to the SecurityLevel parameter.
- 2) The Key Identifier Mode field of the Security Control field shall be set to the KeyIdMode parameter.
- 3) If TSCH mode is being used, the Frame Counter Suppression field in the Security Control field shall be set to one. Otherwise, the Frame Counter Suppression field in the Security Control field shall be set to zero.
- 4) The Frame Counter field shall be set as follows:
  - i) If TSCH mode is being used, the Frame Counter field shall be omitted.
  - ii) If the *secFrameCounterPerKey* in the KeyDescriptor is set to TRUE, the Frame Counter field shall be set to *secKeyFrameCounter*.
  - iii) Otherwise, the Frame Counter field shall be set to *secFrameCounter*.
- 5) If the KeyIdMode parameter is set to a value not equal to zero, the Key Source and Key Index fields of the Key Identifier field shall be set to the KeySource and KeyIndex parameters, respectively.

f) **Secure the frame.** For the frames specified in Table 9-1, the Private Payload field and Open Payload field shall be set as indicated in the table. For frames not specified in Table 9-1, the Private Payload shall be set to the MAC Payload field and Open Payload field shall be empty. The procedure shall then use the Private Payload field, the Open Payload field, the *macExtendedAddress*, the Frame Counter field (if TSCH is not being used), the ASN (if TSCH is being used), the SecurityLevel parameter, and the *secKey* element of the KeyDescriptor to produce the secured frame according to the CCM\* transformation process defined in 9.3.4.

**Table 9-1—Exceptions to Private Payload field and Open Payload field definitions**

| Frame type                         | Private Payload field | Open Payload field                        |
|------------------------------------|-----------------------|-------------------------------------------|
| Beacon<br>(Frame Version < 2)      | Beacon Payload field  | All other fields in the MAC Payload field |
| MAC Command<br>(Frame Version < 2) | Content field         | Command ID field                          |

g) **Store frame counter.**

- 1) If not using TSCH mode and *secFrameCounterPerKey* in the KeyDescriptor is set to TRUE, the procedure shall increment *secKeyFrameCounter* element of the *secKeyDescriptor* by one.

- 2) If not using TSCH mode and *secFrameCounterPerKey* in the *KeyDescriptor* is set to FALSE, the procedure shall increment *secFrameCounter* by one.
- h) **Finish procedure.** The procedure shall return with a Status of SUCCESS.

### 9.2.2 KeyDescriptor lookup procedure

The inputs to this procedure are the *KeyIdMode*, *KeySource*, *KeyIndex*, *DeviceAddressingMode*, *DevicePanId*, and *DeviceAddress*. The outputs from this procedure are a Status and, if Status is set to SUCCESS, a *KeyDescriptor*.

This procedure involves the following steps:

- a) If the *KeyIdMode* parameter is set to 0x00, then for each *secKeyIdLookupDescriptor* with *secKeyIdMode* set to 0x00 in the *secKeyIdLookupList*:
  - 1) If the *DeviceAddressingMode* is set to NONE or if the *DevicePanId* is not set, then the *DevicePanId* shall be set to *macPanId*.
  - 2) If the *DeviceAddressingMode* is set to NONE and the frame type is beacon, then the *DeviceAddress* shall be *macCoordExtendedAddress*.
  - 3) If the *DeviceAddressingMode* is set to NONE and the frame type is not beacon, then:
    - i) If the *macCoordShortAddress* attribute is set to 0xfffe, then the *DeviceAddress* shall be set to the *macCoordExtendedAddress*.
    - ii) If the *macCoordShortAddress* attribute is set to a value of 0x0000–0xfffd, then the *DeviceAddress* shall be set to the *macCoordShortAddress*.
    - iii) If the *macCoordShortAddress* attribute is set to 0xffff, the procedure shall return with Status set to FAILED.
  - 4) If the *DeviceAddressingMode*, *DevicePanId*, and *DeviceAddress* match the *secKeyDeviceAddrMode*, *secKeyDevicePanId*, and *secKeyDeviceAddress* of a *secKeyIdLookupDescriptor*, then the procedure returns with the corresponding *secKeyDescriptor* and Status set to SUCCESS.
- b) If the *KeyIdMode* parameter is set to 0x01 and *KeyIndex* matches the *secKeyIndex* of a *secKeyIdLookupDescriptor* that has *secKeyIdMode* set to 0x01, then the procedure returns with the *KeyDescriptor* set to the corresponding *secKeyDescriptor* and Status set to SUCCESS.
- c) If the *KeyIdMode* parameter is set to 0x02 or 0x03 and *KeySource*, *KeyIdMode* and *KeyIndex* match *secKeySource*, *secKeyIdMode*, and *secKeyIndex*, respectively, of a *secKeyIdLookupDescriptor*, then the procedure returns with the *KeyDescriptor* set to the corresponding *secKeyDescriptor* and Status set to SUCCESS.
- d) The procedure shall return with Status set to FAILED.

NOTE—For broadcast frames, the *KeyDescriptor* lookup procedure will result in Status set to FAILED if implicit key identification is used. Hence, explicit key identification should be used for broadcast frames.

### 9.2.3 Incoming frame security procedure, Security Enabled field is set to one

This procedure shall only be used for incoming frames in which the Security Enabled field is set to one. For frames in which the Security Enabled field is set to zero, the procedure in 9.2.4 is used instead.

The input to this procedure is the frame to be unsecured. The outputs from this procedure are the Status of the procedure, the unsecured frame (including all IEs), *SecurityLevel*, *KeyIdentifierMode*, *KeySource*, *KeyIndex* and *IeStatusList*. The status for an IE in the *IeStatusList* is set to PASSED if the IE conforms to the security policy for that IE and is set to FAILED otherwise.

All outputs of this procedure are assumed to be invalid unless and until explicitly set in this procedure.

This procedure involves the following steps:

- a) **Legacy security.** If the Frame Version field of the frame to be unsecured is set to zero, the procedure shall return with a Status of UNSUPPORTED\_LEGACY.
- b) **Check for *macSecurityEnabled*.** If *macSecurityEnabled* is set to FALSE, the procedure shall return with a Status of UNSUPPORTED\_SECURITY.
- c) **Parse Auxiliary Security Header field.** the procedure shall set SecurityLevel and KeyIdentifierMode to the Security Level field and Key Identifier Mode field, respectively, of the frame to be unsecured. If required by the KeyIdentifierMode, the KeySource and KeyIndex shall be set to the Key Source field and Key Index field, respectively, of the Key Identifier field of the frame to be unsecured. If the resulting SecurityLevel is zero, the procedure shall return with a Status of UNSUPPORTED\_SECURITY.
- d) **Obtain source address.** DevicePanId shall be set to the Source PAN ID field, if it is present. Otherwise, DevicePanId shall be set to the Destination PAN ID field, if present. If neither PAN ID field is present, then DevicePanId shall be set to *macPanId*. The DeviceAddressingMode shall be set based on the Source Addressing Mode field using the mapping in Table 9-2. The DeviceAddress shall be set to the Source Address field, if present.

**Table 9-2—Mapping of Source Addressing Mode field to DeviceAddressingMode**

| Source Addressing Mode field | DeviceAddressingMode |
|------------------------------|----------------------|
| 0x00                         | NONE                 |
| 0x02                         | SHORT                |
| 0x03                         | EXTENDED             |

- e) **Obtain KeyDescriptor.** The procedure shall obtain the KeyDescriptor using the KeyDescriptor lookup procedure as described in 9.2.2 with using the KeyIdMode, KeyIndex, KeySource, DeviceAddressingMode, DevicePanId, and DeviceAddress. If KeyDescriptor lookup procedure fails, the procedure shall return with a Status of UNAVAILABLE\_KEY.
- f) **Obtain DeviceDescriptor.** The procedure shall obtain the DeviceDescriptor using the DeviceDescriptor lookup procedure described in 9.2.5 using the DeviceAddressingMode, DevicePanId and DeviceAddress. If that procedure fails, then the procedure shall return with a Status of UNAVAILABLE\_DEVICE.
- g) **Obtain frame counter.** If TSCH mode is being used, then this step is skipped.
  - 1) If *secFrameCounterPerKey* of the KeyDescriptor is FALSE, the FrameCounterCheck value shall be set to be the *secDeviceMinFrameCounter* element of the DeviceDescriptor.
  - 2) If *secFrameCounterPerKey* of the KeyDescriptor is TRUE and there is a *secKeyDeviceFrameCounter* in the *secKeyDeviceFrameCounterList* in which *secDeviceExtAddress* matches *secExtAddress* of the DeviceDescriptor, then the procedure shall set the FrameCounterCheck value to the *secDeviceFrameCounter* of that *secKeyDeviceFrameCounter*.
  - 3) Otherwise, the procedure shall return with a Status of UNAVAILABLE\_DEVICE.
- h) **Check frame counter.** If TSCH mode is being used, then this step is skipped. If the Frame Counter field of the frame to be unsecured has the value 0xffffffff or the Frame Counter field of the frame to be unsecured is less than the FrameCounterCheck value, the procedure shall return with a Status of COUNTER\_ERROR.

- i) **Unsecure frame.** For frames specified in Table 9-1, the Private Payload field and Open Payload field shall be set as indicated in the table. Otherwise, the Private Payload field shall be set to the MAC payload field and the Open Payload field shall be empty. The procedure shall then use the Private Payload field, the Open Payload field, *secExtAddress* of the DeviceDescriptor, the Frame Counter field of the frame to be unsecured (if TSCH is not being used), the ASN (if TSCH is being used), SecurityLevel, and *secKey* of the KeyDescriptor to produce the unsecured frame, according to the CCM\* inverse transformation process described in the security operations, as described in 9.3.5. If the CCM\* inverse transformation process fails, the procedure shall return with a Status of SECURITY\_ERROR.
- j) **Store frame counter.** If not using TSCH mode and *secFrameCounterPerKey* of the KeyDescriptor is FALSE, then *secDeviceMinFrameCounter* of the *secDeviceDescriptor* shall be set to the value of the Frame Counter plus one. If not using TSCH mode and *secFrameCounterPerKey* of the KeyDescriptor is TRUE, then *secDeviceFrameCounter* of the *secKeyDeviceFrameCounter* corresponding to *secExtAddress* shall be set to the value Frame Counter plus one.
- k) **Obtain SecurityLevelDescriptor.** The procedure shall obtain the SecurityLevelDescriptor by passing the Frame Type field and, if the frame is a MAC command, the Command ID field, of the frame to be unsecured to the SecurityLevelDescriptor lookup procedure described in 9.2.6. If that procedure fails, the procedure shall return with a Status of UNAVAILABLE\_SECURITY\_LEVEL.
- l) **Check IE security.** If the IE present field of the frame to be unsecured is set to one, the procedure shall determine whether the IEs in the frame to be unsecured conforms to the security level policy by passing the IEs from the frame, DeviceDescriptor, SecurityLevelDescriptor and the SecurityLevel to the incoming IE security level checking procedure, as described in 9.2.7.
- m) **Check IE Key Usage Policy.** If the IE present field of the frame to be unsecured is set to one, the procedure shall determine whether the frame to be unsecured conforms to the key usage policy by passing the IeStatusList, KeyDescriptor, the IEs from the frame, the Frame Type field, and, if the frame is a MAC command, the Command ID field, to the incoming IE key usage policy checking procedure, as described in 9.2.8.
- n) **Check security level.** The procedure shall determine whether the frame to be unsecured conforms to the security level policy by passing the SecurityLevelDescriptor and the SecurityLevel to the incoming security level checking procedure, as described in 9.2.9. If that procedure returns with a Status of FAILED, the procedure shall return with the IEs from the unsecured frame, the IeStatusList and a Status of IMPROPER\_SECURITY\_LEVEL.
- o) **Check key usage policy.** The procedure shall determine whether the frame to be unsecured conforms to the key usage policy by passing the KeyDescriptor, the Frame Type field, and, if the frame is a MAC command, the Command ID field, to the incoming key usage policy checking procedure, as described in 9.2.10. If that procedure fails, the procedure shall return with the IEs from the unsecured frame, the IeStatusList and a Status of IMPROPER\_KEY\_TYPE.
- p) **Return unsecured frame.** The procedure shall return with the unsecured frame, SecurityLevel, KeyIdentifierMode, KeySource, KeyIndex, IeStatusList, and a Status of SUCCESS.

#### 9.2.4 Incoming frame security procedure, Security Enabled field is set to zero

This procedure shall only used for incoming frames in which the Security Enabled field set to zero. For frames in which the Security Level field is set to one, the procedure in 9.2.3 is used instead.

The input to this procedure is the frame to be validated. The outputs from this procedure are the Status of the procedure, the validated frame (including all IEs) and IeStatusList. The status for an IE in the IeStatusList is set to PASSED if the IE conforms to the security policy for that IE and is set to FAILED otherwise.

All outputs of this procedure are assumed to be invalid unless and until explicitly set in this procedure.



This procedure involves the following steps:

- a) **Check for *macSecurityEnabled*.** If *macSecurityEnabled* is set to FALSE, the procedure shall set the validated frame to be the frame to be validated and return with a Status of SUCCESS.
- b) **Obtain source address.** DevicePanId shall be set to the Source PAN ID field, if it is present. Otherwise, DevicePanId shall be set to the Destination PAN ID field, if present. If neither PAN ID field is present, then DevicePanId shall be set to *macPanId*. DeviceAddressingMode shall be set to the Source Addressing Mode field. The DeviceAddress shall be set to the Source Address, if present.
- c) **Obtain DeviceDescriptor.** The procedure shall obtain the DeviceDescriptor using the DeviceDescriptor lookup procedure described in 9.2.5 using DeviceAddressingMode, DevicePanId and DeviceAddress with the SecurityLevel set to zero. If that procedure fails, then the procedure shall return with a Status of UNAVAILABLE\_DEVICE.
- d) **Obtain SecurityLevelDescriptor.** The procedure shall obtain the SecurityLevelDescriptor by passing the Frame Type field and, if the frame is a MAC command, the Command ID field, of the frame to be validated to the SecurityLevelDescriptor lookup procedure described in 9.2.6. If that procedure fails, the procedure shall return with a Status of UNAVAILABLE\_SECURITY\_LEVEL.
- e) **Check IE security.** If the IE present field of the frame to be validated is set to one, the procedure shall determine whether the IEs in the frame to be validated conforms to the security level policy by passing the DeviceDescriptor, the SecurityLevelDescriptor, the SecurityLevel set to zero and the IEs from the frame to the incoming IE security level checking procedure, as described in 9.2.7. That procedure will return the IeStatusList.
- f) **Check security level.** The procedure shall determine whether the frame to be validated conforms to the security level policy by passing the SecurityLevelDescriptor and the SecurityLevel set to zero to the incoming security level checking procedure, as described in 9.2.9. If incoming security level checking procedure returns with a Status of FAILED, the procedure shall return with a Status of IMPROPER\_SECURITY\_LEVEL. If the incoming security level checking procedure returned with a Status of CONDITIONALLY\_PASSED and the *secExempt* element of the DeviceDescriptor is set to FALSE, the procedure shall return with a status of IMPROPER\_SECURITY\_LEVEL.
- g) **Return frame.** The procedure shall set the validated frame to be the frame to be validated and return with the frame to be validated, IeStatusList, and a status of SUCCESS.

### 9.2.5 DeviceDescriptor lookup procedure

The inputs to this procedure are DeviceAddressingMode, the DevicePanId, and the DeviceAddress. The output from this procedure is a Status of either PASSED or FAILED, and, if PASSED, a DeviceDescriptor.

This procedure involves the following steps:

- a) If the DeviceAddressingMode is set to NONE, then the DevicePanId shall be set to *macPanId*.
- b) If the DeviceAddressingMode is set to NONE, then:
  - 1) If the *macCoordShortAddress* attribute is set to 0xffff, then the DeviceAddress shall be set to the *macCoordExtendedAddress*.
  - 2) If the *macCoordShortAddress* attribute is set to a value of 0x0000–0xffffd, then the DeviceAddress shall be set to the *macCoordShortAddress*.
  - 3) If the *macCoordShortAddress* attribute is set to 0xffff, the procedure shall return with a Status set to FAILED.
- c) For each DeviceDescriptor, if DevicePanId matches the *secPanId* and DeviceAddress matches *secShortAddress*, if the DeviceAddressingMode is set to SHORT, or the *secExtAddress*, if the DeviceAddressingMode is set to EXTENDED, then the procedure shall return with the corresponding DeviceDescriptor and Status set to PASSED.



- d) The procedure shall return with a Status set to FAILED.

### 9.2.6 SecurityLevelDescriptor lookup procedure

The inputs to this procedure are the Frame Type field and, if the frame is a MAC command, the Command ID field. The output from this procedure are a Status of either PASSED or FAILED, and, if PASSED, a SecurityLevelDescriptor.

This procedure involves the following steps:

- a) For each SecurityLevelDescriptor in the *secSecurityLevelTable* attribute:
  - 1) If the frame type indicates that the frame is not a MAC command and the Frame Type field is equal to the *secFrameType* element of the *secSecurityLevelDescriptor*, the procedure shall return with the *secSecurityLevelDescriptor* and Status set to PASSED.
  - 2) If the Frame Type field indicates that the frame is a MAC command and the Frame Type is equal to the *secFrameType* element of the *secSecurityLevelDescriptor* and the Command ID field is equal to the *secCommandId* element of the *secSecurityLevelDescriptor*, the procedure shall return with the *secSecurityLevelDescriptor* and Status set to PASSED.
- b) The procedure shall return with Status set to FAILED.

### 9.2.7 Incoming IE security level checking procedure

The inputs to this procedure are DeviceDescriptor, SecurityLevelDescriptor, SecurityLevel, and the IEs in the frame. The output from this procedure is an *IeStatusList*, in which each element in the list is set to either PASSED or FAILED for each IE.

A match is found if the *secIeType* of the *secIeSecurityLevelDescriptor* matches the type of the IE and the *secIeId* of the *secIeSecurityLevelDescriptor* entry matches the ID of the IE.

This procedure involves the following steps:

- a) If *secIeSecurityLevelDescriptorList* of the SecurityLevelDescriptor is empty, then set the *IeStatus* in the *IeStatusList* to PASSED for each IE in the frame, and return *IeStatusList*.
- b) Set the *IeStatus* in the *IeStatusList* to FAILED for each IE in the frame.
- c) For each IE in the frame and for each *secIeSecurityLevelDescriptor* in *secIeSecurityLevelDescriptorList*, if the IE matches the *secIeSecurityLevelDescriptor* entry, then
  - 1) If *secIeAllowedSecurityLevels* of the *secIeSecurityLevelDescriptor* is empty, then the procedure shall compare the SecurityLevel (as SEC1) with the *secIeSecurityMinimum* of the *secIeSecurityLevelDescriptor* (as SEC2) according to the algorithm described in 9.4.1.1. If this comparison evaluates to TRUE, the procedure shall set the *IeStatus* in the *IeStatusList* for this IE to PASSED.
  - 2) If *secIeAllowedSecurityLevels* of the *secIeSecurityLevelDescriptor* is not empty, the procedure shall check whether the SecurityLevel is equal to any of the elements of the *secIeAllowedSecurityLevels* of the *secIeSecurityLevelDescriptor*. If this check is successful, the procedure shall set the *IeStatus* in the *IeStatusList* for this IE to PASSED.
  - 3) If the SecurityLevel is equal to 0x00 and *seIecDeviceOverrideSecurityMinimum* of the *secIeSecurityLevelDescriptor* is set to TRUE, and the *secExempt* of the DeviceDescriptor is set to TRUE, the procedure shall set the *IeStatus* in the *IeStatusList* for this IE to PASSED.
- d) Return *IeStatusList*.

### 9.2.8 Incoming IE key usage policy checking procedure

The inputs to this procedure are KeyDescriptor, IeStatusList, Frame Type field, the IEs in the frame, and, if the frame is a MAC command, the Command ID field. The output from this procedure is an IeStatusList, in which each element in the list is set to either PASSED or FAILED for each IE.

A match is found for an IE if the *secIeType* of the *secKeyIeUsageDescriptor* matches the type of the IE and the *secIeId* of the *secKeyIeUsageDescriptor* entry matches the ID of the IE.

This procedure involves the following steps:

- a) Find the *secKeyUsageDescriptor* entry for which the Frame Type field matches *secKeyUsageFrameType* and, if the frame is a MAC command, the Command ID field matches *secKeyUsageCommandId*. If a matching *secKeyUsageDescriptor* entry was not found, or if the *secKeyIeUsageList* is empty in the *secKeyUsageDescriptor* that was found, then return IeStatusList.
- b) For each IE in the frame, if the IE does not match any of the *secKeyIeUsageDescriptor* entries, then set the IeStatus in the IeStatusList for this IE to FAILED.
- c) Return IeStatusList.

### 9.2.9 Incoming security level checking procedure

The inputs to this procedure are SecurityLevelDescriptor and SecurityLevel. The output from this procedure is Status set to one of PASSED, FAILED, or CONDITIONALLY\_PASSED.

The incoming security level checking procedure involves the following steps:

- a) If *secAllowedSecurityLevels* in SecurityLevelDescriptor is empty, then the procedure shall compare the SecurityLevel (as SEC1) with the *secSecurityMinimum* element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 9.4.1.1. If this comparison evaluates to TRUE, the procedure shall return with Status set to PASSED.
- b) If *secAllowedSecurityLevels* in SecurityLevelDescriptor is not empty, the procedure shall check whether the SecurityLevel is equal to any of the elements of the *secAllowedSecurityLevels* of the SecurityLevelDescriptor. If this check is successful, the procedure shall return with Status set to PASSED.
- c) If SecurityLevel is equal to 0x00 and the *secDeviceOverrideSecurityMinimum* element of the SecurityLevelDescriptor is set to TRUE, the procedure shall return with Status set to CONDITIONALLY\_PASSED.
- d) The procedure shall return with Status set to FAILED.

### 9.2.10 Incoming key usage policy checking procedure

The inputs to this procedure are the KeyDescriptor, the Frame Type field, and the Command ID field. The output from this procedure is Status set to PASSED or FAILED.

The incoming key usage policy checking procedure involves the following steps:

- a) For each *secKeyUsageDescriptor* in the *secKeyUsageList* of the KeyDescriptor:
  - 1) If the Frame Type field indicates that the frame is not a MAC command and the Frame Type field is equal to the *secKeyUsageFrameType* element of the *secKeyUsageDescriptor*, the procedure shall return with Status set to PASSED.
  - 2) If the Frame Type field indicates that the frame is a MAC command, the Frame Type is equal to the *secFrameType* element of the *secKeyUsageDescriptor*, and the Command ID field is equal

to the *secKeyUsageCommandId* element of the *secKeyUsageDescriptor*, the procedure shall return with Status set to PASSED.

- b) The procedure shall return with Status set to FAILED.

### 9.3 Security operations

This subclause describes the parameters for the CCM\* security operations, as specified in B.3.2.

#### 9.3.1 Integer and octet representation

The integer and octet representation conventions specified in B.2 are used throughout this subclause.

#### 9.3.2 CCM\* nonce

##### 9.3.2.1 CCM\* nonce for non-TSCH mode

The CCM\* nonce for non-TSCH mode shall be formatted as shown in Figure 9-1, with the leftmost field in the figure defining the first octets and the rightmost field defining the last octet of the nonce.

|                |               |                      |
|----------------|---------------|----------------------|
| Octets: 8      | 4             | 1                    |
| Source Address | Frame Counter | Nonce Security Level |

**Figure 9-1—CCM\* nonce for non-TSCH mode**

The Source Address field shall be set to the extended address of the device originating the frame.

The Frame Counter field shall be set to the value of the respective field in the Auxiliary Security Header field, as defined in 9.4.

The Nonce Security Level field is an unsigned integer that shall be set to the value of the Security Level field of the Security Control field, as defined in 9.4.1.

The Source Address field, Frame Counter field, and Security Level field shall be represented as specified in 9.3.1.

##### 9.3.2.2 CCM\* nonce for TSCH mode

When TSCH mode is enabled, the nonce shall be formatted as shown in Figure 9-2.

|                |     |
|----------------|-----|
| Octets: 8      | 5   |
| Source Address | ASN |

**Figure 9-2—CCM\* nonce in TSCH mode**

The Source Address shall either be set to the extended address of the device originating the frame or shall be formatted as illustrated in Figure 9-3.

| Octets: 3       | 1    | 2      | 2             |
|-----------------|------|--------|---------------|
| IEEE 802.15 CID | 0x00 | PAN ID | Short address |

**Figure 9-3—Source Address field for TSCH mode with short addressing**

The IEEE 802.15 CID field contains the CID for IEEE 802.15.

The PAN ID field contains the PAN ID.

The Short Address field contains the short address of the device originating the frame.

NOTE—When using short addresses in the nonce, it is important that the coordinator assign unique short addresses.

The ASN shall be set to the ASN of the timeslot during which the frame is sent.

### 9.3.2.3 CCM\* nonce for Fragment frames

The CCM\* nonce for Fragment frames shall be formatted as shown in Figure 9-4, with the leftmost field in the figure defining the first octets and the rightmost field defining the last octet of the nonce.

| Octets: 8      | Bits: 0–25   | 26–31           | 32–35    | 36                 | 37–39          |
|----------------|--------------|-----------------|----------|--------------------|----------------|
| Source Address | PSDU Counter | Fragment Number | Reserved | Fragment Indicator | Security level |

**Figure 9-4—CCM\* nonce for Fragment frames**

The Source Address field is defined in 9.3.2.1.

The Security Level field is defined in 9.3.2.1.

The PSDU Counter field shall be set to the value of the PSDU Counter field of the FSCD IE that was used to set up this fragment transaction.

The Fragment Number field shall be set to the value of the Fragment Number field of the Fragment packet, as defined in 23.3.3.

The Fragment Indicator field shall be set to one.

### 9.3.3 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 transformation, as described in B.4.2.

The length  $M$  of the Authentication field for the CCM\* forward transformation and the CCM\* inverse transformation is determined from Table 9-6, using the Security Level field of the Security Control field of the auxiliary security header of the frame.

**9.3.4 CCM\* transformation data representation**

This subclause describes how the inputs and outputs of the CCM\* forward transformation, as described in B.4.1, are formed.

The inputs are as follows:

- Key
- Nonce
- *a* data
- *m* data

The output is *c* data.

**9.3.4.1 Key and nonce data inputs**

The Key data for the CCM\* forward transformation is passed by the outgoing frame security procedure described in 9.2.1. The Nonce data for the CCM\* transformation is constructed as described in 9.3.2.

**9.3.4.2 *a* data and *m* data**

In the CCM\* transformation process, the data fields shall be applied as in Table 9-3.

**Table 9-3—*a* data and *m* data for all security levels**

| Security level | <i>a</i> data                                                | <i>m</i> data                   |
|----------------|--------------------------------------------------------------|---------------------------------|
| 0              | None                                                         | None                            |
| 1              | MHR    Open Payload field    Unsecured Private Payload field | None                            |
| 2              | MHR    Open Payload field    Unsecured Private Payload field | None                            |
| 3              | MHR    Open Payload field    Unsecured Private Payload field | None                            |
| 4              | —                                                            | —                               |
| 5              | MHR    Open Payload field                                    | Unsecured Private Payload field |
| 6              | MHR    Open Payload field                                    | Unsecured Private Payload field |
| 7              | MHR    Open Payload field                                    | Unsecured Private Payload field |

NOTE—The MHR contains the Auxiliary Security Header field, as defined in 7.2.

**9.3.4.3 *c* data output**

In the CCM\* transformation process, the data fields that are applied, or right-concatenated and applied, represent octet strings.

The Private Payload field of the original unsecured frame shall be replaced by the right-concatenation of that field and the *c* field if data confidentiality is not provided and shall be replaced by the *c* field otherwise. The contents of the *c* data for each of the security levels is defined in Table 9-4.

**Table 9-4—*c* data for all security levels**

| Security level | <i>c</i> data                              |
|----------------|--------------------------------------------|
| 0              | None                                       |
| 1              | MIC-32                                     |
| 2              | MIC-64                                     |
| 3              | MIC-128                                    |
| 4              | —                                          |
| 5              | Encrypted Private Payload field    MIC-32  |
| 6              | Encrypted Private Payload field    MIC-64  |
| 7              | Encrypted Private Payload field    MIC-128 |

**9.3.5 CCM\* inverse transformation data representation**

This subclause describes how the inputs and outputs of the CCM\* inverse transformation, as described in B.4.2, are formed.

The inputs are as follows:

- Key
- Nonce
- *c* data
- *a* data

The output is *m* data.

**9.3.5.1 Key and nonce data inputs**

The Key data for the CCM\* inverse transformation is passed by the incoming frame security procedure described in 9.2.3. The Nonce data for the CCM\* transformation is constructed as described in 9.3.2.

**9.3.5.2 *c* data and *a* data**

In the CCM\* inverse transformation process, the data fields shall be applied as in Table 9-5.

**Table 9-5—*c* data and *a* data for all security levels**

| Security level | <i>c</i> data | <i>a</i> data                                      |
|----------------|---------------|----------------------------------------------------|
| 0              | None          | None                                               |
| 1              | MIC-32        | MHR    Open Payload field    Private Payload field |
| 2              | MIC-64        | MHR    Open Payload field    Private Payload field |
| 3              | MIC-128       | MHR    Open Payload field    Private Payload field |
| 4              | —             | —                                                  |

**Table 9-5—*c* data and *a* data for all security levels (continued)**

| Security level | <i>c</i> data                              | <i>a</i> data             |
|----------------|--------------------------------------------|---------------------------|
| 5              | Encrypted Private Payload field    MIC-32  | MHR    Open Payload field |
| 6              | Encrypted Private Payload field    MIC-64  | MHR    Open Payload field |
| 7              | Encrypted Private Payload field    MIC-128 | MHR    Open Payload field |

NOTE—The MHR contains the Auxiliary Security Header field, as defined in 7.2.

### 9.3.5.3 *m* data output

The Private Payload field of the MAC Payload shall be set to the *m* data if frame security includes providing confidentiality and shall be set to the Private Payload field of the MAC Payload, with the rightmost substring, *c*, deleted, otherwise.

## 9.4 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 field is set to one. The Auxiliary Security Header field shall be formatted as illustrated in Figure 9-5.

| Octets: 1        | 0/4           | 0/1/5/9        |
|------------------|---------------|----------------|
| Security Control | Frame Counter | Key Identifier |

**Figure 9-5—Format of the auxiliary security header**

The auxiliary security header uses the representation conventions specified in 7.1.

### 9.4.1 Security Control field

The Security Control field is used to provide information about what protection is applied to the frame. The Security Control field shall be formatted as shown in Figure 9-6.

| Bit: 0–2       | 3–4                 | 5                         | 6            | 6–7      |
|----------------|---------------------|---------------------------|--------------|----------|
| Security Level | Key Identifier Mode | Frame Counter Suppression | ASN in Nonce | Reserved |

**Figure 9-6—Security Control field format**

#### 9.4.1.1 Security Level field

The Security Level field 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 9-6.

**Table 9-6—Security levels available to the MAC sublayer**

| Security level | Security level field<br>b2 b1 b0 | Security attributes | Data confidentiality | Data authenticity | MIC length<br>(octets) |
|----------------|----------------------------------|---------------------|----------------------|-------------------|------------------------|
| 0              | 000                              | None                | OFF                  | NO                | 0                      |
| 1              | 001                              | MIC-32              | OFF                  | YES               | 4                      |
| 2              | 010                              | MIC-64              | OFF                  | YES               | 8                      |
| 3              | 011                              | MIC-128             | OFF                  | YES               | 16                     |
| 4              | 100                              | Reserved            |                      |                   |                        |
| 5              | 101                              | ENC-MIC-32          | ON                   | YES               | 4                      |
| 6              | 110                              | ENC-MIC-64          | ON                   | YES               | 8                      |
| 7              | 111                              | ENC-MIC-128         | ON                   | YES               | 16                     |

In previous version of the standard, security level 4 was a level which provided only data confidentiality but without data authenticity. This security level is deprecated and shall not be used in implementation compliant with this standard. Devices that receive frames with security level 4 shall discard them, as described in 9.2.3. The CCM used allows trivial changes to the underlying encrypted data unless data authenticity is provided, thus using data confidentiality only is not useful. In the case of TSCH mode, security level 4 allows higher security level frames to be downgraded to security level 4 frames.

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:

- Bit position b2 in SEC1 is greater than or equal to bit position b2 in SEC2 (where Encryption OFF < Encryption ON).
- The integer value of bit positions b1 b0 in SEC1 is greater than or equal to the integer value of bit positions b1 b0 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  $\geq$  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  $\geq$  ENC-MIC-64 is FALSE because even though MIC-128 offers stronger data authenticity than ENC-MIC-64, it offers no confidentiality.

#### 9.4.1.2 Key Identifier Mode field

The Key Identifier Mode field is an unsigned integer that 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, as defined in 9.4.3, if derived explicitly. The Key Identifier Mode field shall be set to one of the values listed in Table 9-7. The Key Identifier field of the auxiliary security header, as defined in 9.4.3, shall be present only if this field has a value that is not equal to 0x00.



**Table 9-7—Values of the Key Identifier Mode field**

| Key identifier mode | Key Identifier Mode field<br>b1 b0 | Description                                                                                                       | Key Identifier field length<br>(octets) |
|---------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| 0x00                | 00                                 | Key is determined implicitly from the originator and recipient(s) of the frame, as indicated in the frame header. | 0                                       |
| 0x01                | 01                                 | Key is determined from the Key Index field.                                                                       | 1                                       |
| 0x02                | 10                                 | Key is determined explicitly from the 4-octet Key Source field and the Key Index field.                           | 5                                       |
| 0x03                | 11                                 | Key is determined explicitly from the 8-octet Key Source field and the Key Index field.                           | 9                                       |

**9.4.1.3 Frame Counter Suppression field**

The Frame Counter Suppression field is set to zero when the frame counter is carried in the frame. When set to one, the frame counter is not carried in the frame, and the frame counter used to construct the nonce defined in 9.3.2 is an incrementing shared global frame counter such as ASN.

**9.4.1.4 ASN in Nonce**

The ASN in Nonce field is set to zero when the frame counter is used to generate the nonce, as described in 9.3.2.1. When set to one, the ASN is used to generate the nonce, as described in 9.3.2.2.

**9.4.2 Frame Counter field**

The Frame Counter field is an unsigned integer value used in CCM\* nonce generation as defined in 9.3.2.1 and to provide replay protection.

The Frame Counter field may be included in each secured frame and is one of the elements required for the unsecuring operation at the recipient(s). The value of the Frame Counter field is determined by the outgoing frame security procedure, 9.2.1.

Previous versions of this standard used a single frame counter for outgoing frames for each device. In the current revision of the standard allows multiple outgoing frame counters, each of which is associated with a key. If the next higher layer has been informed that the target device supports per key frame counters, then *secFrameCounterPerKey* of the *secKeyDescriptor* may be set to TRUE. If the next higher layer of the device does not know that the target device supports per key frame counters, then *secFrameCounterPerKey* of the *secKeyDescriptor* shall be set to FALSE.

When a frame counter, either per device or per key, reaches its maximum value, the associated keying material shall no longer be used, thus requiring this key to be updated by changing to use a new key. This provides a mechanism for ensuring that the keying material for every frame is unique and, thereby, provides for sequential freshness.

### 9.4.3 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 field, as defined in 9.4.1.2, is set to a value different from 0x00. The Key Identifier field shall be formatted as illustrated in Figure 9-7.

|               |           |
|---------------|-----------|
| Octets: 0/4/8 | 1         |
| Key Source    | Key Index |

Figure 9-7—Format for the Key Identifier field, if present

#### 9.4.3.1 Key Source field

The KeySource field, when present, indicates the originator of a group key. If the Key Identifier Mode field indicates a 4 octet Key Source field, then the Key Source field shall be the *macPanId* of the originator of the group key right concatenated with the *macShortAddress* of the originator of the group key. If the Key Identifier Mode field indicates an 8 octet Key Source field, then the Key Source field shall be set to the *macExtendedAddress* of the originator of the group key.

#### 9.4.3.2 Key Index field

The Key Index field allows unique identification of different keys with the same originator.

It is the responsibility of each key originator to make sure that the actively used keys that it issues have distinct key indices and that the key indices are all different from 0x00.

## 9.5 Security-related MAC PIB attributes

The PIB security-related attributes are defined in Table 9-8. The model for updates to the PIB described in this standard allow the MAC to update the PIB values asynchronously to write and read requests from the next higher layer. Thus, to maintain security, implementations should use a method to update the security-related PIB attributes in a way to ensure synchronization between the next higher layer and the MAC.

Table 9-8—Security-related MAC PIB attributes

| Attribute                   | Type                                                                  | Range | Description                                                                                     | Default |
|-----------------------------|-----------------------------------------------------------------------|-------|-------------------------------------------------------------------------------------------------|---------|
| <i>secKeyIdLookupList</i>   | List of <i>secKeyIdLookupDescriptors</i> , as defined in Table 9-9    | —     | A list of <i>secKeyIdLookupDescriptors</i> containing keys and security policy information.     | (empty) |
| <i>secDeviceList</i>        | List of <i>secDeviceDescriptors</i> , as defined in Table 9-14        | —     | List of device information for each remote device with which this device securely communicates. | (empty) |
| <i>secSecurityLevelList</i> | List of <i>secSecurityLevelDescriptors</i> , as defined in Table 9-15 | —     | Provides information about the security level required for each MAC frame type and subtype.     | (empty) |

**Table 9-8—Security-related MAC PIB attributes (continued)**

| Attribute                          | Type                                                         | Range                    | Description                                                                                                                                                            | Default    |
|------------------------------------|--------------------------------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <i>secFrameCounter</i>             | Integer                                                      | 0x00000000–0xffffffff    | The outgoing frame counter for this device to be used for keys which do not have <i>secFrameCounterPerKey</i> set to TRUE.                                             | 0x00000000 |
| <i>secAutoRequestSecurityLevel</i> | Integer                                                      | As defined in Table 9-6  | The security level used for automatic data requests.                                                                                                                   | 0x06       |
| <i>secAutoRequestKeyIdMode</i>     | Integer                                                      | As defined in Table 9-10 | The key identifier mode used for automatic data requests. This attribute is invalid if the <i>secAutoRequestSecurityLevel</i> attribute is set to 0x00.                | 0x00       |
| <i>secAutoRequestKeySource</i>     | As specified by the <i>secAutoRequestKeyIdMode</i> parameter | —                        | The originator of the key used for automatic data requests. This attribute is invalid if the <i>secAutoRequestKeyIdMode</i> element is invalid or set to 0x00 or 0x01. | —          |
| <i>secAutoRequestKeyIdIndex</i>    | Integer                                                      | 0x01–0xff                | The index of the key used for automatic data requests. This attribute is invalid if the <i>secAutoRequestKeyIdMode</i> attribute is invalid or set to 0x00.            | —          |

Table 9-9 defines the elements of a *secKeyIdLookupDescriptor*, which is used to find the *secKeyDescriptor* based on key identifiers.

**Table 9-9—Elements of *secKeyIdLookupDescriptor***

| Name                        | Type          | Range                   | Description                                                                                     |
|-----------------------------|---------------|-------------------------|-------------------------------------------------------------------------------------------------|
| <i>secKeyIdMode</i>         | Integer       | As defined in Table 9-7 | The mode used to for this descriptor.                                                           |
| <i>secKeySource</i>         | 4 or 8 octets | As defined in 9.4.3.1   | Information to identify the key. Present only if <i>secKeyIdMode</i> is equal to 0x02 or 0x03   |
| <i>secKeyIdIndex</i>        | Integer       | As defined in 9.4.3.1   | Information used to identify the key. Present only if <i>secKeyIdMode</i> is not equal to 0x00. |
| <i>secKeyDeviceAddrMode</i> | Enumeration   | NONE, SHORT, EXTENDED   | The addressing mode for this descriptor. Present only if <i>secKeyIdMode</i> is equal to 0x00.  |
| <i>secKeyDevicePanId</i>    | Integer       | 0x0000–0xffff           | The PAN ID for this descriptor. Present only if <i>secKeyIdMode</i> is equal to 0x00.           |

**Table 9-9—Elements of *secKeyIdLookupDescriptor* (continued)**

| Name                       | Type                                                  | Range                                           | Description                                                                                         |
|----------------------------|-------------------------------------------------------|-------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <i>secKeyDeviceAddress</i> | —                                                     | As specified by the <i>secKeyDeviceAddrMode</i> | The address for this descriptor. Present only if <i>secKeyIdMode</i> is equal to 0x00.              |
| <i>secKeyDescriptor</i>    | An <i>secKeyDescriptor</i> , as defined in Table 9-10 | As defined in Table 9-10                        | An <i>secKeyDescriptor</i> associated with the parameters in this <i>secKeyIdLookupDescriptor</i> . |

Table 9-10 defines the elements of a *secKeyDescriptor*. The *secKeyDescriptor* contains one entry for each key in use by the device. More than one *secKeyIdLookupDescriptor* can point to a single *secKeyDescriptor*.

**Table 9-10—Elements of *secKeyDescriptor***

| Name                                | Type                                                                      | Range                 | Description                                                                                                                                                                                                                  |
|-------------------------------------|---------------------------------------------------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>secKeyUsageList</i>              | List of <i>secKeyUsageDescriptor</i> entries, as defined in Table 9-12    | —                     | A list of <i>secKeyUsageDescriptor</i> entries indicating the frame types with which this key may be used.                                                                                                                   |
| <i>secKey</i>                       | 16 octets                                                                 | —                     | The value of the key.                                                                                                                                                                                                        |
| <i>secKeyFrameCounter</i>           | Integer                                                                   | 0x00000000–0xffffffff | The outgoing frame counter for this key. This not used if <i>secFrameCounterPerKey</i> is FALSE.                                                                                                                             |
| <i>secFrameCounterPerKey</i>        | Boolean                                                                   | TRUE, FALSE           | If this value is TRUE, this key will use per key frame counters stored in <i>secKeyDescriptor</i> and <i>secKeyDeviceFrameCounter</i> entries. If this value is FALSE, then per device frame counters are used for this key. |
| <i>secKeyDeviceFrameCounterList</i> | List of <i>secKeyDeviceFrameCounter</i> entries, as defined in Table 9-11 | —                     | A list of <i>secKeyDeviceFrameCounter</i> entries containing the per key frame counters used for this <i>secKeyDescriptor</i> .                                                                                              |

Table 9-11 defines the elements of a *secKeyDeviceFrameCounter*, which contains the incoming frame counter for a device specified by the associated extended address.

**Table 9-11—Elements of *secKeyDeviceFrameCounter***

| Name                         | Type         | Range                    | Description                                                                              |
|------------------------------|--------------|--------------------------|------------------------------------------------------------------------------------------|
| <i>secDeviceExtAddress</i>   | IEEE address | An extended IEEE address | The extended address of the device associated with the <i>secKeyDeviceFrameCounter</i> . |
| <i>secDeviceFrameCounter</i> | Integer      | 0x00000000–0xffffffff    | The incoming frame counter for the device indicated by <i>secDeviceExtAddress</i> .      |

Table 9-12, defines the elements of a *secKeyUsageDescriptor*. Each entry of a *secKeyUsageDescriptor* corresponds to a key and is used to define the frame types, MAC commands and IEs that are allowed to be used with that key.

**Table 9-12—Elements of *secKeyUsageDescriptor***

| Name                               | Type                                                                      | Range                    | Description                                                                                                                               |
|------------------------------------|---------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <i>secKeyUsageFrameType</i>        | Integer                                                                   | As defined in 7.2.1.1    | As defined in 7.2.1.1.                                                                                                                    |
| <i>secKeyUsageCommandId</i>        | Integer                                                                   | As defined in Table 7-49 | As defined in Table 7-49.                                                                                                                 |
| <i>secKeyIeUsageDescriptorList</i> | List of <i>secKeyIeUsageDescriptor</i> entries, as defined in Table 9-13. | —                        | List of <i>secKeyIeUsageDescriptor</i> entries. If this is empty, then there are no restrictions on the IE usage associated with the key. |

Table 9-13 defines the elements of a *secKeyIeUsageDescriptor*. The *secKeyIeUsageDescriptor* is a list of IEs that are allowed to be used with a key.

**Table 9-13—Elements of *secKeyIeUsageDescriptor***

| Name                | Type        | Range                                                          | Description                                                                                    |
|---------------------|-------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <i>secKeyIeType</i> | Enumeration | HEADER, PAYLOAD, NESTED_SHORT, NESTED_LONG                     | The type of IE, header, payload, nested long format or nested short format, as defined in 7.4. |
| <i>secKeyIeId</i>   | Integer     | As defined in Table 7-7, Table 7-15, Table 7-16, or Table 7-17 | The IE ID for the IE type indicated by <i>secKeyIeType</i> .                                   |

Table 9-14 defines the elements of a *secDeviceDescriptor*. Each entry in *secDeviceDescriptor* contains information about a device in the PAN with which this device has communicated.

**Table 9-14—Elements of *secDeviceDescriptor***

| Name                            | Type                 | Range                           | Description                                                                                                                                                                                        |
|---------------------------------|----------------------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>secPanId</i>                 | Device PAN ID        | 0x0000–0xffff                   | The PAN ID of the device in this DeviceDescriptor.                                                                                                                                                 |
| <i>secShortAddress</i>          | Device short address | 0x0000–0xffff                   | The 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. |
| <i>secExtAddress</i>            | IEEE address         | Any valid extended IEEE address | The extended IEEE address of the device.                                                                                                                                                           |
| <i>secExempt</i>                | Boolean              | TRUE, FALSE                     | Indication of whether the device may override the minimum security level settings defined in Table 9-15.                                                                                           |
| <i>secDeviceMinFrameCounter</i> | Integer              | 0x00000000–0xffffffff           | The smallest frame counter allowed to be sent by the other device for this key. If incoming frame counter is smaller than this then the frame is replay.                                           |

The security policies for frames and IEs are contained in the *secSecurityLevelDescriptor*. The elements of *secSecurityLevelDescriptor* are defined in Table 9-15.

**Table 9-15—Elements of *secSecurityLevelDescriptor***

| Name                                    | Type    | Range                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------|---------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>secFrameType</i>                     | Integer | As defined in 7.2.1.1    | As defined in 7.2.1.1.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>secCommandId</i>                     | Integer | As defined in Table 7-49 | As defined in Table 7-49.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>secSecurityMinimum</i>               | Integer | As defined in Table 9-6  | The minimal required/expected security level, as defined in Table 9-6, for incoming MAC frames with the indicated frame type and, if present, Command ID. This is only used if <i>secAllowedSecurityLevels</i> is empty.                                                                                                                                                                                                                                    |
| <i>secDeviceOverrideSecurityMinimum</i> | Boolean | TRUE, FALSE              | Indication of whether originating devices for which the <i>secExempt</i> is set may override the security level indicated by the <i>AllowedSecurityLevels</i> or <i>SecurityMinimum</i> . 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 <i>SecurityMinimum</i> element. |

**Table 9-15—Elements of *secSecurityLevelDescriptor* (continued)**

| Name                                    | Type                                                                           | Range | Description                                                                                                                                                                                                                   |
|-----------------------------------------|--------------------------------------------------------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>secAllowedSecurityLevels</i>         | Set of integers                                                                | —     | A set of allowed security levels, as defined in Table 9-6, for incoming MAC frames with the indicated frame type, and, if present, Command ID field. If the set is empty, then the SecurityMinimum parameter applies instead. |
| <i>secIeSecurityLevelDescriptorList</i> | List of <i>secIeSecurityLevelDescriptor</i> entries, as defined in Table 9-16. | —     | The allowed security levels for IEs. If this is empty, then there are no restrictions on security levels for IEs.                                                                                                             |

The *secIeSecurityLevelDescriptor* contains the security policies for IEs. The elements of the *secIeSecurityLevelDescriptor* are defined in Table 9-16.

**Table 9-16—Elements of *secIeSecurityLevelDescriptor***

| Name                                      | Type                      | Range                                                          | Description                                                                                    |
|-------------------------------------------|---------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <i>secIeType</i>                          | Enumeration               | HEADER, PAYLOAD, NESTED_SHORT, NESTED_LONG                     | The type of IE, header, payload, nested long format or nested short format, as defined in 7.4. |
| <i>secIeId</i>                            | Integer                   | As defined in Table 7-7, Table 7-15, Table 7-16, or Table 7-17 | The IE ID for the IE type indicated by <i>secIeType</i> .                                      |
| <i>secIeSecurityMinimum</i>               | As defined in Table 9-15. | As defined in Table 9-15                                       | As defined in Table 9-15.                                                                      |
| <i>secIeDeviceOverrideSecurityMinimum</i> | As defined in Table 9-15. | As defined in Table 9-15                                       | As defined in Table 9-15.                                                                      |
| <i>secIeAllowedSecurityLevels</i>         | As defined in Table 9-15. | As defined in Table 9-15                                       | As defined in Table 9-15.                                                                      |

## 10. General PHY requirements

### 10.1 General requirements and definitions

Unless otherwise specified, all PHYs use a 2-octet FCS.

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
- Precision ranging for ultra-wide band (UWB) PHYs

The PHYs defined in this standard are as follows:

- **O-QPSK PHY:** direct sequence spread spectrum (DSSS) PHY employing offset quadrature phase-shift keying (O-QPSK) modulation, operating in the 780 MHz band, 868 MHz band, 915 MHz band, 2380 MHz band, and 2450 MHz band, as defined in Clause 12.
- **BPSK PHY:** DSSS PHY employing binary phase-shift keying (BPSK) modulation, operating in the 868 MHz band, and 915 MHz band, as defined in Clause 13.
- **ASK PHY:** parallel sequence spread spectrum (PSSS) PHY employing amplitude shift keying (ASK) and BPSK modulation, operating in the 868 MHz band and 915 MHz band, as defined in Clause 14.
- **CSS PHY:** chirp spread spectrum (CSS) employing differential quadrature phase-shift keying (DQPSK) modulation, operating in the 2450 MHz band, as defined in Clause 15.
- **HRP UWB PHY:** combined burst position modulation (BPM) and BPSK modulation, operating in the sub-gigahertz and 3–10 GHz bands, as defined in Clause 16.
- **MPSK PHY:** M-ary phase-shift keying (MPSK) modulation, operating in the 780 MHz band, as defined in Annex E.
- **GFSK PHY:** Gaussian frequency-shift keying (GFSK), operating in the 920 MHz band, as defined in Clause 17.
- **MSK PHY:** minimum shift keying (MSK) PHY as defined in Clause 18.
- **LRP UWB PHY:** low rate pulse UWB PHY as defined in Clause 19.
- **SUN FSK PHY:** SUN FSK PHY operating in multiple over-the-air data rates in support of SUN applications, as defined in Clause 20.
- **SUN OFDM PHY:** SUN OFDM PHY operating in multiple over-the-air data rates in support of SUN applications, as defined in Clause 21.
- **SUN O-QPSK PHY:** SUN O-QPSK PHY operating in multiple over-the-air data rates in support of SUN applications, as defined in Clause 22.
- **LECIM DSSS PHY:** DSSS PHY operating with characteristics that enable support of LECIM applications, as defined in Clause 23.
- **LECIM FSK PHY:** FSK PHY operating with characteristics that enable support of LECIM applications, as defined in Clause 24.



- **TVWS-FSK PHY:** FSK PHY operating in multiple over-the-air data rates in support of various applications in TVWS, as defined in Clause 25.
- **TVWS-OFDM PHY:** OFDM PHY operating in multiple over-the-air data rates in support of various applications in TVWS, as defined in Clause 26.
- **TVWS-NB-OFDM PHY:** narrow band OFDM (NB-OFDM) PHY operating in multiple over-the-air data rates in support of various applications in TVWS, as defined in Clause 27.
- **RCC LMR PHY:** land mobile radio (LMR) for use in rail communications and control (RCC) applications using one of Gaussian minimum shift keying (GMSK), 4FSK, QPSK,  $\pi/4$  differential quadrature phase-shift keying (DQPSK), or DSSS employing DPSK, as defined in Clause 28.
- **RCC DSSS BPSK PHY:** a DSSS BPSK PHY for use in RCC applications, as defined in Clause 29.

### 10.1.1 Operating frequency range

A compliant device shall operate in one or several frequency bands summarized in this subclause.

For devices other than LECIM, TVWS, and RCC PHYs, the frequency bands are listed in Table 10-1.

**Table 10-1—Frequency band designations**

| Band designation      | Frequency band (MHz)   |
|-----------------------|------------------------|
| 169 MHz               | 169.400–169.475        |
| 433 MHz               | 433.05–434.79          |
| 450 MHz               | 450–470                |
| 470 MHz               | 470–510                |
| 780 MHz               | 779–787                |
| 863 MHz               | 863–870                |
| 868 MHz               | 868–868.6              |
| 896 MHz               | 896–901                |
| 901 MHz               | 901–902                |
| 915 MHz               | 902–928                |
| 917 MHz               | 917–923.5              |
| 920 MHz               | 920–928                |
| 928 MHz               | 928–960 <sup>a</sup>   |
| 1427 MHz              | 1427–1518 <sup>a</sup> |
| 2380 MHz              | 2360–2400              |
| 2450 MHz              | 2400–2483.5            |
| HRP UWB sub-gigahertz | 250–750                |
| HRP UWB low band      | 3244–4742              |
| HRP UWB high band     | 5944–10 234            |
| LRP UWB               | 6289.6–9185.6          |

<sup>a</sup>Noncontiguous.

The frequency bands for devices supporting the LECIM DSSS PHY are shown in Table 10-2.

**Table 10-2—LECIM DSSS PHY operating frequency ranges**

| Band designation | Bandwidth (kHz) | Modulation | Chip rate (kchips/s) |
|------------------|-----------------|------------|----------------------|
| 470 MHz          | 100             | BPSK       | 100                  |
|                  |                 | O-QPSK     | 200                  |
| 780 MHz          | 1000            | BPSK       | 1000                 |
|                  |                 | O-QPSK     | 2000                 |
| 863 MHz          | 100             | BPSK       | 100                  |
|                  |                 | O-QPSK     | 200                  |
| 915 MHz          | 200             | BPSK       | 200                  |
|                  |                 | O-QPSK     | 400                  |
|                  | 400             | BPSK       | 400                  |
|                  |                 | O-QPSK     | 800                  |
|                  | 600             | BPSK       | 600                  |
|                  |                 | O-QPSK     | 1200                 |
|                  | 800             | BPSK       | 800                  |
|                  |                 | O-QPSK     | 1600                 |
|                  | 1000            | BPSK       | 1000                 |
|                  |                 | O-QPSK     | 2000                 |
| 922 MHz          | 200             | BPSK       | 200                  |
|                  |                 | O-QPSK     | 400                  |
|                  | 400             | BPSK       | 400                  |
|                  |                 | O-QPSK     | 800                  |
|                  | 600             | BPSK       | 600                  |
|                  |                 | O-QPSK     | 1200                 |
|                  | 800             | BPSK       | 800                  |
|                  |                 | O-QPSK     | 1600                 |
|                  | 1000            | BPSK       | 1000                 |
|                  |                 | O-QPSK     | 2000                 |

**Table 10-2—LECIM DSSS PHY operating frequency ranges (continued)**

| Band designation | Bandwidth (kHz) | Modulation | Chip rate (kchips/s) |
|------------------|-----------------|------------|----------------------|
| 917 MHz          | 200             | BPSK       | 200                  |
|                  |                 | O-QPSK     | 400                  |
|                  | 400             | BPSK       | 400                  |
|                  |                 | O-QPSK     | 800                  |
|                  | 600             | BPSK       | 600                  |
|                  |                 | O-QPSK     | 1200                 |
|                  | 800             | BPSK       | 800                  |
|                  |                 | O-QPSK     | 1600                 |
| 920 MHz          | 200             | BPSK       | 200                  |
|                  |                 | O-QPSK     | 400                  |
|                  | 400             | BPSK       | 400                  |
|                  |                 | O-QPSK     | 800                  |
|                  | 600             | BPSK       | 600                  |
|                  |                 | O-QPSK     | 1200                 |
|                  | 800             | BPSK       | 800                  |
|                  |                 | O-QPSK     | 1600                 |
| 921 MHz          | 200             | BPSK       | 200                  |
|                  |                 | O-QPSK     | 400                  |
|                  | 400             | BPSK       | 400                  |
|                  |                 | O-QPSK     | 800                  |
|                  | 600             | BPSK       | 600                  |
|                  |                 | O-QPSK     | 1200                 |
|                  | 800             | BPSK       | 800                  |
|                  |                 | O-QPSK     | 1600                 |
| 2450 MHz         | 1000            | BPSK       | 1000                 |
|                  |                 | O-QPSK     | 2000                 |

The frequency bands for devices supporting the LECIM FSK PHY are show in Table 10-3.

**Table 10-3—LECIM FSK PHY operating frequency ranges**

| Band designation | Bandwidth (kHz) | Modulation    | Bit rate (kb/s) |
|------------------|-----------------|---------------|-----------------|
| 169 MHz          | 25              | FSK/<br>P-FSK | 25              |
|                  | 12.5            |               | 12.5            |
| 433 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 470 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 780 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 863 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 915 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 922 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 917 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 920 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |
| 921 MHz          | 37.5            | FSK/<br>P-FSK | 37.5            |
|                  | 25              |               | 25              |
|                  | 12.5            |               | 12.5            |

The frequency bands for devices supporting the TVWS PHYs are shown in Table 7-39.

The frequency bands for devices supporting the RCC PHYs are shown in Table 10-4.

**Table 10-4—RCC PHY frequency bands and data rates**

| Band designation | Frequency range (MHz) | Modulation and bit rate                                                                                                                                |
|------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 161 MHz          | 160.170–161.580       | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FS 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s                                     |
| 216 MHz          | 216–217               |                                                                                                                                                        |
| 217 MHz          | 217–220               |                                                                                                                                                        |
| 220 MHz          | 220–222               |                                                                                                                                                        |
| 450 MHz          | 450–470               | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FSK: 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s                                   |
| 770 MHz          | 769–775               | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FSK: 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s                                   |
| 800 MHz          | 799–805               |                                                                                                                                                        |
| 806 MHz          | 806–821<br>851–866    |                                                                                                                                                        |
| 896 MHz          | 896–901<br>935–940    | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FSK: 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s                                   |
| 915 MHz          | 902–928               | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FSK: 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s<br>LMR DSSS DPSK<br>LMR DSSS BPSK |
| 928 MHz          | 928–960               | LMR GMSK: 9.6/19.2 kb/s<br>LMR 4-FSK: 9.6/19.2/38.4 kb/s<br>LMR QPSK: 16/32 kb/s<br>LMR $\pi/4$ DQPSK: 16/32/36 kb/s                                   |
| 2450 MHz         | 2400–2483.5           | LMR DSSS BPSK                                                                                                                                          |
| 4965 MHz         | 4940–4990             | LMR DSSS DPSK<br>LMR DSSS BPSK                                                                                                                         |
| 5800 MHz         | 5725–5850             | LMR DSSS DPSK<br>LMR DSSS BPSK                                                                                                                         |

Devices shall start in the PHY mode in which they are instructed to start. If the device is capable of operating in the 868 MHz or 915 MHz bands using one of the optional PHYs described in Clause 12 and Clause 14, it shall be able to switch dynamically between the optional PHY and the mandatory BPSK PHY in that band when instructed to do so.

Operation of the MPSK PHY in the 780 MHz band is described in Annex E. The NITS/CWPAN Part 15.4 specification [B11] also defines operation in the 314–316 MHz and 430–434 MHz bands, which is not described in this standard.

### 10.1.2 Channel assignments

Channel assignments are defined through a combination of channel numbers and channel pages.

The PHY PIB attributes are described in 11.3.

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.

For each PHY supported, a compliant device shall support all channels allowed by regulations for the region in which the device operates. An exception to this is the HRP UWB PHY where specific mandatory and optional behaviors are as defined in 16.4.1. An additional exception to this is the LRP UWB PHY, in which a transmitter device shall not be required to transmit on more than one channel.

#### 10.1.2.1 Channel numbering for 780 MHz band

This subclause does not apply to the SUN PHY, RCC PHY or LECIM PHY specifications. For channel page five, channels numbered zero to seven are available across the 780 MHz band. The center frequency of these channels is defined as follows:

$$F_c = 780 + 2k \text{ in megahertz, for } k = 0, \dots, 3$$

$$F_c = 780 + 2(k - 4) \text{ in megahertz, for } k = 4, \dots, 7$$

where  $k$  is the channel number.

#### 10.1.2.2 Channel numbering for 868 MHz, 915 MHz, and 2450 MHz bands

This subclause does not apply to the SUN PHY or LECIM PHY specifications. For explanations of channel numbering for the SUN PHYs and LECIM PHYs, see 10.1.2.8 and 10.1.2.10, respectively.

For channel page zero, 16 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:

$$F_c = 868.3 \text{ in megahertz, for } k = 0$$

$$F_c = 906 + 2(k - 1) \text{ in megahertz, for } k = 1, 2, \dots, 10$$

$$F_c = 2405 + 5(k - 11) \text{ in megahertz, for } k = 11, 12, \dots, 26$$

where  $k$  is the channel number.

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

$$F_c = 868.3 \text{ in megahertz, for } k = 0$$

$$F_c = 906 + 2(k - 1) \text{ in megahertz, for } k = 1, 2, \dots, 10$$

where  $k$  is the channel number.

**10.1.2.3 Channel numbering for CSS PHY**

The CSS PHY uses channel page three with the channel numbers defined in Table 10-5. Different subsets of these frequency channels are available in different regions of the world. In North America and Europe, three frequency channels can be selected so that the nonoverlapping frequency channels are used.

**Table 10-5—Center frequencies of CSS**

| Channel number | Frequency (MHz) |
|----------------|-----------------|
| 0              | 2412            |
| 1              | 2417            |
| 2              | 2422            |
| 3              | 2427            |
| 4              | 2432            |
| 5              | 2437            |
| 6              | 2442            |
| 7              | 2447            |
| 8              | 2452            |
| 9              | 2457            |
| 10             | 2462            |
| 11             | 2467            |
| 12             | 2472            |
| 13             | 2484            |

**10.1.2.4 Channel numbering for HRP UWB PHY**

The HRP UWB PHY uses channel page four with the channel numbers defined in Table 10-6. A compliant HRP UWB device shall be capable of transmitting in at least one of three specified bands, sub-gigahertz, low, or high. An HRP UWB device that implements the sub-gigahertz band shall implement channel 0. An HRP UWB device that implements the low band shall support channel 3. The remaining low-band channels are optional. An HRP UWB device that implements the high band shall support channel 9. The remaining high-band channels are optional.

**Table 10-6—HRP UWB PHY channel frequencies**

| Channel number | Center frequency (MHz) | HRP UWB band/mandatory |
|----------------|------------------------|------------------------|
| 0              | 499.2                  | Sub-gigahertz          |
| 1              | 3494.4                 | Low band               |
| 2              | 3993.6                 |                        |
| 3              | 4492.8                 |                        |
| 4              | 3993.6                 |                        |

**Table 10-6—HRP UWB PHY channel frequencies (continued)**

| Channel number | Center frequency (MHz) | HRP UWB band/mandatory |
|----------------|------------------------|------------------------|
| 5              | 6489.6                 | High band              |
| 6              | 6988.8                 |                        |
| 7              | 6489.6                 |                        |
| 8              | 7488.0                 |                        |
| 9              | 7987.2                 |                        |
| 10             | 8486.4                 |                        |
| 11             | 7987.2                 |                        |
| 12             | 8985.6                 |                        |
| 13             | 9484.8                 |                        |
| 14             | 9984.0                 |                        |
| 15             | 9484.8                 |                        |

**10.1.2.5 Channel numbering for MSK PHY 433 MHz band**

The MSK PHY 433 MHz band uses channel page 7 with the channel numbers defined in Table 10-7. A total of 15 frequency channels are available in the band from 433.05 MHz to 434.79 MHz. Different subsets of these frequency channels are available in different regions of the world. Compliant receivers shall implement all 15 channels in Table 10-7, defaulting to channel 7 unless modified by higher layers.

**Table 10-7—MSK PHY 433 MHz band channel frequencies**

| Channel number | Center frequency (MHz) |
|----------------|------------------------|
| 0              | 433.164                |
| 1              | 433.272                |
| 2              | 433.380                |
| 3              | 433.488                |
| 4              | 433.596                |
| 5              | 433.704                |
| 6              | 433.812                |
| 7              | 433.920                |
| 8              | 434.028                |
| 9              | 434.136                |
| 10             | 434.244                |
| 11             | 434.352                |



**Table 10-7—MSK PHY 433 MHz band channel frequencies (continued)**

| Channel number | Center frequency (MHz) |
|----------------|------------------------|
| 12             | 434.460                |
| 13             | 434.568                |
| 14             | 434.676                |

The multiple narrowband channels for the MSK PHY 433 MHz band are specified in order to improve coexistence with other potential 433 MHz services and to comply with applicable regulations. The selection of specific channels is out of the scope of this standard, being performed by higher layers. Channel selection methodologies might include the following:

- Selecting permanent channels based on an RF survey at the time of system installation
- Performing regular CCAs during operation to dynamically select optimal channels
- Monitoring other link quality metrics to select optimal channels

#### 10.1.2.6 Channel numbering for MSK PHY 2450 MHz band

The MSK PHY 2450 MHz band uses channel page 7 with the channel numbers defined in Table 10-8. A total of 42 frequency channels numbered 15 to 56 on channel page 7 are available in the band from 2400 MHz to 2483.5 MHz. Different subsets of these frequency channels are available in different regions of the world. Compliant receivers shall implement all channels in Table 10-8, defaulting to channel 47 unless modified by higher layers. The multiple narrow channels for the narrowband MSK 2450 MHz PHY are specified in order to improve coexistence with other 2450 MHz services. The selection of specific channels is out of the scope of this standard, being performed by higher layers.

**Table 10-8—MSK 2450 MHz mandatory PHY channel frequencies**

| Channel number | Center frequency (MHz) |
|----------------|------------------------|
| 15             | 2401.75                |
| 16             | 2403.75                |
| 17             | 2405.75                |
| 18             | 2407.75                |
| 19             | 2409.75                |
| 20             | 2411.75                |
| 21             | 2413.75                |
| 22             | 2415.75                |
| 23             | 2417.75                |
| 24             | 2419.75                |
| 25             | 2421.75                |
| 26             | 2422.5                 |

**Table 10-8—MSK 2450 MHz mandatory PHY channel frequencies** *(continued)*

| Channel number | Center frequency (MHz) |
|----------------|------------------------|
| 27             | 2423.25                |
| 28             | 2425.75                |
| 29             | 2427.75                |
| 30             | 2429.75                |
| 31             | 2431.75                |
| 32             | 2433.75                |
| 33             | 2435.75                |
| 34             | 2437.75                |
| 35             | 2439.75                |
| 36             | 2442                   |
| 37             | 2443.75                |
| 38             | 2445.75                |
| 39             | 2447.75                |
| 40             | 2449.75                |
| 41             | 2451.75                |
| 42             | 2453.75                |
| 43             | 2455.75                |
| 44             | 2457.75                |
| 45             | 2459.75                |
| 46             | 2462                   |
| 47             | 2463.75                |
| 48             | 2465.75                |
| 49             | 2467.75                |
| 50             | 2469.75                |
| 51             | 2471.75                |
| 52             | 2473.75                |
| 53             | 2475.75                |
| 54             | 2477.75                |
| 55             | 2479.75                |
| 56             | 2481.75                |

**10.1.2.7 Channel numbering for LRP UWB PHY**

The LRP UWB PHY uses channel page 8 with the channel numbers defined in Table 10-9. A total of three frequency channels, numbered 0 to 2, are available in the 6289.6 MHz to 9185.6 MHz frequency bands. Different subsets of these frequency channels are available in different regions of the world. In North America and Europe, a shared channel may be used.

**Table 10-9—LRP UWB PHY channel frequencies**

| Channel number | Center frequency (MHz) |
|----------------|------------------------|
| 0              | 6489.6                 |
| 1              | 6988.8                 |
| 2              | 7987.2                 |

**10.1.2.8 Channel numbering for SUN and TVWS PHYs**

The channel center frequency  $ChanCenterFreq$  for all SUN and TVWS PHYs, except the SUN O-QPSK PHY operating in the 868–870 MHz band, shall be derived as follows:

$$ChanCenterFreq = ChanCenterFreq_0 + NumChan \times ChanSpacing$$

where

- $ChanCenterFreq_0$  is the first channel center frequency
- $ChanSpacing$  is the separation between adjacent channels
- $NumChan$  is the channel number from 0 to  $TotalNumChan-1$
- $TotalNumChan$  is the total number of channels for the available frequency band

The parameters  $ChanSpacing$ ,  $TotalNumChan$ , and  $ChanCenterFreq_0$  for different frequency bands and modulation schemes are specified in Table 10-10.

**Table 10-10—Channel numbering for SUN PHYs**

| Frequency band (MHz) | Modulation                          | $ChanSpacing$ (MHz) | $TotalNumChan$ | $ChanCenterFreq_0$ (MHz) |
|----------------------|-------------------------------------|---------------------|----------------|--------------------------|
| 169.400–169.475      | SUN FSK operating mode #1 & #2 & #3 | 0.0125              | 6              | 169.40625                |
| 450–470              | SUN FSK operating mode #1 & #2      | 0.0125              | 1599           | 450.00625                |
| 470–510              | SUN FSK operating mode #1           | 0.2                 | 199            | 470.2                    |
|                      | SUN FSK operating mode #2 & #3      | 0.4                 | 99             | 470.4                    |
|                      | OFDM Option4                        | 0.2                 | 199            | 470.2                    |
|                      | O-QPSK                              | 0.4                 | 99             | 470.4                    |

**Table 10-10—Channel numbering for SUN PHYs (continued)**

| Frequency band (MHz) | Modulation                             | ChanSpacing (MHz)         | TotalNumChan | ChanCenterFreq <sub>0</sub> (MHz) |
|----------------------|----------------------------------------|---------------------------|--------------|-----------------------------------|
| 779–787              | SUN FSK operating mode #1              | 0.2                       | 39           | 779.2                             |
|                      | SUN FSK operating mode #2 & #3         | 0.4                       | 19           | 779.4                             |
|                      | OFDM Option4                           | 0.2                       | 39           | 779.2                             |
|                      | OFDM Option3                           | 0.4                       | 19           | 779.4                             |
|                      | OFDM Option2                           | 0.8                       | 9            | 779.8                             |
|                      | OFDM Option1                           | 1.2                       | 6            | 780.2                             |
|                      | O-QPSK                                 | 2                         | 4            | 780                               |
| 863–870              | SUN FSK operating mode #1              | 0.2                       | 34           | 863.125                           |
|                      | SUN FSK operating mode #2 & #3         | 0.4                       | 17           | 863.225                           |
|                      | OFDM Option4                           | 0.2                       | 34           | 863.125                           |
|                      | OFDM Option3                           | 0.4                       | 17           | 863.225                           |
|                      | OFDM Option2                           | 0.8                       | 8            | 863.425                           |
|                      | OFDM Option1                           | 1.2                       | 5            | 863.625                           |
| 868–870              | O-QPSK                                 | As defined in Table 10-11 |              |                                   |
| 896–901              | SUN FSK operating mode #1 <sup>a</sup> | 0.0125                    | 399          | 896.0125                          |
|                      | SUN FSK operating mode #2 <sup>b</sup> | 0.0125                    | 397          | 896.025                           |
|                      | SUN FSK operating mode #3 <sup>c</sup> | 0.0125                    | 393          | 896.05                            |
| 901–902              | SUN FSK operating mode #1 <sup>a</sup> | 0.0125                    | 79           | 901.0125                          |
|                      | SUN FSK operating mode #2 <sup>b</sup> | 0.0125                    | 77           | 901.025                           |
|                      | SUN FSK operating mode #3 <sup>c</sup> | 0.0125                    | 73           | 901.05                            |
| 902–928              | SUN FSK operating mode #1              | 0.2                       | 129          | 902.2                             |
|                      | SUN FSK operating mode #2 & #3         | 0.4                       | 64           | 902.4                             |
|                      | OFDM Option4                           | 0.2                       | 129          | 902.2                             |
|                      | OFDM Option3                           | 0.4                       | 64           | 902.4                             |
|                      | OFDM Option2                           | 0.8                       | 31           | 902.8                             |
|                      | OFDM Option1                           | 1.2                       | 20           | 903.2                             |
|                      | O-QPSK                                 | 2                         | 12           | 904                               |

**Table 10-10—Channel numbering for SUN PHYs (continued)**

| Frequency band (MHz) | Modulation                             | ChanSpacing (MHz) | TotalNumChan | ChanCenterFreq <sub>0</sub> (MHz) |
|----------------------|----------------------------------------|-------------------|--------------|-----------------------------------|
| 917–923.5            | OFDM Option4                           | 0.2               | 32           | 917.1                             |
|                      | OFDM Option3                           | 0.4               | 16           | 917.3                             |
|                      | OFDM Option2                           | 0.8               | 8            | 917.5                             |
|                      | OFDM Option1                           | 1.2               | 5            | 917.9                             |
|                      | SUN FSK operating mode #1              | 0.2               | 32           | 917.1                             |
|                      | SUN FSK operating mode #2 & #3         | 0.4               | 16           | 917.3                             |
|                      | O-QPSK                                 | 2                 | 3            | 918.1                             |
| 920–928              | SUN FSK operating mode #1              | 0.2               | 38           | 920.6                             |
|                      | SUN FSK operating mode #2              | 0.4               | 18           | 920.9                             |
|                      | SUN FSK operating mode #3 & #4         | 0.6               | 12           | 920.8                             |
|                      | OFDM Option4                           | 0.2               | 39           | 920.2                             |
|                      | OFDM Option3                           | 0.4               | 19           | 920.4                             |
|                      | OFDM Option2                           | 0.8               | 9            | 920.8                             |
|                      | OFDM Option1                           | 1.2               | 6            | 921.2                             |
|                      | O-QPSK                                 | 0.2               | 38           | 920.6                             |
| 928–960              | SUN FSK operating mode #1 <sup>a</sup> | 0.0125            | 2559         | 928.0125                          |
|                      | SUN FSK operating mode #2 <sup>b</sup> | 0.0125            | 2557         | 928.025                           |
|                      | SUN FSK operating mode #3 <sup>c</sup> | 0.0125            | 2553         | 928.05                            |
| 1427–1518            | SUN FSK operating mode #1 <sup>a</sup> | 0.0125            | 7279         | 1427.0125                         |
|                      | SUN FSK operating mode #2 <sup>b</sup> | 0.0125            | 7277         | 1427.025                          |
|                      | SUN FSK operating mode #3 <sup>c</sup> | 0.0125            | 7273         | 1427.05                           |
| 2400–2483.5          | SUN FSK operating mode #1              | 0.2               | 416          | 2400.2                            |
|                      | SUN FSK operating mode #2 & #3         | 0.4               | 207          | 2400.4                            |
|                      | OFDM Option4                           | 0.2               | 416          | 2400.2                            |
|                      | OFDM Option3                           | 0.4               | 207          | 2400.4                            |
|                      | OFDM Option2                           | 0.8               | 97           | 2400.8                            |
|                      | OFDM Option1                           | 1.2               | 64           | 2401.2                            |
|                      | O-QPSK                                 | 5                 | 16           | 2405                              |

<sup>a</sup>Two adjacent *ChanSpacing*(s) are aggregated to form an overlapping channel with bandwidth of 25 kHz.<sup>b</sup>Four adjacent *ChanSpacing*(s) are aggregated to form an overlapping channel with bandwidth of 50 kHz.<sup>c</sup>Eight adjacent *ChanSpacing*(s) are aggregated to form an overlapping channel with bandwidth of 100 kHz.

Three channels are available for the SUN O-QPSK PHY operating in the 868–870 MHz band. The channel center frequency for each of these channels is shown in Table 10-11.

**Table 10-11—Center frequencies for the SUN O-QPSK PHY operating in the 868–870 MHz band**

| NumChan | ChanCenterFreq (MHz) |
|---------|----------------------|
| 0       | 868.300              |
| 1       | 868.950              |
| 2       | 869.525              |

In the case of TVWS PHYs,  $ChanCenterFreq_0$  is derived as follows:

$$ChanCenterFreq_0 = macStartBandEdge + ChanSpacing$$

$TotalNumChan$  is derived as follows:

$$TotalNumChan = \text{floor}((macEndBandEdge - macStartBandEdge) / ChanSpacing - 1)$$

$macStartBandEdge$  and  $macEndBandEdge$  are set by the higher layers, appropriate for the band of operation at a particular time.

Channel page nine is used to specify the standard-defined PHY operating modes, and channel page ten is used to indicate a SUN FSK Generic PHY mode.

#### 10.1.2.9 Channel numbering for 2380 MHz band

For channel page eleven, 15 channels numbered zero to fourteen are available across the 2380 MHz band. The center frequencies of these channels are defined as follows:

$$F_c = 2363 + 5k \text{ in megahertz, for } k = 0, 1, \dots, 6$$

$$F_c = 2367 + 5(k - 7) \text{ in megahertz, for } k = 7, 8, \dots, 13$$

$$F_c = 2395 \text{ in megahertz, for } k = 14$$

where  $k$  is the channel number.

#### 10.1.2.10 Channel numbering for LECIM PHYs

A channel page value of 12 indicates a LECIM PHY.

For the 2.4 GHz band, a device shall support all the channels from the lowest to the highest channels (inclusive) indicated in the LECIM Capabilities IE, as described in 7.4.2.12.

**10.1.2.10.1 Channel numbering for LECIM DSSS PHY**

When *phyCurrentLECIMPHYType* is set to DSSS, the channel plan is described as follows. The channel center frequency, *ChanCenterFreq*, for all LECIM DSSS PHY frequency bands shall be derived as follows:

$$\text{ChanCenterFreq} = \text{FreqBandEdge} + \text{FreqOffset} + (\text{phyCurrentChannel} - 1) \times \text{ChanSpacing}$$

where

|                          |                                                                               |
|--------------------------|-------------------------------------------------------------------------------|
| <i>ChanCenterFreq</i>    | is the channel center frequency                                               |
| <i>FreqBandEdge</i>      | is the band edge for the frequency band in use ( <i>phyLecimCurrentBand</i> ) |
| <i>FreqOffset</i>        | is the frequency offset for each band                                         |
| <i>phyCurrentChannel</i> | is the designated channel identifier number from 1 to <i>N</i>                |
| <i>ChanSpacing</i>       | is the separation between adjacent channels ( <i>phyChannelSpacing</i> )      |

The parameters *FreqBandEdge*, *FreqOffset*, *ChanSpacing*, and the range of valid *phyCurrentChannel* channel numbers for each frequency band are listed in Table 10-12.

**Table 10-12—Frequency band, frequency band offset, and channel spacing for LECIM DSSS PHY**

| Band designation | <i>FreqBandEdge</i> (MHz) | <i>FreqOffset</i> (MHz) | <i>ChanSpacing</i> (MHz) | <i>phyCurrentChannel</i> range |
|------------------|---------------------------|-------------------------|--------------------------|--------------------------------|
| 433 MHz          | 433                       | 0.17                    | 0.1                      | 1–16                           |
|                  | 433                       | 0.22                    | 0.2                      | 1–8                            |
| 470 MHz          | 470                       | 0.2                     | 0.2                      | 1–199                          |
| 780 MHz          | 779                       | 0.2                     | 0.2                      | 1–39                           |
| 863 MHz          | 863                       | 0.075                   | 0.1                      | 1–69                           |
|                  | 863                       | 0.125                   | 0.2                      | 1–34                           |
| 915 MHz          | 902                       | 0.2                     | 0.2                      | 1–129                          |
| 917 MHz          | 917                       | 0.1                     | 0.2                      | 1–32                           |
| 920 MHz          | 920                       | 0.6                     | 0.2                      | 1–36                           |
| 921 MHz          | 921                       | 0.2                     | 0.2                      | 1–34                           |
| 922 MHz          | 915                       | 0.2                     | 0.2                      | 1–64                           |
| 2450 MHz         | 2400                      | 0.2                     | 0.2                      | 1–416                          |

**10.1.2.10.2 Channel numbering for LECIM FSK PHY**

When *phyCurrentLECIMPHYType* is set to FSK, the channel center frequency *ChanCenterFreq* for the LECIM FSK PHY shall be derived as follows:

$$\text{ChanCenterFreq} = \text{ChanSpacing} \times \text{phyCurrentChannel} + \text{ChanCenterFreq}_0$$

where

*ChanSpacing* is the separation between adjacent channels (*phyChannelSpacing*)  
*phyCurrentChannel* is the current channel number occurring in the range of 0 to *TotalNumChan*–1  
*TotalNumChan* is the total number of channels for the available frequency band  
*ChanCenterFreq<sub>0</sub>* is the first channel center frequency of the band in use (*phyLecimCurrentBand*)

Parameters *TotalNumChan* and *ChanCenterFreq<sub>0</sub>* are specified in Table 10-13 and Table 10-14 for different frequency bands and channel spacings.

For band designation 169 MHz, *ChanCenterFreq<sub>0</sub>* shall be 169.4375 MHz and *TotalNumChan* shall be one.

**Table 10-13—*TotalNumChan* and *ChanCenterFreq<sub>0</sub>* when *ChanSpacing* = 200 kHz**

| Band designation | <i>TotalNumChan</i> | <i>ChanCenterFreq<sub>0</sub></i><br>(MHz) |
|------------------|---------------------|--------------------------------------------|
| 433 MHz          | 8                   | 433.22                                     |
| 470 MHz          | 199                 | 470.2                                      |
| 780 MHz          | 39                  | 779.2                                      |
| 863 MHz          | 34                  | 863.125                                    |
| 915 MHz          | 129                 | 902.2                                      |
| 917 MHz          | 32                  | 917.1                                      |
| 920 MHz          | 36                  | 920.6                                      |
| 921 MHz          | 34                  | 921.2                                      |
| 922 MHz          | 64                  | 915.2                                      |

**Table 10-14—*TotalNumChan* and *ChanCenterFreq<sub>0</sub>* when *ChanSpacing* = 100 kHz**

| Band designation | <i>TotalNumChan</i> | <i>ChanCenterFreq<sub>0</sub></i><br>(MHz) |
|------------------|---------------------|--------------------------------------------|
| 433 MHz          | 16                  | 433.170                                    |
| 470 MHz          | 399                 | 470.1                                      |
| 780 MHz          | 79                  | 779.1                                      |
| 863 MHz          | 69                  | 863.075                                    |
| 915 MHz          | 259                 | 902.1                                      |
| 921 MHz          | 69                  | 921.1                                      |
| 922 MHz          | 129                 | 915.1                                      |

#### 10.1.2.11 Channel numbering for RCC PHYs

A channel page value of 13 indicates an RCC PHY.



Applicable regulations shall be used to define channel numbering where indicated in Table 10-15. For all other bands, the channel center frequency,  $ChanCenterFreq$ , for an RCC PHY shall be derived as follows:

$$ChanCenterFreq = ChanCenterFreq_0 + NumChan \times ChanSpacing$$

where

$ChanCenterFreq_0$  is the first channel center frequency  
 $ChanSpacing$  is the separation between adjacent channels  
 $NumChan$  is the channel number from 0 to  $TotalNumChan-1$   
 $TotalNumChan$  is the total number of channels for the available frequency band

The parameters  $ChanSpacing$ ,  $TotalNumChan$ , and  $ChanCenterFreq_0$  for each frequency band is specified in Table 10-15. The information in Table 10-15 the applies to all RCC modulation schemes.

**Table 10-15—RCC PHY channel numbering information**

| Band designation | $ChanSpacing$<br>(MHz)                                    | $TotalNumChan$ | $ChanCenterFreq_0$<br>(MHz) |
|------------------|-----------------------------------------------------------|----------------|-----------------------------|
| 161 MHz          | 0.0075                                                    | 187            | 160.1775                    |
| 216 MHz          | 0.00625                                                   | 159            | 216.00625                   |
| 217 MHz          | 0.00625                                                   | 479            | 217.00625                   |
| 220 MHz          | 0.005                                                     | 400            | 220.0025                    |
| 450 MHz          | 0.00625                                                   | 3199           | 450.00625                   |
| 770 MHz          | 0.00625                                                   | 960            | 769.003125                  |
| 800 MHz          | 0.00625                                                   | 960            | 799.003125                  |
| 806 MHz          | As defined in 47 CFR, Part 90, Subpart S, Section 90.613  |                |                             |
| 896 MHz          | As defined in 47 CFR, Part 90, Subpart S, Section 90.613  |                |                             |
| 915 MHz          | 0.500                                                     | 51             | 902.500                     |
| 928 MHz          | 0.00625                                                   | 5119           | 928.0125                    |
| 2450 MHz         | 0.2                                                       | 416            | 2400.2                      |
| 4965 MHz         | As defined in 47 CFR, Part 90, Subpart Y, Section 90.1213 |                |                             |
| 5800 MHz         | 0.5                                                       | 249            | 5725.5                      |

### 10.1.3 Minimum LIFS and SIFS periods

For all PHYs other than the HRP UWB PHY and RCC PHY, the minimum LIFS period and SIFS period are as follows:<sup>14</sup>

- $macLifsPeriod$  – 40 symbols
- $macSifsPeriod$  – 12 symbols

<sup>14</sup>For the SUN OFDM PHY, the MAC symbol period is defined in 6.1.

For the HRP UWB PHY, the minimum LIFS period and SIFS period are as follows:

- *macLifsPeriod* – 40 preamble symbols
- *macSifsPeriod* – 12 preamble symbols

For the HRP UWB PHY, the actual time for *macLifsPeriod* and *macSifsPeriod* depends on the PRF and channel in use, as described in Table 16-5.

For the RCC PHY, the minimum LIFS period and SIFS period are as follows:

- *macLifsPeriod* – 5 symbols
- *macSifsPeriod* – 5 symbols

#### 10.1.4 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) and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

#### 10.1.5 Transmit power

A compliant device shall have its nominal transmit power level indicated by its PHY parameter, *phyTxPower*, as defined in 11.3.

For HRP UWB PHYs, the parameter *phyTxPower* refers to the total power transmitted across the entire occupied bandwidth.

#### 10.1.6 Out-of-band spurious emission

The out-of-band spurious emissions shall conform with applicable regulations.

#### 10.1.7 Receiver sensitivity definitions

The general conditions for measuring receiver sensitivity are defined in Table 10-16.

**Table 10-16—Receiver sensitivity conditions**

| 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    | Lowest input power for which the PER conditions are met.                 | <p>PSDU length = 250 octets for SUN PHYs with data rates 50 kb/s and greater, 20 octets for all other PHYs.</p> <p>PER &lt; 10% for SUN PHYs. PER &lt; 1% for all other PHYs.</p> <p>Power measured at antenna terminals with interference not present.</p> |

**10.1.8 Common signaling mode (CSM) for SUN PHY**

The CSM is a common PHY mode specified to facilitate the multi-PHY management (MPM) scheme described in 6.14. A SUN device acting as a coordinator and with a duty cycle greater than 1% shall support CSM. The specification of CSM is given in Table 10-17. The modulation and channel specification of CSM are given in 20.3.

**Table 10-17—PHY specification of the CSM for MPM scheme**

| Band designation                                                           | Modulation | Modulation index | Channel spacing | Data rate |
|----------------------------------------------------------------------------|------------|------------------|-----------------|-----------|
| 470 MHz<br>780 MHz<br>863 MHz<br>915 MHz<br>917 MHz<br>920 MHz<br>2450 MHz | 2-FSK      | 1                | 200 kHz         | 50 kB/s   |

The value of the SFD field, as described in 20.2.1.2, for CSM shall be that associated with a value of zero for the PIB attribute *phySunFskSfd*, as defined in 11.3.

**10.2 General radio specifications**

For the SUN FSK PHY, SUN OFDM PHY and SUN O-QPSK PHY, the meaning of a symbol period for timing parameters is described in 6.1.

**10.2.1 TX-to-RX turnaround time**

The TX-to-RX turnaround time shall be less than or equal to *aTurnaroundTime*, as defined in Table 11-1.

The TX-to-RX turnaround time is defined as the time at the air interface from the trailing edge of the last part/chip (of the last symbol) of a transmitted PPDU to the time that the PHY is ready to receive the leading edge of the first part/chip (of the first symbol) of the next received PPDU.

**10.2.2 RX-to-TX turnaround time**

The RX-to-TX turnaround time shall be less than or equal to *aTurnaroundTime*, as defined in Table 11-1.

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

**10.2.3 Error-vector magnitude (EVM) definition**

The modulation accuracy of the transmitter is determined with an EVM measurement. In order to calculate the EVM, a time record of  $N$  received complex chip values  $(I_j, Q_j)$  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, as illustrated in Figure 10-1.

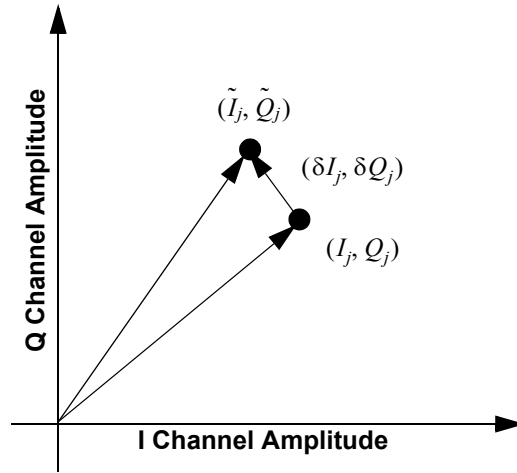


Figure 10-1—Error-vector calculation

Thus, the received vector is the sum of the ideal vector and the error vector as follows:

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

The EVM is defined as follows:

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

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

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.

#### 10.2.4 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 10.1.7 is met.

#### 10.2.5 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 next higher layer using MLME-SCAN.confirm. The minimum ED value (zero) shall indicate received power less than 10 dB above the lowest specified receiver sensitivity, in dBm, for the PHY. 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  $\pm 6$  dB.

#### 10.2.6 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. 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.

#### 10.2.7 Clear channel assessment (CCA)

With the exception of the HRP UWB PHY, a compliant PHY shall provide the capability to perform CCA according to at least one of the following 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.
- *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
  - Energy above the ED threshold, where the logical operator may be AND or OR
- *CCA Mode 4: ALOHA.* CCA shall always report an idle medium.

An HRP UWB PHY shall implement one CCA Mode 1 through CCA Mode 4 or one of the following methods:

- *CCA Mode 5: HRP UWB preamble sense based on the SHR of a frame.* CCA shall report a busy medium upon detection of a preamble symbol as specified in 16.2.5. An idle channel shall be reported if no preamble symbol is detected up to a period not shorter than the maximum packet duration plus the maximum period for acknowledgment.
- *CCA Mode 6: HRP UWB preamble sense based on the packet with the multiplexed preamble as specified in 16.6.* CCA shall report a busy medium upon detection of a preamble symbol as specified in 16.2.5.

CCA mode 4 would typically be used in low duty cycle applications.

For any of the CCA modes, if a request to perform CCA 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.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

NOTE—These modes are used to provide cooperative utilization of the medium in an IEEE 802.15.4 network. They are not designed to provide regulatory compliance, and in some cases only a subset of these modes may meet regulatory requirements.

As an example, EN 300 328 v 1.8.1 requires energy detect for a minimum of 20  $\mu$ s. In this case an implementer could choose to use CCA mode 2 within the CSMA-CA algorithm, followed by a 20  $\mu$ s ED in accordance to the requirements of that standard in order to achieve regulatory compliance. Implementing a design in this manner would provide an optimized network that would not be disadvantaged in a mixed protocol environment with networks other than IEEE 802.15.4.

The PHY PIB attribute *phyCCAMode*, as described in 11.3, shall indicate the appropriate operation mode. The CCA parameters are subject to the following criteria:

- a) Except for the SUN O-QPSK PHY, the ED threshold shall correspond to a received signal power of at most 10 dB greater than the specified receiver sensitivity for that PHY, or in accordance with local regulations. For the SUN O-QPSK PHY, the ED threshold shall comply with the specification in 22.5.13.
- b) Except for the 920 MHz band PHYs and the RCC PHYs, the CCA detection time shall be equal to *aCcaTime*, as defined in Table 11-1. For the 920 MHz band, and the RCC PHYs, *phyCCADuration* symbol periods shall be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018

## 11. PHY services

### 11.1 Overview

The PHY provides an interface between the MAC sublayer and the physical radio channel, via the RF firmware and the 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 11-1 depicts the components and interfaces of the PHY.

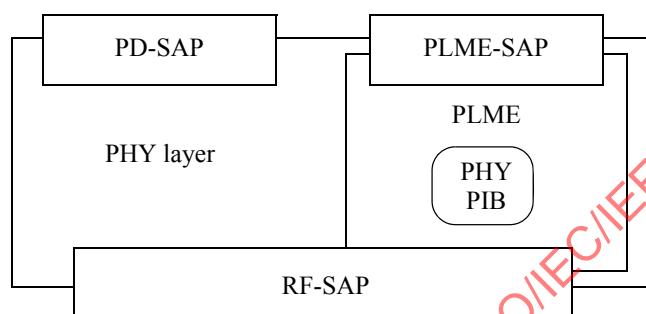


**Figure 11-1—PHY reference model**

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. The PD-SAP and PLME-SAP are not defined in this standard as they are not expected to be exposed in a typical implementation. The PHY PIB attributes are accessed through the MLME-SAP with the MLME-GET and MLME-SET primitives.

Constants and attributes that are specified and maintained by the PHY are written in *italics*. Constants have a general prefix of “a”, e.g., *aMaxPhyPacketSize*, and are listed in Table 11-1. Attributes have a general prefix of “phy”, e.g., *phyCurrentChannel*, and are listed in Table 11-2.

Attributes that have a prefix of “phyHrpUwb”, e.g., *phyHrpUwbDataRatesSupported*, apply only to the HRP UWB PHY and are not used for other PHYs.

### 11.2 PHY constants

The constants that define the characteristics of the PHY are presented in Table 11-1. These constants are hardware dependent and cannot be changed during operation.

### 11.3 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 11-2. 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 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.

**Table 11-1—PHY constants**

| Constant                 | Description                                                                               | Value                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>aMaxPhyPacketSize</i> | The maximum PSDU size (in octets) the PHY shall be able to receive.                       | 2047 for SUN, TVWS, RCC, and LECIM FSK PHYs. For LECIM DSSS PHY, this is not a constant; refer to <i>phyLecimDsssPsduSize</i> . 127 for all other PHYs                                                                                                                                                                                                                                               |
| <i>aTurnaroundTime</i>   | RX-to-TX or TX-to-RX turnaround time (in symbol periods), as defined in 10.2.1 and 10.2.2 | For the SUN, TVWS, and LECIM FSK PHYs, the value is 1 ms expressed in symbol periods, rounded up to the next integer number of symbol periods using the ceiling() function. <sup>a</sup> For the LECIM DSSS PHY, the value is 1 ms expressed in modulation symbol periods, rounded up to the next integer number of symbol periods using the ceiling() function. The value is 12 for all other PHYs. |
| <i>aLeipDelayTime</i>    | The delay between the start of the SFD and the LEIP, as described in 19.6.                | 0.815 ms                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>aCcaTime</i>          | The time required to perform CCA detection.                                               | For the SUN PHYs other than SUN O-QPSK, the duration of 8 symbol periods, as defined in 6.1. For the SUN O-QPSK PHY, this value is defined in Table 22-24. For all other PHYs, the duration of 8 symbol periods.                                                                                                                                                                                     |

<sup>a</sup>The function ceiling() returns the smallest integer value greater than or equal to its argument value.

**Table 11-2—PHY PIB attributes**

| Attribute                       | Type           | Range                  | Description                                                                                                                                                                                      |
|---------------------------------|----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phyCurrentChannel</i>        | Integer        | As defined in 10.1.2   | The RF channel to use for all following transmissions and receptions, 10.1.2.                                                                                                                    |
| <i>phyTxPower</i>               | Signed integer | —                      | The transmit power of the device in dBm.                                                                                                                                                         |
| <i>phyCurrentPage</i>           | Integer        | Any valid channel page | This is the current PHY channel page. This is used in conjunction with <i>phyCurrentChannel</i> to uniquely identify the channel currently being used.                                           |
| <i>phyFskFecEnabled</i>         | Boolean        | TRUE, FALSE            | A value of TRUE indicates that FEC is turned on. A value of FALSE indicates that FEC is turned off. This attribute is only valid for the SUN FSK and TVWS FSK PHY.                               |
| <i>phyFskFecInterleavingRsc</i> | Boolean        | TRUE, FALSE            | A value of TRUE indicates that interleaving is enabled for RSC. A value of FALSE indicates that interleaving is disabled for RSC. This attribute is only valid for the SUN FSK and TVWS FSK PHY. |



**Table 11-2—PHY PIB attributes (continued)**

| Attribute                            | Type             | Range                    | Description                                                                                                                                                                                                                                                            |
|--------------------------------------|------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phyFskFecScheme</i>               | Integer          | 0, 1                     | A value of zero indicates that a non-recursive and non-systematic code (NRNSC) is employed. A value of one indicates that a recursive and systematic code (RSC) is employed. See 20.3.4 for more information on FEC. This attribute is only valid for the SUN FSK PHY. |
| <i>phyModeSwitchParameterEntries</i> | Array            | As defined in Table 11-3 | An array of up to four rows, where each row consists of a set of ModeSwitchDescriptor entries. This attribute is only valid for the SUN FSK PHY.                                                                                                                       |
| <i>phyFskPreambleLength</i>          | Integer          | 4–64                     | The number of repetitions of the preamble pattern, as described in 20.2.1.1 and 25.1.1.1, in the preamble. This attribute is only valid for the SUN FSK and TVWS FSK PHY.                                                                                              |
| <i>phySunFskSfd</i>                  | Integer          | 0, 1                     | Determines which group of SFDs is used, as described in Table 20-2. This attribute is only valid for the SUN FSK PHY.                                                                                                                                                  |
| <i>phyFskScramblePsdu</i>            | Boolean          | TRUE, FALSE              | A value of FALSE indicates that data whitening of the PSDU is disabled. A value of TRUE indicates that data whitening of the PSDU is enabled. This attribute is only valid for the SUN FSK PHY.                                                                        |
| <i>phyOfdmInterleaving</i>           | Integer          | 0, 1                     | A value of zero indicates an interleaving depth of one symbol. A value of one indicates an interleaving depth of the number of symbols equal to the frequency domain spreading factor. This attribute is only valid for the SUN OFDM PHY.                              |
| <i>phyHrpUwbDataRatesSupported†</i>  | List of integers | —                        | A list of the data rates available in the operating channel as defined in Table 16-8.                                                                                                                                                                                  |
| <i>phyHrpUwbCurrentPulseShape</i>    | Enumeration      | MANDATORY, COU, CS, LCP  | Indicates the current pulse shape setting of the HRP UWB PHY. The mandatory pulse is described in 16.4.5. Optional pulse shapes include CoU, as defined in 16.5.1, CS, as defined in 16.5.2, and LCP, as defined in 16.5.3.                                            |
| <i>phyHrpUwbLcpWeight1</i>           | Signed integer   | 0x00–0xff                | The weights are represented in twos-complement form. A value of 0x80 represents –1 while a value of 0x7f represents 1.                                                                                                                                                 |
| <i>phyHrpUwbLcpWeight2</i>           | Signed integer   | 0x00–0xff                | The weights are represented in twos-complement form. A value of 0x80 represents –1 while a value of 0x7f represents 1.                                                                                                                                                 |
| <i>phyHrpUwbLcpWeight3</i>           | Signed integer   | 0x00–0xff                | The weights are represented in twos-complement form. A value of 0x80 represents –1 while a value of 0x7f represents 1.                                                                                                                                                 |
| <i>phyHrpUwbLcpWeight4</i>           | Signed integer   | 0x00–0xff                | The weights are represented in twos-complement form. A value of 0x80 represents –1 while a value of 0x7f represents 1.                                                                                                                                                 |

**Table 11-2—PHY PIB attributes (continued)**

| Attribute                                | Type        | Range                               | Description                                                                                                                                                                                                                                                                                                     |
|------------------------------------------|-------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phyHrpUwbLcpDelay2</i>                | Integer     | 0x00–0xff                           | The range is from 0 to 4 ns with a resolution is 15.625 ps. For example, a value of 0x00 represents 0 while 0x02 represents 31.25 ps, as defined in 16.5.3.                                                                                                                                                     |
| <i>phyHrpUwbLcpDelay3</i>                | Integer     | 0x00–0xff                           | The range is from 0 to 4 ns with a resolution is 15.625 ps. For example, a value of 0x00 represents 0 while 0x02 represents 31.25 ps, as defined in 16.5.3.                                                                                                                                                     |
| <i>phyHrpUwbLcpDelay4</i>                | Integer     | 0x00–0xff                           | The range is from 0 to 4 ns with a resolution is 15.625 ps. For example, a value of 0x00 represents 0 while 0x02 represents 31.25 ps, as defined in 16.5.3.                                                                                                                                                     |
| <i>phyRanging</i> <sup>†</sup>           | Boolean     | TRUE, FALSE                         | TRUE if ranging is supported, FALSE otherwise.                                                                                                                                                                                                                                                                  |
| <i>phyCurrentCode</i>                    | Integer     | 0–24                                | This value is zero for PHYs other than HRP UWB or CSS. For HRP UWB PHYs, this represents the current preamble code index in use by the transmitter, as defined in Table 16-6 and Table 16-7. For the CSS PHY, the value indicates the subchirp, as defined in 15.3.                                             |
| <i>phyHrpUwbScanBinsPerChannel</i>       | Integer     | 0–255                               | Number of frequency intervals used to scan each HRP UWB channel (scan resolution). Set to zero for non-HRP UWB PHYs.                                                                                                                                                                                            |
| <i>phyHrpUwbInsertedPreambleInterval</i> | Enumeration | 0, 4                                | The time interval between two neighboring inserted preamble symbols in the data portion, as defined in 16.6, for HRP UWB PHYs operating with CCA mode 6. The resolution is a data symbol period at a data rate of 850 kb/s for all channels. Set to four for HRP UWB PHY in CCA mode 6; otherwise, set to zero. |
| <i>phyCcaDuration</i>                    | Integer     | 0–1000                              | The duration for CCA, specified in symbols for PHYs operating in the 920 MHz band.                                                                                                                                                                                                                              |
| <i>phyLecimDsssPpduModulationRate</i>    | Enumeration | 100, 200, 400, 600, 800, 1000, 2000 | The modulation rate measured in modulation kilo symbols per second. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                                                        |
| <i>phyLecimDsssPpduTxAt</i>              | Integer     | 0–[2 <sup>32</sup> –1]              | The time, in modulation symbols, relative to the start of the beacon. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                                                      |
| <i>phyLecimDsssPsduSize</i>              | Enumeration | 16, 24, 32                          | The size, in octets, of the PSDU. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                                                                                          |
| <i>phyLecimDsssPreambleSize</i>          | Enumeration | 0, 16, 32                           | The length of the preamble, as illustrated in Table 23-1. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                                                                  |
| <i>phyLecimDsssSfdPresent</i>            | Boolean     | TRUE, FALSE                         | A value of TRUE indicates that the SFD is present. A value of FALSE indicates that the SFD is not present. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                 |
| <i>phyLecimDsssPsduSpreadingFactor</i>   | Integer     | 4–15                                | 2 <sup>x</sup> chips per symbol where x is in the range 4 to 15, inclusive. This attribute is only valid for the LECIM DSSS PHY.                                                                                                                                                                                |

Table 11-2—PHY PIB attributes (continued)

| Attribute                                  | Type        | Range                                                  | Description                                                                                                                                                                                                                         |
|--------------------------------------------|-------------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phyLecimFecTailBitingEnabled</i>        | Boolean     | TRUE, FALSE                                            | A value of TRUE indicates that tail biting is enabled. A value of FALSE indicates that it is disabled. This attribute is only valid for the LECIM DSSS PHY.                                                                         |
| <i>phyLecimDsssPsduOvsfSpreadingFactor</i> | Integer     | 1–256                                                  | The length of the generated code in power of 2. A value of one indicates that OVSF is not enabled. This attribute is only valid for the LECIM DSSS PHY.                                                                             |
| <i>phyLecimDsssPsduOvsfCodeIndex</i>       | Integer     | 0, 1, ..., $N-1$                                       | Specifies the desired code from the available set of codes. The value of $N$ is given by <i>phyLecimDsssPSDUOVSFSpreadingFactor</i> . This attribute is only valid for the LECIM DSSS PHY.                                          |
| <i>phyLecimFskPreambleLength</i>           | Integer     | 4–64                                                   | The number of times the preamble contains the pattern defined in 24.2.1.1. This attribute is only valid for the LECIM FSK PHY.                                                                                                      |
| <i>phyLecimFskPsduPositionMod</i>          | Boolean     | TRUE, FALSE                                            | Indicates whether position-based modulation is enabled. A value of TRUE indicates that position-based modulation is enabled. A value of FALSE indicates that it is not enabled. This attribute is only valid for the LECIM FSK PHY. |
| <i>phyLecimFskSpreading</i>                | Boolean     | TRUE, FALSE                                            | A value of TRUE indicates that spreading is enabled. A value of FALSE indicates that spreading is disabled. This attribute is only valid for the LECIM FSK PHY.                                                                     |
| <i>phyLecimFskSpreadingFactor</i>          | Enumeration | 1, 2, 4, 8, 16                                         | The spreading factor to be used when <i>phyLecimFskSpreading</i> or <i>phyTvwsFskSpreadingEnabled</i> is TRUE. This attribute is only valid for the LECIM FSK and TVWS FSK PHY.                                                     |
| <i>phyLecimFskSpreadingPattern</i>         | Enumeration | ALTERNATING_1/0, NON_ALTERNATING                       | Specifies the type of pattern used for spreading when spreading is enabled. This attribute is only valid for the LECIM FSK and TVWS FSK PHY.                                                                                        |
| <i>phyLecimFecEnabled</i>                  | Boolean     | TRUE, FALSE                                            | A value of TRUE indicates that FEC is turned on. A value of FALSE indicates that FEC is turned off. This attribute is only valid for the LECIM FSK PHY.                                                                             |
| <i>phyLecimFskInterleavingEnabled</i>      | Boolean     | TRUE, FALSE                                            | A value of TRUE indicates that interleaving is turned on. A value of FALSE indicates that interleaving is turned off. This attribute is only valid for the LECIM FSK PHY.                                                           |
| <i>phyLecimCurrentBand</i>                 | Enumeration | 169, 433, 470, 780, 863, 915, 917, 920, 921, 922, 2450 | The operating frequency band currently selected.                                                                                                                                                                                    |
| <i>phyLecimFskSymbolRate</i>               | Float       | As defined in Table 10-3                               | The currently selected symbol rate in $k$ -symbols per second. The valid symbol rates per band are given in Table 10-3.                                                                                                             |
| <i>phyCurrentLecimPhyType</i>              | Enumeration | DSSS, FSK                                              | Specifies the LECIM PHY type in use.                                                                                                                                                                                                |

**Table 11-2—PHY PIB attributes (continued)**

| Attribute                          | Type        | Range                 | Description                                                                                                                                                                                                                                                                                                        |
|------------------------------------|-------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>phyChannelSpacing</i>           | Enumeration | 100, 200              | The channel spacing, measured in kilohertz, that is used with <i>phyCurrentBand</i> and <i>phyCurrentChannel</i> to specify the frequency channel being used.                                                                                                                                                      |
| <i>phyTvwsFskSpreadingEnabled</i>  | Boolean     | TRUE, FALSE           | A value of TRUE indicates that spreading is turned on. A value of FALSE indicates that spreading is turned off. This attribute is only valid for the TVWS-FSK PHY.                                                                                                                                                 |
| <i>phyTvwsFskWhiteningEnabled</i>  | Boolean     | TRUE, FALSE           | A value of TRUE indicates that whitening is turned on. A value of FALSE indicates that whitening is turned off. This attribute is only valid for the TVWS-FSK PHY.                                                                                                                                                 |
| <i>phyTvwsSfdLength</i>            | Integer     | 16 or 24              | Length of the TVWS SFD field in bits. This attribute is only valid for the TVWS-FSK PHY.                                                                                                                                                                                                                           |
| <i>phyTvwsFskFecScheme</i>         | Integer     | 0–2                   | A value of zero indicates that the first FEC scheme as defined in 25.2.2 is employed. A value of one indicates that the second FEC scheme as defined in 25.2.2 is employed. A value of two indicates that the third FEC scheme as defined in 25.2.2 is employed. The attribute is only valid for the TVWS-FSK PHY. |
| <i>phyTvwsChannelAggregation</i>   | Boolean     | TRUE, FALSE           | A value of TRUE indicates that channel aggregation is enabled. A value of FALSE indicates that channel aggregation is disabled. This attribute is only valid for the TVWS-NB-OFDM PHY.                                                                                                                             |
| <i>phyLmrCodingRate</i>            | Float       | 1/2, 2/3, 3/4, 7/8, 1 | Controls which puncturing pattern is used for the PSDU, as described in 28.3. If the attribute value is one, then FEC shall not be applied.                                                                                                                                                                        |
| <i>phyFragmentSize</i>             | Integer     | PHY dependent         | The number of octets in each fragment.                                                                                                                                                                                                                                                                             |
| <i>phyPsduFragSecure</i>           | Boolean     | TRUE, FALSE           | When set to TRUE, a MIC shall be used as the FICS, as described in 7.4.2.9. When set to FALSE, the FICS shall be calculated as in 7.2.10.                                                                                                                                                                          |
| <i>phyFragmentFrameCounter</i>     | Integer     | 0x00 0000–0x3f ffff   | The outgoing PSDU counter to use when <i>phyPSDUFragSecure</i> is TRUE. The counter is not used when <i>phyPSDUFragSecure</i> is FALSE.                                                                                                                                                                            |
| <i>phyFrakProgressTimeout</i>      | Integer     | —                     | The duration, in modulated symbols, at which to generate a Frak frame when Frak policy 1 is in use.                                                                                                                                                                                                                |
| <i>phyPsduFragPadValue</i>         | Integer     | 0–255                 | The value used to pad out the last fragment when MPDU fragmentation is enabled.                                                                                                                                                                                                                                    |
| <i>phyPsduFragmentationEnabled</i> | Boolean     | TRUE, FALSE           | When TRUE, PSDU fragmentation is enabled. See 5.4. When FALSE, PSDU fragmentation is disabled.                                                                                                                                                                                                                     |

**Table 11-3—Elements of ModeSwitchDescriptor**

| Name                              | Type    | Valid range | Description                                                                                                                                                                                                                             |
|-----------------------------------|---------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>SettlingDelay</i>              | Integer | 0–510       | The settling delay, in $\mu\text{s}$ , between the end of the final symbol of the PPDU initiating the mode switch and the start of the PPDU transmitted using the new PHY mode.                                                         |
| <i>SecondaryFskPreambleLength</i> | Integer | 0–16        | The number of 1-octet patterns, as described in 20.2.1.1, in the secondary preamble if the new mode is SUN FSK. This parameter does not apply if the new mode is SUN OFDM or SUN O-QPSK.                                                |
| <i>SecondaryFskSfd</i>            | Boolean | TRUE, FALSE | If the new mode is SUN FSK, a value of TRUE indicates that a secondary SFD is transmitted. A value of FALSE indicates that a secondary SFD is not transmitted. This parameter does not apply if the new mode is SUN OFDM or SUN O-QPSK. |

## 12. O-QPSK PHY

### 12.1 PPDU format

The PPDU shall be formatted as illustrated in Figure 12-1.

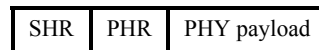


Figure 12-1—Format of the PPDU

#### 12.1.1 SHR field format

The SHR field shall be formatted as illustrated in Figure 12-2.

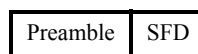


Figure 12-2—Format of the SHR

##### 12.1.1.1 Preamble field

The length of the Preamble field for the O-QPSK PHYs shall be 8 symbols (i.e., 4 octets), and the bits in the Preamble field shall be binary zeros.

##### 12.1.1.2 SFD field

The SFD is a field indicating the end of the SHR and the start of the packet data. The SFD shall be formatted as illustrated in Figure 12-3.

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

Figure 12-3—Format of the SFD field

#### 12.1.2 PHR field format

The PHR field shall be formatted as illustrated in Figure 12-4.

| Bits: 0–6    | 7        |
|--------------|----------|
| Frame Length | Reserved |

Figure 12-4—Format of the PHR

##### 12.1.2.1 Frame Length field

The Frame Length field specifies the total number of octets contained in the PSDU (i.e., PHY payload).

##### 12.1.2.2 PHY Payload field

The PHY Payload field carries the PSDU.

## 12.2 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 1 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).

### 12.2.1 Data rate

The data rate of the O-QPSK PHY shall be 250 kb/s when operating in the 2450 MHz, 915 MHz, 780 MHz or 2380 MHz bands and shall be 100 kb/s when operating in the 868 MHz band.

Support for the 2450 MHz O-QPSK PHY is mandatory when operating in the 2450 MHz band.

Support for the 2380 MHz O-QPSK PHY is mandatory when operating in the 2380 MHz band.

The O-QPSK PHY is not mandatory in the 868 MHz or 915 MHz band. If the O-QPSK PHY is used in the 868 MHz or 915 MHz band, then the same device shall be capable of signaling using the BPSK PHY as well.

### 12.2.2 Reference modulator diagram

The functional block diagram in Figure 12-5 is provided as a reference for specifying the O-QPSK PHY modulation and spreading functions.

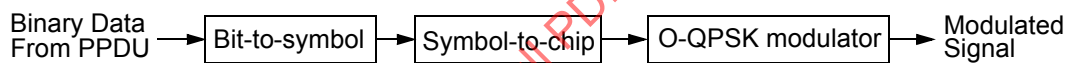


**Figure 12-5—Modulation and spreading functions for the O-QPSK PHYs**

### 12.2.3 Bit-to-symbol mapping

All binary data contained in the PPDU shall be encoded using the modulation and spreading functions shown in Figure 12-5. This subclause describes how binary information is mapped into data symbols.

The 4 LSBs (b0, b1, b2, b3) of each octet shall map into one data symbol, and the 4 MSBs (b4, b5, b6, b7) of each octet shall map into the next data symbol. Each octet of the PPDU is processed through the modulation and spreading functions, as illustrated in Figure 12-5, sequentially, beginning with the Preamble field and ending with the last octet of the PSDU. Within each octet, the least significant symbol (b0, b1, b2, b3) is processed first and the most significant symbol (b4, b5, b6, b7) is processed second.

### 12.2.4 Symbol-to-chip mapping

In the 2450 MHz and 2380 MHz bands, each data symbol shall be mapped into a 32-chip PN sequence as specified in Table 12-1. The PN sequences are related to each other through cyclic shifts and/or conjugation (i.e., inversion of odd-indexed chip values).

**Table 12-1—Symbol-to-chip mapping for the 2450 MHz and 2380 MHz bands**

| Data symbol | Chip values ( $c_0$ $c_1$ ... $c_{30}$ $c_{31}$ )               |
|-------------|-----------------------------------------------------------------|
| 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 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 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           | 0 0 1 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 1 0 1 0 0 1 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 |
| 5           | 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 |
| 6           | 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 1 1 0 0 1 |
| 7           | 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 1 |
| 8           | 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 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 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          | 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 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          | 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          | 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 |
| 15          | 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 1 0 0 0 |

In the 915 MHz, 868 MHz, and 780 MHz bands, each data symbol shall be mapped into a 16-chip PN sequence as specified in Table 12-2.

**Table 12-2—Symbol-to-chip mapping for the 915 MHz, 868 MHz, and 780 MHz bands**

| Data symbol | Chip values ( $c_0$ $c_1$ ... $c_{14}$ $c_{15}$ ) |
|-------------|---------------------------------------------------|
| 0           | 0 0 1 1 1 1 1 0 0 0 1 0 0 1 0 1                   |
| 1           | 0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 1                   |
| 2           | 0 1 0 1 0 0 1 1 1 1 1 0 0 0 1 0                   |
| 3           | 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0                   |
| 4           | 0 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0                   |
| 5           | 1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1                   |
| 6           | 1 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1                   |
| 7           | 1 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0                   |
| 8           | 0 1 1 0 1 0 1 1 0 1 1 1 0 0 0 0                   |
| 9           | 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0                   |

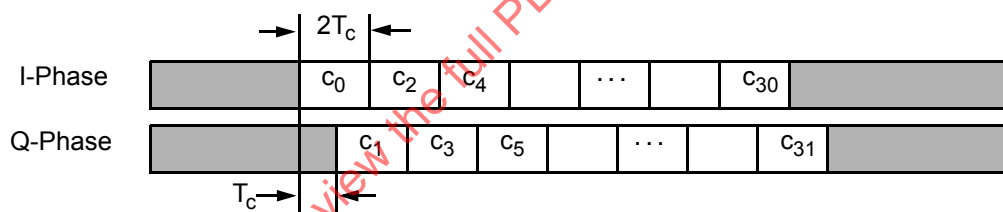


**Table 12-2—Symbol-to-chip mapping for the 915 MHz, 868 MHz, and 780 MHz bands (continued)**

| Data symbol | Chip values ( $c_0$ $c_1$ ... $c_{14}$ $c_{15}$ ) |
|-------------|---------------------------------------------------|
| 10          | 0 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1                   |
| 11          | 1 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1                   |
| 12          | 0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1                   |
| 13          | 1 1 0 1 1 1 0 0 0 0 0 1 1 0 1 0                   |
| 14          | 1 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0                   |
| 15          | 1 0 1 0 1 1 0 1 1 1 0 0 0 0 0 1                   |

**12.2.5 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. In the 2450 MHz and 2380 MHz bands, each data symbol is represented by a 32-chip sequence, and so the chip rate is 32 times the symbol rate. In the 915 MHz, 868 MHz, and 780 MHz bands, each data symbol is represented by a 16-chip sequence, and so the chip rate 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, as illustrated in Figure 12-6, where  $T_c$  is the inverse of the chip rate.

**Figure 12-6—O-QPSK chip offsets****12.2.6 Pulse shape**

In the 2450 MHz, 915 MHz, 868 MHz, and 2380 MHz bands, the half-sine pulse shape is used to represent each baseband chip and is as follows:

$$p(t) = \begin{cases} \sin\left(\pi \frac{t}{2T_c}\right), & 0 \leq t \leq 2T_c \\ 0, & \text{otherwise} \end{cases}$$

$$p(t) = \begin{cases} \frac{\sin(\pi t/T_c)}{\pi t/T_c} \times \frac{\cos(r\pi t/T_c)}{1-4r^2t^2/T_c^2}, & t \neq 0 \\ 1, & t = 0 \end{cases}$$
$$y(t) = \sum_{k=-\infty}^{\infty} c_{2k} p(t - 2kT_c) + j c_{2k+1} p(t - 2kT_c - T_c)$$

- 779–787 MHz
- 868.0–868.6 MHz
- 902–928 MHz
- 2360–2400 MHz
- 2400.0–2483.5 MHz

**12.3.2 Transmit power spectral density (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  $r = 0.2$ , as follows:

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

When operating in the 915 MHz or 780 MHz band, the transmitted spectral products shall be less than the limits specified in Table 12-3. 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  $\pm 600$  kHz of the carrier frequency  $f_c$ .

**Table 12-3—O-QPSK PHY transmit PSD limits for the 915 MHz and 780 MHz bands**

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

When operating in the 2380 MHz or 2450 MHz band, the transmitted spectral products shall be less than the limits specified in Table 12-4. 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  $\pm 1$  MHz of the carrier frequency.

**Table 12-4—O-QPSK transmit PSD limits for the 2380 MHz and 2450 MHz bands**

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

**12.3.3 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 780 MHz, 915 MHz, 2380 MHz, or 2450 MHz band with an accuracy of  $\pm 40 \times 10^{-6}$ .

**12.3.4 Receiver sensitivity**

Under the conditions specified in 10.1.7, a compliant device shall be capable of achieving a receiver sensitivity of −85 dBm or better.

**12.3.5 Receiver interference rejection**

This subclause applies only to the 780 MHz, 915 MHz, 2380 MHz, and 2450 MHz bands as there is only one channel available in the 868 MHz band.

The minimum receiver interference rejection levels are given in Table 12-5. 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 12-5—Minimum receiver interference rejection requirements for the 780 MHz, 915 MHz, 2380 MHz, and 2450 MHz bands**

| 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 O-QPSK PHY signal, as defined by 12.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB greater than the maximum allowed receiver sensitivity given in 12.3.4.

In either the adjacent or the alternate channel, a compliant O-QPSK PHY signal, as defined by 12.2, is input at the level specified in Table 12-5 relative to the desired signal. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

#### 12.3.6 TX-to-RX turnaround time

The O-QPSK PHY shall have a TX-to-RX turnaround time as defined in 10.2.1.

#### 12.3.7 RX-to-TX turnaround time

The O-QPSK PHY shall have an RX-to-TX turnaround time as defined in 10.2.2.

#### 12.3.8 EVM

The O-QPSK PHY shall have EVM values of less than 35% when measured for 1000 chips using the measurement process defined in 10.2.3.

#### 12.3.9 Transmit center frequency tolerance

The O-QPSK PHY transmit center frequency tolerance shall be  $\pm 40 \times 10^{-6}$  maximum.

#### 12.3.10 Transmit power

The O-QPSK PHY shall be capable of transmitting at a power level of at least  $-3$  dBm.

#### 12.3.11 Receiver maximum input level of desired signal

The O-QPSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

#### 12.3.12 Receiver ED

The O-QPSK PHY shall provide the receiver ED measurement as described in 10.2.5.

#### 12.3.13 LQI

The O-QPSK PHY shall provide the LQI measurement as described in 10.2.6.

## 13. Binary phase-shift keying (BPSK) PHY

### 13.1 PPDU format

The BPSK PHY shall use the PPDU format described in 12.1, except that the Preamble field is 32 symbols (4 octets).

### 13.2 Modulation and spreading

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

#### 13.2.1 BPSK PHY data rates

The data rate of the 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.

#### 13.2.2 Reference modulator

The functional block diagram of the reference modulator in Figure 13-1 is provided as a reference for specifying the BPSK PHY modulation and spreading functions. 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 PHY Payload field.

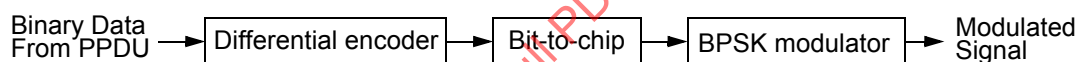


Figure 13-1—BPSK reference modulator block diagram

#### 13.2.3 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 as follows:

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

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 as follows:

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

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

### 13.2.4 Bit-to-chip mapping

Each input bit shall be mapped into a 15-chip PN sequence as specified in Table 13-1.

**Table 13-1—Symbol-to-chip mapping**

| Input bits | Chip values ( $c_0$ $c_1$ ... $c_{14}$ ) |
|------------|------------------------------------------|
| 0          | 1 1 1 1 0 1 0 1 1 1 0 0 1 0 0 0          |
| 1          | 0 0 0 0 1 0 1 0 0 1 1 0 1 1 1 1          |

### 13.2.5 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.

#### 13.2.5.1 Pulse shape

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

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

#### 13.2.5.2 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.

## 13.3 BPSK PHY RF requirements

### 13.3.1 Operating frequency range

The BPSK PHY operates in the following frequency bands:

- 868.0–868.6 MHz
- 902–928 MHz

### 13.3.2 915 MHz band transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 13-2. For the 915 MHz band, 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  $\pm 600$  kHz of the carrier frequency.

**Table 13-2—915 MHz band transmit PSD limits**

| Frequency band | Frequency                     | Relative limit | Absolute limit |
|----------------|-------------------------------|----------------|----------------|
| 915 MHz        | $ f - f_c  > 1.2 \text{ MHz}$ | −20 dB         | −20 dBm        |

**13.3.3 Symbol rate**

The symbol rate of a 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  $\pm 40 \times 10^{-6}$ .

**13.3.4 Receiver sensitivity**

Under the conditions specified in 10.1.7, a compliant device shall be capable of achieving a receiver sensitivity of −92 dBm or better.

**13.3.5 Receiver interference rejection**

This subclause applies only to the 915 MHz band as there is only one channel available in the 868 MHz band.

The minimum receiver interference rejection levels are given in Table 13-3.

**Table 13-3—Minimum receiver interference rejection requirements for the 915 MHz BPSK PHYs**

| Frequency band | Adjacent channel rejection | Alternate channel rejection |
|----------------|----------------------------|-----------------------------|
| 915 MHz band   | 0 dB                       | 30 dB                       |

For the 915 MHz band, 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.

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant 915 MHz BPSK PHY signal, as defined by 13.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB greater than the maximum allowed receiver sensitivity given in 13.3.4.

In either the adjacent or the alternate channel, a compliant 915 MHz BPSK PHY signal, as defined by 13.2, is input at the relative level specified in Table 13-3. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

**13.3.6 TX-to-RX turnaround time**

The BPSK PHY shall have a TX-to-RX turnaround time as defined in 10.2.1.

**13.3.7 RX-to-TX turnaround time**

The BPSK PHY shall have an RX-to-TX turnaround time as defined in 10.2.2.

**13.3.8 EVM**

The BPSK PHY shall have EVM values of less than 35% when measured for 1000 chips using the measurement process defined in 10.2.3.

**13.3.9 Transmit center frequency tolerance**

The BPSK PHY transmit center frequency tolerance shall be  $\pm 40 \times 10^{-6}$  maximum.

**13.3.10 Transmit power**

The BPSK PHY shall be capable of transmitting at a power level of at least  $-3$  dBm.

**13.3.11 Receiver maximum input level of desired signal**

The BPSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

**13.3.12 Receiver ED**

The BPSK PHY shall provide the receiver ED measurement as described in 10.2.5.

**13.3.13 LQI**

The BPSK PHY shall provide the LQI measurement as described in 10.2.6.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018



## 14. Amplitude shift keying (ASK) PHY

### 14.1 Status of ASK PHY

The use of mechanisms described in this clause is deprecated. Consequently, this clause may be removed in a later revision of the standard.

### 14.2 PPDU format

The ASK PHY shall use the PPDU format described in 12.1, except that the SHR field is defined differently.

The SHR field uses a subset of the PSSS coding used for the PHR field and PHY Payload field. The chips in the SHR field shall be transmitted with BPSK modulation using the same chip rates, as described in 14.3.5, and pulse shaping, as described in 14.3.6, that are used for the PHR field and PHY Payload field. However, the symbol-to-chip mapping for the SHR field is different, as described in 14.2.1 and 14.2.2.

#### 14.2.1 Preamble field for ASK PHY

The Preamble field is generated by repeating 2 times for 868 MHz and repeating 6 times for 915 MHz sequence number 0 from Table 14-1 and Table 14-2, respectively. The resulting preamble duration is 160  $\mu$ s for 868 MHz and 120  $\mu$ s 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 14-1. The pulse shaping is the same as for the PHY Payload field as defined in 14.3.6.

#### 14.2.2 SFD for ASK PHY

The SFD for 868 MHz and 915 MHz is the inverted sequence 0 from Table 14-1 and Table 14-2, respectively. The SFD will be BPSK modulated as shown in Figure 14-1. The pulse shaping is the same as for the PHY Payload field as defined in 14.3.6.

The preamble length for the ASK PHY shall be 2 symbols (5 octets). The SFD for the ASK PHY shall be 0.625 octets (1 symbol).

The preamble length and SFD for the ASK PHY is expressed in equivalent octet times as the ASK is defined using a special symbol.

### 14.3 Modulation and spreading

The ASK PHY employs a multi-code modulation technique named parallel sequence spread spectrum (PSSS). 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.

#### 14.3.1 ASK PHY 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 MHz or 915 MHz band. If the ASK PHY is used in the 868 MHz or 915 MHz band, then the same device shall be capable of signaling using the BPSK PHY as well.

### 14.3.2 Reference modulator

The functional block diagram of the reference modulator in Figure 14-1 is provided as a reference for specifying the PSSS modulation and spreading functions.

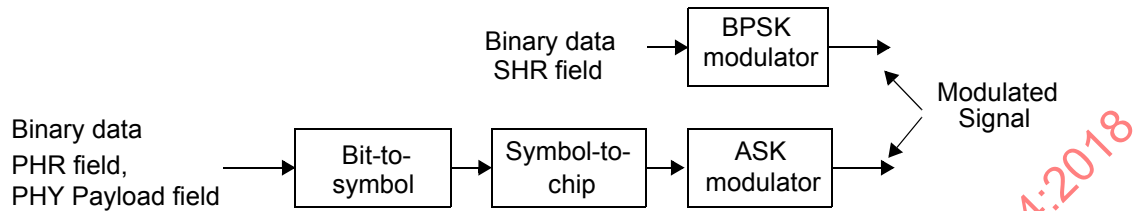


Figure 14-1—ASK reference modulator block diagram

Each octet of the PPDU is sequentially processed through the spreading and modulation functions. The SHR field shall be encoded using the BPSK modulator, and the PHR field and PHY Payload field shall be processed using the bit-to-symbol mapping, and symbol-to-chip spreading and the ASK modulator as shown in Figure 14-1.

### 14.3.3 Bit-to-symbol mapping

The first 20 bits for 868 MHz and the first 5 bits for 915 MHz, starting with the LSB (b0) of the first octet of the PHR field and continuing with the subsequent octet of the PHR field, 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 PHR field and PHY Payload field 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.

### 14.3.4 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 14-2 provides an overview of the symbol-to-chip mapping.

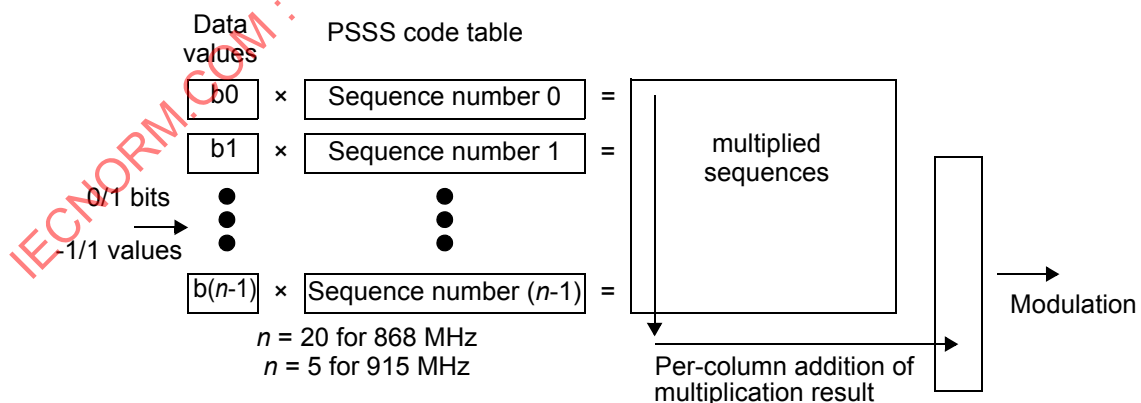


Figure 14-2—Symbol-to-chip mapping for PHR field and PHY Payload field

Each bit of the data stream is multiplied with its corresponding sequence of the code table defined in Table 14-1 for 868 MHz and Table 14-2 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 b0 of the data symbol is multiplied with sequence number “0”, bit b1 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 14-1 or Table 14-2, 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 sequence is produced by linearly summing chip 0 from each of the modulated sequences. The per-column results are 32 chips  $c_0 \dots c_{31}$  for 915 MHz and 64 half-chips  $c_0 \dots c_{63}$  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 follows:

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

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. Let  $p''(m)$  be the output of the second precoding step:

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

where Max' is the maximum within  $p'(m)$ . An example is given in 14.4.14.

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 14.3.5.

### 14.3.5 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.

Table 14-1—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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |
| 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 |    |    |    |

Table 14-2—PSSS code table used in symbol-to-chip mapping for 915 MHz

| Sequence number | Chip number |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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  | -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 | 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 | 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 | -1 |

### 14.3.6 Pulse shape

The root-raised-cosine pulse shape used to represent each baseband chip is described as follows:

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

with a roll-off 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.  $T_c$  is the chip duration (not half-chip) for 868 MHz and 915 MHz.

### 14.3.7 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.

## 14.4 ASK PHY RF requirements

### 14.4.1 Operating frequency range

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

### 14.4.2 915 MHz band transmit PSD mask

The transmitted spectral products shall be less than the limits specified in Table 14-3. 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  $\pm 600$  kHz of the carrier frequency.

Table 14-3—915 MHz band ASK PHY transmit PSD limits

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

### 14.4.3 Symbol rate

The ASK PHY symbol rate shall be 12.5 ksymbol/s  $\pm 40 \times 10^{-6}$  for 868 MHz and 50 ksymbol/s  $\pm 40 \times 10^{-6}$  for 915 MHz.

**14.4.4 Receiver sensitivity**

Under the conditions specified in 10.1.7, a compliant device shall be capable of achieving a receiver sensitivity of  $-85$  dBm or better.

**14.4.5 Receiver interference rejection**

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 receiver interference rejection levels are given in Table 14-4. 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 14-4—Minimum receiver interference rejection 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 ASK PHY signal, as defined by 14.3, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB greater than the maximum allowed receiver sensitivity given in 14.4.4.

In either the adjacent or the alternate channel, a compliant ASK PHY signal, as defined by 14.3, is input at the relative level specified in Table 14-4. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

**14.4.6 TX-to-RX turnaround time**

The ASK PHY shall have a TX-to-RX turnaround time as defined in 10.2.1.

**14.4.7 RX-to-TX turnaround time**

The ASK PHY shall have an RX-to-TX turnaround time as defined in 10.2.2.

**14.4.8 EVM**

The ASK PHY shall have EVM values of less than 35% when measured for 1000 chips using the measurement process defined in 10.2.3.

**14.4.9 Transmit center frequency tolerance**

The ASK PHY transmit center frequency tolerance shall be  $\pm 40 \times 10^{-6}$  maximum.

**14.4.10 Transmit power**

The ASK PHY shall be capable of transmitting at a power level of at least  $-3$  dBm.

**14.4.11 Receiver maximum input level of desired signal**

The ASK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

**14.4.12 Receiver ED**

The ASK PHY shall provide the receiver ED measurement as described in 10.2.5.

**14.4.13 LQI**

The ASK PHY shall provide the LQI measurement as described in 10.2.6.

**14.4.14 Example of PSSS encoding**

Table 14-5 shows the example 5-bit symbol that will be encoded using Table 14-2, the code table for 915 MHz.

**Table 14-5—Example 5 bit symbol**

| Data bits |           |
|-----------|-----------|
| Bit       | Bit value |
| b0        | 1         |
| b1        | −1        |
| b2        | 1         |
| b3        | −1        |
| b4        | −1        |

By weighting sequence 0 with b0, sequence 1 with b1, ..., sequence 4 with b4, the weighted sequences in Table 14-6 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 14-6, the  $\text{Max}\{p(m)\} = 5$  and the  $\text{Min}\{p(m)\} = -5$ . The precoding of one symbol is executed independently of the precoding of any other symbol with the two steps described in 14.3.4. The resulting  $p'(m)$  and  $p''(m)$  are shown in Table 14-7.

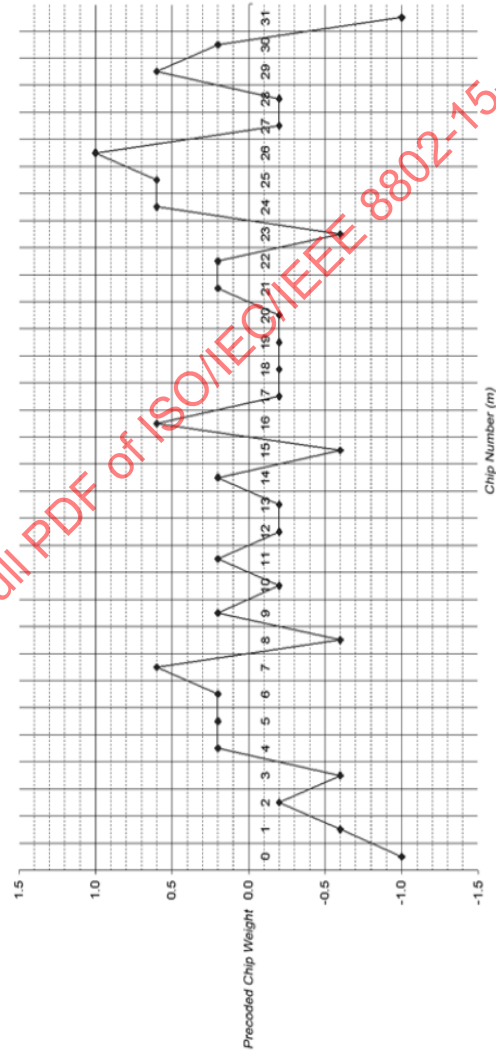
A graph illustrating the example  $p''(m)$  chip weights that will be applied to the pulse shaping filter in 14.3.6 is shown in Figure 14-3.

Table 14-6—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 14-7— $p'(m)$ , 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 14-3—Precoded PSSS symbol  $p''(m)$



## 15. Chirp spread spectrum (CSS) PHY

### 15.1 CSS PPDU format

The CSS PPDU shall be formatted as illustrated in Figure 15-1.



**Figure 15-1—Format of the CSS PPDU**

The SHR field shall be formatted as illustrated in Figure 15-2.

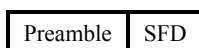


**Figure 15-2—Format of the SHR**

#### 15.1.1 Preamble field

The Preamble field for 1 Mb/s consists of 8 chirp symbols, and the Preamble field for optional 250 kb/s consists of 20 chirp symbols as specified in Table 15-1. The preamble sequence from Table 15-1 should be applied directly to both the I input and the Q input of QPSK.

**Table 15-1—Preamble sequence**

| Data rate | Preamble sequence |
|-----------|-------------------|
| 1 Mb/s    | ones (0:31)       |
| 250 kb/s  | ones (0:79)       |

where ones(0: $N$ ) for integer number  $N$  is defined as an 1-by- $N$  matrix of ones.

#### 15.1.2 SFD field

SFD field bit sequences for the CSS PHY type are defined in Table 15-2. Different SFD sequences are defined for the two different data rates. An SFD sequence from Table 15-2 shall be applied directly to both inputs (I and Q) of the QPSK mapper. An SFD sequence starts with bit 0.

**Table 15-2—CSS SFD field bit sequence**

| Data rate | Bit (0:15)                              |
|-----------|-----------------------------------------|
| 1 Mb/s    | -1 1 1 1 -1 1 -1 -1 1 -1 -1 1 1 1 -1 -1 |
| 250 kb/s  | -1 1 1 1 1 -1 1 -1 -1 -1 1 -1 -1 -1 1 1 |

### 15.1.3 PHR field

The CSS PHR field shall be formatted as illustrated in Figure 15-3.

| Bits: 0–6 | 7–11     |
|-----------|----------|
| Length    | Reserved |

**Figure 15-3—Format of the CSS PHR**

The Length field is an unsigned integer and shall be set to the length of the PHY Payload field, in octets. The Length field shall be transmitted LSB first.

### 15.1.4 PHY Payload field

The PHY Payload field carries the PSDU.

## 15.2 Modulation and spreading

The CSS PHY uses CSS techniques in combination with differential quadrature phase-shift keying (DQPSK) and 8-ary or 64-ary bi-orthogonal coding for 1 Mb/s data rate or 250 kb/s data rate, respectively. By using alternating time gaps in conjunction with sequences of chirp signals (subchirps) in different frequency subbands with different chirp directions, this CSS PHY provides subchirp sequence division as well as frequency division.

### 15.2.1 Data rates

The data rate of the CSS (2450 MHz) PHY shall be 1 Mb/s. An additional data rate of 250 kb/s shall be optional.

### 15.2.2 Reference modulator

The functional block diagram of the reference modulator in Figure 15-4 is provided as a reference for specifying the 2450 MHz CSS PHY modulation for both 1 Mb/s and optional 250 kb/s. All binary data contained in the PHR field and PHY Payload field shall be encoded using the modulation shown in Figure 15-4.

### 15.2.3 De-multiplexer (DEMUX)

For each packet, the initial position of the DEMUX shown in Figure 15-4 shall be set to serve the I path (upper path). Thus, the first bit of the incoming stream of information bits of a packet shall be switched to the I path, and the second bit shall be switched to the Q path.

### 15.2.4 Serial-to-parallel mapping

By using two serial-to-parallel converters, the substreams are independently partitioned into sets of bits to form data symbols. For the mandatory data rate of 1 Mb/s, a data symbol shall consist of three bits. Within the binary data symbol (b0,b1,b2), the first input data bit for each of I and Q is assigned b0, and the third input data bit is assigned b2. For the optional data rate of 250 kb/s, a data symbol shall consist of 6 bits. Within the binary data symbol (b0,b1,b2,b3,b4,b5), the first input data bit for each of I and Q is assigned b0, and the sixth input data bit is assigned b5.

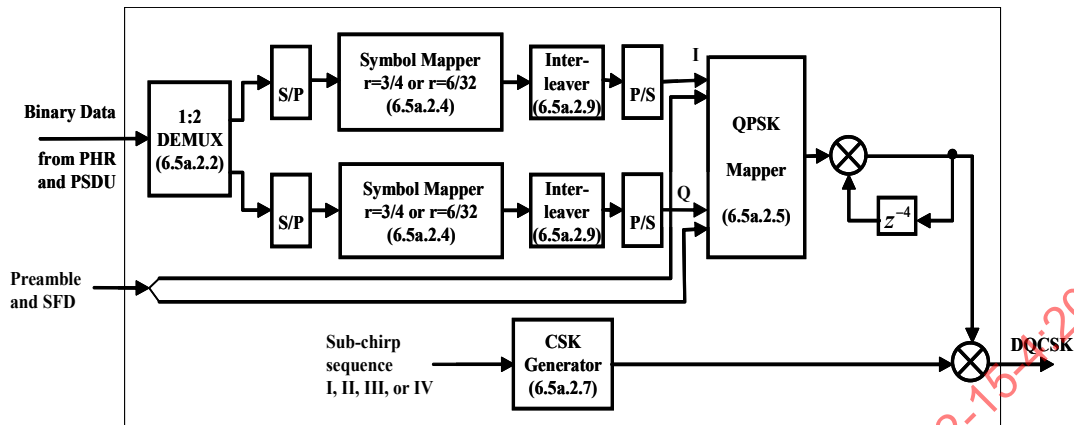


Figure 15-4—Differential bi-orthogonal quaternary-chirp-shift-keying modulator and spreading ( $r = 3/4$  for 8-ary 1 Mb/s,  $r = 3/16$  for 64-ary 250 kb/s)

15.2.5 Data-symbol-to-bi-orthogonal-codeword mapping

Each 3-bit data symbol shall be mapped onto a 4-chip bi-orthogonal codeword ( $c_0, c_1, c_2, c_3$ ) for the 1 Mb/s data rate as specified in Table 15-3. Each 6-bit data symbol shall be mapped onto a 32-chip bi-orthogonal codeword ( $c_0, c_1, c_2, \dots, c_{31}$ ) for the optional 250 kb/s data rate as specified in Table 15-4.

Table 15-3—8-ary bi-orthogonal mapping ( $r = 3/4$ , 1 Mb/s)

| Data symbol | Codeword ( $c_0$ $c_1$ $c_2$ $c_3$ ) |
|-------------|--------------------------------------|
| 0           | 1 1 1 1                              |
| 1           | 1 -1 1 -1                            |
| 2           | 1 1 -1 -1                            |
| 3           | 1 -1 -1 1                            |
| 4           | -1 -1 -1 -1                          |
| 5           | -1 1 -1 1                            |
| 6           | -1 -1 1 1                            |
| 7           | -1 1 1 -1                            |

Table 15-4—64-ary bi-orthogonal mapping ( $r = 3/16$ , 250 kb/s)

| Data symbol | Codeword ( $c_0$ $c_1$ $c_2$ ... $c_{31}$ )                                     |
|-------------|---------------------------------------------------------------------------------|
| 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 |

**Table 15-4—64-ary bi-orthogonal mapping ( $r = 3/16$ , 250 kb/s) (continued)**

| Data symbol | Codeword (c0 c1 c2 ... c31)                                                          |
|-------------|--------------------------------------------------------------------------------------|
| 3           | 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>-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<br>-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<br>-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<br>-1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1   |
| 20          | 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1<br>-1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1   |
| 21          | 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1 -1 1<br>-1 1 -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1   |
| 22          | 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1<br>-1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1   |
| 23          | 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1 -1 1 1 -1<br>-1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1   |

Table 15-4—64-ary bi-orthogonal mapping ( $r = 3/16$ , 250 kb/s) (continued)

| Data symbol | Codeword (c0 c1 c2 ... c31)                                                                        |
|-------------|----------------------------------------------------------------------------------------------------|
| 24          | 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1<br>-1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1 1 1                 |
| 25          | 1 -1 1 -1 1 -1 1 -1 -1 1 -1 1 -1 1 -1 1<br>-1 1 -1 1 -1 1 -1 1 1 -1 1 -1 1 -1 1 -1                 |
| 26          | 1 1 -1 -1 1 1 -1 -1 -1 -1 1 1 -1 -1 1 1<br>-1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1                 |
| 27          | 1 -1 -1 1 1 -1 -1 1 -1 1 1 -1 -1 1 1 -1<br>-1 1 1 -1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 1                 |
| 28          | 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1<br>-1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 -1 -1 -1                 |
| 29          | 1 -1 1 -1 -1 1 -1 1 -1 1 -1 1 1 -1 1 -1<br>-1 1 -1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1                 |
| 30          | 1 1 -1 -1 -1 -1 1 1 -1 -1 1 1 1 1 -1 -1<br>-1 -1 1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 1 1                 |
| 31          | 1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 -1 -1 -1 1<br>-1 1 1 -1 1 -1 -1 1 1 -1 -1 -1 1 1 -1 -1               |
| 32          | -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1<br>-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 |
| 33          | -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1<br>-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1                 |
| 34          | -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1<br>-1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1                 |
| 35          | -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1<br>-1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1                 |
| 36          | 1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1<br>-1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1                  |
| 37          | -1 1 -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1<br>-1 1 -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1                 |
| 38          | -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1<br>-1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1                 |
| 39          | -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1<br>-1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1                 |
| 40          | -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1 1 1<br>-1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1 1 1                 |
| 41          | -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 -1<br>-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 -1               |
| 42          | -1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1<br>-1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1                 |
| 43          | -1 1 1 -1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 1<br>-1 1 1 -1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 1                 |
| 44          | -1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 -1 -1 -1<br>-1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 -1 -1 -1                 |

**Table 15-4—64-ary bi-orthogonal mapping ( $r = 3/16$ , 250 kb/s) (continued)**

| Data symbol | Codeword (c0 c1 c2 ... c31)                                                         |
|-------------|-------------------------------------------------------------------------------------|
| 45          | -1 1 -1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1<br>-1 1 -1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1        |
| 46          | -1 -1 1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 1 1<br>-1 -1 1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 1 1  |
| 47          | -1 1 1 -1 1 -1 -1 1 1 -1 -1 1 -1 1 1 -1<br>-1 1 1 -1 1 -1 -1 1 1 -1 -1 1 -1 1 1 -1  |
| 48          | -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1<br>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  |
| 49          | -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1<br>1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1  |
| 50          | -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1<br>1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1  |
| 51          | -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1<br>1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1  |
| 52          | -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 -1 1 1 1<br>1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 |
| 53          | -1 1 -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1<br>1 -1 1 -1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1  |
| 54          | -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1<br>1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1  |
| 55          | -1 1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1<br>1 -1 -1 1 1 1 -1 1 -1 1 -1 1 -1 1 -1 1   |
| 56          | -1 -1 1 -1 -1 -1 -1 1 1 1 1 1 1 1 1 1<br>1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1    |
| 57          | -1 1 -1 1 -1 1 -1 1 1 -1 1 -1 1 -1 1 -1<br>1 -1 1 -1 1 -1 1 -1 -1 1 -1 1 -1 1 -1 1  |
| 58          | -1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1<br>1 1 -1 -1 1 1 -1 -1 -1 -1 1 1 -1 -1 1 1  |
| 59          | -1 1 1 -1 -1 1 1 -1 1 -1 -1 1 1 -1 -1 1<br>1 -1 -1 1 1 -1 -1 1 1 1 -1 -1 1 1 -1 1   |
| 60          | -1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 -1 -1 -1<br>1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1   |
| 61          | -1 1 -1 1 1 -1 1 -1 1 -1 1 -1 -1 1 -1 1<br>1 -1 1 -1 -1 1 -1 1 -1 1 1 -1 1 1 -1 1   |
| 62          | -1 -1 1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1 1<br>1 1 -1 -1 -1 -1 1 1 -1 -1 1 1 1 1 -1 -1   |
| 63          | -1 1 1 -1 1 -1 -1 1 1 -1 -1 1 -1 1 1 -1<br>1 -1 -1 1 -1 1 1 -1 -1 1 1 -1 1 -1 1 1   |

**15.2.6 Parallel-to-serial converter and QPSK symbol mapping**

Each bi-orthogonal codeword shall be converted to a serial chip sequence. Within each 4-chip codeword ( $c_0, c_1, c_2, c_3$ ) for the 1 Mb/s data rate, the least significant chip  $c_0$  is processed first, and the most significant chip  $c_3$  is processed last for I and Q, respectively. Within each 32-chip codeword ( $c_0, c_1, c_2, \dots, c_{31}$ ) for the 250 kb/s data rate, the least significant chip  $c_0$  is processed first, and the most significant chip  $c_{31}$  is processed last for I and Q, respectively. Each pair of I and Q chips shall be mapped onto a QPSK symbol as specified in Table 15-5.

**Table 15-5—QPSK symbol mapping**

| Input chips<br>( $I_{n,k}, Q_{n,k}$ ) | Magnitude | Output phase<br>(rad) |
|---------------------------------------|-----------|-----------------------|
| 1, 1                                  | 1         | 0                     |
| -1, 1                                 | 1         | $\pi/2$               |
| 1 -1                                  | 1         | $-\pi/2$              |
| -1, -1                                | 1         | $\pi$                 |

**15.2.7 DQPSK coding**

The stream of QPSK symbols shall be differentially encoded by using a differential encoder with a QPSK symbol feedback memory of length 4. (In other words, the phase differences between QPSK symbol 1 and 5, 2 and 6, 3 and 7, 4 and 8, and so on are computed.) For a detailed explanation of the index variables  $n$  and  $k$ , as described in 15.3.3.

DQPSK output:

$$e^{j\theta_{n,k}} = e^{j\theta_{n-1,k}} \times e^{j\phi_{n,k}}$$

where

$e^{j\phi_{n,k}}$  is DQPSK input

$e^{j\theta_{n-1,k}}$  is stored in feedback memory

For every packet, the initial values of all four feedback memory stages of the differential encoder shall be set as follows:

$$e^{j\pi/4}$$

**15.2.8 DQPSK-to-DQCSK modulation**

The stream of DQPSK symbols shall be modulated onto the stream of subchirps that is generated by the chirp-shift keying (CSK) generator. The effect of the differential quadrature chirp-shift keying (DQCSK) modulation shall be that each subchirp is multiplied with a DQPSK value that has unit magnitude and has constant phase for the duration of the subchirp. An example of this operation can be found in 15.3.6.

**15.2.9 CSK generator**

The CSK generator shall periodically generate one of the four defined subchirp sequences (chirp symbols) as specified in 15.3.3. Since each chirp symbol consists of four subchirps, the subchirp rate is four times higher than the chirp symbol rate.

**15.2.10 Bit interleaver**

The bit interleaver is applied only for the optional data rate of 250 kb/s. The 32-chip bi-orthogonal codewords for the optional 250 kb/s data rate are interleaved prior to the parallel-to-serial converter. Bit interleaving provides robustness against double intra-symbol errors caused by the differential detector. The interleaver permutes the chips across two consecutive codewords for each of I and Q, independently.

The memory of the interleaver shall be initialized with zeros before the reception of a packet.

The data stream going into the interleaver shall be padded with zeros if the number of octets to be transmitted does not align with the bounds of the interleaver blocks.

The input-output relationship of this interleaver shall be given as follows:

Input:

even-symbol (c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15,  
c16, c17, c18, c19, c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30, c31)  
odd-symbol (d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15,  
d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26, d27, d28, d29, d30, d31)

Output:

even-symbol (c0, c1, c2, c3, d20, d21, d22, d23, c8, c9, c10, c11, d28, d29, d30, d31,  
c16, c17, c18, c19, d4, d5, d6, d7, c24, c25, c26, c27, d12, d13, d14, d15)  
odd-symbol (d0, d1, d2, d3, c20, c21, c22, c23, d8, d9, d10, d11, c28, c29, c30, c31,  
d16, d17, d18, d19, c4, c5, c6, c7, d24, d25, d26, d27, c12, c13, c14, c15)

NOTE—As shown in Figure 15-4, coding is applied to every bit following the SFD. The first codeword generated shall be counted as zero and thus is even.



### 15.3 Waveform and subchirp sequences

Four individual chirp signals, here called subchirps, shall be concatenated to form a full chirp symbol (subchirp sequence), which occupies two adjacent frequency subbands. Four different subchirp sequences are defined. Each subchirp is weighted with a raised cosine window in the time domain.

#### 15.3.1 Graphical presentation of chirp symbols (subchirp sequences)

Four different sequences of subchirp signals are available for use. Figure 15-5 shows the four different chirp symbols (subchirp sequences) as time frequency diagrams. It can be seen that four subchirps, which have either a linear down-chirp characteristic or a linear up-chirp characteristic, and a center frequency, which has either a positive or a negative frequency offset, are concatenated. The frequency discontinuities between subsequent chirps will not affect the spectrum because the signal amplitude will be zero at these points.

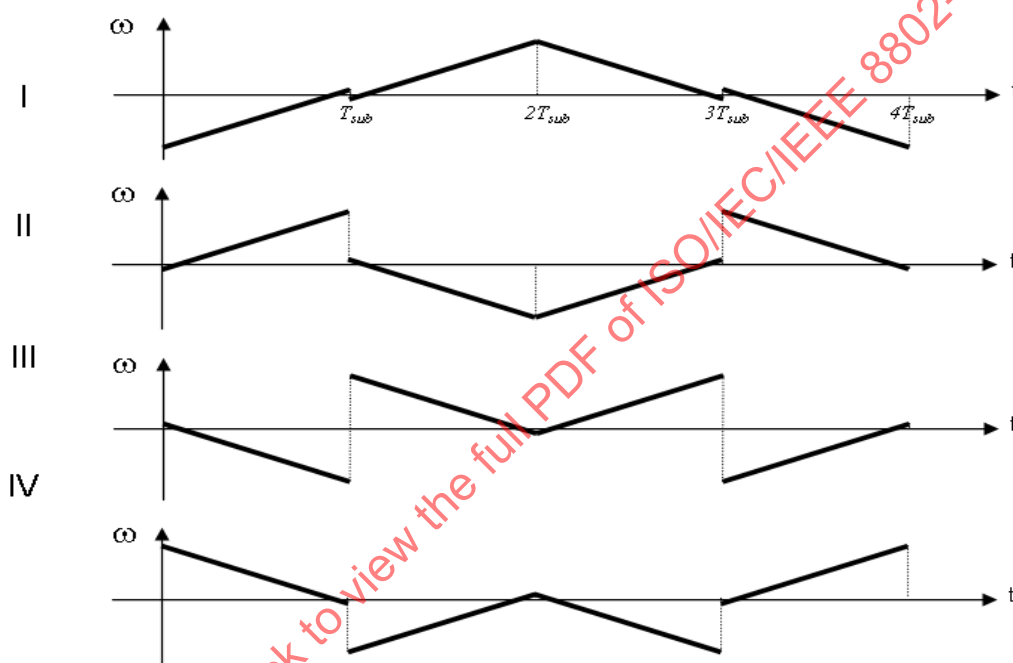


Figure 15-5—Four different combinations of subchirps

#### 15.3.2 Active usage of time gaps

In conjunction with the subchirp sequence, different pairs of time gaps are defined. The time gaps are chosen to make the four sequences even closer to being orthogonal. The time gaps shall be applied alternatively between subsequent chirp symbols as shown in Figure 15-6. The values of the time gaps are calculated from the timing parameters specified in Table 15-8.

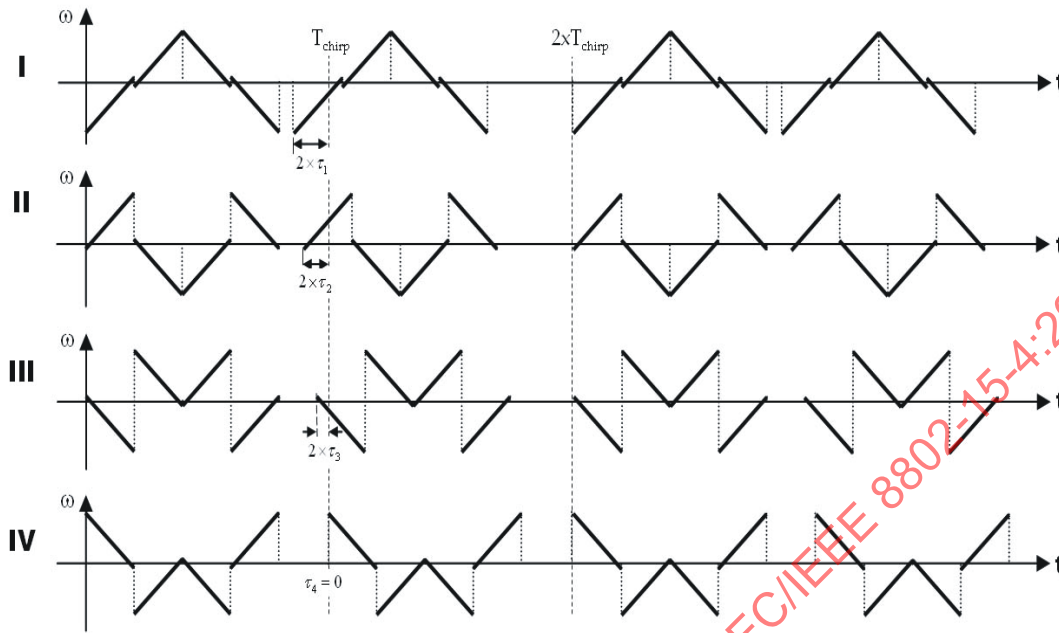


Figure 15-6—Four different time-gap pairs for the four different subchirp sequences

### 15.3.3 Mathematical representation of the continuous time CSS base-band signal

The mathematical representation of the continuous time-domain base-band signal  $\tilde{s}^m(t)$  built of chirp symbols (subchirp sequences) as shown in Figure 15-5 with alternating time gaps as shown in Figure 15-6. The subchirp sequence with its associated time gap is defined to be a chirp symbol as follows:

$$\begin{aligned}\tilde{s}^m(t) &= \sum_{n=0}^{\infty} \tilde{s}^m(t, n) \\ &= \sum_{n=0}^{\infty} \sum_{k=1}^4 \tilde{c}_{n,k} \exp \left[ j \left( \hat{\omega}_{k,m} + \frac{\mu}{2} \xi_{k,m} (t - T_{n,k,m}) \right) (t - T_{n,k,m}) \right] \times P_{RC}(t - T_{n,k,m})\end{aligned}$$

where  $m = 1, 2, 3, 4$  (I, II, III, and IV in Figure 15-5) defines which of the four different possible chirp symbols (subchirp sequences) is used, and  $n = 0, 1, 2, \dots$ , is the sequence number of the chirp symbols.

The  $\tilde{c}_{n,k}$  is the sequence of the complex data that consists of in-phase data  $a_{n,k}$  and quadrature-phase data  $b_{n,k}$  as the output of DQPSK coding.

The possible value of  $a_{n,k}$  and  $b_{n,k}$  are +1 or -1.

$$\tilde{c}_{n,k} = a_{n,k} + jb_{n,k}$$

where

$n$  is the sequence number of chirp symbols  
 $k = 0, 1, 2, \text{ and } 3$  is the subchirp index  
 $j$  is  $\sqrt{-1}$

$\hat{\omega}_{k,m} = 2\pi \times f_{k,n}$  are the center frequencies of the subchirp signals. This value depends on  $m$  and  $k = 1, 2, 3, 4$ , which defines the subchirp number in the subchirp sequence.

$T_{n,k,m}$  defines the starting time of the actual subchirp signal to be generated. It is determined by  $T_{chirp}$ , which is the average duration of a chirp symbol, and by  $T_{sub}$ , which is the duration of a subchirp signal.

$$T_{n,k,m} = \left(k + \frac{1}{2}\right)T_{sub} + nT_{chirp} - (1 - (-1)^n)\tau_m$$

The constant  $\mu$  defines the characteristics of the subchirp signal. A value of  $\mu = 2\pi \times 7.3158 \times 10^{12}$  [rad/s<sup>2</sup>] shall be used.

The function  $P_{RC}$ , which is defined in 15.3.4, is a windowing function that is equal to zero at the edges and outside of the subchirp centered at time zero.

The constant  $\tau_m$  is either not added or added twice and thus determines (but is not identical to) the time gap that was applied between two subsequent subchirp sequences as shown in Figure 15-6.

Table 15-6 shows the values for the subband center frequencies, Table 15-7 the subchirp directions, and Table 15-8 the timing parameters. These time and frequency parameters are assumed to be derived from a reference crystal in a locked manner. In other words, any relative errors in chirp subband center frequencies, chirp rate, and time gaps are equal.

**Table 15-6—Subband center frequencies,  $f_{k,m}$  (MHz, numerical parameters)**

| $m \backslash k$ | 1            | 2            | 3            | 4            |
|------------------|--------------|--------------|--------------|--------------|
| 1                | $f_c - 3.15$ | $f_c + 3.15$ | $f_c + 3.15$ | $f_c - 3.15$ |
| 2                | $f_c + 3.15$ | $f_c - 3.15$ | $f_c - 3.15$ | $f_c + 3.15$ |
| 3                | $f_c - 3.15$ | $f_c + 3.15$ | $f_c + 3.15$ | $f_c - 3.15$ |
| 4                | $f_c + 3.15$ | $f_c - 3.15$ | $f_c - 3.15$ | $f_c + 3.15$ |

**Table 15-7—Subchirp directions,  $\zeta_{k,m}$ , numerical parameters**

| $m \backslash k$ | 1  | 2  | 3  | 4  |
|------------------|----|----|----|----|
| 1                | +1 | +1 | −1 | −1 |
| 2                | +1 | −1 | +1 | −1 |
| 3                | −1 | −1 | +1 | +1 |
| 4                | −1 | +1 | −1 | +1 |

**Table 15-8—Timing parameters for baseband signal**

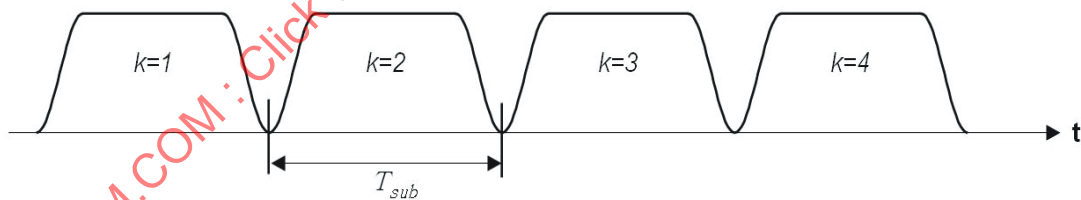
| Symbol             | Value                | Multiple of 1/32MHz |
|--------------------|----------------------|---------------------|
| $T_{\text{chirp}}$ | 6 $\mu\text{s}$      | 192                 |
| $T_{\text{sub}}$   | 1.1875 $\mu\text{s}$ | 38                  |
| $\tau_1$           | 468.75 $\mu\text{s}$ | 15                  |
| $\tau_2$           | 312.5 ns             | 10                  |
| $\tau_3$           | 156.25 ns            | 5                   |
| $\tau_4$           | 0 ns                 | 0                   |

**15.3.4 Raised cosine window for chirp pulse shaping**

The raised-cosine time-window described here shall be used to shape the subchirp. The raised cosine window  $P_{RC}(t)$  is applied to every subchirp signal in the time domain, as illustrated in Figure 15-7.

$$P_{RC}(t) = \begin{cases} 1 & |t| \leq \frac{(1-\alpha)T_{sub}}{(1+\alpha)2} \\ \frac{1}{2} \left[ 1 + \cos \left( \frac{(1+\alpha)\pi}{\alpha T_{sub}} \left( |t| - \frac{(1-\alpha)T_{sub}}{(1+\alpha)2} \right) \right) \right] & \frac{(1-\alpha)T_{sub}}{(1+\alpha)2} < |t| \leq \frac{T_{sub}}{2} \\ 0 & |t| > \frac{T_{sub}}{2} \end{cases}$$

where  $\alpha = 0.25$ .

**Figure 15-7—Subchirp time-domain pulse shaping****15.3.5 Subchirp transmission order**

During each chirp symbol period, subchirp 1 ( $k = 1$ ) is transmitted first, and subchirp 4 ( $k = 4$ ) is transmitted last.

**15.3.6 Example of CSK signal generation**

An example for the modulation of one chirp symbol is provided in this subclause to illustrate each step from DEMUX to the output of the reference modulator as shown in Figure 15-4. The scenario parameters are as follows:

- The initial values of all four feedback memory stages of the differential encoder is set  $e^{(j\pi)/4}$ .
- The data bit rate is 1 Mb/s.

Input binary data:

0 1 0 1 1 0

Demux:

I-path: 0 0 1

Q-path: 1 1 0

Serial-to-parallel mapping:

I-path: {1 0 0}

Q-path: {1 1 0}

Bi-orthogonal mapping ( $r = 3/4$ ):

I-path: 1 -1 1 -1

Q-path: -1 -1 1 1

Parallel-to-serial and QPSK symbol mapping:

Mapper input:  $(1 - j)$ ,  $(-1 - j)$ ,  $(1 + j)$ ,  $(-1 + j)$

QPSK output phase:  $-\pi/2$ ,  $\pi$ ,  $0$ ,  $\pi/2$

D-QPSK coding:

Initial phase of four feedback memory for D-QPSK: all  $\pi/4$

D-QPSK coder output phase:  $-\pi/4$ ,  $-3\pi/4$ ,  $\pi/4$ ,  $3\pi/4$

D-QPSK-to-D-QCSK modulation output and subchirp sequence of D-QCSK output:

$[\exp(-j\pi/4) \times \text{subchirp}(k = 1), \exp(-j3\pi/4) \times \text{subchirp}(k = 2), \exp(j\pi/4) \times \text{subchirp}(k = 3), \exp(j3\pi/4) \times \text{subchirp}(k = 4)]$

## 15.4 CSS RF requirements

In addition to meeting regional regulatory requirements, CSS devices operating in the 2450 MHz band shall also meet the requirements in this subclause.

### 15.4.1 Transmit power spectral density (PSD) mask and signal tolerance

The transmitted spectral power density of a CSS signal  $s(t)$  shall be within the relative limits specified in the template shown in Figure 15-8. The average spectral power shall be made using 100 kHz resolution bandwidth and a 1 kHz video bandwidth. For the relative limit, the reference level shall be the highest average spectral power measured within  $\pm 11$  MHz of the carrier frequency. Specifically, the normalized frequency spectrum to the peak value in the signal bandwidth  $|f - f_c| \leq 7$  MHz shall be less than or equal to  $-30$  dB in the stop band  $11 \text{ MHz} \leq |f - f_c| \leq 22$  MHz and shall be less than or equal to  $-50$  dB in the stop band  $|f - f_c| > 22$  MHz. For testing the transmitted spectral power density, a  $2^{15} - 1$  pseudo-random binary sequence (PRBS) shall be used as input data.

As additional criteria for the compliance of a CSS signal, the mean square error shall be used. Let  $s(t)$  be the baseband CSS signal that is given in 15.3.3. Then the implemented signal,  $s_{impl}(t)$ , shall satisfy the following equation:

$$\text{mmse} = \min_{A, \tau_d, \phi} \left( \frac{\int_0^T |s^m(t) - A \times s_{impl}^m(t - \tau_d) e^{j\phi}|^2 dt}{\int_0^T |s^m(t)|^2 dt} \right) \leq 0.005$$

where the constants  $A$ ,  $\tau_d$ , and  $\phi$  are used to minimize the mean squared error. The constant  $T_{chirp}$  is the period of the CSS symbol. The  $c_{n,k}$  of  $s(t)$  is the constant data  $(1 + j)$  for the measurement for all  $n$  and  $k$ .

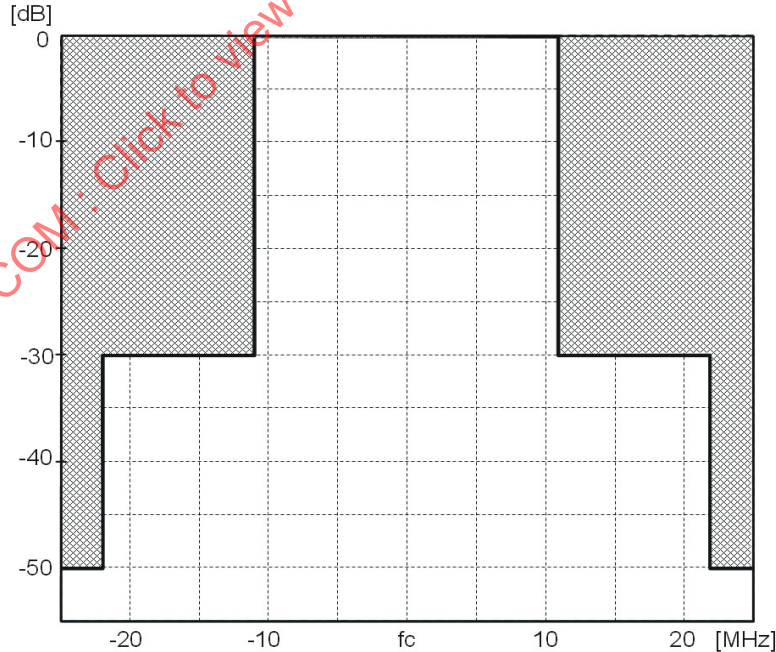


Figure 15-8—Transmit PSD mask

**15.4.2 Symbol rate**

The 2450 MHz PHY DQCSK symbol rate shall be 166.667 ksymbol/s ( $1/6$  Msymbol/s)  $\pm 40 \times 10^{-6}$ .

**15.4.3 Receiver sensitivity**

Under the conditions specified in 6.1.6, a compliant device shall be capable of achieving a receiver sensitivity of  $-85$  dBm or better for 1 Mb/s and  $-91$  dBm or better for 250 kb/s.

**15.4.4 Receiver interference rejection**

Table 15-9 gives minimum receiver interference rejection levels. A nonoverlapping adjacent channel is defined to have a center frequency offset of 25 MHz. A nonoverlapping alternate channel is defined to have a center frequency offset of 50 MHz. The adjacent channel rejection shall be measured as follows: The desired signal shall be a compliant 2450 MHz CSS signal 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 15.4.3. In the adjacent or the alternate channel, a CSS signal of the same or a different subchirp sequence as the victim device is input at the relative level specified in Table 15-9. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 6.1.6 under these conditions.

**Table 15-9—Minimum receiver interference rejection levels for 2450 MHz CSS PHY**

| Data rate           | Nonoverlapping adjacent channel rejection (25 MHz offset) (dB) | Nonoverlapping alternate channel rejection (50 MHz offset) (dB) |
|---------------------|----------------------------------------------------------------|-----------------------------------------------------------------|
| 1 Mb/s              | 34                                                             | 48                                                              |
| 250 kb/s (optional) | 38                                                             | 52                                                              |

**15.4.5 TX-to-RX turnaround time**

The TX-to-RX turnaround time shall be less than or equal to  $aTurnaroundTime$ , as defined in 11.2.

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

**15.4.6 RX-to-TX turnaround time**

The RX-to-TX turnaround time shall be less than or equal to  $aTurnaroundTime$ , as defined in 11.2.

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

**15.4.7 Transmit center frequency tolerance**

The CSS PHY transmit center frequency tolerance shall be  $\pm 40 \times 10^{-6}$  maximum.

**15.4.8 Transmit power**

The CSS PHY shall be capable of transmitting at a power level of at least  $-3$  dBm.

**15.4.9 Receiver maximum input level of desired signal**

The CSS PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

**15.4.10 Receiver ED**

The CSS PHY shall provide the receiver ED measurement as described in 10.2.5.

**15.4.11 LQI**

The CSS PHY shall provide the LQI measurement as described in 10.2.6.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018



## 16. HRP UWB PHY

### 16.1 General

The HRP UWB PHY waveform is based upon an impulse radio signaling scheme using band-limited pulses. The HRP UWB PHY supports three independent bands of operation:

- The sub-gigahertz band, which consists of a single channel and occupies the spectrum from 249.6 MHz to 749.6 MHz
- The low band, which consists of four channels and occupies the spectrum from 3.1 GHz to 4.8 GHz
- The high band, which consists of 11 channels and occupies the spectrum from 6.0 GHz to 10.6 GHz

Within each channel, there is support for at least two complex channels that have unique length 31 preamble codes. The combination of a channel and a preamble code is termed a *complex channel*. A compliant device shall implement support for at least one of the channels (0, 3, or 9) in Table 16-11. In addition, each device shall support the two unique length 31 preamble codes for the implemented channels as defined in Table 16-6. Support for the other channels listed in Table 16-11 is optional.

A combination of burst position modulation (BPM) and binary phase-shift keying (BPSK) is used to support both coherent and noncoherent receivers using a common signaling scheme. The combined BPM-BPSK is used to modulate the symbols, with each symbol being composed of an active burst of UWB pulses. The various data rates are supported through the use of variable-length bursts. Figure 16-1 shows the sequence of processing steps used to create and modulate an HRP UWB PPDU. The sequence of steps indicated here for the transmitter is used as a basis for explaining the creation of the HRP UWB PHY waveform specified in the PHY of this standard. Note that the receiver portion of Figure 16-1 is informative and meant only as a guide to the essential steps that any compliant HRP UWB receiver needs to implement in order to successfully decode the transmitted signal.

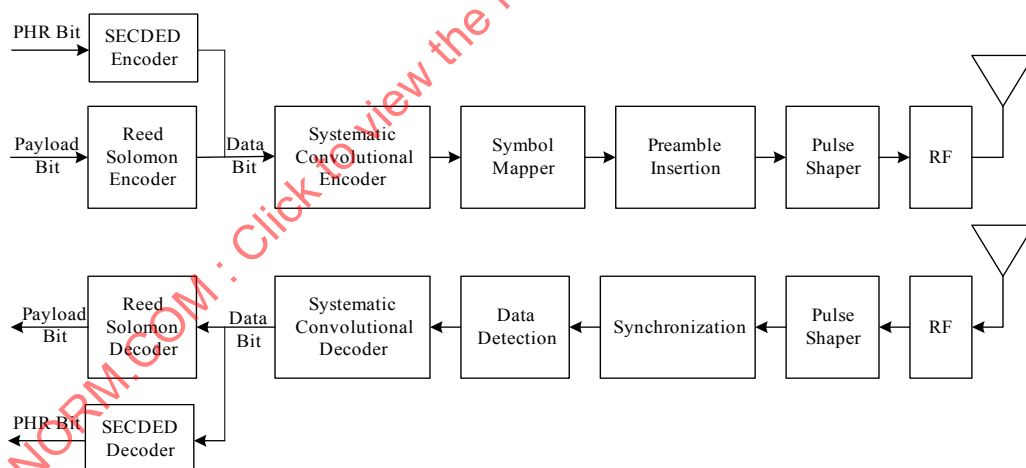


Figure 16-1—HRP UWB PHY signal flow

### 16.2 HRP UWB PPDU format

Figure 16-2 shows the format for the HRP UWB PPDU.

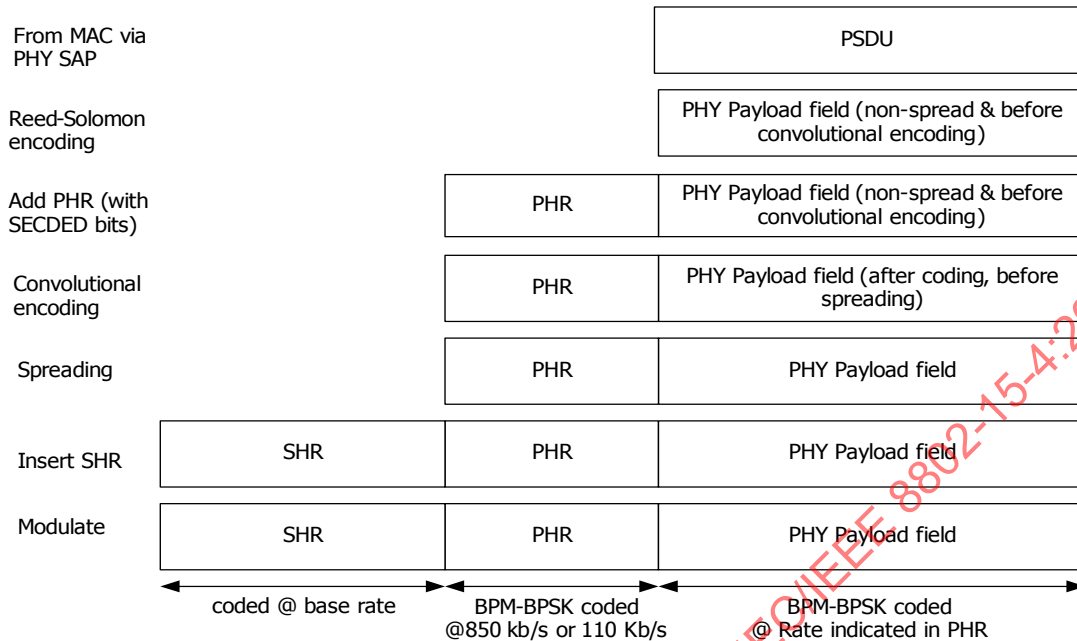


Figure 16-2—HRP UWB PPDU encoding process

Note that each HRP UWB-compliant device shall support the length 31 preamble codes specified in Table 16-6 and that two base rates corresponding to the two mandatory PRFs result for this code length. The mandatory SHR base rates are, therefore, 1.01 Msymbol/s and 0.25 Msymbol/s as indicated in Table 16-4.

The PHR is sent at 850 kb/s for all data rates greater than or equal to 850 kb/s and at 110 kb/s for the data rate of 110 kb/s. The PSDU is sent at the desired information data rate as defined in Table 16-3.

### 16.2.1 PPDU encoding process

The encoding process is composed of many steps, as illustrated in Figure 16-2. The details of these steps are fully described in later subclauses, as noted in the following list:

- Perform Reed-Solomon encoding on the PSDU as described in 16.3.3.1.
- Produce the PHR as described in 16.2.6, including the single error correct, double error detect (SEC-DED) field to and prepend to the PSDU.
- Perform further convolutional coding as described in 16.3.3.2. Note that in some instances at the 27 Mb/s data rate, the convolutional encoding of the PHY Payload field is effectively bypassed and two data bits are encoded per BPM-BPSK symbol.
- Modulate and spread PSDU according to the method described in 16.3.1 and 16.3.2. The PHR is modulated using BPM-BPSK at either 850 kb/s or 110 kb/s and the PHY Payload field is modulated at the rate specified in the PHR.
- Produce the SHR field from the SYNC field, as described in 16.2.5.1, and the SFD field, as described in 16.2.5.2.

Table 16-1 and Table 16-2 show how the SHR field,  $H_0$ - $H_{18}$ , PHY Payload field,  $D_0$ - $D_{N-1}$ , and Tail field,  $T_0$ - $T_1$ , are mapped onto the symbols. In these tables, the polarity bit column operation is an XOR. The tables also show when the transition from the header bit rate to the data bit rate takes place. Note that the delay line of the convolutional code is initialized to zero. For this reason, the position bit of Symbol 0 shall always be zero. This means that Symbol 0 is always transmitted in the first half of the first header symbol.

**Table 16-1—Mapping of SHR field bits, PHY Payload field bits, and Tail field bits onto symbols with Viterbi rate 0.5**

| Symbol # | Input data | Position bit | Polarity bit             |                                                |
|----------|------------|--------------|--------------------------|------------------------------------------------|
| 0        | $H_0$      | 0            | $H_0$                    | 21 symbols of header at 850 kb/s or 110 kb/s   |
| 1        | $H_1$      | $H_0$        | $H_1$                    |                                                |
| 2        | $H_2$      | $H_1$        | $H_0 \oplus H_2$         |                                                |
| 3        | $H_3$      | $H_2$        | $H_1 \oplus H_3$         |                                                |
| ...      | ...        | ...          | ...                      |                                                |
| 16       | $H_{16}$   | $H_{15}$     | $H_{14} \oplus H_{16}$   |                                                |
| 17       | $H_{17}$   | $H_{16}$     | $H_{15} \oplus H_{17}$   |                                                |
| 18       | $H_{18}$   | $H_{17}$     | $H_{16} \oplus H_{18}$   |                                                |
| 19       | $D_0$      | $H_{18}$     | $H_{17} \oplus D_0$      |                                                |
| 20       | $D_1$      | $D_0$        | $H_{18} \oplus D_1$      |                                                |
| 21       | $D_2$      | $D_1$        | $D_0 \oplus D_2$         | N symbols of data at data rate, e.g., 6.8 Mb/s |
| ...      | ...        | ...          | ...                      |                                                |
| N+17     | $D_{N-2}$  | $D_{N-3}$    | $D_{N-4} \oplus D_{N-2}$ |                                                |
| N+18     | $D_{N-1}$  | $D_{N-2}$    | $D_{N-3} \oplus D_{N-1}$ |                                                |
| N+19     | $T_0$      | $D_{N-1}$    | $D_{N-2} \oplus T_0$     |                                                |
| N+20     | $T_1$      | $T_0$        | $D_{N-1} \oplus T_1$     |                                                |

**Table 16-2—Mapping of SHR field bits, PHY Payload field bits and Tail field bits onto symbols with Viterbi rate 1**

| Symbol # | Input data | Position bit | Polarity bit           |                                  |
|----------|------------|--------------|------------------------|----------------------------------|
| 0        | $H_0$      | 0            | $H_0$                  | 21 symbols of header at 850 kb/s |
| 1        | $H_1$      | $H_0$        | $H_1$                  |                                  |
| 2        | $H_2$      | $H_1$        | $H_0 \oplus H_2$       |                                  |
| 3        | $H_3$      | $H_2$        | $H_1 \oplus H_3$       |                                  |
| ...      | ...        | ...          | ...                    |                                  |
| 16       | $H_{16}$   | $H_{15}$     | $H_{14} \oplus H_{16}$ |                                  |
| 17       | $H_{17}$   | $H_{16}$     | $H_{15} \oplus H_{17}$ |                                  |
| 18       | $H_{18}$   | $H_{17}$     | $H_{16} \oplus H_{18}$ |                                  |
| 19       | $T_0$      | $H_{18}$     | $H_{17} \oplus T_0$    |                                  |
| 20       | $T_1$      | $T_0$        | $H_{18} \oplus T_1$    |                                  |

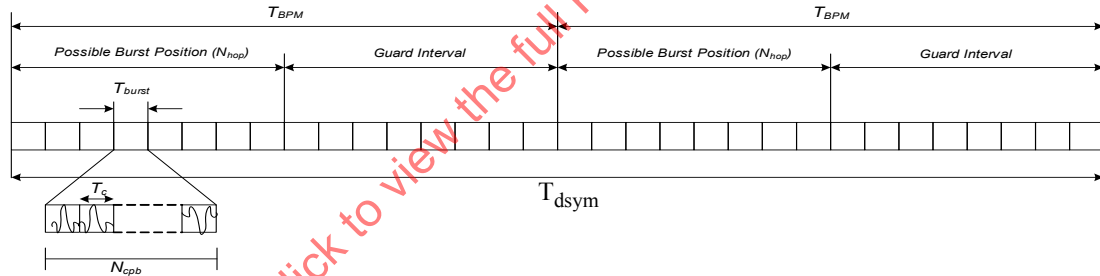
**Table 16-2—Mapping of SHR field bits, PHY Payload field bits and Tail field bits onto symbols with Viterbi rate 1 (continued)**

| Symbol # | Input data         | Position bit | Polarity bit |                                                    |
|----------|--------------------|--------------|--------------|----------------------------------------------------|
| 21       | $D_0, D_1$         | $D_0$        | $D_1$        | 1/2 N symbols of data at data rate, e.g., 6.8 Mb/s |
| ...      | $D_2, D_3$         | $D_2$        | $D_3$        |                                                    |
| ...      | ...                | ...          | ...          |                                                    |
| 1/2 N+19 | $D_{N-6}, D_{N-5}$ | $D_{N-6}$    | $D_{N-5}$    |                                                    |
| 1/2 N+20 | $D_{N-4}, D_{N-3}$ | $D_{N-4}$    | $D_{N-3}$    |                                                    |
| 1/2 N+21 | $D_{N-2}, D_{N-1}$ | $D_{N-2}$    | $D_{N-1}$    |                                                    |

### 16.2.2 Symbol structure

In the BPM-BPSK modulation scheme, each symbol is capable of carrying two bits of information: one bit is used to determine the position of a burst of pulses, while an additional bit is used to modulate the phase (polarity) of this same burst.

The structure and timing of a symbol is illustrated in Figure 16-3. Each symbol shall consist of an integer number of possible chip positions,  $N_c$ , each with duration  $T_c$ . The overall symbol period denoted by  $T_{dsym}$  is given by  $T_{dsym} = N_c T_c$ . Furthermore, each symbol is divided into two BPM intervals each with duration  $T_{BPM} = T_{dsym}/2$ , which enables binary position modulation.

**Figure 16-3—HRP UWB PHY symbol structure**

A burst is formed by grouping  $N_{cpb}$  consecutive chips and has duration  $T_{burst} = N_{cpb} T_c$ . The location of the burst in either the first half or the second half of the symbol indicates one bit of information. Additionally, the phase of the burst (either  $-1$  or  $+1$ ) is used to indicate a second bit of information.

In each symbol interval, a single burst event shall be transmitted. The fact that burst duration is typically much shorter than the BPM duration, i.e.,  $T_{burst} \ll T_{BPM}$ , provides for some multi-user access interference rejection in the form of time hopping. The total number of burst durations per symbol,  $N_{burst}$ , is given by  $N_{burst} = T_{dsym}/T_{burst}$ . In order to limit the amount of inter-symbol interference caused by multipath, only the first half of each  $T_{BPM}$  period shall contain a burst. Therefore, only the first  $N_{hop} = N_{burst}/4$  possible burst positions are candidate hopping burst positions within each BPM interval. Each burst position can be varied on a symbol-to-symbol basis according to a time hopping code as described in 16.3.

### 16.2.3 PSDU timing parameters

The PSDU rate-dependent parameters and timing-related parameters are summarized in Table 16-3. Within each HRP UWB channel {0:15}, the peak PRF shall be 499.2 MHz. This rate corresponds to the highest frequency at which a compliant transmitter shall emit pulses. Additionally, the mean PRF is defined as the total number of pulses emitted during a symbol period divided by the length of the symbol period.

There are two possible preamble code lengths (31 or 127) and three possible mean PRFs (15.6 MHz, 3.90 MHz, and 62.4 MHz). A compliant device shall implement support for the preamble code length of 31 and shall also support both the 15.6 MHz and 3.90 MHz mean PRFs for the PSDU as depicted in Table 16-3. The use of the length 127 code is optional; when implemented, the mean PRF of the PSDU shall be 62.4 MHz.

Channels {4, 7, 11, 15} are all optional channels and are differentiated from other channels by the larger bandwidth (>500 MHz) of the transmitted signals. These channels overlap the existing lower bandwidth channels. The larger bandwidth enables devices operating in these channels to transmit at a higher power (for fixed PSD constraints), and thus they may achieve a longer communication range. The larger bandwidth pulses offer enhanced multipath resistance. Additionally, larger bandwidth leads to more accurate range estimates. The admissible data rates, preamble code lengths, PRFs, and modulation timing parameters are listed in Table 16-3.

Each channel allows for several data rates that are obtained by modifying the number of chips within a burst, while the total number of possible burst positions remains constant. Therefore, the symbol period,  $T_{\text{sym}}$ , changes to obtain the stated symbol rate and bit rates.

Each row in Table 16-3 completely describes all timing parameters shown in Figure 16-3 for each permitted combination of channel number, preamble code length, and PRF.

The channel number parameter column identifies the HRP UWB PHY channel numbers where the remaining PSDU timing parameters in the current row are valid. Association between channel number and center frequency is given in Table 16-11.

The peak PRF states the highest frequency in megahertz at which a compliant transmitter shall emit pulses. The peak PRF is also used to derive the chip duration  $T_c$  by the formula  $T_c = 1/(\text{peakPRF})$ . The value of  $T_c$  is approximately 2 ns.

The bandwidth denotes the 3 dB bandwidth of the HRP UWB pulses. Note that the bandwidth is not necessarily the inverse of the chip duration  $T_c$ . Pulse shape and bandwidth are further defined in 16.4.5.

The preamble code length parameter denotes the length of the preamble code length to be used during the SHR portion of a PPDU. The code length together with the channel number defines a complex channel. Individual codes to be used on each channel are given in Table 16-6 (length 31) and Table 16-7 (length 127).

This Viterbi rate parameter determines the rate of the convolutional code applied to the PHY Payload bits. A value of 1 indicates that no convolutional coding is applied, while a value of 0.5 indicates that a rate 1/2 code as described in 16.3.3.2 is applied.

The RS rate parameters indicates the (63,55) Reed-Solomon code rate, which is approximately 0.87. The Reed-Solomon code is applied to the entire PSDU. Reed-Solomon encoding is further described in 16.3.3.1.

The overall FEC rate is determined by the product of the Viterbi rate and the Reed-Solomon rate and has either a value of 0.44 or 0.87.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

Table 16-3—HRP UWB PHY rate-dependent and timing-related parameters

| Channel Number              | Peak PRF MHz | Bandwidth MHz | Preamble Code Length | Modulation & Coding |         |                  | Data Symbol Structure                   |                       |                            |                   |                                 |                                | Data              |               |                |
|-----------------------------|--------------|---------------|----------------------|---------------------|---------|------------------|-----------------------------------------|-----------------------|----------------------------|-------------------|---------------------------------|--------------------------------|-------------------|---------------|----------------|
|                             |              |               |                      | Viterbi Rate        | RS Rate | Overall FEC Rate | #Burst Positions per Symbol $N_{burst}$ | #Hop Bursts $N_{hop}$ | #Chips Per Burst $N_{cpb}$ | #Chips Per Symbol | Burst Duration $T_{burst}$ (ns) | Symbol Duration $T_{sym}$ (ns) | Symbol Rate (MHz) | Bit Rate Mb/s | Mean PRF (MHz) |
| {0, 3, 5, 6, 8, 10, 12, 14} | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 128                        | 4096              | 256.41                          | 8205.13                        | 0.12              | 0.11          | 15.60          |
|                             | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 16                         | 512               | 32.05                           | 1025.64                        | 0.98              | 0.85          | 15.60          |
|                             | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 2                          | 64                | 4.01                            | 128.21                         | 7.80              | 6.81          | 15.60          |
|                             | 499.2        | 499.2         | 31                   | 1                   | 0.87    | 0.87             | 32                                      | 8                     | 1                          | 32                | 2.00                            | 64.10                          | 15.60             | 27.24         | 15.60          |
| {0, 3, 5, 6, 8, 10, 12, 14} | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 128                                     | 32                    | 32                         | 4096              | 64.10                           | 8205.13                        | 0.12              | 0.11          | 3.90           |
|                             | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 128                                     | 32                    | 4                          | 512               | 8.01                            | 1025.64                        | 0.98              | 0.85          | 3.90           |
|                             | 499.2        | 499.2         | 31                   | 0.5                 | 0.87    | 0.44             | 128                                     | 32                    | 2                          | 256               | 4.01                            | 512.82                         | 1.95              | 1.70          | 3.90           |
|                             | 499.2        | 499.2         | 31                   | 1                   | 0.87    | 0.87             | 128                                     | 32                    | 1                          | 128               | 2.00                            | 256.41                         | 3.90              | 6.81          | 3.90           |
| {0, 3, 5, 6, 8, 10, 12, 14} | 499.2        | 499.2         | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 512                        | 4096              | 1025.64                         | 8205.13                        | 0.12              | 0.11          | 62.40          |
|                             | 499.2        | 499.2         | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 64                         | 512               | 128.21                          | 1025.64                        | 0.98              | 0.85          | 62.40          |
|                             | 499.2        | 499.2         | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 8                          | 64                | 16.03                           | 128.21                         | 7.80              | 6.81          | 62.40          |
|                             | 499.2        | 499.2         | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 2                          | 16                | 4.01                            | 32.05                          | 31.20             | 27.24         | 62.40          |
| {4, 11}                     | 499.2        | 1331.2        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 128                        | 4096              | 256.41                          | 8205.13                        | 0.12              | 0.11          | 15.60          |
|                             | 499.2        | 1331.2        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 16                         | 512               | 32.05                           | 1025.64                        | 0.98              | 0.85          | 15.60          |
|                             | 499.2        | 1331.2        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 2                          | 64                | 4.01                            | 128.21                         | 7.80              | 6.81          | 15.60          |
|                             | 499.2        | 1331.2        | 31                   | 1                   | 0.87    | 0.87             | 32                                      | 8                     | 1                          | 32                | 2.00                            | 64.10                          | 15.60             | 27.24         | 15.60          |
| {4, 11}                     | 499.2        | 1331.2        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 512                        | 4096              | 1025.64                         | 8205.13                        | 0.12              | 0.11          | 62.40          |
|                             | 499.2        | 1331.2        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 64                         | 512               | 128.21                          | 1025.64                        | 0.98              | 0.85          | 62.40          |
|                             | 499.2        | 1331.2        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 8                          | 64                | 16.03                           | 128.21                         | 7.80              | 6.81          | 62.40          |
|                             | 499.2        | 1331.2        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 2                          | 16                | 4.01                            | 32.05                          | 31.20             | 27.24         | 62.40          |
| 7                           | 499.2        | 1081.6        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 128                        | 4096              | 256.41                          | 8205.13                        | 0.12              | 0.11          | 15.60          |
|                             | 499.2        | 1081.6        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 16                         | 512               | 32.05                           | 1025.64                        | 0.98              | 0.85          | 15.60          |
|                             | 499.2        | 1081.6        | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 2                          | 64                | 4.01                            | 128.21                         | 7.80              | 6.81          | 15.60          |
|                             | 499.2        | 1081.6        | 31                   | 1                   | 0.87    | 0.87             | 32                                      | 8                     | 1                          | 32                | 2.00                            | 64.10                          | 15.60             | 27.24         | 15.60          |
| 7                           | 499.2        | 1081.6        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 512                        | 4096              | 1025.64                         | 8205.13                        | 0.12              | 0.11          | 62.40          |
|                             | 499.2        | 1081.6        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 64                         | 512               | 128.21                          | 1025.64                        | 0.98              | 0.85          | 62.40          |
|                             | 499.2        | 1081.6        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 8                          | 64                | 16.03                           | 128.21                         | 7.80              | 6.81          | 62.40          |
|                             | 499.2        | 1081.6        | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 2                          | 16                | 4.01                            | 32.05                          | 31.20             | 27.24         | 62.40          |
| 15                          | 499.2        | 1354.97       | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 128                        | 4096              | 256.41                          | 8205.13                        | 0.12              | 0.11          | 15.60          |
|                             | 499.2        | 1354.97       | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 16                         | 512               | 32.05                           | 1025.64                        | 0.98              | 0.85          | 15.60          |
|                             | 499.2        | 1354.97       | 31                   | 0.5                 | 0.87    | 0.44             | 32                                      | 8                     | 2                          | 64                | 4.01                            | 128.21                         | 7.80              | 6.81          | 15.60          |
|                             | 499.2        | 1354.97       | 31                   | 1                   | 0.87    | 0.87             | 32                                      | 8                     | 1                          | 32                | 2.00                            | 64.10                          | 15.60             | 27.24         | 15.60          |
| 15                          | 499.2        | 1354.97       | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 512                        | 4096              | 1025.64                         | 8205.13                        | 0.12              | 0.11          | 62.40          |
|                             | 499.2        | 1354.97       | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 64                         | 512               | 128.21                          | 1025.64                        | 0.98              | 0.85          | 62.40          |
|                             | 499.2        | 1354.97       | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 8                          | 64                | 16.03                           | 128.21                         | 7.80              | 6.81          | 62.40          |
|                             | 499.2        | 1354.97       | 127                  | 0.5                 | 0.87    | 0.44             | 8                                       | 2                     | 2                          | 16                | 4.01                            | 32.05                          | 31.20             | 27.24         | 62.40          |

The burst positions per symbol parameter is the total number of possible burst positions in a data symbol period.  $N_{burst}$  has been chosen so that for each mean PRF a data symbol consists of a fixed number of burst durations.

The hop bursts parameter is the number of burst positions that may contain an active burst, that is, a burst containing HRP UWB pulses. The value is computed as  $N_{hop} = N_{burst}/4$ .

The chips per burst parameter is the number of chip  $T_c$  durations within each burst period  $T_{burst}$ . Each burst consists of a multiple number of consecutive chips, as illustrated in Figure 16-3. Depending on the data rate to be used in the transmission of the PSDU, the number of chips in a burst varies, e.g., for low data rates, the burst consists of more chip periods than for high data rates. Particular, values of  $N_{cpb}$  have been selected so that the following is a valid data rate:  $(2 \times \text{Overall FEC Rate})/(N_{cpb} \times N_{burst} \times T_c)$ .

The burst duration parameter is simply the duration of a burst and is computed as  $T_{burst} = N_{cpb} \times T_c$ .

The symbol period parameter is the duration of a modulated and coded PSDU symbol on the air and is computed as follows:  $T_{dsym} = N_{burst} \times T_{burst}$ .

The symbol rate parameter is the inverse of the PSDU symbol period  $1/T_{dsym}$ .

The bit rate parameter is the user information rate considering FEC and is computed as follows:

$$\text{Bit Rate} = 2 \times (\text{Overall FEC Rate})/T_{dsym}$$

The mean PRF parameter is the average PRF during the PSDU portion of a PHY frame and is computed as follows:

$$\text{Mean PRF} = N_{cpb}/T_{dsym}$$

#### 16.2.4 Preamble timing parameters

Due to the variability in the preamble code length and the PRF, there are several admissible values for the timing parameters of a preamble symbol. These values are summarized in Table 16-4. In this subclause, a preamble symbol is defined as the waveform consisting of one whole repetition of the modulated preamble code (either length 31 or 127). Details on the construction of the preamble symbol for various code lengths and PRFs are given in 16.2.5. For each target PRF, the preamble is constructed from a preamble code,  $C_i$ , by inserting a number of chip durations between code symbols. The number of chip durations to insert is denoted by  $\delta_L$ , values for each code length and PRF are given in Table 16-4, and the chip insertion is detailed in 16.2.5.1.

**Table 16-4—Preamble parameters**

| Channel Number          | $C_i$ Code Length | Peak PRF (MHz) | Mean PRF (MHz) | Delta Length $\delta_L$ | #Chips Per Symbol | Symbol Duration $T_{psym}$ (ns) | Base Rate Msymbol/s |
|-------------------------|-------------------|----------------|----------------|-------------------------|-------------------|---------------------------------|---------------------|
| {0:15}                  | 31                | 31.20          | 16.10          | 16                      | 496               | 993.59                          | 1.01                |
| {0:3, 5:6, 8:10, 12:14} | 31                | 7.80           | 4.03           | 64                      | 1984              | 3974.36                         | 0.25                |
| {0:15}                  | 127               | 124.80         | 62.89          | 4                       | 508               | 1017.63                         | 0.98                |

Table 16-4 presents the timing parameters during the SHR field portion of an HRP UWB PPDU, while Table 16-3 presents the timing parameters for the PSDU portion of the frame. First, note that the preamble is sent at a slightly higher mean PRF than the data, as defined in Table 16-3. This is due to the fact that length 31 or 127 ternary codes are being used within the SHR, and the number of chips within the SHR is no longer a power of 2. For example, for the two mandatory PRFs in channels {0:3, 5:6, 8:10, 12:14}, the peak PRFs during the preamble are 31.2 MHz and 7.8 MHz, respectively, and the corresponding mean PRFs during the preamble are 16.10 MHz and 4.03 MHz, respectively. The corresponding mean PRFs during the PSDU are 15.60 MHz and 3.90 MHz, respectively. The remaining peak and mean PRF values for other optional HRP UWB channels and the optional length 127 code are listed in Table 16-4.

The base symbol rate is defined as the rate at which the preamble symbols are sent. The base rates corresponding to the two mandatory mean PRFs of 16.10 MHz and 4.03 MHz are 1 Msymbol/s and 0.25 Msymbol/s, respectively, and are listed in the column with the heading “Base Rate” in Table 16-4. These symbol rates correspond to a preamble symbol period of 993.59 ns and 3974.36 ns for the two mandatory PRFs.

Finally, for each HRP UWB PPDU, there are four possible durations of the SHR. This is due to the four possible lengths of SYNC field in the SHR, as described in 16.2.5. The SYNC field consists of repetitions of the preamble symbol. The number of preamble symbol repetitions are 16, 64, 1024, and 4096. These different SYNC field lengths yield different time durations of the PPDU. The relationship between SYNC field length and frame duration is shown in Table 16-5. For each channel, the number of chips in an individual preamble symbol is shown in the row titled “ $N_c$ .”  $N_c$  is a function of the PRF used within the channel and, therefore, has either two or three values. For each value of  $N_c$ , the admissible preamble symbol periods are defined, and the duration of the SYNC portion of the SHR for each length (16, 64, 1024, or 4096) is denoted as  $T_{sync}$ . The values of the frame duration parameters are shown in Table 16-5 for each of the channels.

**Table 16-5—HRP UWB PHY frame-dependent parameters**

| Parameter           | Description                                   | Value       |             |               |
|---------------------|-----------------------------------------------|-------------|-------------|---------------|
| Channel             | HRP UWB PHY channel number                    | {0:15}      |             |               |
| PRF <sub>mean</sub> | Mean PRF (MHz)                                | 16.10       | 62.89       | 4.03          |
| $N_c$               | Number of chips per preamble symbol           | 496         | 508         | 1984          |
| $T_{psym}$          | Preamble symbol period (ns)                   | 993.6       | 1017.6      | 3974.4        |
| $N_{sync}$          | Number of symbols in the packet sync sequence | Short       | 16          |               |
|                     |                                               | Default     | 64          |               |
|                     |                                               | Medium      | 1024        |               |
|                     |                                               | Long        | 4096        |               |
| $T_{sync}$          | Duration of the packet sync sequence (μs)     | Short       | 15.9        | 16.3          |
|                     |                                               | Default     | 63.6        | 65.1          |
|                     |                                               | Medium      | 1017.4      | 1042.1        |
|                     |                                               | Long        | 4069.7      | 4168.2        |
| $N_{sfd}$           | Number of symbols in the SFD                  | 8 or 64     |             |               |
| $T_{sfd}$           | Duration of the SFD (μs)                      | 7.9 or 63.6 | 8.1 or 65.1 | 31.8 or 254.4 |



**Table 16-5—HRP UWB PHY frame-dependent parameters (continued)**

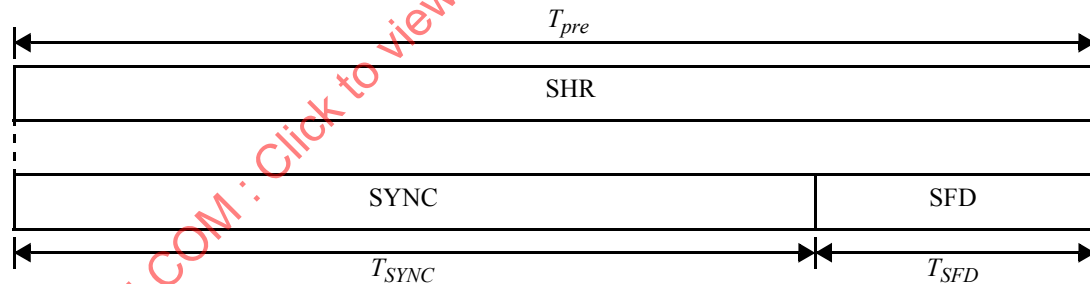
| Parameter       | Description                                                     | Value                           |                  |                  |                  |
|-----------------|-----------------------------------------------------------------|---------------------------------|------------------|------------------|------------------|
| $N_{pre}$       | Number of symbols in the SHR                                    | Short                           | 24 or 80         |                  |                  |
|                 |                                                                 | Default                         | 72 or 128        |                  |                  |
|                 |                                                                 | Medium                          | 1032 or 1088     |                  |                  |
|                 |                                                                 | Long                            | 4104 or 4160     |                  |                  |
| $T_{pre}$       | Duration of the SHR preamble ( $\mu$ s)                         | Short                           | 23.8 or 79.5     | 24.4 or 270.6    | 95.4 or 128.7    |
|                 |                                                                 | Default                         | 71.5 or 127.2    | 73.3 or 319.5    | 286.2 or 319.5   |
|                 |                                                                 | Medium                          | 1025.4 or 081.0  | 1050.2 or 1296.4 | 4101.5 or 4134.9 |
|                 |                                                                 | Long                            | 4077.7 or 4133.3 | 4176.3 or 4422.6 | N/A <sup>a</sup> |
| $N_{CCA\_PHR}$  | Number of multiplexed preamble symbols in PHR                   | 4 or 32                         |                  |                  |                  |
| $N_{CCA\_data}$ | Number of multiplexed preamble symbols in the PHY Payload field | $T_{pre}/(4 \times T_{dsym/M})$ |                  |                  |                  |

<sup>a</sup>The use of the long SYNC sequence is not allowed when operating at a mean PRF of 4.03 MHz.

### 16.2.5 SHR field

Four mandatory preambles are defined: a default preamble, a short preamble, a medium preamble, and a long preamble. The preamble to be used in the transmission of the current frame is determined by the value of the `HrpUwbPreambleSymbolsRepetitions` parameter in the `MCPS-DATA.request` primitive.

The SHR field shall be formatted as illustrated in Figure 16-4.

**Figure 16-4—SHR field structure**

#### 16.2.5.1 SYNC field

Each PAN operating on one of the HRP UWB PHY channels is also identified by a preamble code. The preamble code is used to construct symbols that constitute the SYNC field as shown in Figure 16-4. The HRP UWB PHY supports two lengths of preamble code: a length 31 code and an optional length 127 code. Each preamble code is a sequence of code symbols drawn from a ternary alphabet  $\{-1, 0, 1\}$  and selected for use in the HRP UWB PHY because of their periodic autocorrelation properties. The length 31 code sequences are shown in Table 16-6, while the length 127 code sequences are shown in Table 16-7. The codes that may be used in each of the HRP UWB PHY channels is restricted, and the particular code

assignments are made in Table 16-6 and Table 16-7. Specifically, the last column in each table indicates the set of HRP UWB channel numbers that permit use of the code. This restriction of codes is to ensure that codes with the lowest cross-correlation are used in the same HRP UWB PHY channel. Additionally, 8 of the length 127 codes are reserved for use with the private ranging protocol only and are not used during normal WPAN operation. This restriction is indicated in the third column of Table 16-7 as well.

**Table 16-6—Length 31 ternary codes**

| Code index | Code sequence                   | Channel number <sup>a</sup> |
|------------|---------------------------------|-----------------------------|
| 1          | -0000+0-0+++0+-000+---+00+0-00  | 0, 1, 8, 12                 |
| 2          | 0+0+-0+0+000-++0+---00+00++000  | 0, 1, 8, 12                 |
| 3          | -+0++000-+-++00++0+00-0000-0+0- | 2, 5, 9, 13                 |
| 4          | 0000+-00-00-++++0+-+000+0-0++0- | 2, 5, 9, 13                 |
| 5          | -0+-00+++--000-+0+++0-0+0000-00 | 3, 6, 10, 14                |
| 6          | ++00+00---+0++-000+0+0-+0+0000  | 3, 6, 10, 14                |
| 7          | +0000+-0+0+00+000+0++---0-+00+  | 4, 7, 11, 15                |
| 8          | 0+00-0-0++0000--00-+0+-++0+00   | 4, 7, 11, 15                |

<sup>a</sup> Note that codes indexed 1 through 6 may also be used for HRP UWB channels 4, 7, 11, and 15 (i.e., channels whose bandwidth is wider than 500 MHz) if interchannel communication is desired.

**Table 16-7—Optional length 127 ternary codes**

| Code index | Code sequence                                                                                                                  | Channel number <sup>a</sup> |
|------------|--------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| 9          | +00+000-0--00--0+0+00-++0+0000++-000+00-00--0-+0+0--0-+++0++000+0+00-0++-0+++00-+00+0+0-0++-++000000+00000-+0000-0-000--+      | 0-3, 5, 6, 8-10, 12-14      |
| 10         | ++00+0-+00+00+000000-000-00--000-0+-+0-0+-0-+00000+-00++0-0+00--+00++-+0+-0+0000-0-0-0+-+0+00+0+000-+0+++000----++0000+++0--   | 0-3, 5, 6, 8-10, 12-14      |
| 11         | -+-0000+00--00000-0+0+0+-0+00+00+0-00-+++00+000-+0+0-0000+++++-+0+-0+-0+-0--0-000+0-+00+0+----000-000000-+00+-0+000++-00++-0-0 | 0-3, 5, 6, 8-10, 12-14      |
| 12         | -+0+000000-0+0+0-+-+00-+0+0+0+0+000-00-00-+00+-+000-+-0-++0-0+000-0-00-0+00+0+00+-00+000+-0000-0-000000-+000000+-              | 0-3, 5, 6, 8-10, 12-14      |
| 13         | +000--0000--++0-++++0-0++0+0-00-+0+00++-0+0+-+0-00+00-0--000-+-00+0000-0++-00000+-0-000000-00-+-+000-0+0+0+++00--00+0+000      | 0-15; DPS only              |
| 14         | +000++0-0+0-00+-0-+0-00+0+0000+0+-0000++00+0+++++0-0+-0-+0+-+000---0+000+0+0-+-000000+-+0--00++000-00+00++-00--++-00-00000     | 0-15; DPS only              |
| 15         | 0+-00+0-000-++0000---++000+0+-0-+00-+000--0-00--0-+++-+0-++00+-+0+00000+0-0+++00+00+000-0000+00--0+0+0+0-00-0-+0+0++00000      | 0-15; DPS only              |
| 16         | ++0000+000+00+-0+-+0-000--00+-0+00++000++0+0+0-0+-0-0+00+00+0++----+00++-+0+-0--000000-0-0000-0-00+00000+-+000-0-+0+0          | 0-15; DPS only              |

**Table 16-7—Optional length 127 ternary codes (continued)**

| Code index | Code sequence                                                                                                                | Channel number <sup>a</sup> |
|------------|------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| 17         | +--000-0-0000+-00000+000000+--++0-0+0+00+-00+++0-++0-00+0-+000++0+++0-0-0+0+-0-00-00+000-++0000+0++-+00+0+0+-00--0-000+00+   | 4, 7, 11, 15                |
| 18         | --0+++0000++++--000+++0+-000+0+00+0+-++0-0-0-0000+0-+0+-++00+-00+0-0++00-+00000+-0-+0-0+-+0-000-00-000-000000+00+00+-0+00++  | 4, 7, 11, 15                |
| 19         | -0-++00-++000+0-+00+-000000-000----+0+0+-0+000-0-++0-+0--0+0-+++++0000-0+0+-000+00+++0-0-0+00+00+0-+0+0+0-00000-00+0000+-0   | 4, 7, 11, 15                |
| 20         | --+00000+0--0000-0000+-0-000-+000+00-++00+0+00++0-00-0++++0-0++-0+-000++-+00+-00-00-000+0+0+0++0-00+++-+--0+-0+0-000000++0+- | 4, 7, 11, 15                |
| 21         | +0+00-00-++0+0+0-000+-++-+00-000000-0-+00000-++0-0000+00-+-000--0-00+00-0+-+0+0+-+00++0+-00-0+0+0-0++++0++-+0000--000+000    | 0–15; DPS only              |
| 22         | 0-00-++-00-++00+00-000++00--0-+-+000000-+0+0+000+0---000-++0+-0-+0-0+-+++++0+00++0000-+0+0000+0+00-0+-0-+00-0+0-+000+0000    | 0–15; DPS only              |
| 23         | 000++0+0+-0-00-0+0+0+0+-00+0000-000+00+00-++0-0+00000+0+-+00++-0+-+++-0-00-0-000+-00+-0+0+000++-0000++-000-0+00-+000         | 0–15; DPS only              |
| 24         | +0+-0-000++-+00000+00--0+-0000-0-000000+-0-+0+-++00+-+--0+00+00+0-0+-0-0+0+00+++000++00+0+-00-000-0++-0-++00+000+0000++0     | 0–15; DPS only              |

<sup>a</sup> Note that codes indexed 9 through 13 may also be used for HRP UWB channels 4, 7, 11, and 15 (i.e., channels whose bandwidth is wider than 500 MHz) if interchannel communication is desired.

Note that the assignment of preamble codes to channels has been done to enable interchannel communication. In other words, it is possible that a device operating on a wideband channel {4,7,11,15} may communicate with a device on a channel with which it overlaps.

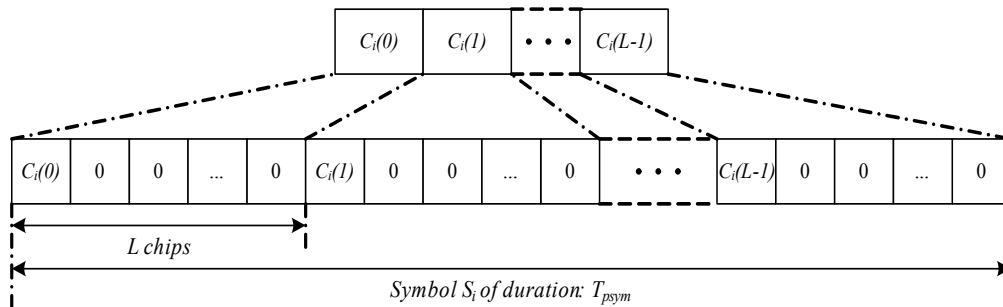
For a WPAN using the ternary code indexed by  $i$ , the SYNC field shall consist of  $N_{sync}$  repetitions of the symbol  $S_i$ , where  $S_i$  is the code  $C_i$  spread by the delta function  $\delta_L$  of length  $L$  as shown in Table 16-4. The spreading operation, where code  $C_i$  is extended to the preamble symbol period indicated in Table 16-4, is described mathematically as follows:

$$S_i = C_i \otimes \delta_L(n)$$

$$\delta_L(n) = \begin{cases} 1 & n = 0 \\ 0 & n = 1, 2, \dots, L-1 \end{cases}$$

where the operator  $\otimes$  indicates a Kronecker product. After the Kronecker operation, a preamble symbol is formed as depicted in Figure 16-5, where  $L-1$  zeros have been inserted between each ternary element of  $C_i$ .

The spreading factor  $L$ , number of chips per symbol, preamble symbol period  $T_{psym}$ , and base symbol rate for different channels are given in Table 16-4.

Figure 16-5—Construction of symbol  $S_i$  from code  $C_i$ 

### 16.2.5.2 SFD field

The short SFD is used for the default and medium data rates while the long SFD is used for the optional low data rate of 110 kb/s, as shown in Figure 16-4. The short SFD shall be  $[0 +1 0 -1 +1 0 0 -1]$  spread by the preamble symbol  $S_i$ , where the leftmost bit shall be transmitted first in time. The long SFD shall be obtained by spreading the sequence  $[0 +1 0 -1 +1 0 0 -1 0 +1 0 -1 +1 0 0 -1 -1 0 0 +1 0 -1 0 +1 0 +1 0 0 0 -1 0 -1 0 -1 0 0 +1 0 -1 -1 0 -1 +1 0 0 0 0 +1 +1 0 0 -1 -1 -1 +1 -1 +1 +1 0 0 0 0 +1 +1]$  by the preamble symbol  $S_i$ . The structure of the SHR and the two possible SFDs are shown in Figure 16-4.

### 16.2.6 PHR field

A PHR field shall be formatted as illustrated in Figure 16-6.

| Bits: 0–1 | 2–8          | 9       | 10       | 11–12             | 13–18  |
|-----------|--------------|---------|----------|-------------------|--------|
| Data Rate | Frame Length | Ranging | Reserved | Preamble Duration | SECDED |

Figure 16-6—PHR field format

The Data Rate field indicates the data rate of the received PHY Payload field. The bits shall be set, depending on the mean PRF, according to Table 16-8. Support for the 850 kb/s data rate is mandatory, all others are optional.

Table 16-8—Nominal data rates

| b0 b1 | Mean PRF 15.60 or 62.40 MHz | Mean PRF 3.90 MHz |
|-------|-----------------------------|-------------------|
| 00    | 0.11                        | 0.11              |
| 01    | 0.85                        | 0.85              |
| 10    | 6.81                        | 1.70              |
| 11    | 27.24                       | 6.81              |

The Frame Length field shall be an unsigned integer number that indicates the number of octets in the PSDU field. The Frame Length field shall be passed to the reference modulator most significant bit first.

The Ranging field shall be set to one if the current frame is an RFRAME and shall be set to zero otherwise.

The Preamble Duration field represents the length (in preamble symbols) of the SYNC portion of the SHR field. The Preamble Duration field shall be set according to Table 16-9.

**Table 16-9—Preamble Duration field values**

| Field value, b11 b12 | SYNC length (symbols) |
|----------------------|-----------------------|
| 00                   | 16                    |
| 01                   | 64                    |
| 10                   | 1024                  |
| 11                   | 4096                  |

The Preamble Duration field is intended for use during ranging operations and is used by a receiver of the PPDU to help determine at which preamble symbol the PHY acquired and began tracking the preamble. A receiver may use the Preamble Duration field to set the value of its own preamble duration based upon the received value when communicating a ranging Ack frame.

The SECDED (single error correct, double error detect) field is a simple Hamming block code that enables the correction of a single error and the detection of two errors at the receiver. The SECDED bit values, b13, ..., b18 depend on PHR bits, b0, ..., b12 and are computed as follows:

$$b18 = \text{XOR}(b1, b0, b8, b6, b4, b3, b10, b11)$$

$$b17 = \text{XOR}(b0, b6, b5, b3, b2, b9, b10, b12)$$

$$b16 = \text{XOR}(b1, b8, b7, b3, b2, b9, b10)$$

$$b15 = \text{XOR}(b8, b7, b6, b5, b4, b9, b10)$$

$$b14 = \text{XOR}(b12, b11)$$

$$b13 = \text{XOR}(b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b14, b15, b16, b17, b18)$$

#### 16.2.7 PHY Payload field

The PHY Payload field is encoded as shown in Figure 16-7.

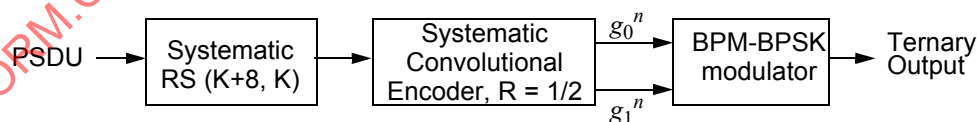


**Figure 16-7—PHY Payload field encoding process**

The PHY Payload field shall be formed as follows:

- Encode the PSDU using systematic Reed-Solomon block code, which adds 48 parity bits as described in 16.3.3.1.

- Encode the output of the Reed-Solomon block code using a systematic convolutional encoder as described in 16.3.3.210.2, except in the cases where the Viterbi rate for the modulation is one, as defined in Table 16-3. In these cases, the convolutional encoder is bypassed.
- Spread and modulate the encoded block using BPM-BPSK modulation as described in 16.3.

## 16.3 Modulation

### 16.3.1 Modulation mathematical framework

The transmit waveform during the  $k$ th symbol interval may be expressed as follows:

$$x^{(k)}(t) = [1 - 2g_1^{(k)}] \sum_{n=1}^{N_{cpb}} [1 - 2s_{n+kN_{cpb}}] \times p(t - g_0^{(k)}T_{BPM} - h^{(k)}T_{burst} - nT_c)$$

This equation describes the time hopping with polarity scrambling, which improves the interference rejection capabilities of the HRP UWB PHY. The  $k$ th symbol interval carries two information bits  $g_0^{(k)}$  and  $g_1^{(k)} \in \{0, 1\}$ . Bit  $g_0^{(k)}$  is encoded into the burst position, whereas bit  $g_1^{(k)}$  is encoded into the burst polarity. The sequence  $s_{n+kN_{cpb}} \in \{0, 1\}$ ,  $n = 0, 1, \dots, N_{cpb} - 1$  is the scrambling code used during the  $k$ th symbol interval,  $h^{(k)} \in \{0, 1 - N_{hop} - 1\}$  is the  $k$ th burst hopping position, and  $p(t)$  is the transmitted pulse shape at the antenna input. The burst hopping sequence  $h^{(k)}$  provides for multiuser interference rejection. The chip scrambling sequence  $s_{n+kN_{cpb}}$  provides additional interference suppression among coherent receivers as well as spectral smoothing of the transmitted waveform. Note that the equation defines the transmitted signal during the valid burst interval; at all other possible burst positions, no signal shall be transmitted. A reference modulator illustrating the BPM-BPSK modulation is shown in Figure 16-8.

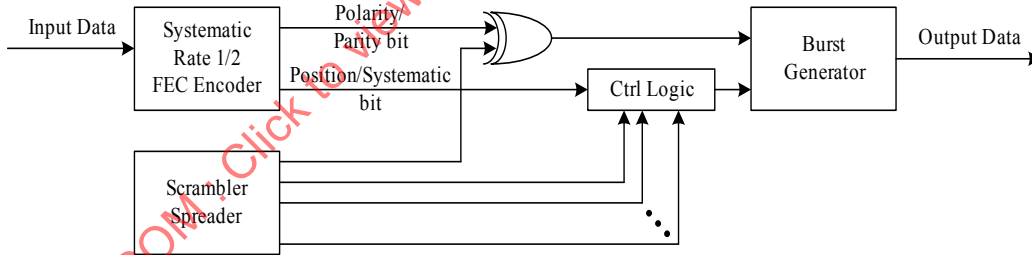


Figure 16-8—Reference modulator

Note here that the FEC encoder is not included if the modulation Viterbi rate is one, as described in 16.2.7. In this case, the FEC encoder is replaced by a multiplexer that shall apply even bits to the position input and odd bits to the polarity input.

### 16.3.2 Spreading

The time-varying spreader sequence  $s_{n+kN_{cpb}}$  and the time-varying burst hopping sequence  $h^{(k)}$  shall be generated from a common PRBS scrambler.

The polynomial for the scrambler generator shall be  $g(D) = 1 + D^{14} + D^{15}$ .

where  $D$  is a single chip delay,  $T_c$  element. This polynomial forms not only a maximal length sequence, but also it is a primitive polynomial. By the given generator polynomial, the corresponding scrambler output is generated as follows:

$$s_n = s_{n-14} \oplus s_{n-15} \quad n = 0, 1, 2, \dots$$

where  $\oplus$  denotes modulo-2 addition.

A linear feedback shift register (LFSR) realization of the scrambler is shown in Figure 16-9. The LFSR shall be initialized upon the transmission of bit 0 of the PHR. Note that  $N_{cpb}$  may change depending on the data rate and PRF in use during the PSDU. The LFSR shall not be reset after transmission of the PHR.

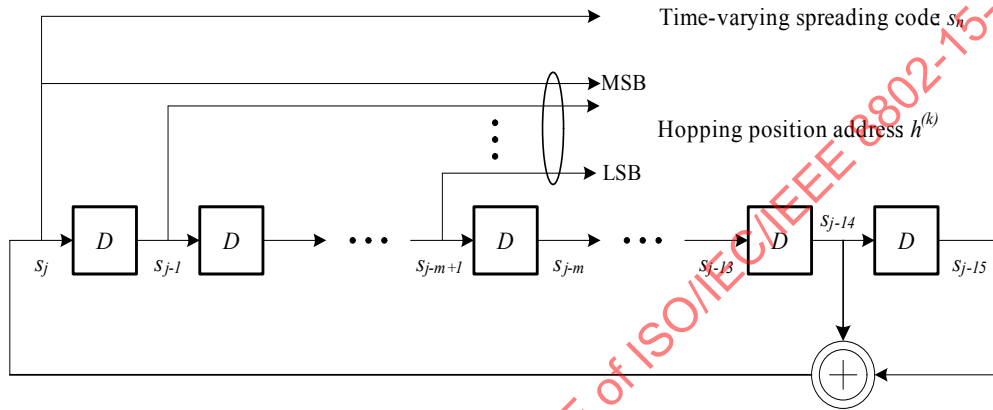


Figure 16-9—LFSR implementation of the scrambler

The initial state of the LFSR shall be determined from the preamble code by first removing all the zeros in the ternary code and then replacing all the negative ones with a zero. The first 15 bits of the resulting binary state shall be loaded into the LFSR. Table 16-10 shows an example of this procedure for preamble code,  $C_6$  (length 31, preamble code index 6, as defined in Table 16-6). The table shows the initial state as well as the first 16 output bits from the scrambler.

Table 16-10—Example LFSR initial state for preamble code 6

| Initial state<br>( $s_{-15}, s_{-14}, \dots, s_{-1}$ ) | LFSR output: First 16 bits<br>$s_0, s_1, \dots, s_{15}$ ( $s_0$ first in time) |
|--------------------------------------------------------|--------------------------------------------------------------------------------|
| 111000101101101                                        | 0010011101101110                                                               |

Note that even though each device within a PAN uses the same initial LFSR setting, the communication in WPAN is asynchronous so that the hopping and scrambling provides interference rejection.

The LFSR shall be clocked at the peak PRF of 499.2 MHz as specified in Table 16-3. During the  $k$ th symbol interval, the LFSR shall be clocked  $N_{cpb}$  times, and the scrambler output shall be the  $k$ th scrambling code  $s_{n+kN_{cpb}}$ ,  $n = 0, 1, \dots, N_{cpb} - 1$ . Furthermore, the  $k$ th burst hopping position shall be computed as follows:

$$h^{(k)} = 2^0 s_{kN_{cpb}} + 2^1 s_{1+kN_{cpb}} + \dots + 2^{m-1} s_{m-1+kN_{cpb}}$$

where  $m = \log_2(N_{hop})$ .

As shown in Table 16-3, the number of hopping burst  $N_{hop}$  is always a power of two, and consequently,  $m$  is always an integer. Note that for  $N_{cpb} < m$ , the LFSR is clocked  $N_{cpb}$  times, not  $m$  times.

For the mandatory modes with mean data PRFs of 15.60 MHz and 3.90 MHz, the numbers of hopping bursts are 8 and 32, respectively, as indicated in Table 16-3, and consequently,  $m$  takes on the values 3 and 5, respectively. The corresponding hopping sequences are as follows:

$$h^{(k)} = s_{kN_{cpb}} + 2s_{1+kN_{cpb}} + 4s_{2+kN_{cpb}} \quad \text{Mean PRF} = 15.60 \text{ MHz}$$

$$h^{(k)} = s_{kN_{cpb}} + 2s_{1+kN_{cpb}} + 4s_{2+kN_{cpb}} + 8s_{3+kN_{cpb}} + 16s_{4+kN_{cpb}} \quad \text{Mean PRF} = 3.90 \text{ MHz}$$

### 16.3.3 FEC

The FEC used by the HRP UWB PHY is a concatenated code consisting of an outer Reed-Solomon systematic block code and an inner half-rate systematic convolutional code. The inner convolutional code is not necessarily enabled at all data rates; the rows of Table 16-3 that have a Viterbi rate of 1 indicate that the inner convolutional code is disabled for the PSDU part of the PHY frame.

The FEC encoding of a block of  $M$  PSDU bits,  $b_0, b_1, \dots, b_{M-1}$ , is shown in Figure 16-10. The Reed-Solomon encoder shall append 48 parity bits,  $p_0, p_1, \dots, p_{47}$ , to the original block. This results in a Reed-Solomon encoded block of length  $M + 48$ . When the Viterbi rate is 0.5, a half-rate systematic convolutional encoder shall encode the Reed-Solomon encoded block into a systematic coded block of length  $2M + 96$  bits. The convolutional systematic bits shall be used to encode the position of the burst, whereas the convolutional parity bits shall be used to encode the polarity of the pulses within a burst. When the Viterbi rate is one, even outputs of the Reed-Solomon encoder ( $b_0, b_2, \dots, b_{M-2}$ ,  $p_0, p_2, \dots, p_{46}$ ) shall be used to encode the position of the burst, and odd outputs ( $b_1, b_3, \dots, b_{M-1}$ ,  $p_1, p_3, \dots, p_{47}$ ) shall be used to encode the polarity of the pulses. Note here that  $M$  is always an even number.

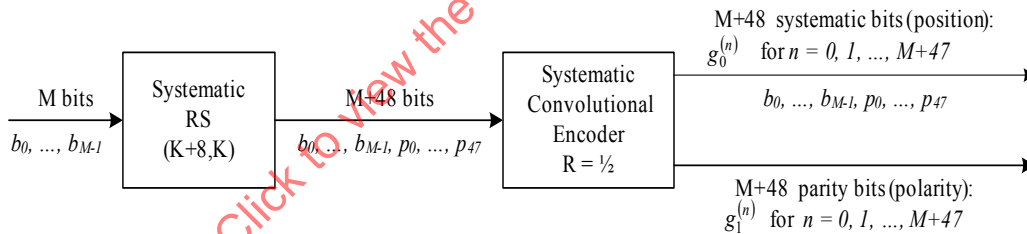


Figure 16-10—FEC encoding process

A noncoherent receiver cannot see the convolutional parity bits (parity bits), and consequently, a noncoherent receiver may use only a Reed-Solomon decoder to improve its performance. A coherent receiver may use either or both Reed-Solomon and convolutional decoding algorithms. Note here that since both the Reed-Solomon and the convolutional codes are both systematic, a receiver (either coherent or noncoherent) may be implemented without an FEC decoder. In this case, the information bits are simply recovered by demodulating the position of the burst. There will be additional parity check bits as a result of the Reed-Solomon encoding, but these may be simply ignored.

#### 16.3.3.1 Reed-Solomon encoding

The systematic Reed-Solomon code is over the Galois field,  $GF(2^6)$ , which is built as an extension of  $GF(2)$ . The systematic Reed-Solomon code shall use the following generator polynomial:



$$g(x) = \prod_{k=1}^8 (x + \alpha^k) = x^8 + 55x^7 + 61x^6 + 37x^5 + 48x^4 + 47x^3 + 20x^2 + 6x + 22$$

where  $\alpha = 010000$  is a root of the binary primitive polynomial  $1 + x + x^6$  in  $\text{GF}(2^6)$ .

In Reed-Solomon encoding  $\text{RS}_6(K+8, K)$ , a block of  $I$  bits (with  $K = \lceil I/6 \rceil$ ) is encoded into a codeword of  $I + 48$  bits. The Reed-Solomon encoding procedure is performed in the following steps:

- a) *Addition of dummy bits.* The block of  $I$  information bits is expanded by adding  $330 - I$  dummy (zero) bits to the beginning of the block. The expanded block is denoted as  $\{d_0, d_1, \dots, d_{329}\}$  where  $d_0$  is the first in time.
- b) *Bit-to-symbol conversion.* The 330 bits  $\{d_0, d_1, \dots, d_{329}\}$  are converted into 55 Reed-Solomon symbols  $\{D_0, D_1, \dots, D_{54}\}$  having the following polynomial representation:

$$D_k = \alpha^5 d_{6k+5} + \alpha^4 d_{6k+4} + \alpha^3 d_{6k+3} + \alpha^2 d_{6k+2} + \alpha d_{6k+1} + d_{6k}, \text{ where } k = 0, 54$$

Resulting 6-bit symbols are presented as  $D_k = \{d_{6k+5}, d_{6k+4}, d_{6k+3}, d_{6k+2}, d_{6k+1}, d_{6k}\}$ , where  $d_{6k+5}$  is the MSB and  $d_{6k}$  is the LSB.

- c) *Encoding.* The information symbols  $\{D_0, D_1, \dots, D_{54}\}$  are encoded by systematic  $\text{RS}_6(63,55)$  code with output symbols  $\{U_0, U_1, \dots, U_{62}\}$  ordered as follows:

$$U_k = \begin{cases} D_k & (k = 0, 1, \dots, 54) \\ P_k & (k = 55, 56, \dots, 62) \end{cases}$$

where  $P_k$  are parity check symbols added by  $\text{RS}_6(63,55)$  encoder.

The information polynomial associated with the information symbols  $\{D_0, D_1, \dots, D_{54}\}$  is denoted as  $D(x) = x^{54}D_0 + x^{53}D_1 + \dots + xD_{53} + D_{54}$ . The parity check polynomial associated with the parity check symbols is denoted as  $P(x) = x^7P_{55} + x^6P_{56} + \dots + xP_{61} + P_{62}$ . The parity check symbols are calculated as follows:

$$P(x) = \text{remainder}[x^8 D(x) / g(x)]$$

$$U(x) = x^8 D(x) + P(x)$$

- d) *Symbol-to-bit conversion.* The output symbols  $\{U_0, U_1, \dots, U_{62}\}$  are converted into binary form with LSB coming out first, resulting in a block of 378 bits  $\{u_0, u_1, \dots, u_{377}\}$ .
- e) *Removal of dummy bits.* The  $330 - I$  dummy bits added in the first step are removed. Only the last  $I + 48$  bits are transmitted, i.e.,  $\{u_{330-I}, u_{331-I}, \dots, u_{377}\}$  with  $u_{330-I}$  being first in time.

### 16.3.3.2 Systematic convolutional encoding

The inner convolutional encoder shall use the rate  $R = 1/2$  code with generator polynomials  $g_0 = [010]_2$  and  $g_1 = [101]_2$  as shown in Figure 16-11. Upon transmission of each PPDU, the encoder shall be initialized to the all zero state. Additionally, the encoder shall be returned to the all zero state by appending two zero bits to the PPDU. Note that since the generator polynomials are systematic, they are also noncatastrophic.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

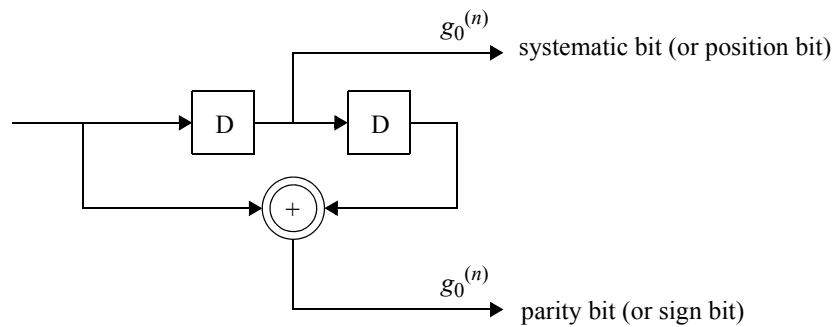


Figure 16-11—Systematic convolutional encoder

## 16.4 RF requirements

### 16.4.1 Operating frequency bands

The set of operating frequency bands is defined in Table 16-11. For the sub-gigahertz operation, channel 0 is defined as the mandatory channel; for the low-band operation, channel 3 is the mandatory channel; and for the high-band operation, channel 9 is the mandatory channel.

Table 16-11—HRP UWB PHY band allocation

| Band group <sup>a</sup><br>(decimal) | Channel number<br>(decimal) | Center frequency, $f_c$<br>(MHz) | Band width<br>(MHz) | Mandatory/Optional     |
|--------------------------------------|-----------------------------|----------------------------------|---------------------|------------------------|
| 0                                    | 0                           | 499.2                            | 499.2               | Mandatory below 1 GHz  |
| 1                                    | 1                           | 3494.4                           | 499.2               | Optional               |
|                                      | 2                           | 3993.6                           | 499.2               | Optional               |
|                                      | 3                           | 4492.8                           | 499.2               | Mandatory in low band  |
|                                      | 4                           | 3993.6                           | 1331.2              | Optional               |
| 2                                    | 5                           | 6489.6                           | 499.2               | Optional               |
|                                      | 6                           | 6988.8                           | 499.2               | Optional               |
|                                      | 7                           | 6489.6                           | 1081.6              | Optional               |
|                                      | 8                           | 7488.0                           | 499.2               | Optional               |
|                                      | 9                           | 7987.2                           | 499.2               | Mandatory in high band |
|                                      | 10                          | 8486.4                           | 499.2               | Optional               |
|                                      | 11                          | 7987.2                           | 1331.2              | Optional               |
|                                      | 12                          | 8985.6                           | 499.2               | Optional               |
|                                      | 13                          | 9484.8                           | 499.2               | Optional               |
|                                      | 14                          | 9984.0                           | 499.2               | Optional               |
|                                      | 15                          | 9484.8                           | 1354.97             | Optional               |

<sup>a</sup> Note that bands indicate a sequence of adjacent HRP UWB center frequencies: band 0 is the sub-gigahertz channel, band 1 has the low-band HRP UWB channels, and band 2 has the high-band channels.

Figure 16-12 is a graphical representation of the data presented in Table 16-11. Each HRP UWB PHY channel is shown as a heavy black line centered on the channel's center frequency. The length of the lines depicts the channel bandwidth.

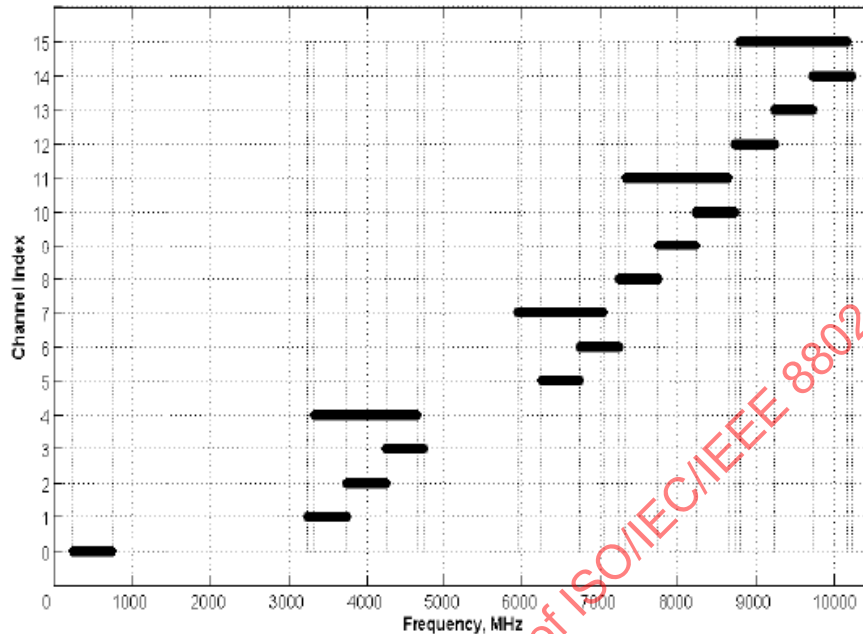


Figure 16-12—HRP UWB PHY band plan

#### 16.4.2 Channel assignments

A total of 32 complex channels are assigned for operation, two channels in each of the 16 defined operating frequency bands. A compliant implementation shall support at least the two channels for one of the mandatory bands.

#### 16.4.3 Regulatory compliance

The maximum allowable output PSD shall be in accordance with practices specified by the appropriate regulatory bodies.

#### 16.4.4 Operating temperature range

A conformant implementation shall meet all of the specifications in this standard for ambient temperatures from 0 °C to 40 °C.

#### 16.4.5 Baseband impulse response

The transmitted pulse shape  $p(t)$  shall be constrained by the shape of its cross-correlation function with a standard reference pulse,  $r(t)$ . The normalized cross-correlation between two waveforms is defined as follows:

$$\phi(\tau) = \frac{1}{\sqrt{E_r E_p}} \operatorname{Re} \int_{-\infty}^{\infty} r(t) p^*(t + \tau) dt$$

where  $E_r$  and  $E_p$  are the energies of  $r(t)$  and  $p(t)$ , respectively. The reference  $r(t)$  pulse used in the calculation of  $|\phi(\tau)|$  is a root raised cosine pulse with a roll-off factor of  $\beta = 0.5$ . Mathematically this is as follows:

$$r(t) = \frac{4\beta}{\pi\sqrt{T_p}} \frac{\cos[(1+\beta)\pi t/T_p] + \frac{\sin[(1-\beta)\pi t/T_p]}{4\beta(t/T_p)}}{1 - (4\beta t/T_p)^2}$$

where  $T_p$  is the inverse of the chip frequency. Table 16-12 shows the required pulse duration for each channel.

**Table 16-12—Required reference pulse durations in each channel**

| Channel number          | Pulse duration, $T_p$<br>(ns) | Main lobe width, $T_w$<br>(ns) |
|-------------------------|-------------------------------|--------------------------------|
| {0:3, 5:6, 8:10, 12:14} | 2.00                          | 0.5                            |
| 7                       | 0.92                          | 0.2                            |
| {4, 11}                 | 0.75                          | 0.2                            |
| 15                      | 0.74                          | 0.2                            |

In order for an HRP UWB PHY transmitter to be compliant with this standard, the transmitted pulse  $p(t)$  shall have a magnitude of the cross-correlation function  $|\phi(\tau)|$  whose main lobe is greater than or equal to 0.8 for a duration of at least  $T_w$ , as defined in Table 16-12, and any sidelobe shall be no greater than 0.3. For the purposes of testing a pulse for compliance, the following are defined: Let  $|\phi(\tau)|$  be the magnitude of the cross-correlation of  $p(t)$  and  $r(t)$ , and let  $\tau_i$ , for  $i = 1, 2, \dots$ , be a set of critical points as follows:

$$\frac{d}{d\tau} |\phi(\tau)| \big|_{\tau = \tau_i} = 0$$

The maximum of the function occurs at one of these critical points,  $\tau_{max}$ , where  $|\phi(\tau_{max})| \geq |\phi(\tau)|$  for all values of  $\tau$ . The requirement thus states that for some continuous set of values that contain the point  $\tau_{max}$ , the function  $|\phi(\tau)|$  is greater than 0.8. In addition, the second constraint on the value of sidelobes may be stated mathematically as  $|\phi(\tau_i)| \leq 0.3$  for all  $\tau_i$ .

Figure 16-13 shows an example HRP UWB-compliant pulse,  $p(t)$  (left plot), along with the root raised cosine reference pulse  $r(t)$  (middle plot) with  $T_p = 2.0$  ns and the magnitude of the cross-correlation  $|\phi(\tau)|$  (right plot). The pulse  $p(t)$  is an 8 order butterworth pulse with a 3 dB bandwidth of 500 MHz. Figure 16-13 is intended to show that this example pulse meets the requirements for compliance. Specifically, the main lobe is above 0.8 for nearly 1 ns, and no sidelobe is greater than 0.3 (in this case, the largest sidelobe peak is 0.2). The pulse  $p(t)$  is a compliant pulse for channels {0:3, 5:6, 8:10, 12:14}.

Note that it is not the intention of this standard to imply that pulse shaping only occurs at baseband, but rather that the measurements described here occur on the pulse envelope if shaping is done at passband.

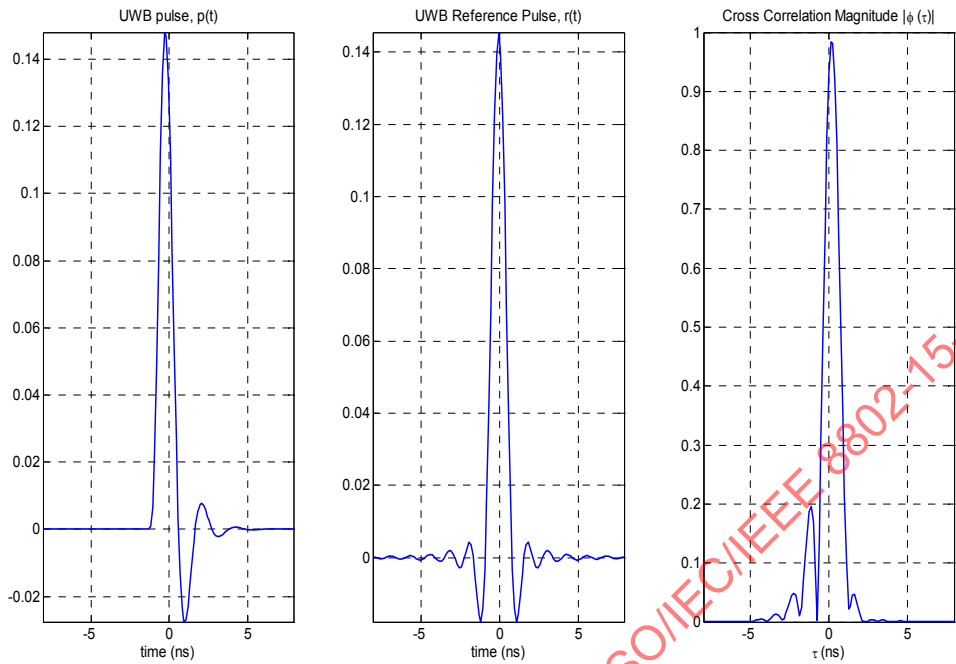


Figure 16-13—Compliant pulse example

16.4.6 Transmit PSD mask

The transmitted spectrum shall be less than  $-10$  dB relative to the maximum spectral density of the signal for  $0.65/T_p < |f-f_c| < 0.8/T_p$  and  $-18$  dB for  $|f-f_c| > 0.8/T_p$ . For example, the transmit spectrum mask for channel 4 is shown in Figure 16-14. The measurements shall be made using a 1 MHz resolution bandwidth and a 1 kHz video bandwidth.

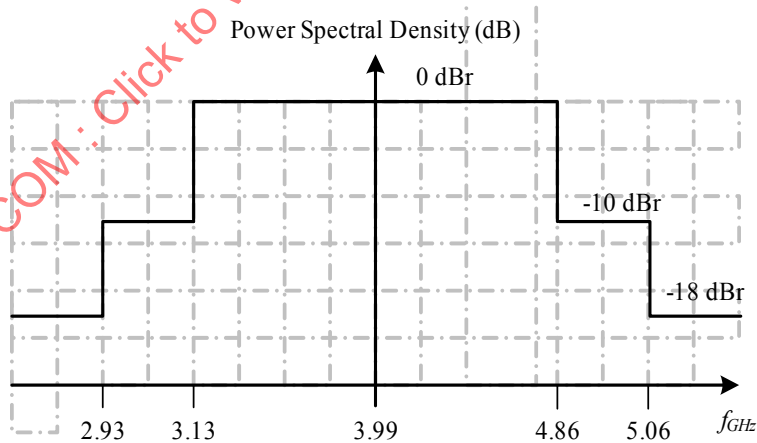


Figure 16-14—Transmit spectrum mask for band 4

**16.4.7 Chip rate clock and chip carrier alignment**

An HRP UWB transmitter shall be capable of chipping at the peak PRF given in Table 16-3 with an accuracy of  $\pm 20 \times 10^{-6}$ . In addition, for each HRP UWB PHY channel, the center of transmitted energy shall be within the values listed in Table 16-11 also with an accuracy of  $\pm 20 \times 10^{-6}$ . The measurements shall be made using a 1 MHz resolution bandwidth and a 1 kHz video bandwidth.

**16.4.8 TX-to-RX turnaround time**

The HRP UWB PHY shall have a TX-to-RX turnaround time as defined in 10.2.1.

**16.4.9 RX-to-TX turnaround time**

The HRP UWB PHY shall have a TX-to-RX turnaround time as defined in 10.2.2.

**16.4.10 Transmit center frequency tolerance**

The HRP UWB PHY transmit center frequency tolerance shall be  $\pm 20 \times 10^{-6}$ . The tolerance on the chipping clock given in 16.4.7 takes precedence over this requirement.

**16.4.11 Receiver maximum input level of desired signal**

The HRP UWB PHY shall have a receiver maximum input level greater than or equal to -45 dBm/MHz, using the measurement defined in 10.2.4.

**16.4.12 Receiver ED**

The HRP UWB PHY shall provide the receiver ED measurement as described in 10.2.5. The averaging period for the receiver ED measurement is implementation specific.

The ED measurement for each channel may be performed as a series of measurements, each made at a fraction of the total channel bandwidth, in which case *phyHrpUwbScanBinsPerChannel* specifies the number of frequency increments used. When this value is greater than one, the ED result reported using the MLME-SCAN.confirm primitive shall be a list of ED measurements, one for each frequency increment measurement. An implementation may provide multiple ED measurements, for example, to provide information to a higher layer that detects non-HRP UWB services for the purpose of active detect and avoid (DAA) procedures as may be required in some environments.

**16.4.13 LQI**

The HRP UWB PHY shall provide the LQI measurement as described in 10.2.6.

**16.4.14 CCA**

For CCA mode 6, the CCA detection time for the HRP UWB PHY shall be equal to 40 mandatory symbol periods, which includes at least 8 (multiplexed) preamble symbols, as described in 16.6.

**16.5 HRP UWB PHY optional pulse shapes**

The HRP UWB PHY offers the capability to transmit several optional pulse types. The use of these options is controlled by the PAN coordinator and shall be limited to the nonbeacon frames. In other words, beacon frames shall be transmitted using the mandatory pulse shape as defined in 16.4.5, but all other frames may be transmitted using the optional pulse shapes if all devices in the PAN are capable of supporting the

optional pulse shapes. PANs that use the optional pulse shapes shall indicate the use of a specific option via the *phyHrpUwbCurrentPulseShape* PIB attribute. Devices choosing to join a PAN using one of the optional pulse shapes should make their decision based on the value of *phyHrpUwbCurrentPulseShape* that is reported during the scan procedure.

### 16.5.1 HRP UWB PHY optional chirp on UWB (CoU) pulses

The purpose of CoU pulses is to provide an additional dimension (besides frequency and DS codes) to support simultaneously operating piconets. Because CoU is an optional mode of pulse shapes in addition to the mandatory pulse shape, all modulation specifications shall be the same as they are for the mandatory pulse shape except those defined for the CoU pulses when a device implements the CoU option.

A mathematical representation of a CoU pulse at baseband is as shown below:

$$p_{CoU}(t) = \begin{cases} p(t) \exp\left(-j\frac{\pi\beta t^2}{2}\right) & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{otherwise} \end{cases}$$

where

$p(t)$  denotes a mandatory pulse shape that satisfies constraints in 16.4.5  
 $\beta = B/T$  is the chirping rate (chirping slope). Moreover,  $B$  and  $T$  are the bandwidth and time duration of the CoU pulse, respectively.

A graphical example of CoU pulse is shown in Figure 16-15.

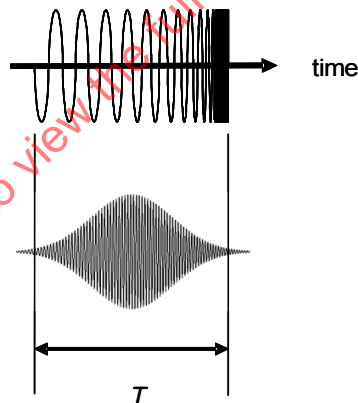


Figure 16-15—Graphical view of a CoU pulse

The CoU is an operation added to the mandatory pulse. When a CoU pulse is transmitted, the receiver needs to perform a matched de-chirp operation to demodulate the signal.

The optional CoU pulses are admitted with two slopes per each DS code per each 500 MHz bandwidth. The chirp slopes are denoted as CCh.1 and CCh.2. Within channels 4, 7, 11, and 15, there are chirp slopes admitted per each DS code. These are denoted as CCh.3 through CCh.6. The values for each chirp slope are listed in Table 16-13.

**Table 16-13—CoU channel slopes**

| CoU number | $\beta$ (slopes) |
|------------|------------------|
| CCh.1      | 500 MHz/2.5 ns   |
| CCh.2      | –500 MHz/2.5 ns  |
| CCh.3      | 1 GHz/5 ns       |
| CCh.4      | –1 GHz/5 ns      |
| CCh.5      | 1 GHz/10 ns      |
| CCh.6      | –1 GHz/10 ns     |

**16.5.2 HRP UWB PHY optional continuous spectrum (CS) pulses**

This subclause specifies optional CS pulses. A CS pulse is obtained by passing the mandatory pulse through an all-passing CS filter. The CS filter introduces controlled group delays to the input pulse. The purpose of the optional CS pulses is to reduce the interference level between different PANs.

Since CS is an optional mode of pulse shapes in addition to the mandatory pulse shape, all modulation specifications shall be the same as they are for the mandatory pulse shape except those defined for the CS pulses when a device implements the CS option.

An optional CS pulse  $p_{CS}(t)$  is defined as shown below:

$$p_{CS}(t) = \int P(f) \exp[-j2\pi f(t - (\tau \times f))] df$$

where

$\tau$  represents the group delay (s/Hz)

$P(f)$  represents the Fourier transform of  $p(t)$ , where  $p(t)$  is any pulse shape that meets the requirements defined in 16.4.5

The Fourier transform is defined as shown below:

$$P(f) = \int p(t) \exp[-j2\pi ft] dt$$

The CS filtering is an operation added to the mandatory pulse. When a CS pulse is transmitted, the receiver needs to perform an inverse CS filtering operation to demodulate the signal.



Some examples of CS pulses are shown in Figure 16-16.

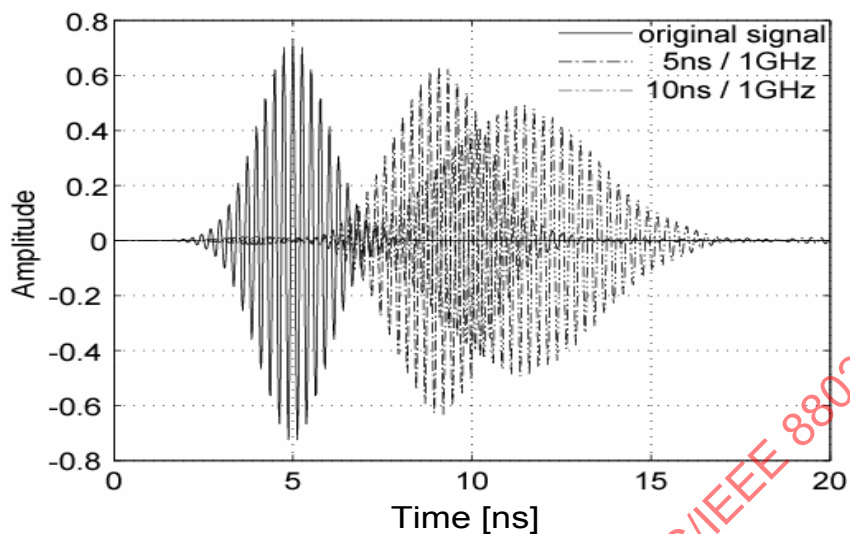


Figure 16-16—Examples of CS pulses

Each 500 MHz band shall use No.1 or No.2 pulses, while each 1.5 GHz band shall use one of No.3 through No.6 pulses, as defined in Table 16-14.

Table 16-14—CS group delays

| CS pulse number | $\tau$ (Group delay) |
|-----------------|----------------------|
| No.1            | 2 ns/500 MHz         |
| No.2            | −2 ns/500 MHz        |
| No.3            | 5 ns/1 GHz           |
| No.4            | −5 ns/1 GHz          |
| No.5            | 10 ns/1 GHz          |
| No.6            | −10 ns/1 GHz         |

### 16.5.3 HRP UWB PHY linear combination of pulses (LCP)

LCP is an optional pulse shape that can be used in regulatory regions where DAA schemes are required by regulators. Using LCP pulses enables a PAN to limit interference to incumbent wireless systems. The pulse shape for LCP is denoted  $p_{LCP}(t)$  and is the sum of  $N$  weighted and delayed pulses  $p(t)$  as follows:

$$p_{LCP}(t) = \sum_{i=1}^N a_i p(t - \tau_i)$$

where  $p(t)$  is any pulse that satisfies the cross-correlation constraints outlined in 16.4.5.

The number of pulses  $N$  that can be combined is set to four (although smaller number of pulses can be realized by setting the amplitudes of some of the pulses to zero). The values of the pulse delays shall be limited to  $0 \leq \tau_i \leq 4$  ns. The value of  $\tau_1$  is assumed to be zero, and thus, the remaining delays are considered as relative delay time with respect to the nominal pulse location. The values for these delays are stored as the PIB values *phyHrpUwbLcpDelay2*, *phyHrpUwbLcpDelay3*, and *phyHrpUwbLcpDelay4*. The values for the amplitudes  $a_i$  are stored as the PIB values *phyHrpUwbLcpweight1* through *phyHrpUwbLcpDelay4*, as defined in Table 11-2. The amplitudes,  $a_i$ , shall be selected so that the energy in the combined pulse is the same as the energy in the mandatory pulse. The numerical values of the delays and amplitudes of the pulses shall be transmitted following the general framework of optional pulse shapes, as defined in 16.5. The method to compute the weights and delay values is outside the scope of this standard.

## 16.6 Extended preamble for optional CCA mode 6

The PHY may provide the capability to perform the optional CCA mode 6, as defined in 10.2.7. This CCA mode shall be supported by the modified frame structure where preamble symbols are multiplexed with the data symbols in the PHR field and the PHY Payload field of a frame.

Figure 16-17 shows the modified frame structure with multiplexed preamble symbols. One preamble symbol is inserted after each PHR and PHY Payload segment. The inserted preamble symbol shall be the same as the symbol used in the SHR, as described in 16.2.5.1, of the same frame. The time interval between two neighboring inserted preamble symbols, which is also the time duration of each PHR or PHY Payload segment, is independent of the current data rate. The PIB attribute *phyHrpUwbInsertedPreambleInterval* defines a constant time interval based on the data rate of 850 kb/s for all operation bands. The data rate of 850 kb/s is listed in Table 16-8 with the data rate field index of 01. The value of the PIB attribute *phyHrpUwbInsertedPreambleInterval* is fixed to four. Distinguished from the CCA in narrowband systems, which is used to detect the energy of carrier waveforms, CCA mode 6 is based on the frame with multiplexed preamble is used to detect the presence of preamble symbols. The processing gain can be enhanced by exploiting the spreading characteristics and repetition of the preamble symbols.

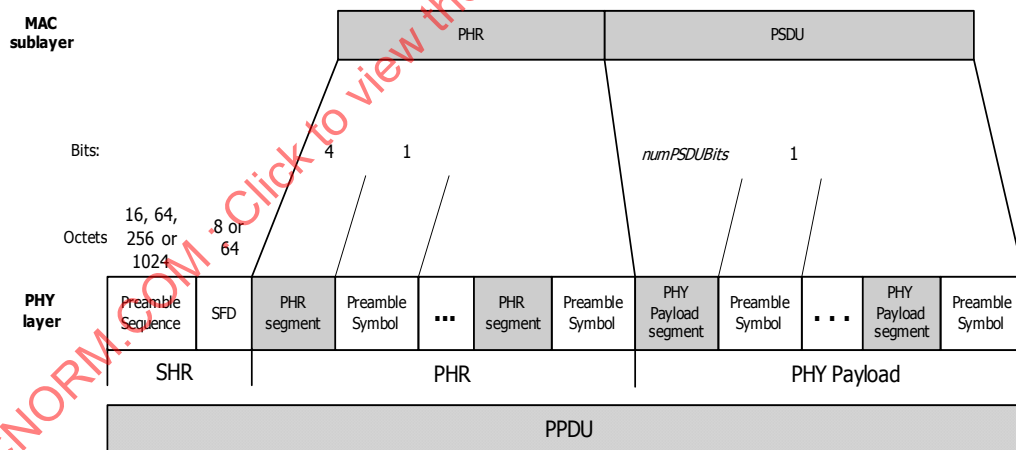


Figure 16-17—Illustration of the modified frame structure with multiplexed preamble

The PAN coordinator of a PAN shall coordinate all nodes in the PAN before the CCA mode 6 is enabled. The modified frame structure with multiplexed preamble shall be applied to a Data frame or a MAC Command frame in the CAP only when the CCA mode 6 is enabled.

The CCA detection time shall be equivalent to 40 data symbol periods,  $T_{dsym}$ , for a nominal 850 kb/s, or equivalently, at least 8 (multiplexed) preamble symbols should be captured in the CCA detection time.

In addition to enabling the CCA mode 6, the multiplexed preamble symbols can help to improve ranging accuracy or assist data demodulation. This function is similar to that of the pilot tone in narrowband systems.

## 16.7 Ranging

Support for ranging is optional. A PHY that supports ranging is called a ranging-capable device (RDEV), and it has optional and mandatory capabilities. An RDEV shall support the ranging counter described in 16.7.1 and the FoM described in 16.7.3. An RDEV may support optional crystal characterization described in 16.7.2 and the optional DPS, as describe in “Applications of IEEE Std 802.15.4” [B3].

RDEVs produce results, called timestamp reports, that are used by higher layers to compute the ranges between devices. An RDEV timestamp report shall consist of a 4-octet ranging counter start value, 4-octet ranging counter stop value, 4-octet ranging tracking interval, 3-octet ranging tracking offset, and 1-octet ranging FoM. These numbers are always reported together in the same primitive and remain together for their entire processing lifetime. It is not acceptable to have any pipelining of the individual results where (for example) in a timestamp report the ranging tracking offset and ranging tracking interval might be associated with the ranging counter value of the previous timestamp report and the ranging FoM might be associated with the ranging counter value of the timestamp report before that.

### 16.7.1 Ranging counter

The ranging counter supported by an RDEV is a set of behavioral properties and capabilities of the RDEV that produce ranging counter values. A ranging counter value is a 32-bit unsigned integer. The LSB of the counter value shall represent 1/128 of a chip time at the mandatory chipping rate of 499.2 MHz.

### 16.7.2 Crystal characterization

An RDEV that implements optional crystal characterization shall produce a tracking offset value and a tracking interval value for every timestamp report that is produced. The tracking offset and the tracking interval are computed from measurements taken during an interval that includes the interval bounded by the ranging counter start value and the ranging counter stop value. Note that crystal characterization is relevant only if it is characterizing the crystal that affects the ranging counter.

#### 16.7.2.1 Ranging tracking offset

The HRP UWB ranging tracking offset is a signed magnitude integer. The integer magnitude part of the number shall be 19 bits. The LSB of the integer represents a “part.” The sign bit of the signed magnitude integer shall be equal to zero when the oscillator at the transmitter is a higher frequency than the oscillator at the receiver, and the sign bit shall be 1 when the oscillator at the receiver is a higher frequency than the transmitter. The value of the integer shall be a number that represents the difference in frequency between the receiver’s oscillator and the transmitter’s oscillator after the tracking offset integer is divided by the ranging tracking interval integer of 16.7.2.2. For example, if the difference between the oscillators is  $10 \times 10^{-6}$ , then an acceptable value of the ranging tracking offset would be 10 when the ranging tracking interval is 1 million. Another acceptable value for the ranging tracking offset is 15 when the ranging tracking interval is 1.5 million.

#### 16.7.2.2 Ranging tracking interval

The HRP UWB ranging tracking interval shall be a 32-bit unsigned integer. The LSB of the ranging tracking interval represents a “part” that shall be exactly equal to the “part” in the LSB of the ranging tracking offset

of 16.7.2.1. The size of the “part” is a time period that shall be smaller than or equal to a chip time at the mandatory chipping rate of 499.2 MHz. Use of smaller “parts” for the LSB is encouraged, as described in “Applications of IEEE Std 802.15.4” [B3].

### 16.7.3 Ranging FoM

An RDEV shall produce a ranging FoM for every ranging counter value that is produced. The HRP UWB ranging FoM shall be formatted as shown in Figure 16-18. The FoM Confidence Level field is defined in Table 16-15. The confidence level is the probability that the leading edge of the pulse will arrive during the confidence interval. The FoM Confidence Interval field is defined in Table 16-16. The confidence interval width in Table 16-16 is the entire interval width, not a plus or minus number. The FoM Confidence Interval Scaling Factor field is defined in Table 16-17. Thus, the overall confidence interval is obtained according to the formula *overall confidence interval = confidence interval × confidence interval scaling field*. The MSB of the FoM octet is the extension bit. When the extension bit is set to zero, the fields have the normal meanings given in Table 16-15, Table 16-16, and Table 16-17. When the extension bit is 1, the FoM has the meaning given in Table 16-18.

| Bit 7     | 6                                        | 5 | 4                         | 3 | 2                      | 1 | 0 |
|-----------|------------------------------------------|---|---------------------------|---|------------------------|---|---|
| Extension | Confidence Interval Scaling Factor field |   | Confidence Interval field |   | Confidence Level field |   |   |

Figure 16-18—Ranging FoM

Table 16-15—Confidence Level field

| Confidence level | Bit 2 | Bit 1 | Bit 0 |
|------------------|-------|-------|-------|
| No FoM           | 0     | 0     | 0     |
| 20%              | 0     | 0     | 1     |
| 55%              | 0     | 1     | 0     |
| 75%              | 0     | 1     | 1     |
| 85%              | 1     | 0     | 0     |
| 92%              | 1     | 0     | 1     |
| 97%              | 1     | 1     | 0     |
| 99%              | 1     | 1     | 1     |

Table 16-16—Confidence Interval field

| Confidence interval | Bit 4 | Bit 3 |
|---------------------|-------|-------|
| 100 ps              | 0     | 0     |
| 300 ps              | 0     | 1     |
| 1 ns                | 1     | 0     |
| 3 ns                | 1     | 1     |

**Table 16-17—Confidence Interval Scaling Factor field**

| Confidence interval scaling factor | Bit 6 | Bit 5 |
|------------------------------------|-------|-------|
| Confidence interval $\times 1/2$   | 0     | 0     |
| Confidence interval $\times 1$     | 0     | 1     |
| Confidence interval $\times 2$     | 1     | 0     |
| Confidence interval $\times 4$     | 1     | 1     |

**Table 16-18—FoM values with the extension bit set**

|                                     | Bit 7 | 6                 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------------------------|-------|-------------------|---|---|---|---|---|---|
| The timestamp report is uncorrected | 1     | 0                 | 0 | 0 | 0 | 0 | 0 | 0 |
| Reserved                            | 1     | Any nonzero value |   |   |   |   |   |   |

The FoM characterizes the accuracy of the PHY estimate of the arrival time of the RMARKER at the antenna. The FoM within a timestamp report shall characterize the accuracy of the timer counter value in the same timestamp report.

The FoM value of 0x80 is specifically used to signal the higher layer that the RangingCounterStart value is not correct and the higher layer should use the sounding primitives. The FoM value of 0x00 is special and means “no FoM.” No FoM means that there simply is no information about the quality of a ranging measurement. That is different from reporting a very low quality measurement, but it is known that the measurement cannot be trusted. The FoM value 0x00 is not used to report untrustworthy measurements. The most untrustworthy measurement reportable is 0x79.

## 17. GFSK PHY

### 17.1 PPDU formats

The GFSK PHY shall use the PPDU format described in 12.1, except that the Preamble field is 32 symbols (4 octets) and the bits in each octet shall be “01010101”.

### 17.2 Modulation

#### 17.2.1 GFSK PHY data rates

The data rate of the GFSK PHY shall be 100 kb/s.

#### 17.2.2 Reference modulator diagram

The functional block diagram in Figure 17-1 is provided as a reference for specifying the GFSK PHY modulation and spreading functions. Each bit in the PPDU shall be processed through the data whitening 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, b0, is processed first and the MSB, b7, is processed last.

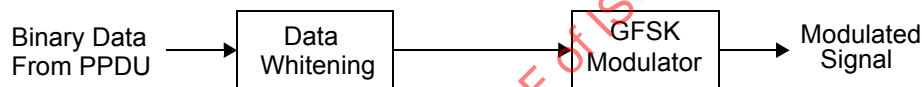


Figure 17-1—GFSK modulation and data whitening functions

#### 17.2.3 Data whitening

Data whitening shall be the exclusive OR of the PHY Payload field with the PN9 sequence. This shall be performed by the transmitter and is as follows:

$$E_n = R_n \oplus \text{PN9}_n$$

where

$R_n$  is the data bit being whitened

$E_n$  is the whitened bit

$\text{PN9}_n$  is the PN9 sequence bit

For each packet transmitted with data whitening,  $R_0$  is the first bit of the PSDU and index  $n$  increments for subsequent bits of the PSDU. The receiver decodes the scrambled data in the following way:

$$R_n = RE_n \oplus \text{PN9}_n$$

where

$R_n$  is the PSDU bit after de-whitening

$RE_n$  is the PSDU bit at the output of the demodulator

The PN generator is defined by the schematic in Figure 17-2.

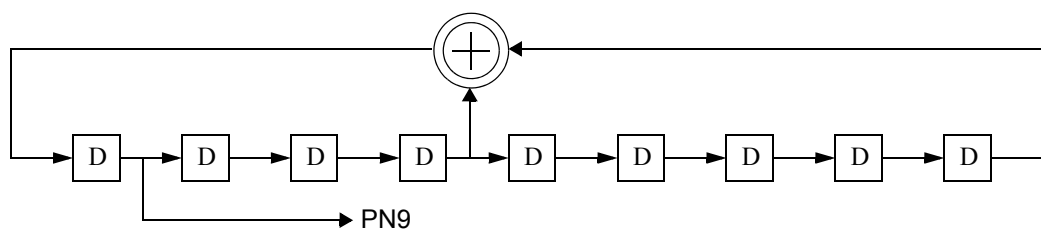


**Figure 17-2—Schematic of the PN generator**

The seed in the PN9 shall be all ones: “1 1 1 1 1 1 1 1”. The PN9 shall be reinitialized to the seed after each packet (either transmit or receive).

The PN9 generator is clocked using the seed as the starting point and enabled after the first clock cycle. For example, the first 30 bits out of the PN9, once it is enabled, would be as follows:

$$\text{PN9}_n = 0_0 0_1 0_2 0_3 1_4 1_5 1_6 1_7 0_8 1_9 1_{10} 1_{11} 0_{12} 0_{13} 0_{14} 0_{15} 1_{16} 0_{17} 1_{18} 1_{19} 0_{20} 0_{21} 1_{22} 1_{23} 0_{24} \\ 1_{25} 1_{26} 0_{27} 1_{28} 1_{29}$$

#### 17.2.4 GFSK modulation

The bit sequences are modulated onto the carrier using GFSK with a modulation index of one where the Gaussian filter BT is 0.5, where a bit value of one is transmitted by shifting the frequency higher than the channel center frequency and a bit value of zero is transmitted by shifting the frequency lower than the current channel center frequency.

The nominal frequency deviation shall be 50 kHz. The deviation shall be between 70% and 130% of the nominal deviation. For the sequence 0101, the deviation shall be between 70% and 110% of the nominal deviation. for the sequence 00011111, the deviation shall be between 80% and 130% of the nominal deviation.

### 17.3 GFSK PHY RF requirements

#### 17.3.1 Operating frequency range

The GFSK PHY operates in the 920.8–928 MHz frequency band.

#### 17.3.2 Transmit PSD mask

The PSD mask for the GFSK PHY is specified as follows:

- The average power measured within  $\pm 100$  kHz of the frequency 300 kHz apart from the center frequency shall be  $-26$  dBm or less for a 1 mW device or  $-18$  dBm or less for a 10 mW device.
- The average power measured with a 100 kHz resolution bandwidth in the frequency band from 920 MHz to 928 MHz except for the frequency band within  $\pm 300$  kHz of the carrier frequency,  $f_c$ , shall be less than  $-39$  dBm

**17.3.3 Symbol rate**

The GFSK PHY symbol rate shall be 100 ksymbol/s with an accuracy of  $\pm 40 \times 10^{-6}$ .

**17.3.4 Receiver sensitivity**

Under the conditions specified in 10.1.7, a compliant GFSK PHY device shall be capable of achieving a sensitivity of  $-85$  dBm or better.

**17.3.5 Receiver interference rejection**

The minimum receiver interference rejection levels are given in Table 17-1. 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 15 is the desired channel, channel 14 and channel 6 are the adjacent channels, and channel 13 and channel 17 are the alternate channels.

**Table 17-1—Minimum receiver interference rejection requirements for GFSK PHY**

| Adjacent channel rejection | Alternate channel rejection |
|----------------------------|-----------------------------|
| 0 dB                       | 24 dB                       |

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant GFSK PHY signal, as defined by 17.2, of pseudo-random data. The desired signal is input to the receiver at a level 3 dB greater than the maximum allowed receiver sensitivity given in 17.3.4.

In either the adjacent or the alternate channel, a compliant signal, as defined by 17.2, is input at the level specified in Table 17-1 relative to the desired signal. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

**17.3.6 TX-to-RX turnaround time**

The GFSK PHY shall have a TX-to-RX turnaround time as defined in 10.2.1.

**17.3.7 RX-to-TX turnaround time**

The GFSK PHY shall have an RX-to-TX turnaround time as defined in 10.2.2.

**17.3.8 Transmit center frequency tolerance**

The GFSK PHY transmit center frequency tolerance shall be  $\pm 40 \times 10^{-6}$  maximum.

**17.3.9 Transmit power**

The GFSK PHY shall be capable of transmitting at a power level of at least  $-3$  dBm.

**17.3.10 Receiver maximum input level of desired signal**

The GFSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.



#### **17.3.11 Receiver ED**

The GFSK PHY shall provide the receiver ED measurement as described in 10.2.5.

#### **17.3.12 LQI**

The GFSK PHY shall provide the LQI measurement as described in 10.2.6.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-4:2018

## 18. MSK PHY

### 18.1 PPDU formats

The MSK PHY shall use the PPDU formats described in 12.1, except that the Preamble field is 32 symbols (4 octets) and the bits in each octet shall be “10101010.”

### 18.2 Data rate

The default data rate of the MSK PHY shall be 250 kb/s. Additional optional data rates for the 433 MHz band are shown in Table 18-1.

**Table 18-1—Data rates for MSK PHY**

| DataRate as used in MCPS-DATA primitives | Data rate (kb/s) |
|------------------------------------------|------------------|
| 1                                        | 31.25            |
| 2                                        | 100              |
| 3                                        | 250              |

The optional data rates for the 433 MHz band are not available on all channels. Table 18-2 shows the allowed data rates for the 433 MHz channels and their associated channel numbers.

**Table 18-2—Data rate channel map**

| Data rate (kb/s) | Channel         |
|------------------|-----------------|
| 31.25            | 0 to 14         |
| 100              | 1, 4, 7, 10, 13 |
| 250              | 2, 7, 12        |

### 18.3 SFD for the MSK PHY

The SFD for the MSK PHY shall be contain the value given in Table 18-3. The SFD is transmitted starting from the leftmost bit (b0).

**Table 18-3—Value of the SFD field for the MSK PHY**

| SFD (b0–b15)        |
|---------------------|
| 1110 0110 1101 0000 |

## 18.4 MSK modulation

### 18.4.1 Reference modulator diagram

The functional block diagram in Figure 18-1 is provided as a reference for specifying the MSK PHY modulation. It should be noted the preamble is not included in the whitening scheme.

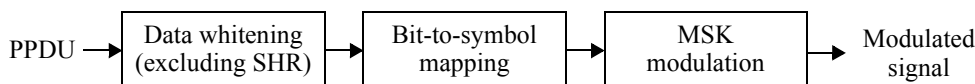


Figure 18-1—MSK functional block diagram

### 18.4.2 Data whitening

Data whitening for the MSK PHY shall use the procedure described in 17.2.3.

### 18.4.3 Bit-to-symbol mapping

The bit rate and symbol rate are equal. The mapping of bits to frequency shall be as described in Table 18-4.

Table 18-4—Bit to frequency mapping

| Bit | Frequency        |
|-----|------------------|
| 0   | $f_c - \Delta f$ |
| 1   | $f_c + \Delta f$ |

In Table 18-4,  $f_c$  is the channel center frequency as defined in Table 10-7 and Table 10-8 and  $\Delta f$  is as follows:

$$\Delta f = \frac{1}{4 \times T_s}$$

where  $T_s$  is the symbol rate.

### 18.4.4 Signal modulation

The MSK modulation shall be FSK with modulation index  $h = 0.5$ . This modulation index corresponds to minimum frequency spacing that allows two FSK signals to be coherently orthogonal.

As shown in Figure 18-2, MSK modulation has two possible frequencies over any symbol interval, which differs in frequency by half the bit rate as follows:

$$f_2 - f_1 = \frac{1}{2} T_b$$

This is the smallest frequency difference that allows two signals to be orthogonal.

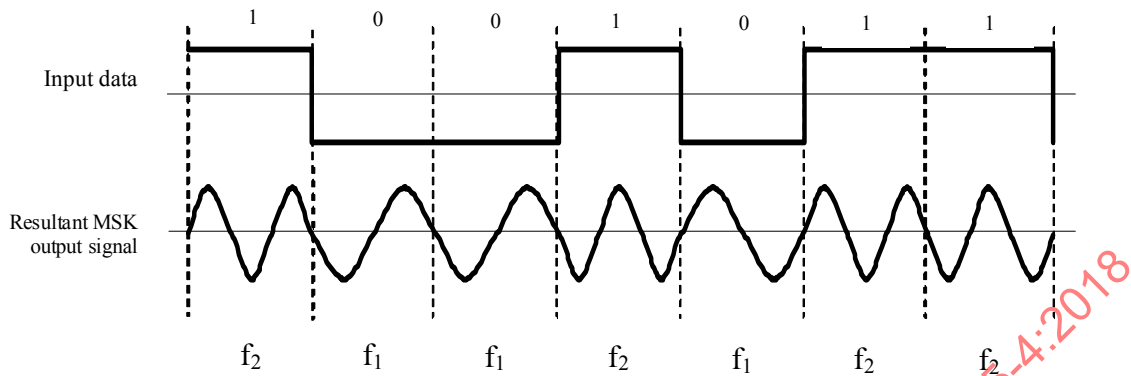


Figure 18-2—Example signal using MSK modulation in the time domain

## 18.5 MSK PHY requirements

### 18.5.1 Operating frequency range

The MSK PHY specifies the following two optional frequency bands:

- 433.05 MHz to 434.79 MHz
- 2400 MHz to 2483 MHz

### 18.5.2 Transmit PSD mask

The PSD mask for the MSK PHY is shown in Table 18-5.

The transmitted spectral products shall be less than the limits specified in Table 18-5. 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  $\pm 600$  kHz,  $\pm 300$  kHz, or  $\pm 100$  kHz of the carrier frequency (relative to data rate).

Table 18-5—MSK PHY transmit PSD limit

| Frequency band | Data rate       | Frequency             | Relative limit | Absolute limit |
|----------------|-----------------|-----------------------|----------------|----------------|
| 433 MHz        | 31.25 ksymbol/s | $ f - f_c  > 200$ kHz | −20 dB         | −20 dBm        |
|                | 100 ksymbol/s   | $ f - f_c  > 600$ kHz | −20 dB         | −20 dBm        |
|                | 250 ksymbol/s   | $ f - f_c  > 1.2$ MHz | −20 dB         | −20 dBm        |
| 2450 MHz       | 250 ksymbol/s   | $ f - f_c  > 1.2$ MHz | −20 dB         | −20 dBm        |

### 18.5.3 Symbol rate

Transmission of the symbol rate for 433 MHz band shall be 31.25 ksymbol/s, 100 ksymbol/s, or 250 ksymbol/s with an accuracy of  $\pm 300 \times 10^{-6}$ .

The transmitted symbol rate for 2450 MHz band shall be 250 ksymbol/s with an accuracy of  $\pm 300 \times 10^{-6}$ .

#### 18.5.4 Transmit center frequency tolerance

The MSK PHY shall have a transmit center frequency tolerance of  $\pm 40 \times 10^{-6}$ .

#### 18.5.5 Transmit power

The MSK PHY in the 433 MHz band shall be capable of transmitting at a power level of at least  $-3$  dBm.

The MSK PHY in the 2450 MHz band shall be capable of transmitting at a power level of at least  $-13$  dBm.

#### 18.5.6 Receiver maximum input level of desired signal

The MSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement as described in 10.1.7.

#### 18.5.7 Modulation frequency deviation tolerance

Modulation frequency tolerance is measured as a percentage of maximum frequency deviation  $\Delta f$ .

Modulation frequency deviation shall be constrained within  $\pm 30\%$  of maximum frequency deviation  $\Delta f$  as defined in 18.4.3.

#### 18.5.8 Zero crossing tolerance

All zero crossing shall be constrained within  $\pm 12.5\%$  of symbol time.

## 19. LRP UWB PHY specification

### 19.1 Overview

The LRP UWB PHY waveform is based upon an impulse radio signaling scheme using band-limited data pulses. It consists of three frequency channels and occupies the spectrum from 6.2896 GHz to 9.1856 GHz. A combination of on-off keying (OOK) modulation or pulse position modulation (PPM) is used to support both coherent and noncoherent receivers using a common signaling scheme. Either OOK or PPM are used to modulate the symbols, as defined by the mode. Symbols are composed of one or more active bursts of UWB pulses. The various data rates are supported through the use of variable-length bursts.

The LRP UWB PHY supports the following three transmission modes:

- Base mode, for highest data rate
- Extended mode, for moderate data rate but improved sensitivity
- Long-range mode, for best sensitivity

All transmit modes are optional, but all modes shall be implemented in the receiver and operational concurrently. Active RFID systems are often simplex systems so mandatory modes are not defined for the PHY but separately for the transmitter (RFD-TX) and receiver (RFD-RX).

The PHY has different characteristics depending on the transmission mode. These characteristics are defined for each mode separately as shown in Table 19-1. Otherwise, the characteristics of the PHY are independent of transmission mode.

**Table 19-1—Signaling modes and data rates for LRP UWB PHY**

| Mode            | PRF (MHz) | DataRate as used in MCPS-DATA primitives | Data rate  | Modulation |
|-----------------|-----------|------------------------------------------|------------|------------|
| Long-range mode | 2.0       | 1                                        | 31.25 kb/s | PPM        |
| Extended mode   | 1.0       | 2                                        | 250 kb/s   | OOK        |
| Base mode       | 1.0       | 3                                        | 1 Mb/s     | OOK        |

### 19.2 LRP UWB PHY symbol structure

In base mode, the LRP UWB PHY symbol consists of presence/absence of pulses in 1 MHz PRF sequence.

In extended mode, the LRP UWB PHY symbol consists of presence/absence of pulses in 1 MHz PRF sequence generated by convolution code with octal generators (5,7,7,7).

In long-range mode, the LRP UWB PHY symbol consists of Manchester-encoded groups of 64 pulses (32 on, 32 off) in 2 MHz PRF sequence.

#### 19.2.1 Base mode LRP UWB PHY symbol structure

In the base mode of LRP UWB modulation scheme, each symbol carries one bit of information. The base mode operates at 1 chip per symbol and with a PRF of 1 MHz, so the symbol time  $T_{DSYM}$  is 1  $\mu$ s and the chip time  $T_{CHIP}$  is also 1  $\mu$ s. Binary data values 0 and 1 are encoded as per Table 19-2. The data rate is thus

1 Mb/s. The pulse duration  $T_{PULSE}$  is much shorter than the symbol time. The pulse is nominally sent in the center of the chip and symbol period  $T_{DSYM}$  as shown in Figure 19-1.

**Table 19-2—Base mode LRP UWB symbol encoding**

| Binary value being encoded | Transmitted signal                                             |
|----------------------------|----------------------------------------------------------------|
| 0                          | No energy is transmitted during the 1 $\mu$ s symbol time      |
| 1                          | A single pulse is transmitted during the 1 $\mu$ s symbol time |

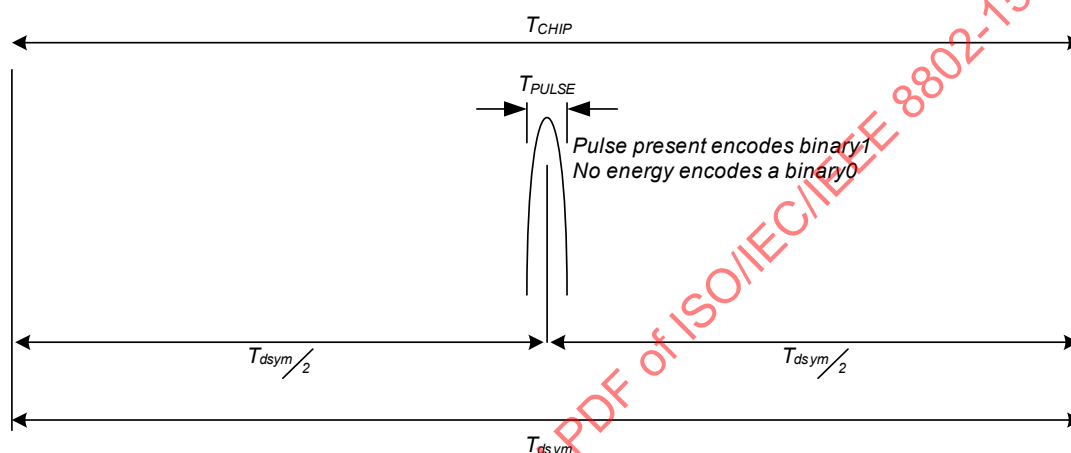


**Figure 19-1—Base mode LRP UWB PHY symbol structure**

#### 19.2.1.1 Base mode LRP UWB PHY PSDU synchronization signal

During the base mode PSDU transmission, after every 128 symbols of user data, the PHY inserts four chips of binary 1. This ensures that the receiver has enough information to retain synchronization when the user data is all zeros. These four chips/symbols are removed in the PHY and not decoded as user data.

#### 19.2.2 Extended mode LRP UWB PHY symbol structure

In the extended mode of LRP UWB modulation scheme, each symbol consists of four chips generated by a rate 1/4 convolutional code using octal generators 5,7,7,7 for  $k = 3$ , as shown in Figure 19-2.

The extended mode receiver may employ a relatively simple Viterbi decoder with hard or soft decisions to make use of the coding gain afforded by the transmitter convolution code.

Extended mode employs a PRF of 1 MHz with a rate 1/4 code giving a symbol time of 4  $\mu$ s. The data rate is thus 250 kb/s. The pulses are nominally centered within the chip periods as shown in Figure 19-3. These four pulses are transmitted in order with pulse 1 transmitted first. The individual pulses shown are present depending on whether the pulse out value is binary 1 or binary 0 as indicated in Table 19-3.

IEEE Std 802.15.4-2015  
IEEE Standard for Low-Rate Wireless Networks

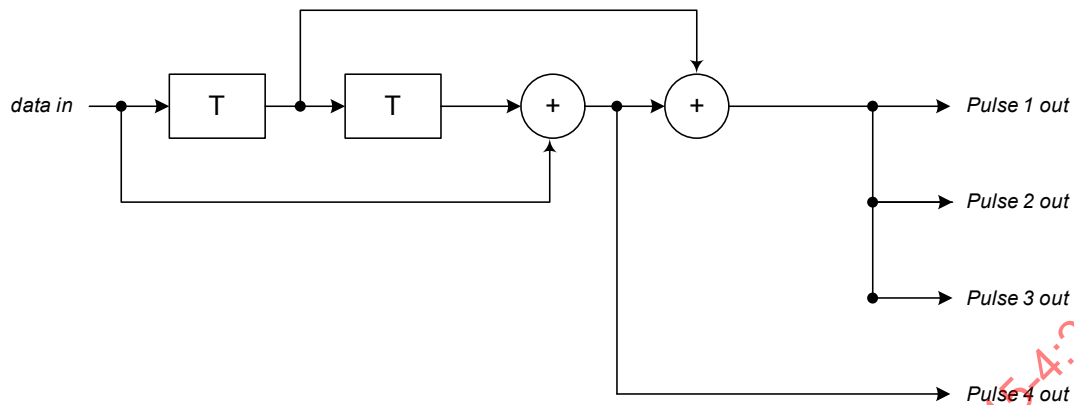


Figure 19-2—Extended mode LRP UWB PHY transmitter convolution code

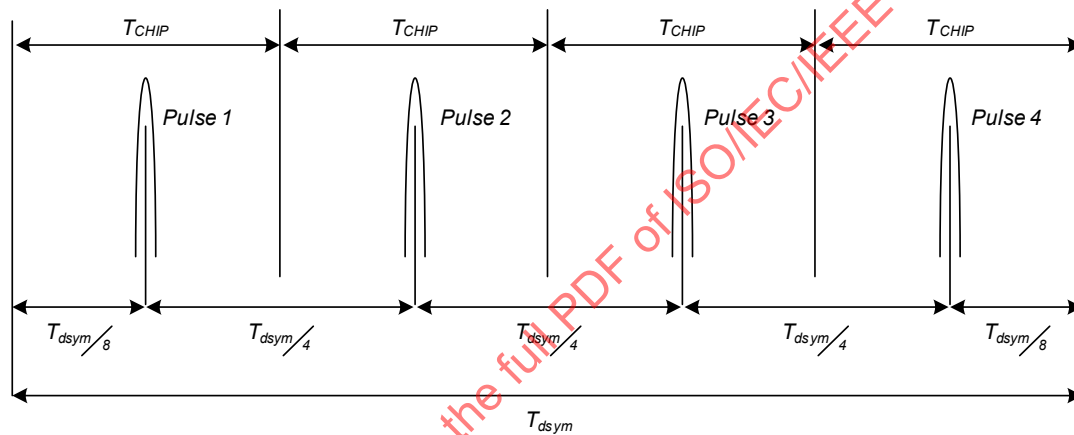


Figure 19-3—Extended mode LRP UWB PHY symbol structure

Table 19-3—Extended mode LRP UWB pulse to chip encoding

| Pulse out value | Transmitted chip                               |
|-----------------|------------------------------------------------|
| 0               | No energy is transmitted during this chip time |
| 1               | A pulse is transmitted during this chip time   |

### 19.2.2.1 Extended mode LRP UWB PHY PSDU synchronization signal

During the extended mode PSDU transmission, after every 32 symbols of user data, which is (128 chips), the PHY inserts four chips of “1” pulse. This ensures that the receiver has enough information to retain synchronization when the encoded output pulses are “0” pulses (i.e., no transmitted energy). These four chips (1 symbol of pulses) are removed in the PHY before the received pulse (Viterbi) decoding.

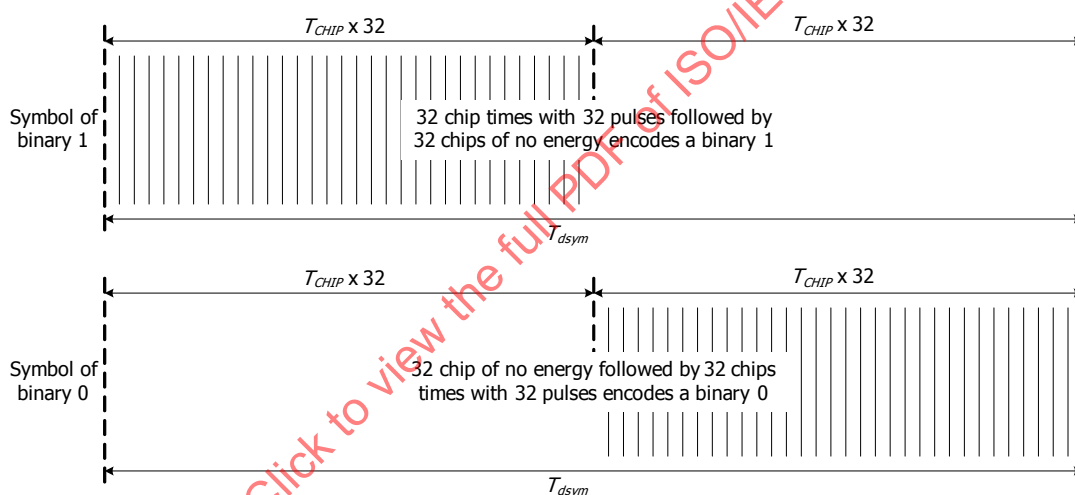


**19.2.3 Long-range mode LRP UWB PHY symbol structure**

In the long-range mode of the LRP UWB modulation scheme, each symbol encodes one bit using 64 chips at a chipping rate of 2 MHz PRF, with Manchester encoding as given in per Table 19-4. The data rate is thus 31.25 kb/s. Figure 19-4 shows this diagrammatically. When a pulse is present, it is nominally centered within the chip period.

**Table 19-4—Long-range mode LRP UWB symbol encoding**

| Binary value being encoded | Transmitted signal                                                                                                                               |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                          | The symbol period is 32 $\mu$ s. No energy is transmitted during first 16 $\mu$ s, and then in the second 16 $\mu$ s, 32 pulses are transmitted. |
| 1                          | The symbol period is 32 $\mu$ s. In the first 16 $\mu$ s, 32 pulses are transmitted, and then no energy is transmitted during second 16 $\mu$ s. |

**Figure 19-4—Long-range mode LRP UWB PHY symbol structure****19.2.3.1 Long-range mode LRP UWB PHY PSDU synchronization signal**

No additional synchronization measures are needed in long-range mode since its Manchester encoding scheme ensures that sufficient pulses are transmitted.

### 19.3 LRP UWB SHR

The SHR consists of two components: the LRP UWB SHR preamble and SFD.

#### 19.3.1 LRP UWB SHR preamble

##### 19.3.1.1 LRP UWB base mode SHR preamble

The LRP UWB base mode SHR preamble consists of a continuous stream of pulses at the base mode PRF of 1 MHz, with a length between 16 and 128.

##### 19.3.1.2 RP UWB extended mode SHR preamble

The LRP UWB extended mode SHR preamble consists of a continuous stream of pulses at the extended mode PRF of 1 MHz, with a length between 16 and 256.

##### 19.3.1.3 LRP UWB long-range mode SHR preamble

The LRP UWB long-range mode SHR preamble consists of three segments, which are transmitted in turn, as follows:

- a) A continuous stream of pulses at the long-range mode PRF of 2 MHz, with a length between 1024 and 8192 pulses.
- b) The following pulse/no-pulse sequence, transmitted at a PRF of 2 MHz where “-” represents “no pulse,” and “P” represents a pulse:  
- - - P - P - - P - - P P P - P
- c) A series of between 16 and 64 “1” symbols, transmitted as per the long-range mode PHY symbol structure defined in 19.2.3.

#### 19.3.2 LRP UWB SHR SFD

The SFD for the LRP UWB PHY is common to all modes and shall be formatted as illustrated in Figure 19-5. The modulation encoding for the SFD shall be 1 pulse per bit for both the base and extended modes. The modulation encoding for the long-range mode shall be Manchester encoded with 64 chips per bit as specified for long-range mode.

| SFD (b0–b15)        |
|---------------------|
| 0001 0100 1001 1101 |

Figure 19-5—Value of the SFD field for LRP UWB PHY

The SFD is transmitted starting from the leftmost bit (b0).

### 19.4 LRP UWB PHR

This is defined in symbols and is therefore common to base, extended, and long-range modes, with the exception of the encoding type (bits 0–2). The encoding type is defined in symbols for long-range mode and as pulses for basic and extended mode.

The PHY Header field of the LRP UWB modulation shall be formatted as illustrated in Figure 19-6.

| Bits: 0–2     | 3                | 4–9    | 10–16        | 17       | 18–20       | 21            |
|---------------|------------------|--------|--------------|----------|-------------|---------------|
| Encoding Type | Header Extension | SECDED | Frame Length | Reserved | LEIP length | LEIP Position |

**Figure 19-6—PHR format for LRP UWB PHY**

#### 19.4.1 Encoding Type field

Long-range mode is specified (and detected) by its own unique symbol mapping, defined in 19.2.3, and the use of a 2 MHz PRF. In long-range mode the Encoding Type field of the PHR is given by Table 19-5.

**Table 19-5—PHR Encoding Type field in long-range mode**

| Encoding Type value<br>b0–b2 | Meaning                                                                                                                      |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 000                          | Each symbol encodes 1 bit. Each symbol consists of 64 chips and uses Manchester encoding. This encoding is defined in 19.2.3 |
| 001 to 111                   | Reserved                                                                                                                     |

In base mode and extended mode where the PHR is sent with a chip rate of 1 MHz, the Encoding Type field of the PHR is given by Table 19-6. These three bits are encoded as per 19.2.1. Only two values are legal, 000 and 111. This allows a receiver to use all three bits in a voting scheme to determine whether it should switch to using extended mode decoding for the remainder of the PHR and the PSDU or should continue decoding the PHR and PSDU in base mode.

**Table 19-6—PHR Encoding Type field base mode and extended mode**

| Encoding Type value<br>b0–b2 | Meaning                                                                                                                                                                         |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 000                          | This value indicates the operating mode is base mode. All remaining bits in the frame continue to be encoded as per 19.2.1.                                                     |
| 111                          | This value indicates the operating mode is extended mode. All bits in the remaining fields of the PHR and PSDU are encoded as per 19.2.2.                                       |
| 001 to 110                   | ILLEGAL—These values can never be legally used. The receiver may use all three bits of the two legal values 000 and 111 in a voting scheme to decide which is actually present. |

#### 19.4.2 Header Extension field

The PHR Header Extension bit shall be set to zero upon transmission. If a PPDU is received with the PHR Header Extension bit set, the device shall discard the PPDU.

**19.4.3 SECDED field**

The SECDED field is a simple Hamming block code that enables the correction of a single error and the detection of two errors at the receiver. The SECDED field is computed as follows:

$$b_9 = \text{XOR}(b_{21}, b_{18}, b_{19}, b_{20}, b_{17})$$

$$b_8 = \text{XOR}(b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16})$$

$$b_7 = \text{XOR}(b_1, b_2, b_3, b_{13}, b_{14}, b_{15}, b_{16}, b_{20}, b_{17})$$

$$b_6 = \text{XOR}(b_0, b_2, b_3, b_{11}, b_{12}, b_{15}, b_{16}, b_{18}, b_{19})$$

$$b_5 = \text{XOR}(b_0, b_1, b_3, b_{10}, b_{12}, b_{14}, b_{16}, b_{21}, b_{19}, b_{17})$$

$$b_4 = \text{XOR}(b_0, \dots, b_3, b_5, \dots, b_{21})$$

**19.4.4 Frame Length field**

The Frame Length field is an unsigned integer that shall be set to the length of the PSDU in octets. The Frame Length field shall be transmitted MSB first.

**19.4.5 LEIP Length field**

This gives the length of the LEIP in pulses. The meaning of this field is defined in Table 19-7.

**Table 19-7—PHR LEIP Length field meaning**

| LEIP Length field value<br>(b18 b19 b20) | LEIP sequence length |
|------------------------------------------|----------------------|
| 000                                      | Not present.         |
| 001                                      | 16 pulses            |
| 010                                      | 64 pulses            |
| 011                                      | 128 pulses           |
| 100                                      | 192 pulses           |
| 101                                      | 256 pulses           |
| 110                                      | 512 pulses           |
| 111                                      | 1024 pulses          |

**19.4.6 LEIP Position field**

The LEIP Position field specifies the position of the optional LEIP sequence. The LEIP Position field only applies if the LEIP Length field indicates that the LEIP sequence length is non-zero, in which case the meaning of this bit is then as defined in Table 19-8. When the LEIP Length field indicates that the LEIP sequence is not present, then the LEIP Position field is reserved.

**Table 19-8—PHR LEIP position bit meaning**

| LEIP Position field value | Meaning                 |
|---------------------------|-------------------------|
| 0                         | The LEIP is delayed     |
| 1                         | The LEIP is not delayed |

When the LEIP sequence is delayed, it is delayed by *aLeipDelayTime* from start of SFD. The LEIP then starts on the first chipping interval after the delay. Where the PSDU is of sufficient length that it has not ended by this time, then the LEIP is deferred to start in the chipping interval immediately following the final chipping interval being used for the PSDU.

When the LEIP is not delayed, the LEIP starts in the chipping interval immediately following the final chipping interval being used for the PSDU.

### 19.5 LRP UWB PSDU

In base mode the PSDU is encoded as per 19.2.1; in extended mode the PSDU is encoded as per 19.2.2; and in long-range mode the PSDU is encoded as per 19.2.3.

### 19.6 LRP UWB location enhancing information postamble

The LEIP consists of a sequence of UWB pulses. The PRF of the LEIP pulse sequence is as follows:

- 1 MHz in the LRP UWB base and extended modes
- 2 MHz in the long-range mode

Information in the PHY header, as defined in 19.4, indicates whether or not the LEIP is appended to a transmitted packet.

If the information in the PHY header indicates that the LEIP is appended to the transmitted packet, further information in the PHY header indicates when the first LEIP pulse occurs immediately. It occurs either

- Immediately after the end of the PSDU, or
- *LeipDelayTime* after the start of the SHR SFD.

The length of the LEIP (in pulses, at the appropriate rate) is defined in Table 19-7 by the PHR LEIP length field.

### 19.7 LRP UWB transmitter specification

#### 19.7.1 Pulse shape

The LRP UWB PHY shall employ an impulse transmitter that instantaneously produces an ultra wideband frequency response. There are no constraints on the specific pulse shape providing that the pulse shall comply with the Transmit PSD Mask defined in 19.7.3.

**19.7.2 Pulse timing**

The transmission time of any individual pulse shall not drift more than 11 ns from its nominal transmission time during a 128 symbol period over the specified operating temperature range of the device.

In order to avoid long sequences of zeros driving the need for high-quality clocks, the symbol structure in the base and extended modes includes a periodic sync marker as described in 19.2.1.1 and 19.2.2.1. No additional sync marker is required in the long-range mode.

**19.7.3 Transmit PSD mask**

The transmitter shall operate with a power spectral density contained by one of three PSD masks defined in Table 19-9 and shown in Figure 19-7. The permitted spectral density is defined in dBr relative to the maximum spectral density of the signal, and shall be made using a 1 MHz resolution bandwidth and a 1 MHz video bandwidth. Additionally, the upper  $-10$  dBr point of the transmitter PSD shall be at least 200 MHz above a nominal frequency,  $f_n$ , and the lower  $-10$  dBr point shall be at most 200 MHz below the same nominal frequency.

**Table 19-9—LRP UWB PHY PSD mask**

| Band number | $f_n$ (MHz) | Frequency (MHz)        | PSD limit (dBr) |
|-------------|-------------|------------------------|-----------------|
| 0           | 6489.6      | $< 5624.32$            | $-18$           |
|             |             | $5624.32$ to $5786.56$ | $-10$           |
|             |             | $5786.56$ to $7192.64$ | $0$             |
|             |             | $7192.64$ to $7354.88$ | $-10$           |
|             |             | $> 7354.88$            | $-18$           |
| 1           | 6988.8      | $< 6090.24$            | $-18$           |
|             |             | $6090.24$ to $6165.12$ | $-10$           |
|             |             | $6165.12$ to $8311.68$ | $0$             |
|             |             | $8311.68$ to $8386.56$ | $-10$           |
|             |             | $> 8386.56$            | $-18$           |
| 2           | 7987.2      | $< 6922.24$            | $-18$           |
|             |             | $6922.24$ to $7121.92$ | $-10$           |
|             |             | $7121.92$ to $8852.48$ | $0$             |
|             |             | $8852.48$ to $9052.16$ | $-10$           |
|             |             | $> 9052.16$            | $-18$           |

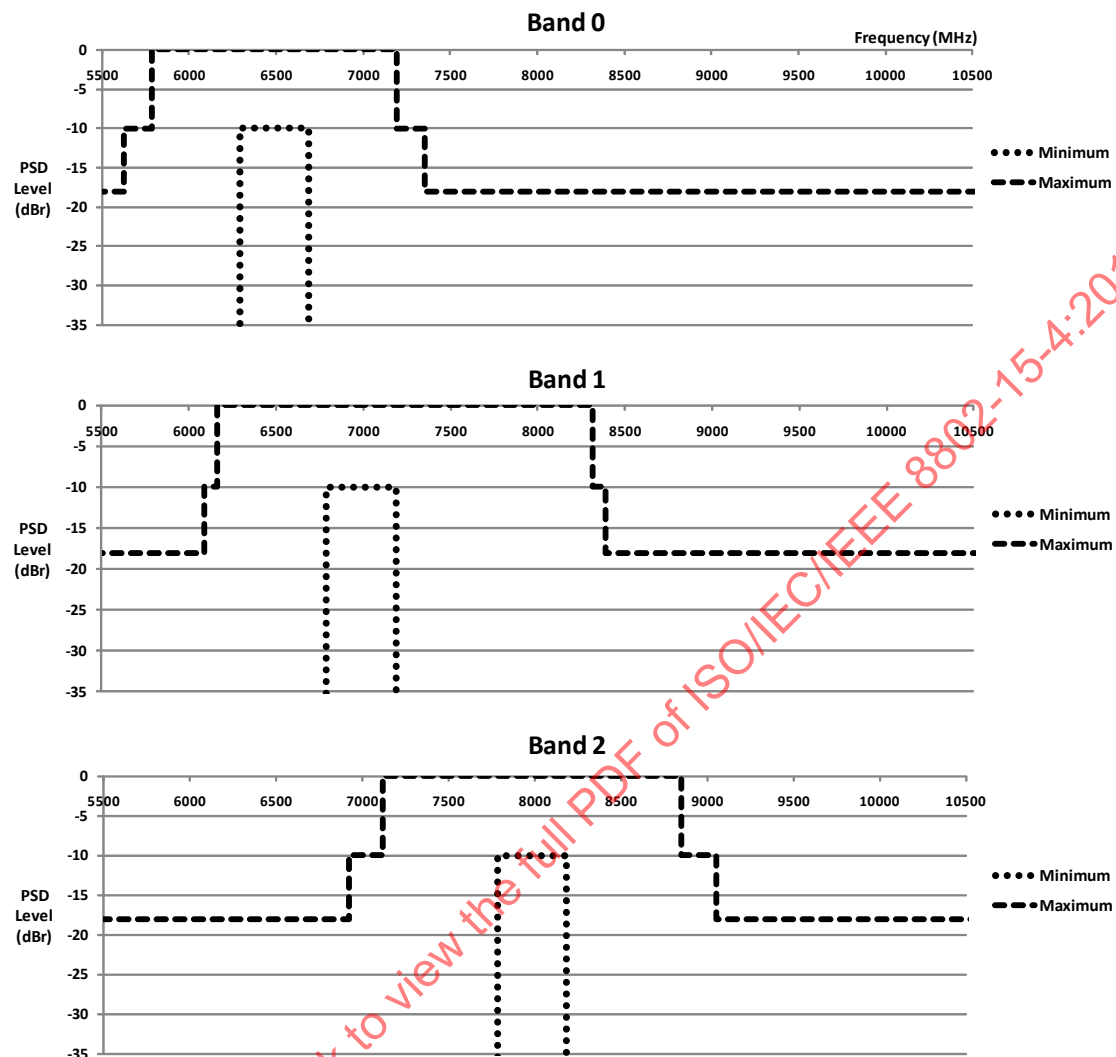


Figure 19-7—Low rate PRF PHY PSD mask

## 19.8 LRP UWB receiver specification

The receiver shall support each mode of operation: base mode, extended mode and long-range mode.

Receiving devices shall be capable of receiving at least one channel allowed by regulations for the region(s) in which the device operates.

## 20. SUN FSK PHY

### 20.1 Introduction

A SUN device shall support the SUN FSK PHY, allowing MPM signaling utilizing the CSM.

For the SUN FSK PHY, the symbol period used for MAC and PHY timing parameters, shown in Table 20-1, shall be the symbol period of operating mode #1 specified in Table 20-6 and Table 20-7.

**Table 20-1—SUN FSK symbol period used for MAC and PHY timing parameters**

| Frequency band (MHz) | symbol period used for MAC and PHY timing parameters (μs) |
|----------------------|-----------------------------------------------------------|
| 169.400–169.475      | 208+1/3                                                   |
| 450–470              | 104+1/6                                                   |
| 470–510              | 20                                                        |
| 779–787              | 20                                                        |
| 863–870              | 20                                                        |
| 896–901              | 100                                                       |
| 901–902              | 100                                                       |
| 902–928              | 20                                                        |
| 917–923.5            | 20                                                        |
| 928–960              | 100                                                       |
| 920–928              | 20                                                        |
| 1427–1518            | 100                                                       |
| 2400–2483.5          | 20                                                        |

For the purposes of calculating the Ack frame timing required in 6.7.5, the default length of the Preamble field shall be 8 octets and mode switch shall not be used for Ack frames.

### 20.2 PPDU format for SUN FSK

The SUN FSK PPDU shall support the format shown in Figure 20-1 and may support the format shown in Figure 20-2 if mode switch is enabled.

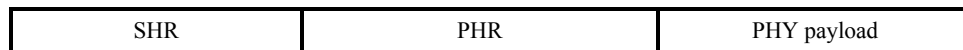


**Figure 20-1—Format of the SUN FSK PPDU (without mode switch)**

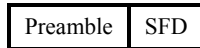


**Figure 20-2—Format of the SUN FSK mode switch PPDU**



**20.2.1 SHR field format**

The SHR field shall be formatted as illustrated in Figure 20-3.

**Figure 20-3—Format of the SHR****20.2.1.1 Preamble field**

The Preamble field shall contain *phyFskPreambleLength* (as defined in 11.3) multiples of the 8-bit sequence “01010101” for 2-FSK. The Preamble field shall contain *phyFskPreambleLength* multiples of the 16-bit sequence “0111 0111 0111 0111” for 4-FSK.

**20.2.1.2 SFD**

The SFD for 2-FSK shall be a 2-octet sequence selected from the values shown in Table 20-2. The SFD for 4-FSK shall be a 4-octet sequence selected from the values shown in Table 20-3. Devices that do not support FEC shall support the SFD associated with uncoded format and a value of zero for the PIB attribute *phySunFskSfd*, as defined in 11.3; these devices may also support the SFD associated with uncoded format and a value of one for the PIB attribute *phySunFskSfd*. Devices that support FEC shall support both SFDs associated with a value of zero for the PIB attribute *phySunFskSfd*; these devices may additionally support both SFDs associated with a value of one for the PIB attribute *phySunFskSfd*.

The SFD is transmitted starting from the leftmost bit (i.e., starting with b0).

**Table 20-2—SUN FSK PHY SFD values for 2-FSK**

|                         | SFD value for coded format<br>(b0–b15) | SFD value for uncoded format<br>(b0–b15) |
|-------------------------|----------------------------------------|------------------------------------------|
| <i>phySunFskSfd</i> = 0 | 0110 1111 0100 1110                    | 1001 0000 0100 1110                      |
| <i>phySunFskSfd</i> = 1 | 0110 0011 0010 1101                    | 0111 1010 0000 1110                      |

**Table 20-3—SUN FSK PHY SFD values for 4-FSK**

|                         | SFD value for coded format<br>(b0–b31)  | SFD value for uncoded format<br>(b0–b31) |
|-------------------------|-----------------------------------------|------------------------------------------|
| <i>phySunFskSfd</i> = 0 | 0111 1101 1111 1111 0111 0101 1111 1101 | 1101 0111 0101 0101 0111 0101 1111 1101  |
| <i>phySunFskSfd</i> = 1 | 0111 1101 0101 1111 0101 1101 1111 0111 | 0111 1111 1101 1101 0101 0101 1111 1101  |

### 20.2.2 PHR field format

The format of the PHR is shown in Figure 20-4.

| Bits: 0     | 1–2      | 3        | 4              | 5–15         |
|-------------|----------|----------|----------------|--------------|
| Mode Switch | Reserved | FCS Type | Data Whitening | Frame Length |

**Figure 20-4—Format of the PHR for SUN FSK**

The Mode Switch field shall be set to zero, indicating that the entire packet shall be transmitted at a single data rate and using a single modulation scheme.

The FCS Type field indicates the length of the FCS field described in 7.2.10 that is included in the MPDU. The FCS Type field shall be set to zero for a 4-octet FCS and shall be set to 1 for a 2-octet FCS.

The Data Whitening field indicates whether data whitening of the PSDU is used upon transmission. The Data Whitening field shall be set to one when data whitening is used and shall be set to zero otherwise.

The Frame Length field is an unsigned integer that shall be set to the total number of octets contained in the PSDU (prior to FEC encoding, if enabled). The Frame Length field shall be transmitted MSB first.

### 20.2.3 Mode Switch PHR

The use of the Mode Switch mechanisms described in this subclause is deprecated. Consequently, this clause may be removed in a later revision of the standard.

The format of the Mode Switch PHR is shown in Figure 20-5.

| Bits: 0     | 1–2                   | 3            | 4–10     | 11–14    | 15           |
|-------------|-----------------------|--------------|----------|----------|--------------|
| Mode Switch | Mode Switch Parameter | New Mode FEC | New Mode | Checksum | Parity Check |

**Figure 20-5—Format of the Mode Switch PHR for SUN FSK**

The Mode Switch field shall be set to one, indicating that a mode switch shall occur. The mode of the next PPDU transmitted (i.e., the new mode PPDU) shall be as described by the remaining fields contained in the PHR in Figure 20-5. If the new mode is SUN FSK, the new mode PPDU is the same as Figure 20-1 except that the preamble and SFD are optional. For SUN OFDM, the new mode PPDU has the same format as Figure 21-1. If the new mode is SUN O-QPSK, the new mode PPDU is the same as Figure 22-1.

The Mode Switch Parameter field is the index of the entry in the *phyModeSwitchParameterEntries* array, as described in Table 11-2, that defines the mode switch parameters to be used, as described in Table 11-3. If the Mode Switch Parameter field indicates an unsupported entry in the *phyModeSwitchParameterEntries* array of the receiver, the receiver shall discard the packet and remain in the present PHY mode.

The New Mode FEC field specifies whether the packet following the mode switch PPDU is transmitted using FEC. A value of zero indicates that the new mode packet is transmitted without FEC, and a value of one indicates that it is transmitted with FEC. If the new mode packet has an SFD and, therefore, packet coding information, as described in 20.3.4, the SFD shall override the value of the New Mode FEC field.

The New Mode field is formatted as shown in Figure 20-6.

| Bits: 0 | 1–2               | 3–6  |
|---------|-------------------|------|
| Page    | Modulation Scheme | Mode |

**Figure 20-6—Format of the New Mode field**

The Page field shall be set to zero to indicate channel page nine or set to one to indicate channel page ten.

The Modulation Scheme field indicates the modulation scheme, as described in Table 20-4, when *phyCurrentChannel* (as defined in 11.3) equals seven; when *phyCurrentChannel* equals eight, the Modulation Scheme field shall be set to zero upon transmission and ignored upon reception.

**Table 20-4—Modulation Scheme field values**

| Field value | Description |
|-------------|-------------|
| 0           | FSK         |
| 1           | OFDM        |
| 2           | O-QPSK      |
| 3           | Reserved    |

The Mode field specifies the new mode of operation. When the Page field is zero (channel page nine), the interpretation of the Mode field is based on the modulation scheme:

- If the modulation scheme is FSK, the Mode field is a bitmap where a bit is set to one if the operating mode given in Table 20-5 is supported and is set to zero otherwise.

**Table 20-5—Mode field values**

| Bit | Description       |
|-----|-------------------|
| 0   | Operating mode #1 |
| 1   | Operating mode #2 |
| 2   | Operating mode #3 |
| 3   | Operating mode #4 |

- If the modulation scheme is not FSK, the Mode field shall be set to zero upon transmission and ignored upon reception. The corresponding data rates are specified in the PHR of the new mode PPDU.

When the Page field is one (channel page ten), the new PHY mode is defined by the SUN FSK Generic PHY mechanism.

The generator polynomial for the Checksum field is a Bose Chaudhuri Hocquenghem (BCH) code. The checksum for the BCH(15,11) code is calculated as follows:

$$G(x) = 1 + x + x^4$$

The Parity Check field provides error detection for the mode switch PPDU. Its value is calculated using the first 11 bits from the PHR, (b0, b1, ... b10), using the following equation:

$$\text{Parity Check} = b0 \oplus b1 \oplus b2 \oplus b3 \oplus b4 \oplus b5 \oplus b6 \oplus b7 \oplus b8 \oplus b9 \oplus b10$$

where  $\oplus$  is modulo-2 addition (addition over GF(2)). The combination of the BCH(15,11) code and one parity bit allows for the achievement of single error correction and double error detection over the first 11 bits of the PHR.

If the receiving device receives a PHR with the Mode Switch field set to one, it first performs the BCH calculation over the first 11 bits of the PHR. If the resulting checksum is valid, and the Mode Switch field is still set to one after error correction, a parity check using the Parity Check field is performed. If the result of the parity check is valid, the receiving device processes the mode switch and decodes the subsequent PPDU. If the result of the parity check is invalid, or if the Mode Switch field is set to zero after the error correction, the receiver terminates the receive procedure.

#### 20.2.4 PHY Payload field

The PHY Payload field carries the PSDU, encoded as described in 20.3.1.

### 20.3 Modulation and coding for SUN FSK

The modulation for the SUN FSK PHY is either a 2- or a 4-level FSK that meets the transmit spectral mask, as defined in 20.6.6.

Table 20-6 shows the modulation and channel parameters for the standard-defined PHY operating modes for the 169 MHz, 450 MHz, 470 MHz, 863 MHz, 896 MHz, 901 MHz, 915 MHz, 928 MHz, 1427 MHz, and 2450 MHz bands. A device shall support operating mode #1 and may additionally support operating modes #2 and #3.

**Table 20-6—SUN FSK modulation and channel parameters<sup>a</sup>**

| Frequency band (MHz) | Parameter             | Operating mode #1 | Operating mode #2 | Operating mode #3 |
|----------------------|-----------------------|-------------------|-------------------|-------------------|
| 169.400–169.475      | Data rate (kb/s)      | 4.8               | 2.4               | 9.6               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 4-FSK             |
|                      | Modulation index      | 0.5               | 2.0               | 0.33              |
|                      | Channel spacing (kHz) | 12.5              | 12.5              | 12.5              |
| 450–470              | Data rate (kb/s)      | 9.6               | 4.8               | —                 |
|                      | Modulation            | 4-FSK             | 2-FSK             | —                 |
|                      | Modulation index      | 0.33              | 1.0               | —                 |
|                      | Channel spacing (kHz) | 12.5              | 12.5              | —                 |

**Table 20-6—SUN FSK modulation and channel parameters<sup>a</sup> (continued)**

| Frequency band (MHz) | Parameter             | Operating mode #1 | Operating mode #2 | Operating mode #3 |
|----------------------|-----------------------|-------------------|-------------------|-------------------|
| 470–510              | Data rate (kb/s)      | 50                | 100               | 200               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 4-FSK             |
|                      | Modulation index      | 1.0               | 1.0               | 0.33              |
|                      | Channel spacing (kHz) | 200               | 400               | 400               |
| 779–787              | Data rate (kb/s)      | 50                | 100               | 200               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 4-FSK             |
|                      | Modulation index      | 1.0               | 1.0               | 0.33              |
|                      | Channel spacing (kHz) | 200               | 400               | 400               |
| 863–870              | Data rate (kb/s)      | 50                | 100               | 200               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 4-FSK             |
|                      | Modulation index      | 1.0               | 1.0               | 0.33              |
|                      | Channel spacing (kHz) | 200               | 400               | 400               |
| 896–901              | Data rate (kb/s)      | 10                | 20                | 40                |
|                      | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                      | Modulation index      | 0.5               | 0.5               | 0.5               |
|                      | Channel spacing (kHz) | 12.5              | 12.5              | 12.5              |
| 901–902              | Data rate (kb/s)      | 10                | 20                | 40                |
|                      | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                      | Modulation index      | 0.5               | 0.5               | 0.5               |
|                      | Channel spacing (kHz) | 12.5              | 12.5              | 12.5              |
| 902–928              | Data rate (kb/s)      | 50                | 150               | 200               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                      | Modulation index      | 1.0               | 0.5               | 0.5               |
|                      | Channel spacing (kHz) | 200               | 400               | 400               |
| 917–923.5            | Data rate (kb/s)      | 50                | 150               | 200               |
|                      | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                      | Modulation index      | 1.0               | 0.5               | 0.5               |
|                      | Channel spacing (kHz) | 200               | 400               | 400               |
| 928–960 <sup>b</sup> | Data rate (kb/s)      | 10                | 20                | 40                |
|                      | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                      | Modulation index      | 0.5               | 0.5               | 0.5               |
|                      | Channel spacing (kHz) | 25                | 25                | 25                |

**Table 20-6—SUN FSK modulation and channel parameters<sup>a</sup> (continued)**

| Frequency band (MHz)   | Parameter             | Operating mode #1 | Operating mode #2 | Operating mode #3 |
|------------------------|-----------------------|-------------------|-------------------|-------------------|
| 1427–1518 <sup>b</sup> | Data rate (kb/s)      | 10                | 20                | 40                |
|                        | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                        | Modulation index      | 0.5               | 0.5               | 0.5               |
|                        | Channel spacing (kHz) | 25                | 25                | 25                |
| 2400–2483.5            | Data rate (kb/s)      | 50                | 150               | 200               |
|                        | Modulation            | 2-FSK             | 2-FSK             | 2-FSK             |
|                        | Modulation index      | 1.0               | 0.5               | 0.5               |
|                        | Channel spacing (kHz) | 200               | 400               | 400               |

<sup>a</sup>Data rates shown are over-the-air data rates (the data rate transmitted over the air regardless of whether the FEC is enabled).

<sup>b</sup>Noncontiguous.

Table 20-7 shows the modulation and channel parameters for the standard-defined PHY operating modes for the 920 MHz Japanese bands. For these bands, a device shall support both operating modes #1 and #2 and may additionally support operating modes #3 and #4.

**Table 20-7—SUN FSK modulation and channel parameters for Japanese band<sup>a</sup>**

| Frequency band (MHz) | Parameter                          | Operating mode #1 | Operating mode #2 | Operating mode #3 | Operating mode #4 |
|----------------------|------------------------------------|-------------------|-------------------|-------------------|-------------------|
| 920–928              | Data rate (kb/s)                   | 50                | 100               | 200               | 400               |
|                      | Modulation                         | 2-FSK             | 2-FSK             | 2-FSK             | 4-FSK             |
|                      | Modulation index                   | 1.0               | 1.0               | 1.0               | 0.33              |
|                      | Channel spacing (kHz) <sup>b</sup> | 200               | 400               | 600               | 600               |

<sup>a</sup>Data rates shown are over-the-air data rates (the data rate transmitted over the air regardless of whether the FEC is enabled).

<sup>b</sup>Channel separation of 200 kHz is used. Channel spacing shows bundling of 200 kHz channels.

In addition to the standard-defined PHY operating modes, the SUN FSK PHY may support an SUN FSK Generic PHY mechanism, which enables the use of a broader set of data rates and PHY parameters to describe a PHY mode. The set of PHY operating mode parameters is defined by the SUN FSK Generic PHY IE, as defined in 7.4.4.11.

### 20.3.1 Reference modulator

The functional block diagram of the reference modulator in Figure 20-7 is provided as a reference for specifying the SUN FSK PHY data flow processing functions. Each bit shall be processed using the bit order rules defined in 20.2.

When FEC is enabled, the PHR and PSDU shall be processed for coding as a single block of data, as described in 20.3.4. When data whitening is enabled, the scrambling shall be only applied over the PSDU, as described in 20.4.

All fields in the PPDU shall use the same symbol rate and modulation order, unless otherwise specified elsewhere in this standard.

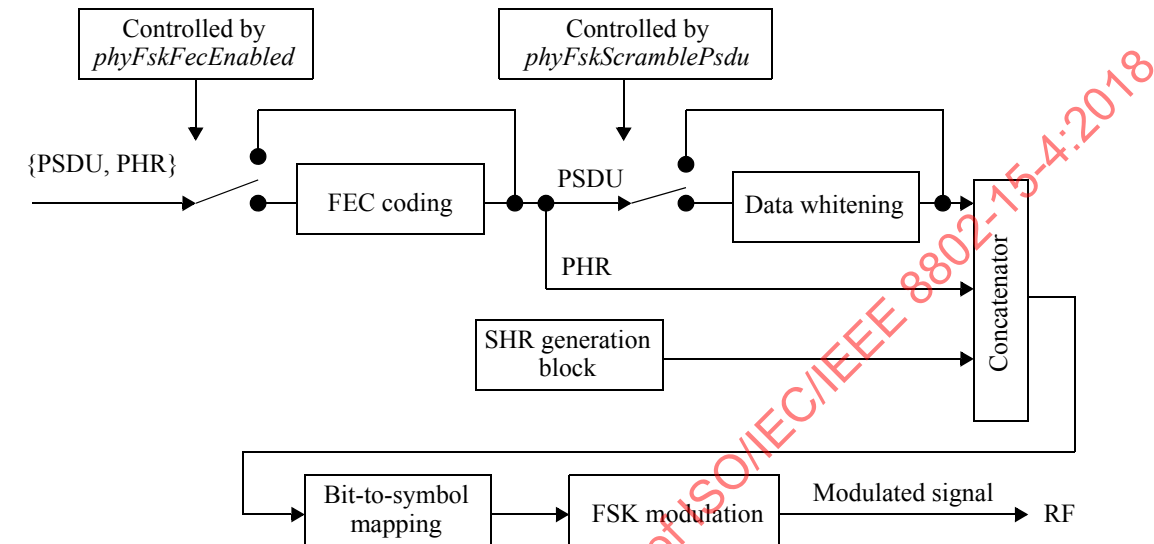


Figure 20-7—SUN FSK reference modulator

20.3.2 Bit-to-symbol mapping

The nominal frequency deviation,  $\Delta f$ , shall be as follows:

$$\left( \frac{\text{symbol rate} \times \text{modulation index}}{2} \right)$$

The symbol encoding for 2-FSK is shown in Table 20-8 and for 4-FSK in Table 20-9, where the frequency deviation,  $f_{\text{dev}}$ , is equal to  $\Delta f$  for 2-FSK and is equal to  $3 \times \Delta f$  for 4-FSK. For 4-FSK modulation, two bits shall be mapped to four frequency deviation levels for the PHR and PHY Payload field. The SHR shall be encoded in the lowest ( $-f_{\text{dev}}$ ) and the highest ( $+f_{\text{dev}}$ ) frequency deviations. The symbol rate shall be the same for the entire PPDU.

Table 20-8—SUN 2-FSK symbol encoding

| Symbol (binary) | Frequency deviation |
|-----------------|---------------------|
| 0               | $-f_{\text{dev}}$   |
| 1               | $+f_{\text{dev}}$   |

**Table 20-9—SUN 4-FSK symbol encoding**

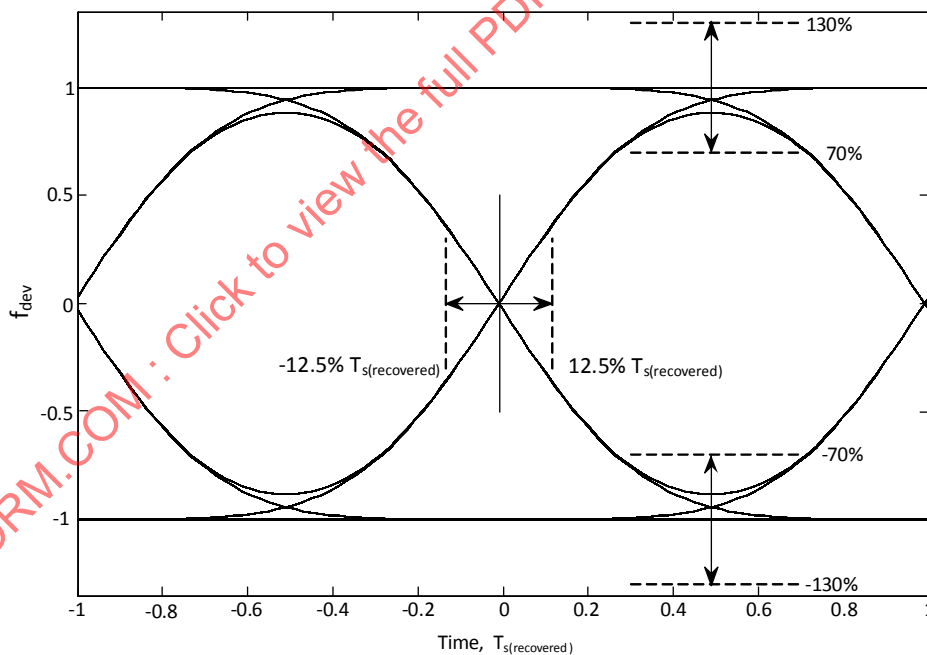
| Symbol (binary) | Frequency deviation   |
|-----------------|-----------------------|
| 01              | $-f_{\text{dev}}$     |
| 00              | $-f_{\text{dev}} / 3$ |
| 10              | $+f_{\text{dev}} / 3$ |
| 11              | $+f_{\text{dev}}$     |

**20.3.3 Modulation quality**

Modulation quality shall be measured by observing the frequency deviation tolerance and the zero crossing tolerance of the eye diagram caused by a PN9 sequence of length 511 bits.

**20.3.3.1 Frequency deviation tolerance**

Modulation frequency tolerance is measured as a percentage of the frequency deviation,  $f_{\text{dev}}$ , dictated by the modulation index. In the case of 2-FSK, the measured frequency deviation,  $f$ , at  $T_{s(\text{recovered})} / 2$  shall be constrained to the range  $70\% f_{\text{dev}} < |f| < 130\% f_{\text{dev}}$ , as shown in Figure 20-8, where  $T_{s(\text{recovered})}$  is the recovered symbol time, corrected for the symbol timing offset. In the case of 4-FSK, the measured frequency deviation,  $f$ , at  $T_{s(\text{recovered})} / 2$  shall be constrained to the range  $8\% f_{\text{dev}} < |f| < 58\% f_{\text{dev}}$  for the inner levels and  $75\% f_{\text{dev}} < |f| < 125\% f_{\text{dev}}$  for the outer levels, as shown in Figure 20-9.

**Figure 20-8—Eye diagram for 2-FSK**



### 20.3.3.2 Zero crossing tolerance

In the case of 2-FSK, the excursions for the zero crossings for all trajectories of the eye diagram shall be constrained to within  $\pm 12.5\%$  rms of the symbol time  $T_s$ , as shown in Figure 20-8. In the case of 4-FSK, the excursions for the zero crossings for all trajectories of the eye diagram shall be constrained to within  $\pm 30\%$  of the symbol time  $T_s$ , as shown in Figure 20-9.

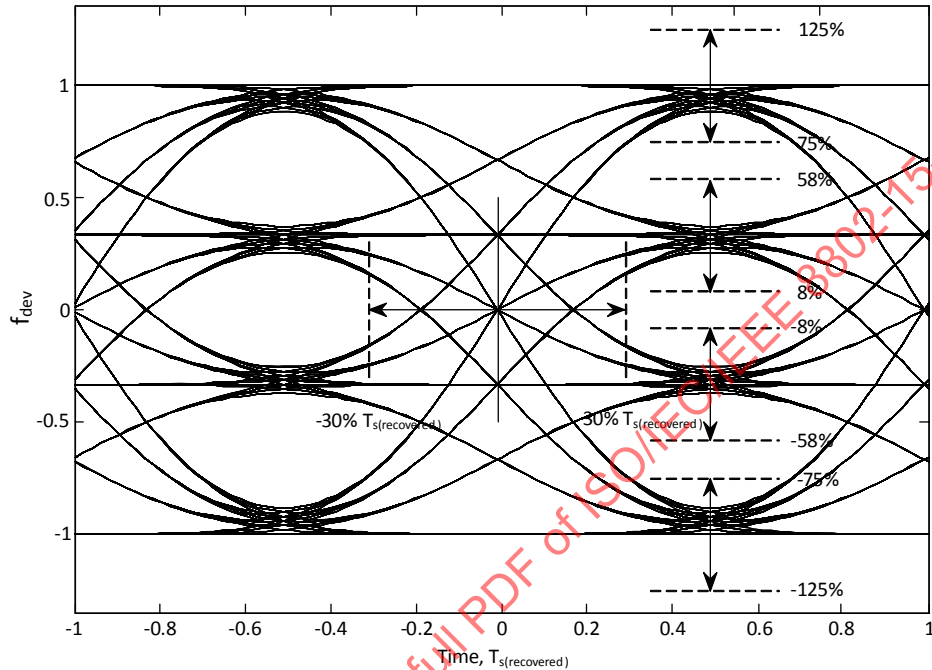


Figure 20-9—Eye diagram for 4-FSK

### 20.3.4 FEC

FEC is optional. If the SFD indicates that FEC is used, as described in Table 20-2, then the FEC is applied to the PHR and PSDU as a single block of data.

Two types of FEC may be applied: a recursive and systematic code (RSC) or a nonrecursive and nonsystematic code (NRNSC). The use of RSC or NRNSC coding shall be controlled by the PIB attribute *phyFskFecScheme*, as defined in 11.3.

When the SFD value indicates a coded packet, FEC shall be employed on the PHR and PSDU bits, applying either a  $1/2$ -rate systematic or nonsystematic convolution coding with constraint length  $K = 4$ , and using the following two generator polynomials:

$$G_0(x) = 1 + x + x^2 + x^3$$

$$G_1(x) = 1 + x^2 + x^3$$

The total number of bits to be encoded,  $N$ , is obtained by summing up the size of the PHR ( $L_{PHR}$ ), the length of the PSDU ( $L_{PSDU}$  is equal to the content of the Frame Length field in Figure 20-4), the number of tail bits ( $L_{TAIL}$ ), and the number of padding bits ( $L_{PAD}$ ).  $N$  shall be computed as follows:

$$N = L_{PHR} + L_{PSDU} + L_{TAIL} + L_{PAD}$$

NOTE—The value of  $L_{PSDU}$  is zero in the case of a mode switch packet.

Immediately after encoding the PHR and PSDU, a termination sequence with length  $L_{TAIL} = 3$  bits shall be inserted into the encoder, as shown in Figure 20-10. The tail bits are required to return the encoder to the zero state.

|     |      |           |
|-----|------|-----------|
| PHR | PSDU | Tail bits |
|-----|------|-----------|

**Figure 20-10—Data block extension with tail bits prior to coding**

The value of the tail bits are dependent on the coding scheme and shall be set as shown in Table 20-10.

**Table 20-10—Tail bit pattern for the RSC and NRNSC encoders**

| Memory state<br>( $M_0$ – $M_2$ ) | Tail bits         |                     |
|-----------------------------------|-------------------|---------------------|
|                                   | RSC<br>(b0 b1 b2) | NRNSC<br>(b0 b1 b2) |
| 000                               | 000               | 000                 |
| 001                               | 100               | 000                 |
| 010                               | 110               | 000                 |
| 011                               | 010               | 000                 |
| 100                               | 111               | 000                 |
| 101                               | 011               | 000                 |
| 110                               | 001               | 000                 |
| 111                               | 101               | 000                 |

When interleaving is used in conjunction with convolutional coding, a padding sequence of  $L_{PAD}$  bits shall be further inserted into the encoder immediately after the tail bits. The padding bits are required to fill up the last interleaver buffer completely, as described in 20.3.5.  $L_{PAD}$  shall be computed as follows:

$$L_{PAD} = 5, \text{ when } \frac{L_{PHR} + L_{PSDU}}{8} \text{ is odd}$$

$$L_{PAD} = 13, \text{ when } \frac{L_{PHR} + L_{PSDU}}{8} \text{ is even}$$

Padding bit patterns should not contain a long series of ‘1’s or ‘0’s. Figure 20-11 and Figure 20-12 illustrate examples of such patterns.

|     |      |           |                              |
|-----|------|-----------|------------------------------|
| PHR | PSDU | Tail bits | 5-bit padding pattern: 01011 |
|-----|------|-----------|------------------------------|

**Figure 20-11—An example of extension with padding bits prior to encoding, when  $(L_{PHR} + L_{PSDU})/8$  is odd**

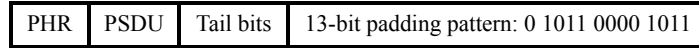


**Figure 20-12—An example of extension with padding bits prior to encoding, when  $(L_{\text{PHR}} + L_{\text{PSDU}})/8$  is even**

The RSC encoder is shown in Figure 20-13, and the NRNSC encoder is shown in Figure 20-14.

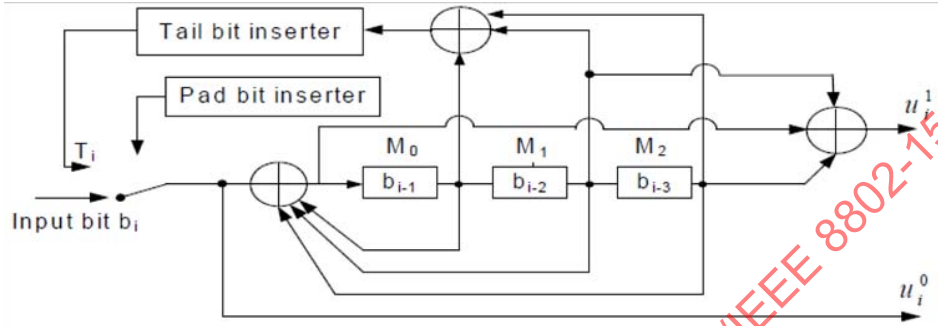


**Figure 20-13—The recursive and systematic code (RSC) encoder**

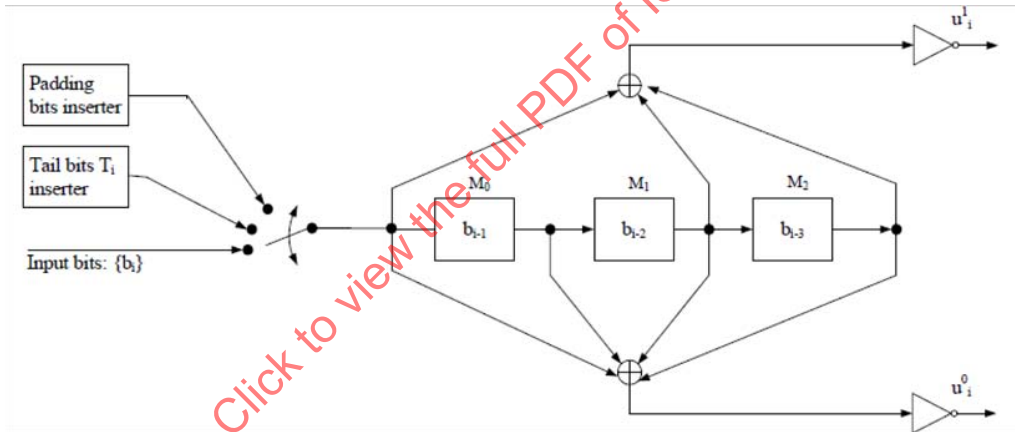
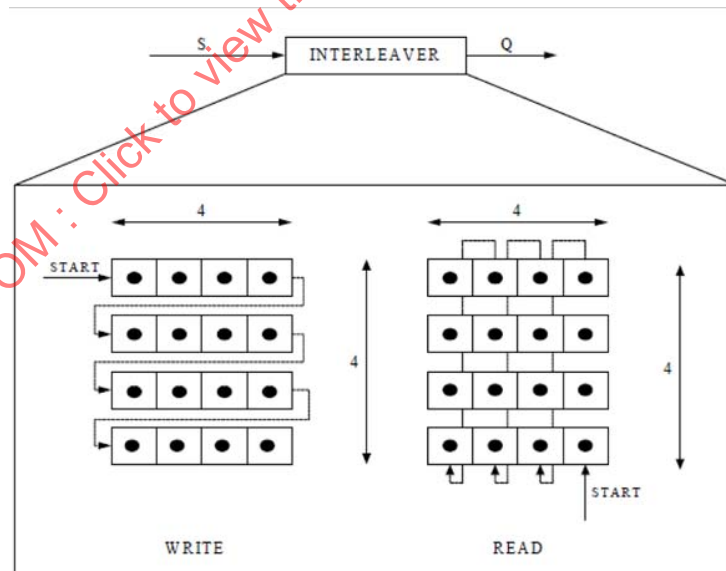


**Figure 20-14—Non-recursive and non-systematic code (NRNSC) encoder**

For an input sequence of bits with length  $N$ ,  $B = \{b_i, i \in [0, 1, 2, \dots, N-1]\}$ , the  $i$ th input bit shall be represented as  $b_i$  and be fed into memory state  $M_0$ ,  $M_1$ , and  $M_2$  in that order. The tail bits  $T_i$  and the pad bits shall be inserted once the encoding of PHR and PSDU is complete. The output sequence  $S$  also comprises  $N$  code-symbols.

$$S = \{s(0), s(1), s(2), \dots, s(N-1)\} = \{u_0^1, u_0^0, \dots, u_i^1, u_i^0, u_{i+1}^1, u_{i+1}^0, \dots, u_{N-1}^1, u_{N-1}^0\}$$

Each code-symbol is denoted by  $s(i) = \{u_i^1, u_i^0\}$ , for all  $i = 0, \dots, N-1$ , where  $s(i)$  is the  $i$ th output code-symbol due to the  $i$ th input bit and  $u_i^1$  and  $u_i^0$  indicate the first and second output bits of the convolutional encoder, respectively. The code-symbol  $s(i)$  shall precede the code-symbol  $s(i+1)$  and the code bit  $u_i^1$  shall precede the code bit  $u_i^0$ .

$$u_i^0 = b_i$$
$$u_i^0 = \overline{b_i \oplus b_{i-1} \oplus b_{i-2} \oplus b_{i-3}}$$


The complete sequence of code-symbols  $S = \{s(i)\}$ ,  $0 \leq i \leq N-1$ , is passed to the interleaver as  $N_{BLOCK}$  consecutive subsequences  $A^{(p)}$ , where  $0 \leq p \leq N_{BLOCK}-1$ ,  $N_{BLOCK} = N/16$ , and  $N$  is a nonzero integer multiple of 16, as described in 20.3.4. The subsequence  $A^{(0)}$  shall be passed to the interleaver first in time, and the subsequence  $A^{(N_{BLOCK}-1)}$  shall be passed to the interleaver last in time.

Each subsequence  $A^{(p)} = \{a^{(p)}(j)\}$  contains exactly 16 code-symbols and shall be derived according to the following equation:

$$a^{(p)}(j) = s(p \times 16 + j)$$

where  $0 \leq j \leq 15$ .

For each subsequence  $A^{(p)}$  passed to the interleaver, the corresponding subsequence  $Q^{(p)} = \{q^{(p)}(k)\}$  exiting the interleaver shall be computed as follows:

$$q^{(p)}(k) = a^{(p)}(t)$$

where

$$0 \leq k \leq 15$$

$$t = 15 - 4 \times (k \bmod 4) - \left\lfloor \frac{k}{4} \right\rfloor$$

The function  $\lfloor x \rfloor = \text{floor}(x)$  returns the largest integer value not greater than  $x$ .

The complete sequence of interleaved code-symbols is derived as  $Q = \{Q^{(0)}, Q^{(1)}, \dots, Q^{(N_{BLOCK}-1)}\}$ .

## 20.4 Data whitening for SUN FSK

Support for data whitening is optional. Data whitening for the SUN FSK PHY shall use the procedure described in 17.2.3.

## 20.5 Mode switch mechanism for SUN FSK

The use of the Mode Switch mechanisms described in this subclause is deprecated. Consequently, this clause may be removed in a later revision of the standard.

The mode switch mechanism is optional.

The mode switch mechanism is enabled by setting the Mode Switch field to one. The SUN FSK mode switch PPDU is transmitted on *phyCurrentChannel*, as defined in 11.3, and the PPDU containing the PSDU is transmitted on the channel that corresponds to the same center frequency used for the SUN FSK mode switch PPDU. When an SUN FSK mode switch PPDU, as described in Figure 20-2, is received, a device that supports mode switching shall change its mode of operation to the new mode defined in the SUN FSK mode switch PPDU, in order to receive the following packet.

When changing from the current operating mode to the new mode, a settling delay may exist. If the modulation scheme of the new mode is FSK, the settling delay shall be in the range of zero to 100  $\mu$ s. If the modulation scheme of the new mode is not FSK, the settling delay shall be in the range of 200  $\mu$ s to 500  $\mu$ s. The settling delay value is part of a ModeSwitchDescriptor, as described in Table 11-3. The value specified in the Mode Switch Parameter field of the PHR, as described in Figure 20-5, is the index of the PIB attribute

array *phyModeSwitchParameterEntries*, as defined in 11.3, which contains the elements of the ModeSwitchDescriptor. How the Mode Switch Parameter field maps to ModeSwitchDescriptor is exemplified in Table 20-11. For the mode switch operation of FSK->FSK, the symbol rate is changed. For the mode switch operation of FSK->4-FSK, the modulation order and/or the symbol rate is changed. The Mode Switch Parameter table may be defined by the next higher layer.

**Table 20-11—An example of mapping between *phyModeSwitchParameterEntries*[] and ModeSwitchDescriptor**

| <i>phyModeSwitchParameterEntries</i> [ ] | Mode Switch Operation<br>(Source mode -> Target mode) | ModeSwitchDescriptor |                              |                  |
|------------------------------------------|-------------------------------------------------------|----------------------|------------------------------|------------------|
|                                          |                                                       | Settling Delay (μs)  | SecondaryFSK Preamble Length | Secondary FSKSFD |
| 0                                        | FSK->FSK                                              | 20                   | 0                            | FALSE            |
| 1                                        | FSK->4-FSK                                            | 40                   | 0                            | FALSE            |
| 2                                        | FSK->OFDM                                             | 160                  | n/a                          | n/a              |
| 3                                        | FSK->O-QPSK                                           | 80                   | n/a                          | n/a              |

Transmission of the new mode PPDU shall start SettlingDelay from the end of the mode switch PPDU. The SettlingDelay shall be the value indicated in the Mode Switch Parameter field in the mode switch PPDU. The reception and rejection of the following packet follows the same mechanism described in 5.1.6.2. When the new mode PPDU has been received, the receiver shall return to the mode specified by *phyCurrentPage*, as defined in 11.3, within a SIFS or LIFS period based on the symbol period of the new mode PPDU, depending on the received frame length, as described in 5.1.1.3. If the transmission of an Ack frame is requested by the transmitter, the Ack frame is transmitted using the PHY mode specified by *phyCurrentPage*.

The sequence of the SUN FSK mode switch PPDU, the optional settling delay, and the PPDU transmitted in the new PHY mode is shown in Figure 20-16.

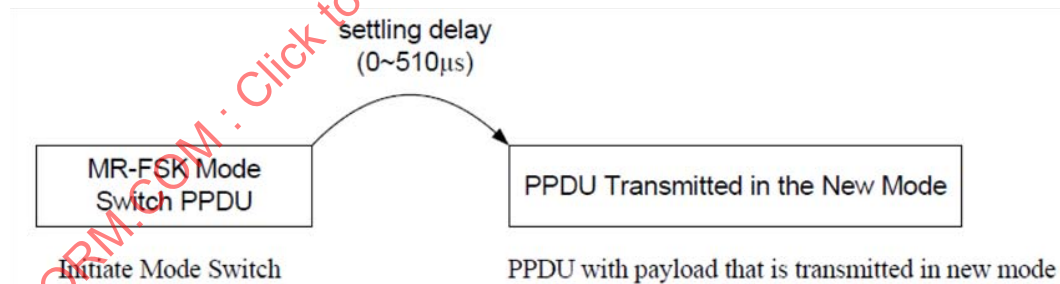


**Figure 20-16—Transmitting sequence between SUN FSK mode switch PPDU and the new mode PPDU**

Devices employing the mode switch mechanism shall meet the MAC timing requirements of 5.1.1.1.1 and 5.1.1.1.2, using the symbol period of the PHY mode prior to the mode switch.

The frequency band is not changed by the PHY mode switch mechanism. The center frequency of the channel is also not changed by a PHY mode switch, and channel center frequency alignment between the various modulation schemes support the mode switch mechanism. For example, the channel frequency alignment for the 915 MHz band is shown in Table 20-12.

**Table 20-12—Channel alignment for 915 MHz band**

| FSK or OFDM<br>(200 kHz channel spacing) | FSK or OFDM<br>(400 kHz channel spacing) | OFDM<br>(800 kHz channel spacing) | O-QPSK |
|------------------------------------------|------------------------------------------|-----------------------------------|--------|
| 902.2                                    | —                                        | —                                 | —      |
| 902.4                                    | 902.4                                    | —                                 | —      |
| 902.6                                    | —                                        | —                                 | —      |
| 902.8                                    | 902.8                                    | 902.8                             | —      |
| 903.0                                    | —                                        | —                                 | —      |
| 903.2                                    | 903.2                                    | —                                 | —      |
| 903.4                                    | —                                        | —                                 | —      |
| 903.6                                    | 903.6                                    | 903.6                             | —      |
| 903.8                                    | —                                        | —                                 | —      |
| 904.0                                    | 904.0                                    | —                                 | 904.0  |
| 904.2                                    | —                                        | —                                 | —      |
| 904.4                                    | 904.4                                    | 904.4                             | —      |
| 904.6                                    | —                                        | —                                 | —      |
| 904.8                                    | 904.8                                    | —                                 | —      |
| 905.0                                    | —                                        | —                                 | —      |
| 905.2                                    | 905.2                                    | 905.2                             | —      |
| 905.4                                    | —                                        | —                                 | —      |
| 905.6                                    | 905.6                                    | —                                 | —      |
| 905.8                                    | —                                        | —                                 | —      |
| 906.0                                    | 906.0                                    | 906.0                             | 906.0  |
| 906.2                                    | —                                        | —                                 | —      |
| 906.4                                    | 906.4                                    | —                                 | —      |
| 906.6                                    | —                                        | —                                 | —      |
| 906.8                                    | 906.8                                    | 906.8                             | —      |
| 907.0                                    | —                                        | —                                 | —      |
| 907.2                                    | 907.2                                    | —                                 | —      |
| 907.4                                    | —                                        | —                                 | —      |
| 907.6                                    | 907.6                                    | 907.6                             | —      |
| 907.8                                    | —                                        | —                                 | —      |
| 908.0                                    | 908.0                                    | —                                 | 908.0  |
| 908.2                                    | —                                        | —                                 | —      |
| 908.4                                    | 908.4                                    | 908.4                             | —      |
| 908.6                                    | —                                        | —                                 | —      |

**Table 20-12—Channel alignment for 915 MHz band (continued)**

| FSK or OFDM<br>(200 kHz channel spacing) | FSK or OFDM<br>(400 kHz channel spacing) | OFDM<br>(800 kHz channel spacing) | O-QPSK |
|------------------------------------------|------------------------------------------|-----------------------------------|--------|
| 908.8                                    | 908.8                                    | —                                 | —      |
| 909.0                                    | —                                        | —                                 | —      |
| 909.2                                    | 909.2                                    | 909.2                             | —      |
| 909.4                                    | —                                        | —                                 | —      |
| 909.6                                    | 909.6                                    | —                                 | —      |
| 909.8                                    | —                                        | —                                 | —      |
| 910.0                                    | 910.0                                    | 910.0                             | 910.0  |
| ⋮                                        | ⋮                                        | ⋮                                 | ⋮      |
| etc.                                     | etc.                                     | etc.                              | etc.   |

## 20.6 SUN FSK PHY RF requirements

### 20.6.1 Operating frequency range

The SUN FSK PHY operates in the bands given in Table 20-6 and Table 20-7.

### 20.6.2 Regulatory compliance

It is the responsibility of the implementer to verify and ensure that the device is in compliance with all regulatory requirements in the geographic region where the device is deployed or sold. Conformance with this standard does not guarantee compliance with the relevant regulatory requirements that may apply.

### 20.6.3 Radio frequency tolerance

The single-sided clock frequency tolerance  $T$  at the transmitter shall be as follows:

$$T \leq \min\left(\frac{T_0 \times R \times h \times F_0}{R_0 \times h_0 \times F}, 50 \times 10^{-6}\right)$$

for all combinations of  $R$ ,  $h$ , and  $F$  and for each mode supported by the device, where

$R$  is the symbol rate, in ksymbol/s

$h$  is the modulation index

$F$  is the carrier frequency, in megahertz

$R_0$  is 50 ksymbol/s

$h_0$  is 1

$F_0$  is 915 MHz

$T_0$  is  $30 \times 10^{-6}$  for modes in all bands, except at 2450 MHz for which the value of  $T_0$  is  $40 \times 10^{-6}$



**20.6.4 Channel switch time**

Channel switch time shall be less than or equal to 500  $\mu$ s. The channel switch time is defined as the time elapsed when changing to a new channel, including any required settling time.

**20.6.5 Transmitter symbol rate**

The transmitter symbol rate tolerance shall be less than or equal to  $\pm 300 \times 10^{-6}$ . The peak transmitter symbol rate jitter shall be less than or equal to  $\pm 40 \times 10^{-6}$ . Transmitted packets shall have symbol rates within the specified symbol rate tolerance, and all symbols within the packet shall be within the symbol rate tolerance relative to the average symbol rate of all the symbols in the packet. The symbol rate jitter is measured as the standard deviation of symbol edges from the nominal symbol edge position for the symbol rate used by the transmitter.

**20.6.6 Transmit spectral mask**

The transmit spectral content is the ratio of the total transmitted out-of-channel power to the total transmitted in-channel power in a given integration bandwidth.

The integration bandwidth shall be equal to  $1.5 \times R$ , where  $R$  is the symbol rate, expressed in units of hertz.

Out-of-channel power shall be measured at two offset frequencies relative to the carrier frequency. The offset frequencies  $M_1$  and  $M_2$  are defined as follows:

$$M_1 = 1.5 \times R \times (1 + h)$$

$$M_2 = 3 \times R \times (1 + h)$$

where  $h$  is the modulation index for 2-level modulation and three times the modulation index for 4-level modulation.

The transmit spectral content at  $M_1$  and  $M_2$  shall be less than  $-25$  dB and  $-35$  dB, respectively.

The modulated signal shall use a PN data pattern of 511 bits or longer.

The spectrum analyzer settings for this measurement shall be as follows: the resolution bandwidth is 1 kHz, the video bandwidth is 1 kHz or greater, and the detector is RMS.

**20.6.7 Receiver sensitivity**

The SUN FSK receiver sensitivity shall be better than  $S$ , where  $S$ , for binary modulation, is defined as follows:

$$S = \left( S_0 + 10 \log \left[ \frac{R}{R_0} \right] \right) \text{ dBm}$$

where

- $S_0$  is  $-91$  without FEC and  $-97$  with FEC
- $R_0$  is 50 kb/s
- $R$  is the bit rate, in kb/s

See 10.1.7 for additional information on receiver sensitivity.

**20.6.8 Receiver interference rejection**

The adjacent designated channels are those on either side of the desired designated channel that are closest in frequency to the desired designated channel. The alternate designated channel is more than one removed from the desired designated channel in the operational frequency band.

The adjacent channel rejection shall be measured as follows: the desired signal shall be a compliant SUN FSK PHY signal, as defined in 20.3, of pseudo-random data at the center frequency of the desired channel. The desired signal is input to the receiver at a level 3 dB above the receiver sensitivity given in 20.6.7.

In either the adjacent or the alternate channel, an unmodulated carrier in the center of that channel is input at the following level relative to the level of the desired signal:

- The adjacent channel rejection shall be greater than or equal to 10 dB.
- The alternate channel rejection shall be greater than or equal to 30 dB.

The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

**20.6.9 TX-to-RX turnaround time**

The SUN FSK PHY shall meet the requirements for TX-to-RX turnaround time as defined in 10.2.1.

**20.6.10 RX-to-TX turnaround time**

The SUN FSK PHY shall meet the requirements for RX-to-TX turnaround time as defined in 10.2.2.

**20.6.11 Transmit power**

A transmitter shall be capable of transmitting at a power level of 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.

**20.6.12 Receiver maximum input level of desired signal**

The SUN FSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

**20.6.13 Receiver ED**

The SUN FSK PHY shall provide the receiver ED measurement as described in 10.2.5.

**20.6.14 LQI**

The SUN FSK PHY shall provide the LQI measurement as described in 10.2.6.

## 21. SUN OFDM PHY

### 21.1 Introduction

The SUN OFDM PHY supports data rates ranging from 50 kb/s to 800 kb/s. The subcarrier spacing is constant and is equal to  $10416\frac{2}{3}$  Hz.

The symbol rate is  $8\frac{1}{3}$  ksymbol/s, which corresponds 120  $\mu$ s per symbol. This symbol period is to be used for *aTurnaroundTime*.

The symbol period used for MAC timing parameters (*macLifsPeriod* and *macSifsPeriod*) shall be the symbol period of SUN FSK operating mode #1 specified in Table 20-6 and Table 20-7.

This PHY includes four options, each one being characterized by the number of active tones during the PHR or PSDU. The total signal bandwidth for each option ranges from 1.2 MHz down to less than 200 kHz.

Two examples of encoding a packet for the SUN OFDM PHY are given in “Examples of IEEE Std 802.15.4 PHY encodings” [B6].

### 21.2 PPDU format for SUN OFDM

The SUN OFDM PPDU shall be formatted as illustrated in Figure 21-1.



Figure 21-1—Format of the SUN OFDM PPDU

The PHY Payload field shall be formatted as illustrated in Figure 21-2.

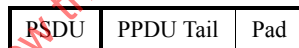


Figure 21-2—Format of the PHY Payload field

#### 21.2.1 Short Training field (STF)

##### 21.2.1.1 Frequency domain STF

Table 21-1 shows the frequency domain representation of the STF for Option 1. The scaling factor used in the Table 21-1 is  $\sqrt{104/12}$ .

Table 21-1—Frequency domain representation of Option 1 STF\_freq(0)

| Tone# | Value | Tone# | Value   | Tone# | Value | Tone# | Value  |
|-------|-------|-------|---------|-------|-------|-------|--------|
| −64   | 0     | −32   | −2.9439 | 0     | 0     | 32    | 2.9439 |
| −63   | 0     | −31   | 0       | 1     | 0     | 33    | 0      |
| −62   | 0     | −30   | 0       | 2     | 0     | 34    | 0      |
| −61   | 0     | −29   | 0       | 3     | 0     | 35    | 0      |
| −60   | 0     | −28   | 0       | 4     | 0     | 36    | 0      |

**Table 21-1—Frequency domain representation of Option 1 STF\_freq(0) (continued)**

| Tone# | Value   | Tone# | Value  | Tone# | Value   | Tone# | Value   |
|-------|---------|-------|--------|-------|---------|-------|---------|
| −59   | 0       | −27   | 0      | 5     | 0       | 37    | 0       |
| −58   | 0       | −26   | 0      | 6     | 0       | 38    | 0       |
| −57   | 0       | −25   | 0      | 7     | 0       | 39    | 0       |
| −56   | 0       | −24   | 2.9439 | 8     | 2.9439  | 40    | −2.9439 |
| −55   | 0       | −23   | 0      | 9     | 0       | 41    | 0       |
| −54   | 0       | −22   | 0      | 10    | 0       | 42    | 0       |
| −53   | 0       | −21   | 0      | 11    | 0       | 43    | 0       |
| −52   | 0       | −20   | 0      | 12    | 0       | 44    | 0       |
| −51   | 0       | −19   | 0      | 13    | 0       | 45    | 0       |
| −50   | 0       | −18   | 0      | 14    | 0       | 46    | 0       |
| −49   | 0       | −17   | 0      | 15    | 0       | 47    | 0       |
| −48   | −2.9439 | −16   | 2.9439 | 16    | −2.9439 | 48    | 2.9439  |
| −47   | 0       | −15   | 0      | 17    | 0       | 49    | 0       |
| −46   | 0       | −14   | 0      | 18    | 0       | 50    | 0       |
| −45   | 0       | −13   | 0      | 19    | 0       | 51    | 0       |
| −44   | 0       | −12   | 0      | 20    | 0       | 52    | 0       |
| −43   | 0       | −11   | 0      | 21    | 0       | 53    | 0       |
| −42   | 0       | −10   | 0      | 22    | 0       | 54    | 0       |
| −41   | 0       | −9    | 0      | 23    | 0       | 55    | 0       |
| −40   | −2.9439 | −8    | 2.9439 | 24    | 2.9439  | 56    | 0       |
| −39   | 0       | −7    | 0      | 25    | 0       | 57    | 0       |
| −38   | 0       | −6    | 0      | 26    | 0       | 58    | 0       |
| −37   | 0       | −5    | 0      | 27    | 0       | 59    | 0       |
| −36   | 0       | −4    | 0      | 28    | 0       | 60    | 0       |
| −35   | 0       | −3    | 0      | 29    | 0       | 61    | 0       |
| −34   | 0       | −2    | 0      | 30    | 0       | 62    | 0       |
| −33   | 0       | −1    | 0      | 31    | 0       | 63    | 0       |

Table 21-2 shows the frequency domain representation of the STF for Option 2. The scaling factor used in the table is  $\sqrt{52/12}$ .

**Table 21-2—Frequency domain representation of Option 2 STF\_freq(1)**

| Tone# | Value   | Tone# | Value   | Tone# | Value   | Tone# | Value   |
|-------|---------|-------|---------|-------|---------|-------|---------|
| −32   | 0       | −16   | −2.0817 | 0     | 0       | 16    | 2.0817  |
| −31   | 0       | −15   | 0       | 1     | 0       | 17    | 0       |
| −30   | 0       | −14   | 0       | 2     | 0       | 18    | 0       |
| −29   | 0       | −13   | 0       | 3     | 0       | 19    | 0       |
| −28   | 0       | −12   | 2.0817  | 4     | 2.0817  | 20    | −2.0817 |
| −27   | 0       | −11   | 0       | 5     | 0       | 21    | 0       |
| −26   | 0       | −10   | 0       | 6     | 0       | 22    | 0       |
| −25   | 0       | −9    | 0       | 7     | 0       | 23    | 0       |
| −24   | −2.0817 | −8    | 2.0817  | 8     | −2.0817 | 24    | 2.0817  |
| −23   | 0       | −7    | 0       | 9     | 0       | 25    | 0       |
| −22   | 0       | −6    | 0       | 10    | 0       | 26    | 0       |
| −21   | 0       | −5    | 0       | 11    | 0       | 27    | 0       |
| −20   | −2.0817 | −4    | 2.0817  | 12    | 2.0817  | 28    | 0       |
| −19   | 0       | −3    | 0       | 13    | 0       | 29    | 0       |
| −18   | 0       | −2    | 0       | 14    | 0       | 30    | 0       |
| −17   | 0       | −1    | 0       | 15    | 0       | 31    | 0       |

Table 21-3 shows the frequency domain representation of the STF for Option 3. The scaling factor used in the table is  $\sqrt{26/6}$ .

**Table 21-3—Frequency domain representation of Option 3 STF\_freq(2)**

| Tone# | Value  | Tone# | Value  | Tone# | Value   | Tone# | Value   |
|-------|--------|-------|--------|-------|---------|-------|---------|
| −16   | 0      | −8    | 2.0817 | 0     | 0       | 8     | 2.0817  |
| −15   | 0      | −7    | 0      | 1     | 0       | 9     | 0       |
| −14   | 0      | −6    | 0      | 2     | 0       | 10    | 0       |
| −13   | 0      | −5    | 0      | 3     | 0       | 11    | 0       |
| −12   | 2.0817 | −4    | 2.0817 | 4     | −2.0817 | 12    | −2.0817 |
| −11   | 0      | −3    | 0      | 5     | 0       | 13    | 0       |
| −10   | 0      | −2    | 0      | 6     | 0       | 14    | 0       |
| −9    | 0      | −1    | 0      | 7     | 0       | 15    | 0       |

Table 21-4 shows the frequency domain representation of the STF for Option 4. The scaling factor used in the table is  $\sqrt{14/6}$ .

**Table 21-4—Frequency domain representation of Option 4 STF\_freq(3)**

| Tone# | Value  | Tone# | Value  | Tone# | Value   | Tone# | Value   |
|-------|--------|-------|--------|-------|---------|-------|---------|
| −8    | 0      | −4    | 1.5275 | 0     | 0       | 4     | 1.5275  |
| −7    | 0      | −3    | 0      | 1     | 0       | 5     | 0       |
| −6    | 1.5275 | −2    | 1.5275 | 2     | −1.5275 | 6     | −1.5275 |
| −5    | 0      | −1    | 0      | 3     | 0       | 7     | 0       |

### 21.2.1.2 Time domain STF generation

Given a sequence of  $N$  samples  $f(n)$ , indexed by  $n = 0, \dots, N - 1$ , the DFT is defined as  $F(k)$ , where  $k = 0, \dots, N - 1$ :

$$F(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) e^{-j2\pi kn/N}$$

The sequence  $f(n)$  can be calculated from  $F(k)$  using the inverse discrete Fourier transform (IDFT), where the  $k$  values numbered from 0 to  $(N/2) - 1$  correspond to tones numbered from 0 to  $(N/2) - 1$  and the  $k$  values numbered from  $(N/2)$  to  $(N - 1)$  correspond to tones numbered from  $-(N/2)$  to  $-1$ , respectively:

$$f(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} F(k) e^{j2\pi nk/N}$$

The time domain STF for Option- $n$  ( $n = 1, 2, 3, 4$ ) is obtained as follows:

$$\text{STF\_time}(\text{Option-}n) = \text{IDFT}(\text{STF\_freq}(\text{Option-}n))$$

The CP is then prepended to the OFDM symbol.

### 21.2.1.3 Time domain STF repetition

There are four STF OFDM symbols, and the last 1/2 of the fourth OFDM symbol is negated in the time domain. For Options 2, 3, and 4, the CP is 1/4 of the OFDM symbol. Therefore, for Options 2 and 3, there are 18 repetitions of the 1/4 STF symbol followed by the last 1/2 of the fourth OFDM symbol, which is negated in the time domain. For Option 4, there are nine repetitions of the 1/2 STF symbol followed by the last 1/2 of the fourth OFDM symbol, which is negated in the time domain.

For Option 1, the CP is also 1/4 symbol, and the STF repetition is eight times per STF symbol. Therefore, there are 36 repetitions of 1/8 STF symbol in the four STF symbols followed by the last 1/2 of the fourth OFDM symbol, which is negated in the time domain.

Figure 21-3 shows the STF structure for all four options. Each “s” in Figure 21-3 represents one time-domain repetition of a subsequence of different length for SUN OFDM Option 1, Option 2 & 3, and Option 4.

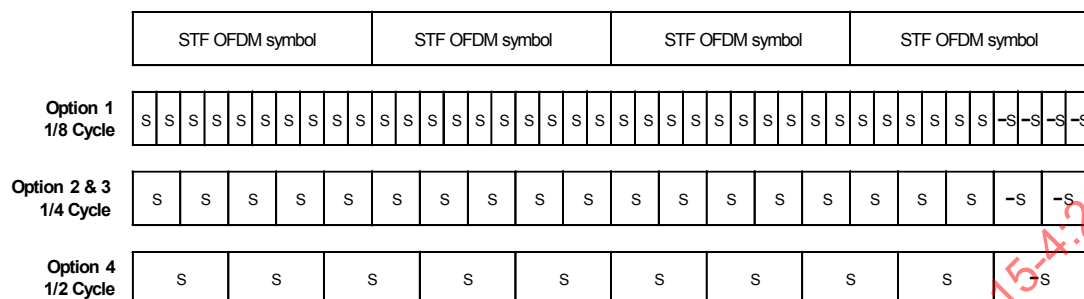


**Figure 21-3—Structure of STF for SUN OFDM for Options 1, 2, 3, and 4**

#### 21.2.1.4 STF normalization

The STF uses a lesser number of tones than the PHY Payload field. Hence, normalization of the frequency domain STF is required to ensure that the STF power is the same as the rest of the packet. In order to have the same power as the PHY Payload field, the normalization value is as follows:

$$\sqrt{N_{active}/N_{stf}}$$

where

$N_{active}$  is the number of used subcarriers in rest of the OFDM packet for the particular DFT option  
 $N_{stf}$  is the number of subcarriers used in the STF

Power boosting shall be applied to the STF symbols in order to aid preamble detection. The boost should be a multiplication by 1.25, which is approximately 1.94 dB.

### 21.2.2 Long Training field (LTF)

### 21.2.2.1 Frequency domain LTF

Table 21-5 shows the frequency domain representation of the LTF for Option 1.

**Table 21-5—Frequency domain representation of Option 1 LTF\_freq(0)**

| Tone# | Value | Tone# | Value | Tone# | Value | Tone# | Value |
|-------|-------|-------|-------|-------|-------|-------|-------|
| -64   | 0     | -32   | -1    | 0     | 0     | 32    | -1    |
| -63   | 0     | -31   | -1    | 1     | 1     | 33    | -1    |
| -62   | 0     | -30   | -1    | 2     | -1    | 34    | -1    |
| -61   | 0     | -29   | 1     | 3     | 1     | 35    | 1     |
| -60   | 0     | -28   | 1     | 4     | -1    | 36    | 1     |
| -59   | 0     | -27   | -1    | 5     | 1     | 37    | 1     |

**Table 21-5—Frequency domain representation of Option 1 LTF\_freq(0) (continued)**

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



Table 21-6 shows the frequency domain representation of the LTF for Option 2.

**Table 21-6—Frequency domain representation of Option 2 LTF\_freq(1)**

| Tone# | Value | Tone# | Value | Tone# | Value | Tone# | Value |
|-------|-------|-------|-------|-------|-------|-------|-------|
| −32   | 0     | −16   | 1     | 0     | 0     | 16    | 1     |
| −31   | 0     | −15   | −1    | 1     | 1     | 17    | −1    |
| −30   | 0     | −14   | 1     | 2     | −1    | 18    | −1    |
| −29   | 0     | −13   | 1     | 3     | 1     | 19    | −1    |
| −28   | 0     | −12   | −1    | 4     | 1     | 20    | −1    |
| −27   | 0     | −11   | −1    | 5     | −1    | 21    | −1    |
| −26   | −1    | −10   | −1    | 6     | 1     | 22    | 1     |
| −25   | −1    | −9    | 1     | 7     | −1    | 23    | −1    |
| −24   | −1    | −8    | 1     | 8     | −1    | 24    | −1    |
| −23   | −1    | −7    | −1    | 9     | 1     | 25    | −1    |
| −22   | 1     | −6    | 1     | 10    | −1    | 26    | 1     |
| −21   | 1     | −5    | 1     | 11    | 1     | 27    | 0     |
| −20   | 1     | −4    | 1     | 12    | 1     | 28    | 0     |
| −19   | −1    | −3    | −1    | 13    | −1    | 29    | 0     |
| −18   | 1     | −2    | −1    | 14    | −1    | 30    | 0     |
| −17   | −1    | −1    | 1     | 15    | 1     | 31    | 0     |

Table 21-7 shows the frequency domain representation of the LTF for Option 3.

**Table 21-7—Frequency domain representation of Option 3 LTF\_freq(2)**

| Tone# | Value | Tone# | Value | Tone# | Value | Tone# | Value |
|-------|-------|-------|-------|-------|-------|-------|-------|
| −16   | 0     | −8    | 1     | 0     | 0     | 8     | −1    |
| −15   | 0     | −7    | 1     | 1     | −1    | 9     | 1     |
| −14   | 0     | −6    | 1     | 2     | −1    | 10    | 1     |
| −13   | 1     | −5    | 1     | 3     | 1     | 11    | −1    |
| −12   | −1    | −4    | 1     | 4     | −1    | 12    | −1    |
| −11   | 1     | −3    | 1     | 5     | 1     | 13    | 1     |
| −10   | −1    | −2    | 1     | 6     | 1     | 14    | 0     |
| −9    | 1     | −1    | −1    | 7     | −1    | 15    | 0     |

Table 21-8 shows the frequency domain representation of the LTF for Option 4.

**Table 21-8—Frequency domain representation of Option 4 LTF\_freq(3)**

| Tone# | Value | Tone# | Value | Tone# | Value | Tone# | Value |
|-------|-------|-------|-------|-------|-------|-------|-------|
| –8    | 0     | –4    | 1     | 0     | 0     | 4     | 1     |
| –7    | 1     | –3    | –1    | 1     | –1    | 5     | –1    |
| –6    | –1    | –2    | 1     | 2     | 1     | 6     | –1    |
| –5    | 1     | –1    | 1     | 3     | 1     | 7     | –1    |

### 21.2.2.2 Time domain LTF generation

The time domain LTF for Option-n ( $n = 1, 2, 3, 4$ ) is obtained as follows:

$$\text{LTF\_time}(\text{Option-n}) = \text{IDFT}(\text{LTF\_freq}(\text{Option-n})).$$

A 1/2 symbol CP is prepended to two consecutive copies of the base symbol as shown in Figure 21-4. For more details, see 21.4.8.

The time-domain LTF structure is shown in Figure 21-4, and  $T_{\text{DFT}}$  is the duration of the base symbol.

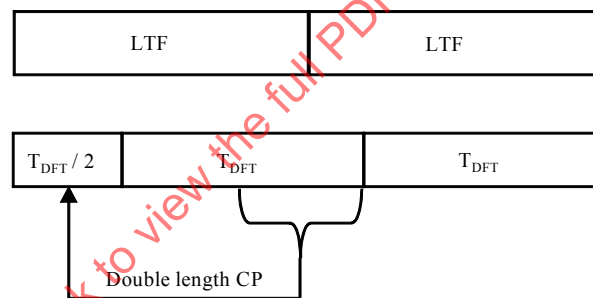


**Figure 21-4—Structure of LTF for SUN OFDM**

### 21.2.2.3 LTF normalization

Power boosting is not used by the LTF.

### 21.2.3 PHR

The PHR field shall be formatted as illustrated in Figure 21-5. All multi-bit fields are unsigned integers and shall be processed MSB first.

| Bits: 0–4 | 5        | 6–16         | 17–18    | 19–20     | 21       | 22–29 | 30–35 |
|-----------|----------|--------------|----------|-----------|----------|-------|-------|
| Rate      | Reserved | Frame Length | Reserved | Scrambler | Reserved | HCS   | Tail  |

**Figure 21-5—PHY header fields for SUN OFDM**

When the PIB attribute *phyOfdmInterleaving*, as defined in 11.3, is zero (i.e., interleaving depth of one symbol), the PHR occupies three OFDM symbols for Option 1 and six OFDM symbols for Options 2, 3, and 4. When the PIB attribute *phyOfdmInterleaving* is one (i.e., interleaving depth of the number of symbols equal to the frequency domain spreading factor), the PHR occupies four OFDM symbols for Option 1, eight OFDM symbols for Option 2, and six OFDM symbols for Options 3 and 4. The PHR shall be transmitted using the lowest supported modulation and coding scheme (MCS) level, as described in Table 21-9, for the option being used. It is sent to the convolutional encoder starting from the leftmost bit in Figure 21-5 to the rightmost bit.

The Rate field is set to the numerical value of the MCS, as described in 21.4, transmitted MSB first. The data rates for each OFDM bandwidth option can be found in 21.3.

The Frame Length field is an unsigned integer that shall be set to the total number of octets contained in the PSDU (prior to FEC encoding). The Frame Length field shall be transmitted MSB first.

The Scrambler field specifies the scrambling seed, as described in 21.4.11.

The header check sequence (HCS) field is an 8-bit CRC taken over the PHR fields.

The HCS shall be computed using the first 22 bits of the PHR. The HCS shall be calculated using the polynomial  $G_8(x) = x^8 + x^2 + x + 1$ .

The HCS is the one's complement of the modulo 2 sum of the two remainders in a) and b):

- a) The remainder resulting from  $[x^k(x^7 + x^6 + \dots + 1)]$  divided (modulo 2) by  $G_8(x)$ , where the value  $k$  is the number of bits in the calculation field.
- b) The remainder resulting from the calculation field contents, treated as a polynomial, multiplied by  $x^8$  and then divided (modulo 2) by  $G_8(x)$ .

At the transmitter, the initial remainder of the division shall be preset to all ones and is then modified via division of the calculation field by the generator polynomial  $G_8(x)$ . The one's complement of this remainder is the HCS field. The HCS field is transmitted commencing with the coefficient of the highest order term. An example of HCS generation is given in "Examples of IEEE Std 802.15.4 PHY encodings" [B6].

The Tail field consists of all zeros, which is intended for Viterbi decoder flushing, as described in 21.4.9.

#### 21.2.4 PSDU field

The PSDU field contains the encoded PSDU.

### 21.3 Data rates for SUN OFDM

There are four OFDM options, each with a different number of active tones. All devices supporting a particular option (1, 2, 3, or 4) shall support all BPSK and QPSK modulation and coding scheme (MCS) levels for that option. All 16 quadrature amplitude modulation (QAM) MCS levels are optional.

The various data rates are shown in Table 21-9. The nominal bandwidth is calculated by multiplying {the number of active tones + 1 for the DC tone} by {the subcarrier spacing}.

**Table 21-9—Data Rates for SUN OFDM PHY**

| Parameter                                                   | OFDM<br>Option 1 | OFDM<br>Option 2 | OFDM<br>Option 3 | OFDM<br>Option 4 |
|-------------------------------------------------------------|------------------|------------------|------------------|------------------|
| Nominal bandwidth (kHz)                                     | 1094             | 552              | 281              | 156              |
| Channel spacing (kHz)                                       | 1200             | 800              | 400              | 200              |
| DFT size                                                    | 128              | 64               | 32               | 16               |
| Active tones                                                | 104              | 52               | 26               | 14               |
| # Pilot tones                                               | 8                | 4                | 2                | 2                |
| # Data tones                                                | 96               | 48               | 24               | 12               |
| MCS0 (kb/s)<br>(BPSK rate 1/2 with 4x frequency repetition) | 100              | 50               | —                | —                |
| MCS1 (kb/s)<br>(BPSK rate 1/2 with 2x frequency repetition) | 200              | 100              | 50               | —                |
| MCS2 (kb/s)<br>(QPSK rate 1/2 and 2x frequency repetition)  | 400              | 200              | 100              | 50               |
| MCS3 (kb/s)<br>(QPSK rate 1/2)                              | 800              | 400              | 200              | 100              |
| MCS4 (kb/s)<br>(QPSK rate 3/4)                              | —                | 600              | 300              | 150              |
| MCS5 (kb/s)<br>(16-QAM rate 1/2)                            | —                | 800              | 400              | 200              |
| MCS6 (kb/s)<br>(16-QAM rate 3/4)                            | —                | —                | 600              | 300              |

## 21.4 Modulation and coding for SUN OFDM

### 21.4.1 Reference modulator diagram

The reference modulator diagram is shown in Figure 21-6.

### 21.4.2 Bit-to-symbol mapping

Figure 21-7 shows the bit-to-symbol mapping for BPSK, QPSK, and 16-QAM.

The output values,  $d$ , are formed by multiplying the resulting  $(I + jQ)$  value by a normalization factor  $K_{MOD}$ :

$$d = (I + jQ) \times K_{MOD}$$

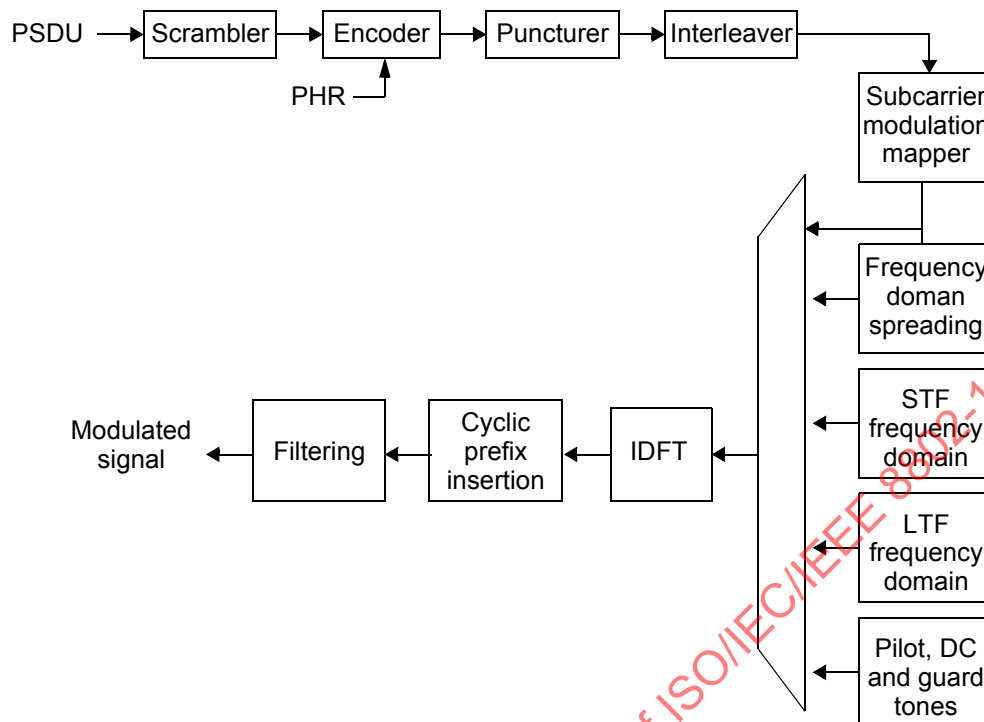


Figure 21-6—Reference modulator diagram for SUN OFDM

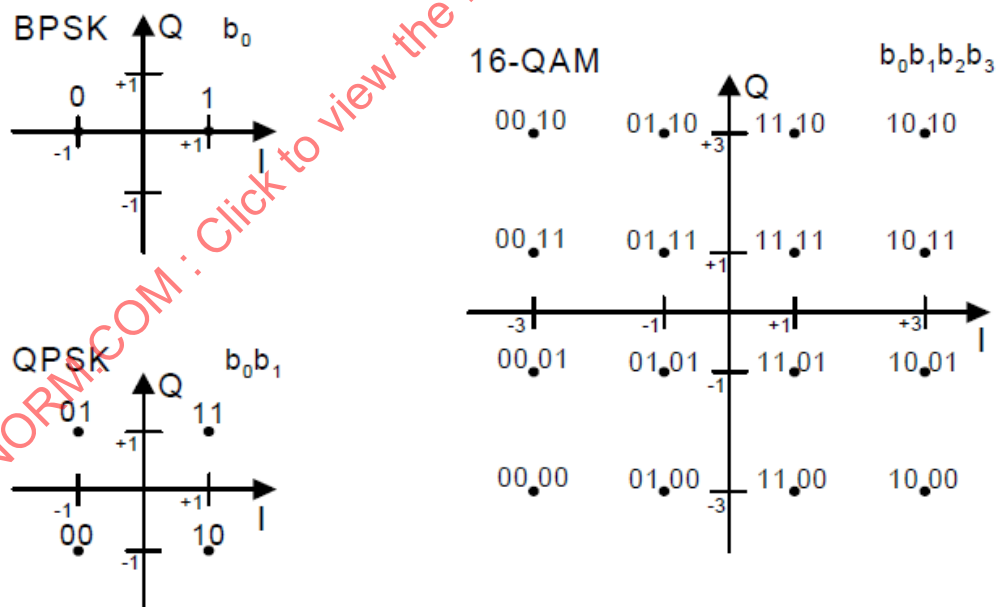


Figure 21-7—Bit-to-symbol mapping for SUN OFDM

The normalization factor,  $K_{MOD}$ , depends on the base modulation mode, as described in Table 21-10. The purpose of the normalization factor is to achieve the same average power for all mappings.

**Table 21-10—Modulation-dependent normalization factor  $K_{MOD}$**

| Modulation | $K_{MOD}$       |
|------------|-----------------|
| BPSK       | 1               |
| QPSK       | $1/(\sqrt{2})$  |
| 16-QAM     | $1/(\sqrt{10})$ |

#### 21.4.3 PIB attribute values for *phySymbolsPerOctet*<sup>15</sup>

The number of symbols per octet depends on both the MCS level and the OFDM option, as represented in Table 21-11.

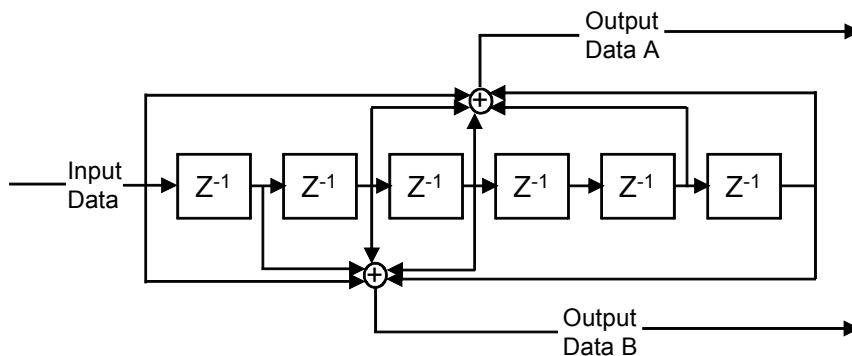
**Table 21-11—*phySymbolsPerOctet* values for SFD OFDM PHY**

| MCS level                                              | OFDM Option |      |     |     |
|--------------------------------------------------------|-------------|------|-----|-----|
|                                                        | 1           | 2    | 3   | 4   |
| MCS0 (BPSK 1/2 rate coded and 4x frequency repetition) | 2/3         | 4/3  | —   | —   |
| MCS1 (BPSK 1/2 rate coded and 2x frequency repetition) | 1/3         | 2/3  | 4/3 | —   |
| MCS2 (QPSK 1/2 rate coded and 2x frequency repetition) | 1/6         | 1/3  | 2/3 | 4/3 |
| MCS3 (QPSK 1/2 rate coded)                             | 1/12        | 1/6  | 1/3 | 2/3 |
| MCS4 (QPSK 3/4 rate coded)                             | —           | 1/9  | 2/9 | 4/9 |
| MCS5 (16-QAM 1/2 rate coded)                           | —           | 1/12 | 1/6 | 1/3 |
| MCS6 (16-QAM 3/4 rate coded)                           | —           | —    | 1/9 | 2/9 |

#### 21.4.4 FEC

The PHY Payload field shall be coded with a convolutional encoder of coding rate  $R = 1/2$  or  $3/4$ , corresponding to the desired data rate. The convolutional encoder shall use the generator polynomials expressed in octal representation,  $g_0 = 133_8$  and  $g_1 = 171_8$ , of rate  $R = 1/2$ , as shown in Figure 21-8. The convolutional encoder shall be initialized to the all zeros state before encoding the PHR and then reset to the all zeros state before encoding the PSDU.

<sup>15</sup>PHY PIB attributes are defined in 11.3.



Convolutional Encoder: Rate  $\frac{1}{2}$ , constraint length  $K=7$   
Octal generator polynomials [133 , 171]

**Figure 21-8—Rate 1/2 convolutional encoder**

The device shall support also coding rates of  $R = 3/4$ , derived by puncturing, as shown in Figure 21-9.

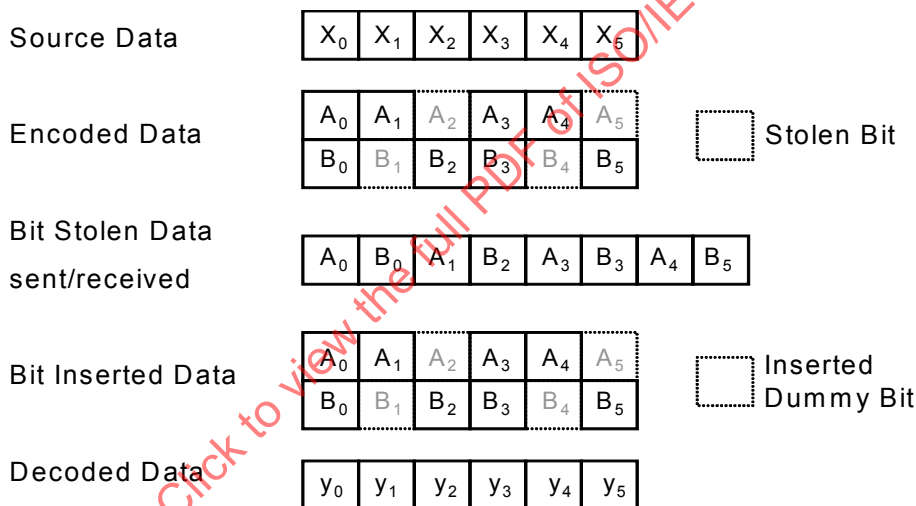


**Figure 21-9—Puncturing for rate 3/4**

#### 21.4.5 Interleaver

The interleaving process consists of two permutations. The index of the coded bit before the first permutation shall be denoted as  $k$ ;  $i$  shall be the index after the first and before the second permutation; and  $j$  shall be the index after the second permutation, just prior to modulation mapping. The coded bits are written at the index given by  $j$ , and read out sequentially. The index  $i$  is defined as follows:

$$i = \left( \frac{N_{cbps}}{N_{row}} \right) \times [k \bmod(N_{row})] + \text{floor}\left(\frac{k}{N_{row}}\right)$$

where

- $N_{cbps}$  is the number of coded bits per symbol before any frequency spreading  
 $k$  is 0, 1, 2, ...,  $(N_{cbps} - 1)$   
 $N_{row}$  is 12 when no frequency spreading is used or 12 / (spreading factor) when spreading is used

The index  $j$  is defined as follows:

$$j = s \times \text{floor}\left(\frac{i}{s}\right) + \left[ i + N_{cbps} - \text{floor}\left(\frac{N_{row} \times i}{N_{cbps}}\right) \right] \text{mod}(s)$$

where

- $N_{cbps}$  is the number of coded bits per symbol before any frequency spreading  
 $i$  is 0, 1, 2, ...,  $N_{cbps}-1$   
 $N_{row}$  is 12 when no frequency spreading is used or 12 / (spreading factor) when spreading is used

and

$$s = \max\left(\frac{N_{bpsc}}{2}, 1\right)$$

where  $N_{bpsc}$  is the number of bits per subcarrier and has the values 1, 2, and 4 for BPSK, QPSK, and 16-QAM, respectively.

Devices shall support an interleaving depth of one symbol, which is associated with a value of zero for the PIB attribute *phyOfdmInterleaving*, as defined in 4.1.3. The values for  $N_{cbps}$  with *phyOfdmInterleaving* set to zero are shown in Table 21-12. In this case,  $N_{cbps}$  is defined as follows: 24, 48, 96, or 192 bits for Option 1; 12, 24, 48, 96, or 192 bits for Option 2; 12, 24, 48, or 96 bits for Option 3; 12, 24, or 48 bits for Option 4.

**Table 21-12— $N_{cbps}$  for SUN OFDM with *phyOfdmInterleaving* = 0**

| MCS level | OFDM<br>Option 1 | OFDM<br>Option 2 | OFDM<br>Option 3 | OFDM<br>Option 4 |
|-----------|------------------|------------------|------------------|------------------|
| MCS0      | 24               | 12               | —                | —                |
| MCS1      | 48               | 24               | 12               | —                |
| MCS2      | 96               | 48               | 24               | 12               |
| MCS3      | 192              | 96               | 48               | 24               |
| MCS4      | —                | 96               | 48               | 24               |
| MCS5      | —                | 192              | 96               | 48               |
| MCS6      | —                | —                | 96               | 48               |

Devices may support an interleaving depth of the number of symbols equal to the frequency domain spreading factor, which is associated with a value of one for the PIB attribute *phyOfdmInterleaving*. The frequency domain spreading factor can be one, two, or four. In this case,  $N_{cbps}$  is defined as follows: 96 bits for BPSK or 192 bits for QPSK in Option 1; 48 bits for BPSK, 96 bits for QPSK, or 192 bits for 16-QAM in



Option 2; 24 bits for BPSK, 48 bits for QPSK, or 96 bits for 16-QAM in Option 3; 24 bits for QPSK or 48 bits for 16-QAM in Option 4.

#### 21.4.6 Frequency spreading

Frequency spreading is a method of replicating PSK symbols on different carriers.

The DFT index 0 is the center of the channel, as defined in 10.1.2.8. The positive DFT indices are mapped to the higher frequencies:

$$\text{center} + N \times \text{tone spacing}$$

where  $N$  is the DFT index.

The negative DFT indices are mapped to the lower frequencies:

$$\text{center} - N \times \text{tone spacing}$$

##### 21.4.6.1 Frequency spreading by 2x

The device shall offer the possibility to create a 2x repetition through frequency spreading.

The spreading is performed by first separating out the data tones from the pilot tones. The data tones are renumbered from  $-N_d/2$  to  $-1$  and  $1$  to  $N_d/2$ , where  $N_d$  is the number of data tones in an OFDM symbol. As an example with Option 3, there are two pilot tones and 24 data tones with indices from  $-13$  to  $13$  excluding the DC tone. Therefore, the data tones are renumbered as  $d_{-12}, d_{-11}, d_{-10}, d_{-9}, d_{-8}, d_{-7}, d_{-6}, d_{-5}, d_{-4}, d_{-3}, d_{-2}, d_{-1}$ , and  $d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}$ . The DC tone is omitted since it is not used in any of the OFDM options.

The data tones to be transmitted in the OFDM symbol are placed into the positive data tones (numbered from  $1$  to  $N_d/2$ ). Phase rotations are applied after copying the data tones to the negative frequencies, in order to reduce the peak-to-average power ratio of the OFDM symbol with frequency spreading. The data tones are as follows:

$$d_{(k-[N_d/2]-1)} = d_k e^{[j2\pi(2 \times k - 1)/4]}$$

where  $k$  is  $1, \dots, N_d/2$ .

##### 21.4.6.2 Frequency spreading by 4x

The device shall offer the possibility to create a 4x repetition through frequency spreading.

As with frequency spreading by 2x, the data tones are separated from the pilot tones and renumbered. The data tones to be transmitted in the OFDM symbol are placed into the lower half of the positive data tones (numbered from  $1$  to  $N_d/4$ ). Phase rotations are applied after copying the data tones to the negative frequencies and upper half of the positive frequencies, in order to reduce the peak-to-average power ratio of the OFDM symbol with frequency spreading.

$$d_{(k+N_d/4)} = d_k e^{[j2\pi(k-1)/4]}$$

where  $k$  is  $1, \dots, N_d/4$ .

$$d_{(k-\lfloor N_d/2 \rfloor - 1)} = d_k e^{[j2\pi(2 \times k - 1)/4]}$$

where  $k$  is 1, ...,  $N_d/4$ .

$$d_{(k-\lfloor N_d/4 \rfloor - 1)} = d_k e^{[j2\pi(3 \times k - 1)/4]}$$

where  $k$  is 1, ...,  $N_d/4$ .

### 21.4.6.3 No spreading

The device shall offer the possibility to map a symbol into tones without frequency spreading.

The data tones to be transmitted in the OFDM symbol are placed into the negative data tones (numbered from  $-N_d/2$  to  $-1$ ) followed by the positive data tones (numbered 1 to  $N_d/2$ ).

### 21.4.7 Pilot tones/null tones

The number of pilot and null tones for each OFDM option are defined as shown in Table 21-13.

**Table 21-13—Number of pilot and null tones for SUN OFDM PHY**

|                | OFDM<br>Option 1 | OFDM<br>Option 2 | OFDM<br>Option 3 | OFDM<br>Option 4 |
|----------------|------------------|------------------|------------------|------------------|
| Active tones   | 104              | 52               | 26               | 14               |
| # Pilot tones  | 8                | 4                | 2                | 2                |
| # Data tones   | 96               | 48               | 24               | 12               |
| #DC null tones | 1                | 1                | 1                | 1                |

The pilot tones shall be transmitted with different shifts in the frequency domain, in order to enable channel estimation when the channel is changing due to Doppler. Immediately after the second LTF, the pilot shifts change every OFDM symbol to the next set. For Options 1, 2, 3, and 4, there are 13, 7, 7, and 4 pilot sets, respectively. Figure 21-10 illustrates how the pilot sets cycle through the sets for Option 1. Figure 21-11 illustrates how the pilot sets cycle through the sets for Options 2 and 3. Figure 21-12 illustrates how the pilot sets cycle through the sets for Option 4. The pilot sets for each option are unique to that option. The long vertical lines show visually when each cycle through the pilots sets is complete.

The DC tone is numbered as 0.

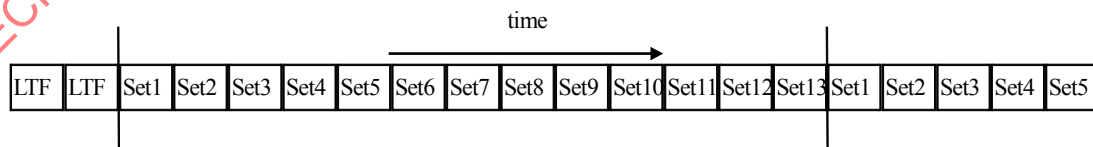


**Figure 21-10—Pilot tone sets for Option 1**

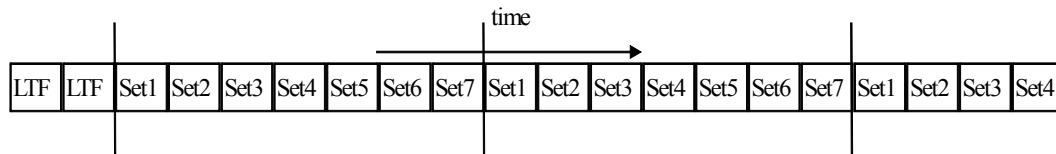


Figure 21-11—Pilot tone sets for Options 2 and 3

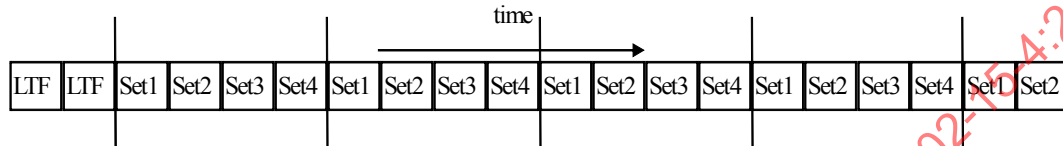


Figure 21-12—Pilot tone sets for Option 4

For Option 1, the device shall use the 13 sets of pilot tones consisting of the subcarriers shown in Table 21-14. The subcarriers for pilot and data are numbered as  $-52$  to  $52$  with the DC tone unused.

Table 21-14—Pilot tones for Option 1

|              |     |     |     |     |    |    |    |    |
|--------------|-----|-----|-----|-----|----|----|----|----|
| Pilot set 1  | −38 | −26 | −14 | −2  | 10 | 22 | 34 | 46 |
| Pilot set 2  | −46 | −34 | −22 | −10 | 2  | 14 | 26 | 38 |
| Pilot set 3  | −42 | −30 | −18 | −6  | 6  | 18 | 30 | 42 |
| Pilot set 4  | −50 | −38 | −26 | −14 | −2 | 10 | 22 | 50 |
| Pilot set 5  | −46 | −34 | −22 | −10 | 2  | 14 | 34 | 46 |
| Pilot set 6  | −42 | −30 | −18 | −6  | 6  | 18 | 26 | 38 |
| Pilot set 7  | −50 | −38 | −26 | −14 | −2 | 30 | 42 | 50 |
| Pilot set 8  | −46 | −34 | −22 | −10 | 10 | 22 | 34 | 46 |
| Pilot set 9  | −42 | −30 | −18 | −6  | 2  | 14 | 26 | 38 |
| Pilot set 10 | −50 | −38 | −26 | 6   | 18 | 30 | 42 | 50 |
| Pilot set 11 | −46 | −34 | −14 | −2  | 10 | 22 | 34 | 46 |
| Pilot set 12 | −42 | −30 | −22 | −10 | 2  | 14 | 26 | 38 |
| Pilot set 13 | −50 | −18 | −6  | 6   | 18 | 30 | 42 | 50 |

For Option 2, the device shall use the seven sets of pilot tones consisting of the subcarriers shown in Table 21-15. The subcarriers for pilot and data are numbered as  $-26$  to  $26$  with the DC tone unused.

**Table 21-15—Pilot tones for Option 2**

|             |     |     |    |    |
|-------------|-----|-----|----|----|
| Pilot set 1 | −14 | −2  | 10 | 22 |
| Pilot set 2 | −22 | −10 | 2  | 14 |
| Pilot set 3 | −18 | −6  | 6  | 18 |
| Pilot set 4 | −26 | −14 | −2 | 26 |
| Pilot set 5 | −22 | −10 | 10 | 22 |
| Pilot set 6 | −18 | −6  | 2  | 14 |
| Pilot set 7 | −26 | 6   | 18 | 26 |

For Option 3, the device shall use the seven sets of pilot tones consisting of the subcarriers shown in Table 21-16. The subcarriers for pilot and data are numbered as  $-13$  to  $13$  with the DC tone unused.

**Table 21-16—Pilot tones for Option 3**

|             |     |    |
|-------------|-----|----|
| Pilot set 1 | −7  | 7  |
| Pilot set 2 | −11 | 3  |
| Pilot set 3 | −3  | 11 |
| Pilot set 4 | −9  | 5  |
| Pilot set 5 | −5  | 9  |
| Pilot set 6 | −13 | 1  |
| Pilot set 7 | −1  | 13 |

For Option 4, the device shall use the four sets of pilot tones consisting of the subcarriers shown in Table 21-17. The subcarriers for pilot and data are numbered as  $-7$  to  $7$  with the DC tone unused.

**Table 21-17—Pilot tones for Option 4**

|             |    |   |
|-------------|----|---|
| Pilot set 1 | −3 | 5 |
| Pilot set 2 | −7 | 1 |
| Pilot set 3 | −5 | 3 |
| Pilot set 4 | −1 | 7 |

The data carried on the pilot tones shall be determined by a pseudo-noise sequence PN9 with the seed “11111111”, as described in 20.4. The first output bit is assigned to the most negative index in Set 1. For example, for Option 3, the first output bit from the PN9 sequence is assigned to the pilot symbol with

index  $-7$  and the second output bit is assigned to the pilot symbol with index  $7$ . Table 21-18 shows the mapping from PN9 bits to the pilot BPSK symbols for all OFDM options and MCS levels. Index  $n$  starts after the LTF from zero and is increased by one every pilot subcarrier.

**Table 21-18—Mapping from PN9 sequence to pilot BPSK symbols**

| Input bit (PN $9_n$ ) | BPSK symbol         |
|-----------------------|---------------------|
| 0                     | $-1 + (0 \times j)$ |
| 1                     | $1 + (0 \times j)$  |

#### 21.4.8 Cyclic prefix (CP)

For the STF, the CP is defined in 21.2.1.3. For the LTF, the CP is defined in 21.2.2.2. For the remaining OFDM symbols, a CP shall be prepended to each base symbol. The duration of the CP ( $24 \mu\text{s}$ ) shall be  $1/4$  of the base symbol ( $96 \mu\text{s}$ ). The CP is a replication of the last  $24 \mu\text{s}$  of the base symbol. The CP is illustrated in Figure 21-13. In Figure 21-13,  $N$  is the number of samples in the base symbol, while  $L$  is the length of the CP that is added to the base symbol.

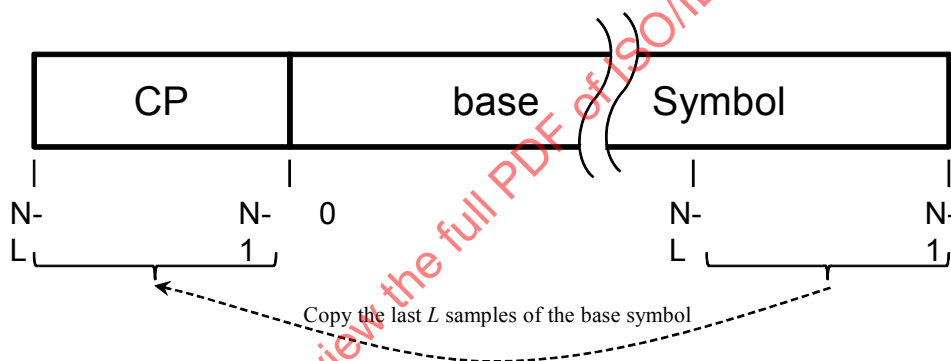


**Figure 21-13—Cyclic prefix (CP)**

#### 21.4.9 PPDU Tail field

The PPDU Tail field shall be six bits of “0,” which are required to return the convolutional encoder to the “zero state.” This procedure reduces the error probability of the convolutional decoder, which relies on future bits when decoding and which may be not be available past the end of the message. The PPDU Tail field shall be produced by replacing six scrambled “zero” bits following the message end with six nonscrambled “zero” bits.

#### 21.4.10 Pad field

The number of bits in the PHY Payload field shall be a multiple of  $N_{\text{cbps}}$ . To achieve that, the length of the message is extended so that it becomes a multiple of  $N_{\text{dbps}}$ , the number of data bits per OFDM symbol; this case is associated with a value of zero for the PIB attribute *phyOfdmInterleaving*, as defined in 11.3. At least six bits are appended to the message, in order to accommodate the PPDU Tail field, as described in 21.4.9. The number of OFDM symbols,  $N_{\text{SYM}}$ , the number of bits in the PHY Payload field,  $N_{\text{PHYPayload}}$ , and the number of bits in the Pad field,  $N_{\text{Pad}}$ , are computed from the length, in octets, of the PSDU (Length is equal to the content of the Frame Length field in Figure 21-5) as follows:

$$N_{SYM} = \text{ceiling}[(8 \times \text{Length} + 6) / N_{dbps}]$$

$$N_{Data} = N_{SYM} \times N_{dbps}$$

$$N_{Pad} = N_{Data} - (8 \times \text{Length} + 6)$$

The function ceiling() returns the smallest integer value greater than or equal to its argument value. The appended bits, the Pad field, are set to zeros and are subsequently scrambled with the rest of the bits in the PHY Payload field.

If a device supports an interleaving depth of the number of symbols equal to the frequency domain spreading factor (SF), which is associated with a value of one for the PIB attribute *phyOfdmInterleaving*, the length of the message is extended so that it becomes a multiple of  $N_{dbps}$ , the number of data bits per SF OFDM symbols, defined as  $N_{dbps} = N_{cbps} \times \text{coding rate (R)}$ , where  $N_{cbps}$  is the number of coded bits per SF OFDM symbols as in 21.4.5. The number of sets of SF OFDM symbols,  $N_{SYMSF}$ , and  $N_{PHYPayload}$  are computed as follows:

$$N_{SYMSF} = SF \times \text{ceiling}[(8 \times \text{Length} + 6) / N_{dbps}]$$

where  $SF$  may be 1, 2, or 4.

$$N_{PHYPayload} = N_{SYMSF} \times N_{dbps} / SF$$

$N_{Pad}$  is computed as before. In the case of the PHR, 36 should be set instead of  $8 \times \text{Length} + 6$  as in 21.2.3.

#### 21.4.11 Scrambler and scrambler seeds

The input to the scrambler is the data bits followed by the PPDU Tail field and then Pad field. The scrambler uses a PN9 sequence that is shown in Figure 17-2. The PN9 scrambler is initialized by one of four seeds. The seed to be used for the scrambler is indicated by two bits in the PHR, as shown in Table 21-19. The leftmost value of the scrambling seed in Table 21-19 is placed into the leftmost delay element in Figure 17-2.

**Table 21-19—Initial seeds to be used for PN9 scrambler**

| MSB scrambler<br>(bit 19 of PHR) <sup>a</sup> | LSB scrambler<br>(bit 20 of PHR) <sup>a</sup> | Scrambling seed   |
|-----------------------------------------------|-----------------------------------------------|-------------------|
| 0                                             | 0                                             | 0 0 0 0 1 0 1 1 1 |
| 1                                             | 0                                             | 0 0 0 0 1 1 1 0 0 |
| 0                                             | 1                                             | 1 0 1 1 1 0 1 1 1 |
| 1                                             | 1                                             | 1 0 1 1 1 1 1 0 0 |

<sup>a</sup>See 21.2.3.

The scrambled bits are found using an XOR operation of each of the input bits with the PN9 sequence:

$$\text{bit}_n = (\text{input bit}_n) \text{ XOR } (\text{PN9}_n)$$

After scrambling, the PPDU Tail field is set to all zeros.

## 21.5 SUN OFDM PHY RF requirements

### 21.5.1 Operating frequency range

The SUN OFDM PHY operates in the following bands:

- 470–510 MHz
- 779–787 MHz
- 863–870 MHz
- 902–928 MHz
- 917–923.5 MHz
- 920–928 MHz
- 2400–2483.5 MHz

### 21.5.2 Transmit power spectral density (PSD) mask

The SUN OFDM transmit PSD mask shall conform with local regulations.

### 21.5.3 Receiver sensitivity

The sensitivity requirements, as described in 10.1.7, for every Option and MCS level are shown in Table 21-20.

**Table 21-20—Sensitivity requirements for OFDM options and MCS levels**

|                                                                  | Option 1 | Option 2 | Option 3 | Option 4 |
|------------------------------------------------------------------|----------|----------|----------|----------|
| MCS0 (BPSK $\frac{1}{2}$ rate coded and 4x frequency repetition) | –103 dBm | –105 dBm | —        | —        |
| MCS1 (BPSK $\frac{1}{2}$ rate coded and 2x frequency repetition) | –100 dBm | –103 dBm | –105 dBm | —        |
| MCS2 (QPSK $\frac{1}{2}$ rate coded and 2x frequency repetition) | –97 dBm  | –100 dBm | –103 dBm | –105 dBm |
| MCS3 (QPSK $\frac{1}{2}$ rate coded)                             | –94 dBm  | –97 dBm  | –100 dBm | –103 dBm |
| MCS4 (QPSK $\frac{3}{4}$ rate coded)                             | —        | –94 dBm  | –97 dBm  | –100 dBm |
| MCS5 (16-QAM $\frac{1}{2}$ rate coded)                           | —        | –91 dBm  | –94 dBm  | –97 dBm  |
| MCS6 (16-QAM $\frac{3}{4}$ rate coded)                           | —        | —        | –91 dBm  | –94 dBm  |

### 21.5.4 Adjacent channel rejection

The definition of an adjacent channel can be found in 11.3.5.

The adjacent channel rejection shall be measured as follows. The desired signal shall be a compliant SUN OFDM PHY signal of pseudo-random data, and the adjacent channel interferer shall be a compliant SUN OFDM PHY signal of pseudo-random data using the same MCS as the desired signal at a power level stronger than the desired signal, as indicated in Table 21-21 for each MCS. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 21.5.3, and the PER shall be as defined in 10.1.7.

**Table 21-21—SUN OFDM adjacent and alternate channel rejection**

| MCS level | Adjacent channel rejection (dB) | Alternate channel rejection (dB) |
|-----------|---------------------------------|----------------------------------|
| 0         | 10                              | 26                               |
| 1         | 10                              | 26                               |
| 2         | 7                               | 23                               |
| 3         | 7                               | 23                               |
| 4         | 5                               | 21                               |
| 5         | 2                               | 18                               |
| 6         | −2                              | 14                               |

**21.5.5 Alternate channel rejection**

The adjacent channels are those on either side of the desired channel that is closest in frequency to the desired channel. The alternate channel is more than one removed from the desired channel in the operational frequency band.

The alternate channel rejection shall be measured as follows. The desired signal shall be a compliant SUN OFDM PHY signal of pseudo-random data, and the alternate channel interferer shall be a compliant SUN OFDM PHY signal of pseudo-random data using the same MCS as the desired signal at a power level stronger than the desired signal, as indicated in Table 21-21, for each MCS. The desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity given in 21.5.3, and the PER shall be as defined in 10.1.7.

**21.5.6 TX-to-RX turnaround time**

The SUN OFDM PHY shall meet the requirements for TX-to-RX turnaround time as defined in 10.2.1.

**21.5.7 RX-to-TX turnaround time**

The SUN OFDM PHY shall meet the requirements for RX-to-TX turnaround time as defined in 10.2.2.

**21.5.8 EVM definition**

The relative constellation RMS error averaged over subcarriers, symbols, and packets shall not exceed the values shown in Table 21-22.

**Table 21-22—EVM requirements for SUN OFDM PHY**

| MCS  | RMS error |
|------|-----------|
| MCS0 | −10 dB    |
| MCS1 | −10 dB    |
| MCS2 | −10 dB    |



**Table 21-22—EVM requirements for SUN OFDM PHY (continued)**

| MCS  | RMS error |
|------|-----------|
| MCS3 | –10 dB    |
| MCS4 | –13 dB    |
| MCS5 | –16 dB    |
| MCS6 | –19 dB    |

The transmit modulation accuracy test shall be performed by instrumentation capable of converting the transmitted signal into a stream of complex samples. The sampled signal shall be processed in a manner similar to an actual receiver, according to the following steps, or an equivalent procedure:

- Detect the start of packet.
- Detect the transition from STF to LTF, and establish fine timing (with one sample resolution).
- Estimate the coarse and fine frequency offsets.
- De-rotate the packet according to estimated frequency offset.
- Estimate the complex channel response coefficients for each of the subcarriers.
- For each data OFDM symbol, transform the symbol into subcarrier received values and divide each subcarrier value with the estimated channel response coefficient.
- For each  $N_d$  data-carrying subcarrier, find the closest constellation point and compute the squared Euclidean distance from it.
- Compute the RMS average of all errors in a packet:

$$\text{RMS}_{\text{error}} = 20 \log_{10} \left( \frac{1}{N_f} \sum_{j=1}^{N_f} \left( \frac{\sum_{k \in U_D} \Delta(i, j, k)^2}{N_d \times N_{\text{SYM}} \times P_0} \right) \right)$$

with

$$\Delta(i, j, k)^2 = [I(i, j, k) - I_0(i, j, k)]^2 + [Q(i, j, k) - Q_0(i, j, k)]^2$$

where

$N_{\text{SYM}}$

$N_f$

$U_D = \{-N_d/2, \dots, -1, 1, \dots, N_d/2\}$

$[I_0((i, j, k), Q_0(i, j, k))]$

$[I((i, j, k), Q(i, j, k))]$

$P_0$

is the number of OFDM symbols in the packet

is the number of packets used for the measurement

is the index set of data tones

denotes the ideal symbol point of the  $i$ th packet,  $j$ th OFDM symbol of the packet, and  $k$ th subcarrier of the OFDM symbol in the complex plane

denotes the observed point of the  $i$ th packet,  $j$ th OFDM symbol of the packet, and  $k$ th tone of the OFDM symbol in the complex plane

is the average power of the constellation

The test shall be performed over at least  $N_f = 20$  packets. The payload of the packets under test shall contain  $N_{SYM} = 16$  OFDM symbols. Random data shall be used for the payload.

#### 21.5.9 Transmit center frequency and symbol tolerance

The transmit center frequency tolerance shall be  $\pm 20 \times 10^{-6}$  maximum. The symbol clock frequency tolerance shall also be  $\pm 20 \times 10^{-6}$  maximum. The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

#### 21.5.10 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.

#### 21.5.11 Receiver maximum input level of desired signal

The SUN OFDM PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

#### 21.5.12 Receiver ED

The SUN OFDM PHY shall provide the receiver ED measurement as described in 10.2.5.

#### 21.5.13 LQI

The SUN OFDM PHY shall provide the LQI measurement as described in 10.2.6.

## 22. SUN O-QPSK PHY

### 22.1 Introduction

The SUN O-QPSK PHY supports multiple PSDU data rates within each supported frequency band, as described in Table 10-1.

For all frequency bands, spreading is obtained by direct sequence spread spectrum (DSSS) applying various spreading factors. For the 780 MHz, 915 MHz, 917 MHz, and 2450 MHz frequency bands, the SUN O-QPSK PHY may support an alternative spreading mode for the PSDU, called multiplexed direct sequence spread spectrum (MDSSS).

For the 780 MHz, 915 MHz, and 2450 MHz frequency bands, the SUN O-QPSK PHY supports communication with legacy devices according to the specifications in Clause 12, as described in 22.4.

An SUN O-QPSK compliant device shall support at least one of the frequency bands designated in Table 10-1.

For the SUN QPSK PHY, the symbol rate is defined as the bit rate of the SHR.

An example of encoding a packet for the SUN O-QPSK PHY is given in “Examples of IEEE Std 802.15.4 PHY encodings” [B6].

### 22.2 PPDU format for SUN O-QPSK

The SUN O-QPSK PPDU shall be formatted as illustrated in Figure 22-1.



Figure 22-1—Format of the SUN O-QPSK PHY PPDU

#### 22.2.1 SHR field format

The SHR field shall be formatted as illustrated in Figure 22-2.

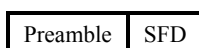


Figure 22-2—Format of the SHR

##### 22.2.1.1 Preamble field format

The Preamble field shall contain a sequence of 56 bits, all zero, for the 780 MHz, 915 MHz, 917 MHz, and 2450 MHz frequency bands. It shall contain a sequence of 32 bits, all zero, for the 470 MHz, 868 MHz, and 920 MHz frequency bands.

**22.2.1.2 SFD field format**

The SFD shall be the sequence described in Figure 22-3.

| SFD value<br>(bits 0–15)        |
|---------------------------------|
| 1 1 1 0 1 0 1 1 0 1 1 0 0 0 1 0 |

**Figure 22-3—Format of the SFD field for the SUN O-QPSK PHY**

**22.2.2 PHR field format**

The format of the PHR is shown in Figure 22-4.

| Bits: 0        | 1–2       | 3–4      | 5–15         | 16–23 |
|----------------|-----------|----------|--------------|-------|
| Spreading Mode | Rate Mode | Reserved | Frame Length | HCS   |

**Figure 22-4—Format of the PHR for SUN O-QPSK**

For the 780 MHz, 915 MHz, 917 MHz, and 2450 MHz frequency bands, the Spreading Mode field shall be set to one if MDSSS is used for PSDU spreading, as described in 22.3.5. Otherwise, the Spreading Mode field shall be set to zero if DSSS is used for PSDU spreading, as described in 22.3.4. For the 470 MHz, 868 MHz, and 920 MHz frequency bands, the Spreading Mode field shall be set to zero, i.e., MDSSS is not supported.

The SUN O-QPSK PHY supports up to four different PSDU rate modes within each frequency band, and the rate mode is given by the Rate Mode field. Table 22-1 shows the mapping of the bit values to the rate mode.

**Table 22-1—Rate mode mapping of the SUN O-QPSK PHY**

| Rate Mode field (b1, b2) | Rate mode |
|--------------------------|-----------|
| (0, 0)                   | 0         |
| (0, 1)                   | 1         |
| (1, 0)                   | 2         |
| (1, 1)                   | 3         |

The Frame Length field specifies the total number of octets contained in the PSDU prior to FEC encoding.

The HCS field is calculated over the first 16 PHR bits, (b0, b1, ..., b15), where b0 is the PHR bit at bit string index 0 and b15 is the PHR bit at bit string index 15, as described in Figure 22-4. The HCS field is defined as follows:

$$HCS = (r_0, r_1, \dots, r_6, r_7)$$

for certain coefficients  $r_0, r_1, \dots, r_6, r_7$ . The computation of those coefficients is shown by the following algorithm.

The HCS shall be calculated using the following standard generator polynomial of degree 8:

$$G_8(x) = x^8 + x^2 + x + 1$$

The HCS shall be calculated as follows:

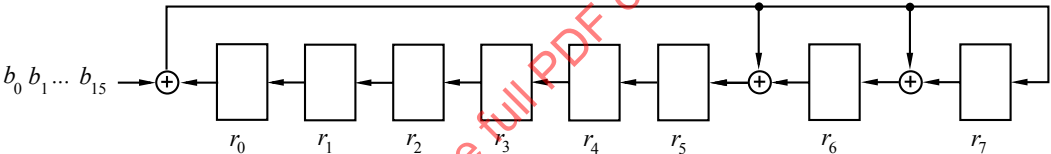
- Let  $M(x) = b_0x^{15} + b_1x^{14} + \dots + b_{14}x + b_{15}$  be the polynomial representing the sequence of bits for which the checksum is to be computed.
- Multiply  $M(x)$  by  $x^8$ , giving the polynomial  $x^8 \times M(x)$ .
- Divide modulo 2 by the generator polynomial,  $G_8(x)$ , to obtain the remainder polynomial,  $R(x) = r_0x^7 + r_1x^6 + \dots + r_6x + r_7$ .

The HCS field is given by the coefficients of the remainder polynomial. An example HCS is shown in Figure 22-5.

| Bits: 0 | 1 | 2 | 3 | 4 | 5–15                  | 16–23           |
|---------|---|---|---|---|-----------------------|-----------------|
| 0       | 0 | 1 | 0 | 0 | 0 0 0 0 0 1 0 1 0 1 0 | 0 1 1 1 1 0 0 0 |

Figure 22-5—Example HCS for SUN O-QPSK

A typical implementation is depicted in Figure 22-6.



- 1) Initialize the remainder register ( $r_0, r_1, \dots, r_7$ ) to zero.
- 2) Shift the sequence  $b_0, b_1, \dots, b_{15}$  into the divider beginning with  $b_0$ .
- 3) After the last bit,  $b_{15}$ , is shifted into the divider, the remainder register contains the HCS:  $(r_0, r_1, \dots, r_7) = (b_{16}, b_{17}, \dots, b_{23})$

Figure 22-6—Typical HCS implementation for SUN O-QPSK

### 22.2.3 PHY Payload field

The PHY Payload field contains the encoded PSDU.

## 22.3 Modulation and coding for SUN O-QPSK

### 22.3.1 Reference modulator

Figure 22-7 shows the reference modulator diagram for the SUN O-QPSK PHY.

The inputs to the reference modulator are the bit sequences of the SHR field, the PHR field, and the PSDU field. Processing of the SHR and PHR gives corresponding chip sequences  $c_{SHR}$ , as described in 22.3.2, and  $c_{PHR}$ , as described in 22.3.3, respectively. The bits of the PSDU field are processed by a dedicated signal flow depending on spreading mode, as described in 22.3.4 and 22.3.5. In either case, the corresponding chip

sequences will be extended by pilots, as described in 22.3.12, resulting in a final PSDU chip sequence  $c_{PSDU}$ . The concatenated sequence of chips belonging to the PPDU

$$c_{PPDU} = \{c_{SHR}, c_{PHR}, c_{PSDU}\}$$

shall be O-QPSK modulated, as described in 22.3.13.

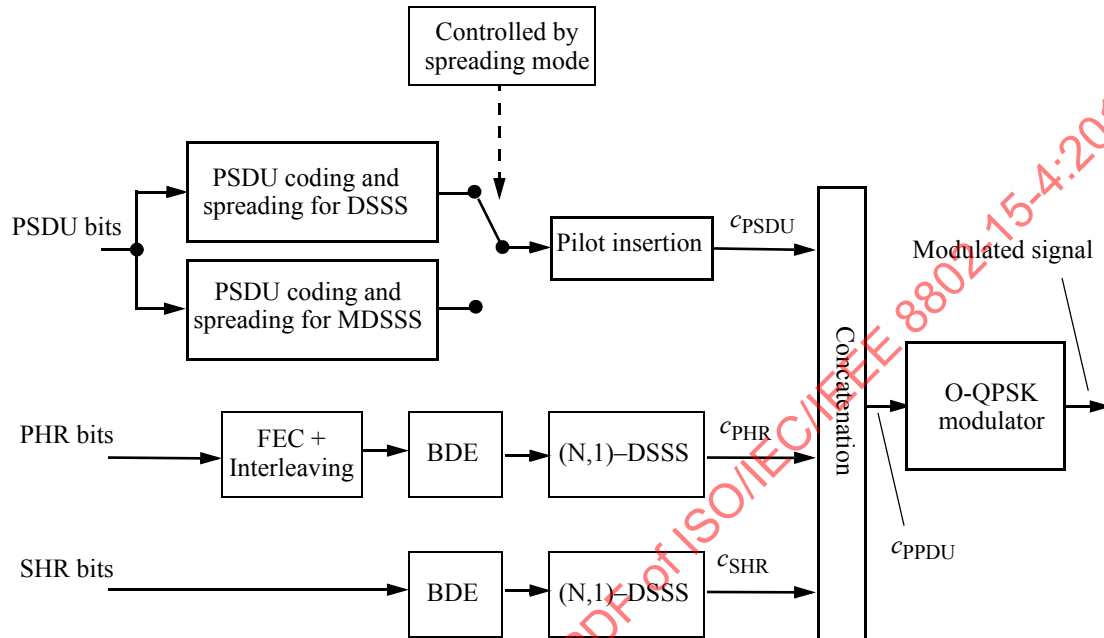


Figure 22-7—Reference modulator diagram

### 22.3.2 SHR coding and spreading

For the SHR bits, bit differential encoding (BDE), as described in 22.3.8, and subsequently  $(N,1)$ -DSSS shall be applied, as described in 22.3.9.

Table 22-2 shows the spreading parameters of  $(N,1)$ -DSSS bit-to-chip mapping.

Table 22-2—SHR coding and spreading parameters

| Frequency band (MHz) | Chip rate (kchip/s) | BDE | Spreading mode   |
|----------------------|---------------------|-----|------------------|
| 470–510              | 100                 | yes | $(32,1)_0$ -DSSS |
| 779–787              | 1000                | yes | $(64,1)$ -DSSS   |
| 868–870              | 100                 | yes | $(32,1)_0$ -DSSS |
| 902–928              | 1000                | yes | $(64,1)$ -DSSS   |
| 917–923.5            | 1000                | yes | $(64,1)$ -DSSS   |
| 920–928              | 100                 | yes | $(32,1)_0$ -DSSS |
| 2400–2483.5          | 2000                | yes | $(128,1)$ -DSSS  |

### 22.3.3 PHR coding and spreading

The PHR field, as described in Figure 22-4, shall be processed using FEC, as described in 22.3.6, and interleaving, as described in 22.3.7, resulting in 60 interleaved code-bits. For the interleaved PHR code-bits, BDE, as described in 22.3.8, and subsequently  $(N,1)$ -DSSS shall be applied, as described in 22.3.9.

Table 22-3 shows the spreading parameters of  $(N,1)$ -DSSS bit-to-chip mapping.

**Table 22-3—PHR coding and spreading parameters**

| Frequency band (MHz) | Chip rate (kchip/s) | BDE | rate $\frac{1}{2}$ FEC + interleaver | Spreading mode       |
|----------------------|---------------------|-----|--------------------------------------|----------------------|
| 470–510              | 100                 | yes | yes                                  | $(8,1)_{0/1}$ -DSSS  |
| 779–787              | 1000                | yes | yes                                  | $(16,1)_{0/1}$ -DSSS |
| 868–870              | 100                 | yes | yes                                  | $(8,1)_{0/1}$ -DSSS  |
| 902–928              | 1000                | yes | yes                                  | $(16,1)_{0/1}$ -DSSS |
| 917–923.5            | 1000                | yes | yes                                  | $(16,1)_{0/1}$ -DSSS |
| 920–928              | 100                 | yes | yes                                  | $(8,1)_{0/1}$ -DSSS  |
| 2400–2483.5          | 2000                | yes | yes                                  | $(32,1)_{0/1}$ -DSSS |

### 22.3.4 PSDU coding and spreading for DSSS

Figure 22-8 shows the signal flow when DSSS is applied to the PSDU (spreading mode set to DSSS).

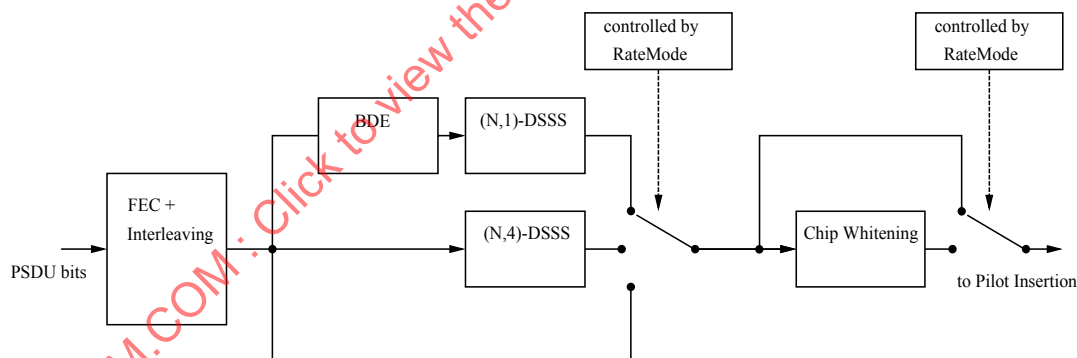


**Figure 22-8—PSDU processing for DSSS**

The supported PSDU parameters for spreading mode DSSS are shown in Table 22-4. An SUN O-QPSK compliant device shall implement at least rate mode zero with spreading mode set to DSSS, as described in 22.3.4. All other possible combinations of rate mode and spreading mode, as described in 22.3.4 and 22.3.5, are optional.

The PSDU information bit shall be first processed using FEC as described in 22.3.6, delivering a sequence of code-bits. The code-bits shall be interleaved as described in 22.3.7. Depending on the frequency band and rate mode, spreading by DSSS with different spreading factors shall be applied.

**Table 22-4—PSDU parameters for spreading mode DSSS**

| Frequency band (MHz) | Chip rate (kchip/s) | Rate mode | BDE | Spreading mode              | rate $\frac{1}{2}$ FEC + interleaver | Data rate (kb/s) |
|----------------------|---------------------|-----------|-----|-----------------------------|--------------------------------------|------------------|
| 470–510              | 100                 | 0         | yes | (8,1) <sub>0/1</sub> -DSSS  | yes                                  | 6.25             |
|                      |                     | 1         | yes | (4,1)-DSSS                  | yes                                  | 12.5             |
|                      |                     | 2         | yes | (2,1)-DSSS                  | yes                                  | 25               |
|                      |                     | 3         | no  | none                        | yes                                  | 50               |
| 779–787              | 1000                | 0         | yes | (16,1) <sub>0/1</sub> -DSSS | yes                                  | 31.25            |
|                      |                     | 1         | no  | (16,4)-DSSS                 | yes                                  | 125              |
|                      |                     | 2         | no  | (8,4)-DSSS                  | yes                                  | 250              |
|                      |                     | 3         | no  | none                        | yes                                  | 500              |
| 868–870              | 100                 | 0         | yes | (8,1) <sub>0/1</sub> -DSSS  | yes                                  | 6.25             |
|                      |                     | 1         | yes | (4,1)-DSSS                  | yes                                  | 12.5             |
|                      |                     | 2         | yes | (2,1)-DSSS                  | yes                                  | 25               |
|                      |                     | 3         | no  | none                        | yes                                  | 50               |
| 902–928              | 1000                | 0         | yes | (16,1) <sub>0/1</sub> -DSSS | yes                                  | 31.25            |
|                      |                     | 1         | no  | (16,4)-DSSS                 | yes                                  | 125              |
|                      |                     | 2         | no  | (8,4)-DSSS                  | yes                                  | 250              |
|                      |                     | 3         | no  | none                        | yes                                  | 500              |
| 917–923.5            | 1000                | 0         | yes | (16,1) <sub>0/1</sub> -DSSS | yes                                  | 31.25            |
|                      |                     | 1         | no  | (16,4)-DSSS                 | yes                                  | 125              |
|                      |                     | 2         | no  | (8,4)-DSSS                  | yes                                  | 250              |
|                      |                     | 3         | no  | none                        | yes                                  | 500              |
| 920–928              | 100                 | 0         | yes | (8,1) <sub>0/1</sub> -DSSS  | yes                                  | 6.25             |
|                      |                     | 1         | yes | (4,1)-DSSS                  | yes                                  | 12.5             |
|                      |                     | 2         | yes | (2,1)-DSSS                  | yes                                  | 25               |
|                      |                     | 3         | no  | none                        | yes                                  | 50               |
| 2400–2483.5          | 2000                | 0         | yes | (32,1) <sub>0/1</sub> -DSSS | yes                                  | 31.25            |
|                      |                     | 1         | no  | (32,4)-DSSS                 | yes                                  | 125              |
|                      |                     | 2         | no  | (16,4)-DSSS                 | yes                                  | 250              |
|                      |                     | 3         | no  | (8,4)-DSSS                  | yes                                  | 500              |

The first DSSS method applies BDE, as described in 22.3.8, to the interleaved code-bits and subsequently the  $(N,1)$ -bit-to-chip mapping as described in 22.3.9. The second DSSS method applies  $(N,4)$ -bit-to-chip mapping to the interleaved code-bits as described in 22.3.9. In this case, BDE shall not be applied, as described in Table 22-4. The highest PSDU data rate is obtained by bypassing BDE and spreading, as described in Figure 22-8 and Table 22-4.



When applying (N,4)-DSSS, the sequence of interleaved PSDU code-bits shall be extended by appending a minimum number of pad bits, such that the length of the extended interleaved PSDU code-bit sequence is a multiple of four. The pad bits shall be set to zero.

Depending on the frequency band and rate mode, the output sequence of the bit-to-chip mapper shall be whitened, as described in 22.3.11.

The relationship between the rate mode and the DataRate parameter of the MCPS-DATA.request primitive is described in Table 8-75.

### 22.3.5 PSDU coding and spreading for MDSSS

Figure 22-9 shows the signal flow when MDSSS is applied to the PSDU (spreading mode set to MDSSS).

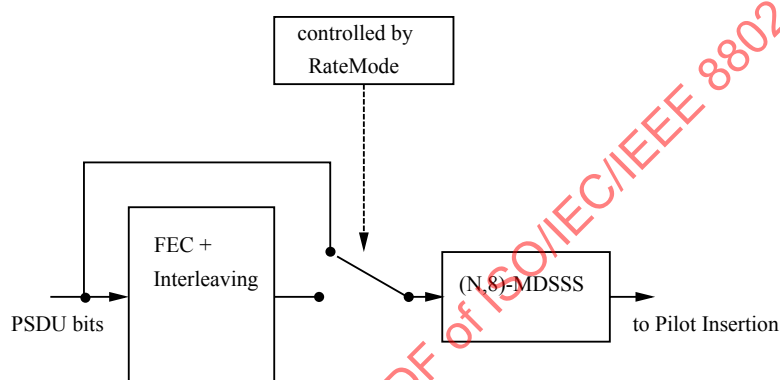


**Figure 22-9—PSDU processing for MDSSS**

The supported PSDU parameters for spreading mode MDSSS are shown in Table 22-5. The combinations of rate mode and spreading mode described in this subclause are optional.

The use of FEC depends on the rate mode chosen, as described in Table 22-5. When FEC is used, the PSDU information bits shall be first processed using FEC, as described in 22.3.6, delivering a sequence of code-bits. When FEC is enabled, the code-bits shall be interleaved, as described in 22.3.7; otherwise, interleaving is bypassed. The rate mode also determines which (N,8)-MDSSS spreading factor shall be used, as described in Table 22-5 and 22.3.10.

**Table 22-5—PSDU parameters for spreading mode MDSSS**

| Frequency band (MHz) | Chip rate (kchip/s) | Rate mode | BDE | Spreading mode | rate $\frac{1}{2}$ FEC + interleaver | Data rate (kb/s) |
|----------------------|---------------------|-----------|-----|----------------|--------------------------------------|------------------|
| 470–510              | not supported       |           |     |                |                                      |                  |
| 779–787              | 1000                | 0         | no  | (64,8)-MDSSS   | yes                                  | 62.5             |
|                      |                     | 1         | no  | (32,8)-MDSSS   | yes                                  | 125              |
|                      |                     | 2         | no  | (32,8)-MDSSS   | no                                   | 250              |
|                      |                     | 3         | no  | (16,8)-MDSSS   | no                                   | 500              |
| 868–870              | not supported       |           |     |                |                                      |                  |

**Table 22-5—PSDU parameters for spreading mode MDSSS (continued)**

| Frequency band (MHz) | Chip rate (kchip/s) | Rate mode | BDE | Spreading mode | rate ½<br>FEC +<br>interleaver | Data rate (kb/s) |
|----------------------|---------------------|-----------|-----|----------------|--------------------------------|------------------|
| 902–928              | 1000                | 0         | no  | (64,8)-MDSSS   | yes                            | 62.5             |
|                      |                     | 1         | no  | (32,8)-MDSSS   | yes                            | 125              |
|                      |                     | 2         | no  | (32,8)-MDSSS   | no                             | 250              |
|                      |                     | 3         | no  | (16,8)-MDSSS   | no                             | 500              |
| 917–923.5            | 1000                | 0         | no  | (64,8)-MDSSS   | yes                            | 62.5             |
|                      |                     | 1         | no  | (32,8)-MDSSS   | yes                            | 125              |
|                      |                     | 2         | no  | (32,8)-MDSSS   | no                             | 250              |
|                      |                     | 3         | no  | (16,8)-MDSSS   | no                             | 500              |
| 920–928              | not supported       |           |     |                |                                |                  |
| 2400–2483.5          | 2000                | 0         | no  | (128,8)-MDSSS  | yes                            | 62.5             |
|                      |                     | 1         | no  | (64,8)-MDSSS   | yes                            | 125              |
|                      |                     | 2         | no  | (64,8)-MDSSS   | no                             | 250              |
|                      |                     | 3         | no  | (32,8)-MDSSS   | no                             | 500              |

When applying (N,8)-MDSSS, the sequence of interleaved PSDU code-bits shall be extended by appending a minimum number of pad bits, such that the length of the extended interleaved PSDU code-bit sequence is a multiple of 8. The pad bits shall be set to zero.

The relationship between the spreading mode variable and the DataRate parameter of the MCPS-DATA.request primitive is described in Table 8-75.

### 22.3.6 FEC

FEC shall be applied to the bits of the PHR field.

For spreading mode set to DSSS, FEC shall be applied to the PSDU bits, as described in Table 22-4. For spreading mode set to MDSSS, FEC is enabled depending on rate mode, as described in Table 22-5.

When used, FEC shall employ rate 1/2 convolutional coding with constraint length  $K = 7$  using the following generator polynomials:

$$G_0(x) = 1 + x^2 + x^3 + x^5 + x^6$$

$$G_1(x) = 1 + x + x^2 + x^3 + x^6$$

The encoder is shown in Figure 22-10, where  $\oplus$  denotes modulo-2 addition.

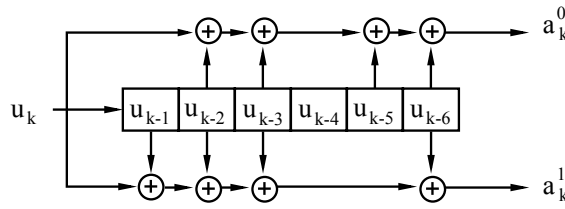


**Figure 22-10—Convolutional encoder**

Prior to convolutional encoding of the PHR information bits, as described in 22.2.2, the initial encoder state at  $k = 0$  shall be set to

$$(u_{-1}, u_{-2}, \dots, u_{-6}) = (0, 0, 0, 0, 0, 0)$$

and the PHR information bit sequence shall be extended by a termination sequences of 6 zero bits as shown in Figure 22-11.



**Figure 22-11—PHR and PSDU extension prior to encoding**

Prior to the convolutional encoding of the PSDU, the sequence of PSDU information bits shall be extended by appending a termination sequence of six bits, all zero, and a sequence of additional pad bits as shown in Figure 22-11. The pad bits shall be set to zero and the number of pad bits,  $N_{\text{PAD}}$ , is computed from the number of blocks,  $N_B$ , the total number of uncoded bits,  $N_D$ , and the interleaver depth,  $N_{\text{INTRLV}}$ , as follows:

$$N_B = \text{ceiling}((8 \times \text{LENGTH} + 6) / (N_{\text{INTRLV}} / 2))$$

$$N_D = N_B \times (N_{\text{INTRLV}} / 2)$$

$$N_{\text{PAD}} = N_D - (8 \times \text{LENGTH} + 6)$$

The function  $\text{ceiling}(\cdot)$  is a function that returns the smallest integer value greater than or equal to its argument value.

The sequence of extended information bits according to Figure 22-11 shall be passed to the convolutional encoder. The corresponding output sequence of code-bits,  $z$ , shall be generated as follows:

$$z = \{\dots, a_k^0, a_k^1, a_{k+1}^0, a_{k+1}^1, a_{k+2}^0, a_{k+2}^1, \dots\} = \{z_0, z_1, \dots, z_{2N_D+59}\}$$

i.e.,  $a_k^0$  is preceding sample  $a_k^1$ . The first sample,  $z_0$ , shall be passed to the interleaver first in time, and the last sample,  $z_{2N_D+59}$ , shall be passed to the interleaver last in time.

The number of code-bits referring to a single interleaving block,  $N_{\text{INTRLV}}$ , is defined in 22.3.7.

**22.3.7 Code-bit interleaving**

Interleaving of PHR code-bits shall be employed and is separated from the interleaving of the PSDU code-bits. Since the PHR information bits are terminated, PHR code-bits and PSDU code-bits are independent code blocks.

Interleaving of PSDU code-bits shall be employed in conjunction with PSDU FEC, in order to improve robustness against burst errors and to break correlation of consecutive bits when applying  $(N,4)$  or  $(N,8)$  bit-to-chip mapping. No PSDU code-bit interleaving shall be employed if PSDU FEC is not used, as described in Table 22-5.

The sequence of PHR code-bits consists of a single sequence

$$z^0 = \{z_0^0, \dots, z_{N_{\text{INTRLV}}-1}^0\}$$

of length  $N_{\text{INTRLV}} = 60$ .

The sequence of PSDU code-bits consists of  $N_B$  subsequences

$$z^j = \{z_0^j, \dots, z_{N_{\text{INTRLV}}-1}^j\} = \{z_{(j-1)N_{\text{INTRLV}}+60}, \dots, z_{jN_{\text{INTRLV}}+59}\} \text{ for } j = 1, \dots, N_B$$

where

$N_B$  is the number of blocks, as defined in 22.3.6

$N_{\text{INTRLV}}$  is shown in Table 22-6

In either case, the interleaver is defined by a permutation. The index of the code-bits before the permutation shall be denoted by  $k$ , where  $k = 0$  refers to the first sample,  $z_0^j$ , and  $k = N_{\text{INTRLV}} - 1$  refers to the last sample,  $z_{N_{\text{INTRLV}}-1}^j$ , passed to the interleaver for a given subsequence  $z^j$ . The index  $i$  shall be the index after the permutation. The permutation is defined by the following rule:

$$i = \frac{N_{\text{INTRLV}}}{\lambda} \times ((N_{\text{INTRLV}} - 1 - k) \bmod \lambda) + \text{floor}\left(\frac{N_{\text{INTRLV}} - 1 - k}{\lambda}\right) \quad k = 0, \dots, N_{\text{INTRLV}} - 1$$

where the degree  $\lambda$  is given in Table 22-6. The function  $\text{floor}(\cdot)$  is a function that returns the largest integer value less than or equal to its argument value.

**Table 22-6—Parameters of the interleaver**

|      | Degree $\lambda$ | Depth $N_{\text{INTRLV}}$ |
|------|------------------|---------------------------|
| PHR  | 6                | $10 \times 6 = 60$        |
| PSDU | 7                | $7 \times 18 = 126$       |

The process of interleaving a subsequence is shown in Figure 22-12. The first subsequence,  $z^0$ , shall be processed first in time and the last subsequence,  $z^{N_B}$ , shall be processed last in time.

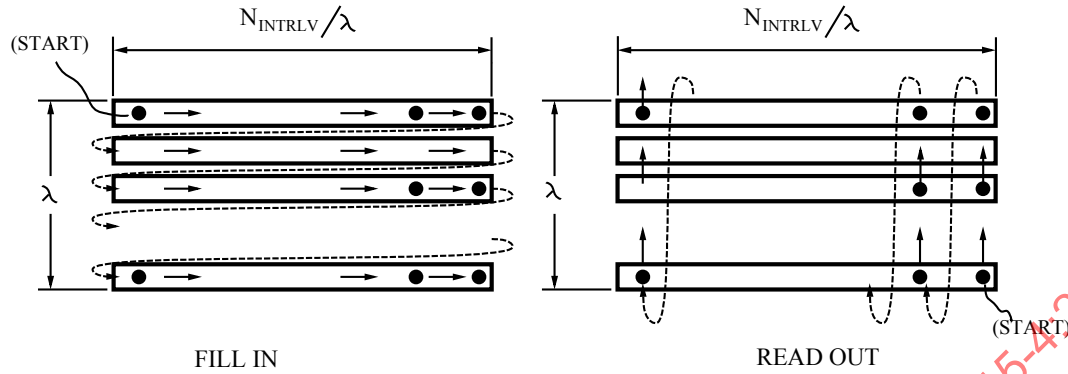


Figure 22-12—Interleaver

The deinterleaver, which performs the inverse relation, is defined by the following rule:

$$k = \lambda \times (N_{\text{INTRLV}} - 1 - i) - (N_{\text{INTRLV}} - 1) \times \text{floor}\left(\frac{\lambda \times (N_{\text{INTRLV}} - 1 - i)}{N_{\text{INTRLV}}}\right) \quad i = 0, \dots, N_{\text{INTRLV}} - 1$$

### 22.3.8 Bit differential encoding (BDE)

In conjunction with  $(N,1)$ -DSSS, BDE supports noncoherent detection, which is beneficial for robust operation at low-chip SNR.

BDE is the modulo-2 addition of a raw bit with the previous encoded bit. This is performed by the transmitter and can be described as follows:

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

where

- $R_n$  is the raw bit being encoded
- $E_n$  is the corresponding differentially encoded bit
- $E_{n-1}$  is the previous differentially encoded bit

BDE shall be applied to the bits of the SHR field resulting in a sequence  $\{E_0^{SHR}, E_1^{SHR}, \dots, E_{N_{SHR}-1}^{SHR}\}$  of differentially encoded bits, where  $N_{SHR}$  is the total number of bits in the SHR, as defined in 22.2.1.1 and 22.2.1.2. The initial state,  $E_{-1}^{SHR}$ , shall be zero.

BDE shall be applied to the 60 interleaved PHR code-bits, resulting in a sequence  $\{E_0^{PHR}, E_1^{PHR}, \dots, E_{59}^{PHR}\}$  of differentially encoded bits. The initial state,  $E_{-1}^{PHR}$ , is assumed to be  $E_{N_{SHR}-1}^{SHR}$ , assuring that during noncoherent differential detection, the very first interleaved PHR code-bit can be referenced to the last SHR bit.

If differential encoding is enabled, depending on the frequency band and rate mode, as described in Table 22-4, the sequence of differentially encoded PSDU bits shall be computed as follows. Let  $R = \{R_0, R_1, \dots, R_{2N_D-1}\}$  be the sequence of interleaved PSDU code-bits obtained by FEC and interleaving. The pilot spacing ratio,  $M$ , is calculated using the following:

$$M = \frac{M_P}{N}$$

where

$M_P$  is the pilot spacing, as described in Table 22-20

$N$  is the parameter of  $(N, 1)$ -DSSS, as described in Table 22-4

Note that  $M$  is always an integer value. Let  $E_n$  be defined as shown:

$$E_n = \begin{cases} R_n \oplus E_{n-1}, & (n \bmod M) \neq 0 \\ R_n \oplus 0, & (n \bmod M) = 0 \end{cases}$$

generating a sequence  $\{E_0^{PSDU}, E_1^{PSDU}, \dots, E_{2N_D-1}^{PSDU}\}$  of differentially encoded PSDU bits.

Referencing to zero for  $(n \bmod M) = 0$  assures that, during noncoherent differential detection, the very first interleaved PSDU code-bit subsequent to a pilot sequence  $p$ , as described in 22.3.13, can be referenced to the pilot sequence.

If BDE is not applied to the PSDU, the frequency band and rate mode, as described in Table 22-4 and Table 22-5, determine whether the sequence of interleaved PSDU code-bits (FEC is enabled) or the raw information PSDU bits (FEC is not enabled) remain unchanged.

### 22.3.9 DSSS bit-to-chip mapping

For  $(N,1)$ -DSSS, a single bit is mapped to a sequence of  $N$  binary valued chips. The number of chips,  $N$ , depends on the frequency band and rate mode, as described in Table 22-4. This mapping defines a binary  $(N,x)$  block code with  $x = 1$ .

Table 22-7 through Table 22-13 show  $(N,1)$ -DSSS used in the SUN O-QPSK PHY. For  $N = 1$ , the chip value is equal to the input bit value (no spreading).

**Table 22-7—(2,1)-DSSS bit-to-chip mapping**

| Input bit | Chip values ( $c_0$ $c_1$ ) |
|-----------|-----------------------------|
| 0         | 10                          |
| 1         | 01                          |

**Table 22-8—(4,1)-DSSS bit-to-chip mapping**

| Input bit | Chip values ( $c_0$ $c_1$ ... $c_3$ ) |
|-----------|---------------------------------------|
| 0         | 1010                                  |
| 1         | 0101                                  |

**Table 22-9— $(8,1)_k$ -DSSS bit-to-chip mapping**

| k | Input bit | Chip values ( $c_0$ $c_1$ ... $c_7$ ) |
|---|-----------|---------------------------------------|
| 0 | 0         | 1011 0001                             |
|   | 1         | 0100 1110                             |
| 1 | 0         | 0110 0011                             |
|   | 1         | 1001 1100                             |

**Table 22-10— $(16,1)_k$ -DSSS bit-to-chip mapping**

| k | Input bit | Chip values ( $c_0$ $c_1$ ... $c_{15}$ ) |
|---|-----------|------------------------------------------|
| 0 | 0         | 0010 0011 1101 0110                      |
|   | 1         | 1101 1100 0010 1001                      |
| 1 | 0         | 0100 0111 1010 1100                      |
|   | 1         | 1011 1000 0101 0011                      |

**Table 22-11— $(32,1)_k$ -DSSS bit-to-chip mapping**

| k | Input bit | Chip values ( $c_0$ $c_1$ ... $c_{31}$ ) |
|---|-----------|------------------------------------------|
| 0 | 0         | 1101 1110 1010 0010 0111 0000 0110 0101  |
|   | 1         | 0010 0001 0101 1101 1000 1111 1001 1010  |
| 1 | 0         | 1110 1111 0101 0001 0011 1000 0011 0010  |
|   | 1         | 0001 0000 1010 1110 1100 0111 1100 1101  |

**Table 22-12— $(64,1)$ -DSSS bit-to-chip mapping**

| Input bit | Chip values ( $c_0$ $c_1$ ... $c_{63}$ ) |
|-----------|------------------------------------------|
| 0         | 1011 0010 0010 0101 1011 0001 1101 0000  |
|           | 1101 0111 0011 1101 1111 0000 0010 1010  |
| 1         | 0100 1101 1101 1010 0100 1110 0010 1111  |
|           | 0010 1000 1100 0010 0000 1111 1101 0101  |

**Table 22-13—(128,1)-DSSS bit-to-chip mapping**

| Input bit | Chip values ( $c_0 c_1 \dots c_{127}$ )                                                                                                                                  |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0         | 1001 1000 1000 1011 0100 1110 0100 0010<br>0101 0010 0110 1101 1100 0111 1010 0000<br>1101 0100 0110 0101 1101 1000 0111 0101<br>1110 0111 1101 1111 1000 0000 1010 1011 |
| 1         | 0110 0111 0111 0100 1011 0001 1011 1101<br>1010 1101 1001 0010 0011 1000 0101 1111<br>0010 1011 1001 1010 0010 0111 1000 1010<br>0001 1000 0010 0000 0111 1111 0101 0100 |

Note that for  $N$  greater than one,  $(N,1)$ -DSSS is always preceded by differential encoding, supporting noncoherent detection of the interleaved code-bits, as described in Table 22-2, Table 22-3, and Table 22-4.

For  $N$  equal to 8, 16, and 32, two spreading codes are defined,  $(N,1)_0$ -DSSS and  $(N,1)_1$ -DSSS. When applied to either the PHR or the PSDU, the two spreading codes are applied in an alternating manner, denoted as  $(N,1)_{0/1}$ -DSSS. The even indexed bits of the PHR and PSDU,  $E_{2k}$ , for  $k=0, 1, \dots$ , shall be spread with  $(N,1)_0$ -DSSS and the odd indexed bits,  $E_{2k+1}$ , for  $k=0, 1, \dots$ , shall be spread with  $(N,1)_1$ -DSSS.

When applying  $(N,4)$ -DSSS, a 4-tuple of bits is mapped to a sequence of  $N$  binary valued chips. This mapping defines a binary  $(N, x)$  block code with  $x = 4$ .

Table 22-14, Table 22-15, and Table 22-16 show  $(N,4)$ -DSSS supported by the SUN O-QPSK PHY.

**Table 22-14—(8,4)-DSSS bit-to-chip mapping**

| Input bits (b0 b1 b2 b3) | Chip values ( $c_0 c_1 \dots c_7$ ) |
|--------------------------|-------------------------------------|
| 0000                     | 0000 0001                           |
| 1000                     | 1101 0000                           |
| 0100                     | 0110 1000                           |
| 1100                     | 1011 1001                           |
| 0010                     | 1110 0101                           |
| 1010                     | 0011 0100                           |
| 0110                     | 1000 1100                           |
| 1110                     | 0101 1101                           |
| 0001                     | 1010 0010                           |
| 1001                     | 0111 0011                           |
| 0101                     | 1100 1011                           |
| 1101                     | 0001 1010                           |



**Table 22-14—(8,4)-DSSS bit-to-chip mapping (continued)**

| Input bits (b0 b1 b2 b3) | Chip values (c <sub>0</sub> c <sub>1</sub> ... c <sub>7</sub> ) |
|--------------------------|-----------------------------------------------------------------|
| 0011                     | 0100 0110                                                       |
| 1011                     | 1001 0111                                                       |
| 0111                     | 0010 1111                                                       |
| 1111                     | 1111 1110                                                       |

**Table 22-15—(16,4)-DSSS bit-to-chip mapping**

| Input bits (b0 b1 b2 b3) | Chip values (c <sub>0</sub> c <sub>1</sub> ... c <sub>15</sub> ) |
|--------------------------|------------------------------------------------------------------|
| 0000                     | 0011 1110 0010 0101                                              |
| 1000                     | 0100 1111 1000 1001                                              |
| 0100                     | 0101 0011 1110 0010                                              |
| 1100                     | 1001 0100 1111 1000                                              |
| 0010                     | 0010 0101 0011 1110                                              |
| 1010                     | 1000 1001 0100 1111                                              |
| 0110                     | 1110 0010 0101 0011                                              |
| 1110                     | 1111 1000 1001 0100                                              |
| 0001                     | 0110 1011 0111 0000                                              |
| 1001                     | 0001 1010 1101 1100                                              |
| 0101                     | 0000 0110 1011 0111                                              |
| 1101                     | 1100 0001 1010 1101                                              |
| 0011                     | 0111 0000 0110 1011                                              |
| 1011                     | 1101 1100 0001 1010                                              |
| 0111                     | 1011 0111 0000 0110                                              |
| 1111                     | 1010 1101 1100 0001                                              |

**Table 22-16—(32,4)-DSSS bit-to-chip mapping**

| Input bits (b0 b1 b2 b3) | Chip values (c <sub>0</sub> c <sub>1</sub> ... c <sub>31</sub> ) |
|--------------------------|------------------------------------------------------------------|
| 0000                     | 1101 1001 1100 0011 0101 0010 0010 1110                          |
| 1000                     | 1110 1101 1001 1100 0011 0101 0010 0010                          |
| 0100                     | 0010 1110 1101 1001 1100 0011 0101 0010                          |
| 1100                     | 0010 0010 1110 1101 1001 1100 0011 0101                          |
| 0010                     | 0101 0010 0010 1110 1101 1001 1100 0011                          |
| 1010                     | 0011 0101 0010 0010 1110 1101 1001 1100                          |

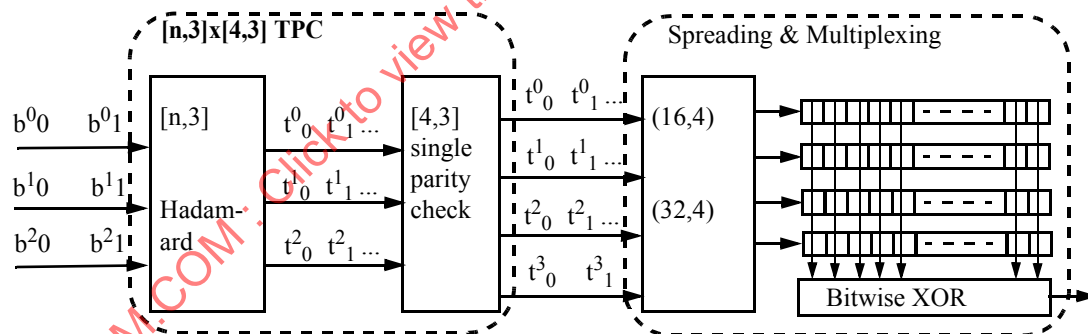
**Table 22-16—(32,4)-DSSS bit-to-chip mapping (continued)**

| Input bits (b0 b1 b2 b3) | Chip values ( $c_0$ $c_1$ ... $c_{31}$ ) |
|--------------------------|------------------------------------------|
| 0110                     | 1100 0011 0101 0010 0010 1110 1101 1001  |
| 1110                     | 1001 1100 0011 0101 0010 0010 1110 1101  |
| 0001                     | 1000 1100 1001 0110 0000 0111 0111 1011  |
| 1001                     | 1011 1000 1100 1001 0110 0000 0111 0111  |
| 0101                     | 0111 1011 1000 1100 1001 0110 0000 0111  |
| 1101                     | 0111 0111 1011 1000 1100 1001 0110 0000  |
| 0011                     | 0000 0111 0111 1011 1000 1100 1001 0110  |
| 1011                     | 0110 0000 0111 0111 1011 1000 1100 1001  |
| 0111                     | 1001 0110 0000 0111 0111 1011 1000 1100  |
| 1111                     | 1100 1001 0110 0000 0111 0111 1011 1000  |

For each codeword, ( $c_0, \dots, c_{N-1}$ ), the first component,  $c_0$ , shall be transmitted first in time, and the last component,  $c_{N-1}$ , shall be transmitted last in time.

### 22.3.10 MDSSS bit-to-chip mapping

The functional block diagram in Figure 22-13 is provided as a reference for specifying the MDSSS. Each bit in the PSDU shall be processed through the turbo product code (TPC) encoding and multiplexing module. For the horizontal code of the TPC, 3 bits are encoded into  $n$  bits with the  $[n, 3]$  Hadamard code for  $n = 4, 8, 16$ , and 32. The  $[4, 3]$  single parity check encoder is employed as the vertical code of the TPC.

**Figure 22-13—MDSSS signal flow**

Each octet of the PSDU shall be mapped into three horizontal input rows, as specified in Table 22-17. The three LSBs ( $b_0, b_1, b_2$ ) of each octet shall be mapped into the first horizontal input row ( $b^0_0, b^0_1, b^0_2$ ), and the next three bits ( $b_3, b_4, b_5$ ) of each octet shall be mapped into the second horizontal input row ( $b^1_0, b^1_1, b^1_2$ ). The last horizontal input row ( $b^2_0, b^2_1, b^2_2$ ) shall be mapped into the last two bits ( $b_6, b_7$ ) of each octet and the reference value of the octet, which is provided by the following equation:

$$p = 0$$

**Table 22-17—PSDU bit stream to horizontal code input mapping**

| Horizontal code input | $b^{00}$ | $b^{01}$ | $b^{02}$ | $b^{10}$ | $b^{11}$ | $b^{12}$ | $b^{20}$ | $b^{21}$ | $b^{22}$ |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| PSDU bit stream       | Bits:0   | 1        | 2        | 3        | 4        | 5        | 6        | 7        | p        |

For the horizontal coding of the TPC, the three parallel bit streams ( $b^{x0}$ ,  $b^{x1}$ ,  $b^{x2}$ ;  $x = 0, 1, 2$ ) are converted to the three parallel  $n$ -bit streams ( $t^x_0, t^x_1, t^x_2, \dots, t^x_{n-1}$ ) through the  $[n, 3]$  Hadamard encoder. An  $[n, 3]$  Hadamard codeword set is given by  $[h_0; h_1; h_2; h_3; h_3; h_2; h_1; h_0]$ , where  $h_i$  is the  $i$ th row of the  $n \times n$  Hadamard matrix and  $\bar{h}_i$  is the bitwise inversion of  $h_i$ .

For example, if  $n = 4$ , the  $[4, 3]$  Hadamard codeword set is obtained from the  $(4 \times 4)$  Hadamard matrix,  $\bar{H}(4)$ , given in the following equation, and the information bits to codeword mapping table is shown in Table 22-18.

$$\bar{H}(4) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

**Table 22-18—Information bits to codeword mapping for  $[4,3]$  Hadamard encoder**

| Information bits<br>( $b^x_0, b^x_1, b^x_2$ )<br>( $x = 0, 1, 2$ ) | Codeword<br>( $t^x_0, t^x_1, t^x_2, t^x_3$ )<br>( $x = 0, 1, 2$ ) |
|--------------------------------------------------------------------|-------------------------------------------------------------------|
| 0 0 0                                                              | 0 0 0 0                                                           |
| 0 0 1                                                              | 0 1 0 1                                                           |
| 0 1 0                                                              | 0 0 1 1                                                           |
| 0 1 1                                                              | 0 1 1 0                                                           |
| 1 0 0                                                              | 1 0 0 1                                                           |
| 1 0 1                                                              | 1 1 0 0                                                           |
| 1 1 0                                                              | 1 0 1 0                                                           |
| 1 1 1                                                              | 1 1 1 1                                                           |

For the vertical coding of the TPC, the single parity check encoder adds one  $n$ -bit parity stream ( $t^3_x$ ,  $x = 0, 1, \dots, n-1$ ) to the original three parallel  $n$ -bit streams ( $t^0_x, t^1_x, t^2_x$ ). For instance, if  $n$  equals four, the single parity check encoder converts the three parallel 4-bit streams to four parallel 4-bit streams as shown:

$$T_{unit} = \begin{bmatrix} t^0_0 & t^0_1 & t^0_2 & t^0_3 \\ t^1_0 & t^1_1 & t^1_2 & t^1_3 \\ t^2_0 & t^2_1 & t^2_2 & t^2_3 \\ t^3_0 & t^3_1 & t^3_2 & t^3_3 \end{bmatrix}$$

where  $T_{unit}$  is the matrix of parallel bit streams and  $t^i_j$ ,  $i = 0, 1, 2, 3$  are obtained based on the following relationships:

$$t^3_0 = \overline{t^0_0 \oplus t^1_0 \oplus t^2_0}$$

$$t^3_1 = \overline{t^0_1 \oplus t^1_1 \oplus t^2_1}$$

$$t^3_2 = \overline{t^0_2 \oplus t^1_2 \oplus t^2_2}$$

$$t^3_3 = \overline{t^0_3 \oplus t^1_3 \oplus t^2_3}$$

As a result of [4,3] horizontal and [4,3] vertical coding with a parity bit per octet, the PSDU bit stream is transformed into a  $[4, 3] \times [4, 3]$  TPC codeword matrix, forming  $(4 \times 4)$  2-dimensional data, as shown in Figure 22-14.

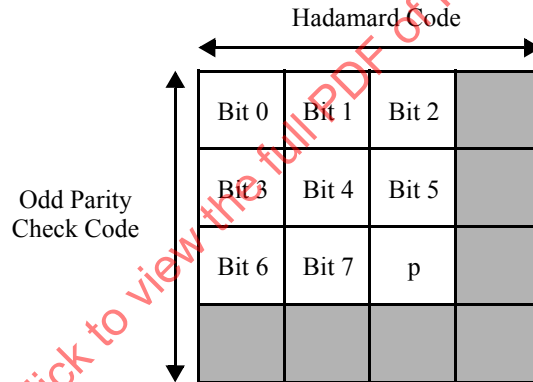


**Figure 22-14—Structure of turbo product codeword**

For  $n = 8, 16$ , and  $32$ , the TPC can be generated by the serial concatenations of  $T_{unit}$ .

$$T_{[n \times 4]} = \begin{cases} [T_{unit}] & , n = 4 \\ [T_{unit} \ T_{unit}] & , n = 8 \\ [T_{unit} \ T_{unit} \ T_{unit} \ T_{unit}] & , n = 16 \\ [T_{unit} \ T_{unit} \ T_{unit} \ T_{unit} \ T_{unit} \ T_{unit} \ T_{unit} \ T_{unit}] & , n = 32 \end{cases}$$

Each  $i$ th row of  $T_{unit}$  is spread with  $h_i$ ; i.e., the spread  $i$ th row of  $T_{unit}$  is given as follows:

$$s^i = [t^i_0 \oplus h_i \ t^i_1 \oplus h_i \ t^i_2 \oplus h_i \ t^i_3 \oplus h_i] \ , \ i = 0, 1, 2, 3$$

Then, the vertically multiplexed bit sequence  $\bar{c}_j$  of length 16 can be expressed as follows:

$$\bar{c}_j = ((s^0_j \ \& \ s^1_j)|(s^2_j \ \& \ s^3_j)) \ , (0 \leq j < 16)$$

$$\bar{c}_0 = ((t^0_0 \ \& \ t^1_0)|(t^2_0 \ \& \ t^3_0))$$

$$\bar{c}_1 = ((t^0_0 \ \& \ \overline{t^1_0})|(t^2_0 \ \& \ \overline{t^3_0}))$$

$$\bar{c}_2 = ((t^0_0 \ \& \ t^1_0)|(\overline{t^2_0} \ \& \ \overline{t^3_0}))$$

$$\bar{c}_3 = ((t^0_0 \ \& \ \overline{t^1_0})|(\overline{t^2_0} \ \& \ t^3_0))$$

$$\bar{c}_4 = ((t^0_1 \ \& \ t^1_1)|(t^2_1 \ \& \ t^3_1))$$

$$\bar{c}_5 = ((t^0_1 \ \& \ \overline{t^1_1})|(t^2_1 \ \& \ \overline{t^3_1}))$$

$$\bar{c}_6 = ((t^0_1 \ \& \ t^1_1)|(\overline{t^2_1} \ \& \ \overline{t^3_1}))$$

$$\bar{c}_7 = ((t^0_1 \ \& \ \overline{t^1_1})|(\overline{t^2_1} \ \& \ t^3_1))$$

$$\bar{c}_8 = ((t^0_2 \ \& \ t^1_2)|(t^2_2 \ \& \ t^3_2))$$

$$\bar{c}_9 = ((t^0_2 \ \& \ \overline{t^1_2})|(t^2_2 \ \& \ \overline{t^3_2}))$$

$$\bar{c}_{10} = ((t^0_2 \ \& \ t^1_2)|(\overline{t^2_2} \ \& \ \overline{t^3_2}))$$

$$\bar{c}_{11} = ((t^0_2 \ \& \ \overline{t^1_2})|(\overline{t^2_2} \ \& \ t^3_2))$$

$$\bar{c}_{12} = ((t^0_3 \ \& \ t^1_3)|(t^2_3 \ \& \ t^3_3))$$

$$\bar{c}_{13} = ((t^0_3 \ \& \ \overline{t^1_3})|(t^2_3 \ \& \ \overline{t^3_3}))$$

$$\bar{c}_{14} = ((t^0_3 \ \& \ t^1_3)|(\overline{t^2_3} \ \& \ \overline{t^3_3}))$$

$$\bar{c}_{15} = ((t^0_3 \ \& \ \overline{t^1_3})|(\overline{t^2_3} \ \& \ t^3_3))$$

If  $n$  is greater than 4, the spread and multiplexed bit sequence  $\bar{c}_j$  of  $T_{[n \times 4]}$  can also be expressed as the repeated form of the preceding equation.

Then, the final output bit stream shall be bitwise XOR-ed by covering code for the chip and symbol synchronization. For each of (16,8), (32,8), (64,8), (128,8) MDSSS, the covering code shall be bit 0 of (16,1)<sub>0</sub>-DSSS, (32,1)<sub>0</sub>-DSSS, (64,1)-DSSS, (128,1)-DSSS code, which are described in Table 22-10, Table 22-11, Table 22-12, and Table 22-13, respectively.

The final output chip sequence,  $c_0 \sim c_{4n-1}$  ( $n = 4, 8, 16, 32$ ), of  $(n,8)$  MDSSS shall be described as follows:

$$c_i = \bar{c}_{\left((i \bmod 4) + \text{floor}\left(\frac{i}{n}\right) \times 4\right)} \oplus m_i \ , (0 \leq i < 4n)$$

where  $m_i$  is the covering code.

**22.3.11 Chip whitening**

When spreading mode is set to DSSS, the PSDU chip sequence shall be whitened, depending on the frequency band and rate mode, as shown in Table 22-19. This improves spectral properties of modes with low spreading gain or insufficient spectral properties (i. e., notches) of the spreading codes. For all other modes, no chip whitening shall be applied.

**Table 22-19—Chip whitening for DSSS**

| Frequency band (MHz) | Rate mode     |
|----------------------|---------------|
| 470–510              | 1 and 2 and 3 |
| 779–787              | 2 and 3       |
| 868–870              | 1 and 2 and 3 |
| 902–928              | 2 and 3       |
| 917–923.5            | 2 and 3       |
| 920–928              | 1 and 2 and 3 |
| 2400–2483.5          | 3             |

Chip whitening is the modulo-2 addition of a chip of the PSDU at the output of the bit-to-chip mapper with the value of a cyclic  $m$ -sequence  $S_{(k \bmod (2^m - 1))}$  of length  $m = 9$ . This shall be performed by the transmitter and is described as follows:

$$c'_k = c_k \oplus S_{(k \bmod 511)}$$

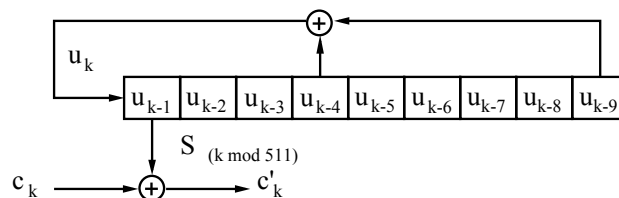
where

$c_k$  is the raw PSDU chip being whitened

$c'_k$  is the whitened chip

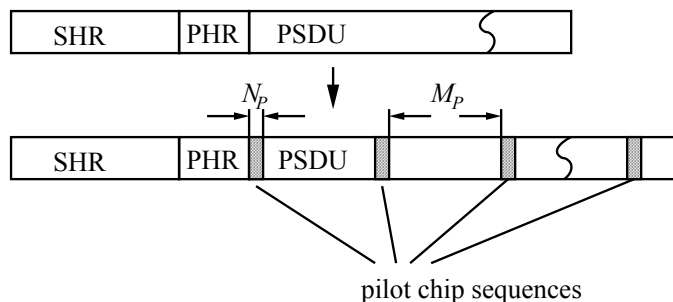
Index  $k$  starts at 0, referring to the first chip of the PSDU at the output of the bit-to-chip mapper and is increased by one at every chip interval. Figure 22-15 shows the whitening process. At  $k = 0$ , the register shall be initialized as follows:

$$(u_{k-1}, u_{k-2}, \dots, u_{k-9}) = (1, 0, 0, 0, 0, 0, 0, 0, 0)$$

**Figure 22-15—Chip whitening**

**22.3.12 Pilot insertion**

Periodic insertion of known chip sequences (pilots) into the stream of PSDU chips shall be used to simplify symbol time, channel or phase tracking during receive, taking the finite coherence time of the radio channel into account. The pilot structure of this PHY is shown in Figure 22-16.

**Figure 22-16—PSDU pilot insertion**

Since the extended PSDU chip sequence always starts with a first pilot sequence, the complexity of PHR decoding can be also reduced.

The pilot length  $N_p$  (in number of chip samples), the pilot spacing  $M_p$  (in number of chip samples), and the pilot sequence  $p = (p_0, p_1, \dots, p_{N_p-1})$  depend on the frequency band and are shown in Table 22-20.

**Table 22-20—Pilot length, spacing and chip sequences**

| Frequency band (MHz) | Length $N_p$ (# of chips) | Spacing $M_p$ (# of chips) | Chip sequence $p = (p_0, p_1, \dots, p_{N_p-1})$                                                                                                                         |
|----------------------|---------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 470–510              | 32                        | 512                        | 1101 1110 1010 0010 0111 0000 0110 0101                                                                                                                                  |
| 779–787              | 64                        | 1024                       | 1011 0010 0010 0101 1011 0001 1101 0000 1101<br>0111 0011 1101 1111 0000 0010 1010                                                                                       |
| 868–870              | 32                        | 512                        | 1101 1110 1010 0010 0111 0000 0110 0101                                                                                                                                  |
| 902–928              | 64                        | 1024                       | 1011 0010 0010 0101 1011 0001 1101 0000 1101<br>0111 0011 1101 1111 0000 0010 1010                                                                                       |
| 917–923.5            | 64                        | 1024                       | 1011 0010 0010 0101 1011 0001 1101 0000 1101<br>0111 0011 1101 1111 0000 0010 1010                                                                                       |
| 920–928              | 32                        | 512                        | 1101 1110 1010 0010 0111 0000 0110 0101                                                                                                                                  |
| 2400–2483.5          | 128                       | 2048                       | 1001 1000 1000 1011 0100 1110 0100 0010 0101<br>0010 0110 1101 1100 0111 1010 0000 1101 0100<br>0110 0111 1101 1000 0111 0101 1110 0111 1101<br>1111 1000 0000 1010 1011 |

Let

$$u = \{u_0, u_1, \dots, u_{N_{PSDU}-1}\}$$

be the sequence of PSDU chips prior to pilot insertion, assuming the first PSDU chip sample,  $u_0$ , is transmitted first in time, and the last PSDU chip sample,  $u_{N_{PSDU}-1}$ , is transmitted last in time. The number  $N_{PSDU}$  can be computed as follows:

$$N_{PSDU}(\text{LENGTH}) = \begin{cases} R_{spread} \times 2 \times N_D(\text{LENGTH}), & \text{if FEC is enabled} \\ R_{spread} \times 8 \times \text{LENGTH}, & \text{if FEC is disabled} \end{cases}$$

where

LENGTH is the number of octets in the PSDU  
 $R_{spread}$  is the spreading rate, i.e., the ratio of number of output chips  $N$  to number of input bits  $x$  for  $(N, x)$  bit-to-chip mapping, as described in Table 22-4 and Table 22-5  
 $N_D$  is the number of uncoded bits, as defined in 22.3.6

Let  $L$  as a function of the variable LENGTH be defined as follows:

$$L(\text{LENGTH}) = \text{ceiling}\left(\frac{N_{PSDU}(\text{LENGTH})}{M_p}\right)$$

Let  $u^j$  be the following subsequences:

$$u^j = \begin{cases} \{u_{jM_p}, u_{jM_p+1}, \dots, u_{(j+1)M_p-1}\} & \text{if } (j+1)M_p \leq N_{PSDU} \\ \{u_{jM_p}, u_{jM_p+1}, \dots, u_{N_{PSDU}-1}\} & \text{if } (j+1)M_p > N_{PSDU} \end{cases} \quad \text{for } j = 0, \dots, L-1$$

The pilot extended PSDU chip sequence is as follows:

$$c_{PSDU} = \{p, u^0, \dots, p, u^{L-1}\}$$

### 22.3.13 Modulation parameters for O-QPSK

A chip value shall be mapped into a binary real-valued symbol out of  $\{-1, 1\}$  as follows:

$$\zeta(c) = \begin{cases} 1, & c = 0 \\ 1, & c = 1 \end{cases}$$

In the 915 MHz and 2450 MHz bands, the half-sine pulse shape is used to represent each baseband chip and is as follows:

$$p(t) = \begin{cases} \sin\left(\frac{\pi t}{2T_c}\right), & \text{for } 0 \leq t \leq 2T_c \\ 0, & \text{otherwise} \end{cases}$$

where the chip duration  $T_c$  is the inverse of the chip rate, as described in Table 22-4 and Table 22-5.



In the 470 MHz, 868 MHz, 780 MHz, 917 MHz, and 920 MHz bands, a raised cosine pulse shape with roll-off factor of  $r = 0.8$  is used to represent each baseband symbol and is described as follows:

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

Let  $c_{PPDU} = \{c_k\}_{0}^{N_{PPDU}-1}$  be the discrete-time sequence of consecutive chip samples of the PPDU, where the first sample,  $c_0$ , is transmitted first in time and the last sample,  $c_{N_{PPDU}-1}$ , is transmitted last in time. The continuous-time, pulse-shaped complex baseband signal is as follows:

$$y(t) = \sum_{k=0}^{N_{PPDU}/2-1} \zeta(c_{2k})p(t-2kT_c) + j\zeta(c_{2k+1})p(t-(2k+1)T_c)$$

with  $j = \sqrt{-1}$ .

## 22.4 Support of legacy devices of the 780 MHz, 915 MHz, and 2450 MHz O-QPSK PHYs

When operating in the 779–787 MHz frequency band, a compliant device of the SUN O-QPSK PHY shall be able to communicate with devices of the 780 MHz band O-QPSK PHY within the specifications given in Clause 12.

When operating in the 902–928 MHz frequency band, a compliant device of the SUN O-QPSK PHY shall be able to communicate with devices of the 915 MHz band O-QPSK PHY within the specifications given in Clause 12.

When operating in the 2400–2483.5 MHz frequency band, a compliant device of the SUN O-QPSK PHY shall be able to communicate with devices of the 2450 MHz band O-QPSK PHY within the specifications given in Clause 12.

## 22.5 SUN O-QPSK PHY RF requirements

### 22.5.1 Operating frequency range

The SUN O-QPSK PHY operates in the following bands:

- 470–510 MHz
- 779–787 MHz
- 868–870 MHz
- 902–928 MHz
- 917–923.5 MHz
- 920–928 MHz
- 2400–2483.5 MHz

### 22.5.2 Transmit power spectral density (PSD) mask

The SUN O-QPSK transmit PSD mask shall conform with local regulations.

### 22.5.3 Receiver sensitivity

Under the conditions specified in 10.1.7, a compliant device shall be capable of achieving the sensitivity values given in Table 22-21 and Table 22-22 or better.

**Table 22-21—Required receiver sensitivity for spreading mode DSSS [dBm]**

| Frequency band (MHz) | Rate mode |      |      |     |
|----------------------|-----------|------|------|-----|
|                      | 0         | 1    | 2    | 3   |
| 470–510              | –110      | –105 | –100 | –95 |
| 779–787              | –105      | –100 | –95  | –90 |
| 868–870              | –110      | –105 | –100 | –95 |
| 902–928              | –105      | –100 | –95  | –90 |
| 917–923.5            | –105      | –100 | –95  | –90 |
| 920–928              | –110      | –105 | –100 | –95 |
| 2400–2483.5          | –105      | –100 | –95  | –90 |

**Table 22-22—Required receiver sensitivity for spreading mode MDSSS [dBm]**

| Frequency band (MHz) | Rate mode     |      |     |     |
|----------------------|---------------|------|-----|-----|
|                      | 0             | 1    | 2   | 3   |
| 470–510              | not supported |      |     |     |
| 779–787              | –105          | –100 | –95 | –90 |
| 868–870              | not supported |      |     |     |
| 902–928              | –105          | –100 | –95 | –90 |
| 917–923.5            | –105          | –100 | –95 | –90 |
| 920–928              | not supported |      |     |     |
| 2400–2483.5          | –105          | –100 | –95 | –90 |

### 22.5.4 Adjacent channel rejection

The interference-to-signal ratio (ISR) is the ratio of the signal power of an interferer relative to the signal power of the desired signal. The adjacent channel rejection shall be measured as follows: the desired signal shall be an SUN O-QPSK compliant signal of pseudo-random PSDU data. For a given rate mode, the desired signal is input to the receiver at a level 3 dB above the maximum allowed receiver sensitivity of Table 22-21 and Table 22-22.

The interfering signal shall be an SUN O-QPSK compliant signal with the following characteristics:

- Pseudo-random PSDU
- Spreading mode set to either DSSS or MDSSS
- The same chip rate as the desired signal
- Chip-whitening enabled

The interferer is separated in frequency by  $|\Delta f|$  from the carrier frequency of the desired channel with an ISR, as shown in Table 22-23. The test shall be performed for only one interfering signal at a time. The receiver shall meet the error rate criteria defined in 10.1.7 under these conditions.

**Table 22-23—Minimum interference-to-signal ratio (ISR) requirements depending on  $|\Delta f|$**

|                                     |                    |      |       |
|-------------------------------------|--------------------|------|-------|
| Frequency band (MHz)<br>470–510     | $ \Delta f $ (MHz) | 0.4  | 0.8   |
|                                     | ISR (dB)           | 10   | 30    |
| Frequency band (MHz)<br>779–787     | $ \Delta f $ (MHz) | 2.0  | 4.0   |
|                                     | ISR (dB)           | 10   | 30    |
| Frequency band (MHz)<br>868–870     | $ \Delta f $ (MHz) | 0.65 | 1.225 |
|                                     | ISR (dB)           | 10   | 30    |
| Frequency band (MHz)<br>902–928     | $ \Delta f $ (MHz) | 2.0  | 4.0   |
|                                     | ISR (dB)           | 10   | 30    |
| Frequency band (MHz)<br>920–928     | $ \Delta f $ (MHz) | 0.2  | 0.4   |
|                                     | ISR (dB)           | 10   | 30    |
| Frequency band (MHz)<br>2400–2483.5 | $ \Delta f $ (MHz) | 5.0  | 10.0  |
|                                     | ISR (dB)           | 10   | 30    |

#### 22.5.5 TX-to-RX turnaround time

The SUN O-QPSK PHY shall meet the requirements for TX-to-RX turnaround time as defined in 10.2.1.

#### 22.5.6 RX-to-TX turnaround time

The SUN O-QPSK PHY shall meet the requirements for RX-to-TX turnaround time as defined in 10.2.2.

#### 22.5.7 EVM definition

A transmitter shall have EVM values of less than 35% when measured for 1000 chips. The EVM measurement shall conform with 10.2.3.

#### 22.5.8 Transmit center frequency and symbol tolerance

The transmit center frequency tolerance shall be  $\pm 20 \times 10^{-6}$  maximum. When communicating with legacy devices, as described in 22.4, the receiver shall be capable of receiving signals with a center frequency offset tolerance of up to  $\pm 40 \times 10^{-6}$ .

The symbol clock frequency tolerance shall be  $\pm 20 \times 10^{-6}$  maximum. When communicating with legacy devices, as described in 22.4, the receiver shall be capable of receiving signals with a symbol clock frequency tolerance of up to  $\pm 40 \times 10^{-6}$ .

The transmit center frequency and the symbol clock frequency shall be derived from the same reference oscillator.

#### 22.5.9 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.

#### 22.5.10 Receiver maximum input level of desired signal

The SUN O-QPSK PHY shall have a receiver maximum input level greater than or equal to  $-20$  dBm using the measurement defined in 10.2.4.

#### 22.5.11 Receiver ED

The SUN O-QPSK PHY shall provide the receiver ED measurement as described in 10.2.5.

#### 22.5.12 LQI

The SUN O-QPSK PHY shall provide the LQI measurement as described in 10.2.6.

#### 22.5.13 CCA

The detection time, *aCcaTime* (as defined in 11.2), for CCA is shown in Table 22-24; see 10.2.7 for information on the 920 MHz band. The ED threshold shall correspond to a received signal power of at most  $-90$  dBm, when applying CCA Mode 1 or CCA Mode 3, as defined in 10.2.7.

**Table 22-24—CCA duration for SUN O-QPSK PHY**

| Frequency band (MHz) | <i>aCcaTime</i> (# of symbols) |
|----------------------|--------------------------------|
| 470–510              | 4                              |
| 779–787              | 8                              |
| 868–870              | 4                              |
| 902–928              | 8                              |
| 917–923.5            | 8                              |
| 2400–2483.5          | 8                              |

## 23. LECIM DSSS PHYs

### 23.1 PPDU format for DSSS

The PSDU field carries the data of the PPDU. The size of the field is set by the value of *phyLecimDsssPsdSize*. The composition of the PSDU field is affected by the optional use of tail biting, as described in 23.2.3. The PPDU shall be formatted as illustrated in Figure 23-1.

| Octets: 0/2/4 | 0/1 | 16/24/32    |
|---------------|-----|-------------|
| Preamble      | SFD | PSDU        |
| SHR           |     | PHY payload |

**Figure 23-1—Format of the LECIM DSSS PPDU**

The Preamble field length may be selected via *phyLecimDsssPreambleSize*. The SFD values and LECIM DSSS preamble values are given in Table 23-1.

**Table 23-1—LECIM DSSS preamble and SFD values**

| <i>phyLecimDsssPreambleSize</i> | Preamble field                          | SFD field (if <i>phyLecimDsssSfdPresent</i> is TRUE) |
|---------------------------------|-----------------------------------------|------------------------------------------------------|
| 0                               | Not present                             | Not present                                          |
| 16                              | 0011 1111 0101 1001                     | 0011 1000                                            |
| 32                              | 0000 1111 1101 1011 0110 0111 0010 1010 | 1000 0100                                            |

The SFD field, if present, indicates the beginning of the frame. One SFD is used when the preamble length is 2 octets, and a second SFD is used when the preamble length is 4 octets. Both SFDs are given in Table 23-1.

For the purposes of calculating the Ack frame timing required in 6.7.5, the default length of the Preamble size shall be 4 octets.

### 23.2 Modulation and spreading

In this subclause, modulation symbol refers to the output of the BPSK/O-QPSK modulator, as shown in Figure 23-2.

#### 23.2.1 Data rate

The information data rate depends on the band in use, the spreading factor, the modulation rate, and the modulation being used, and it is calculated, in kb/s, as follows:

$$\text{DataRate} = 0.5 \times \frac{(\text{phyLECIMDSSSPDDUModulationRate} \times \text{ChipPerSymbol})}{\text{phyLECIMDSSSPDUSpreadingFactor}}$$

where *ChipPerSymbol* = 1 when BPSK modulation is used and *ChipPerSymbol* = 2 when O-QPSK modulation is used. The term 0.5 represents the FEC ½ coding.

### 23.2.2 Reference modulator diagram

The functional block diagram in Figure 23-2 is provided as a reference for specifying the LECIM DSSS PHY modulation. All binary data contained in the SHR and PSDU shall be encoded using the modulation shown in Figure 23-2.

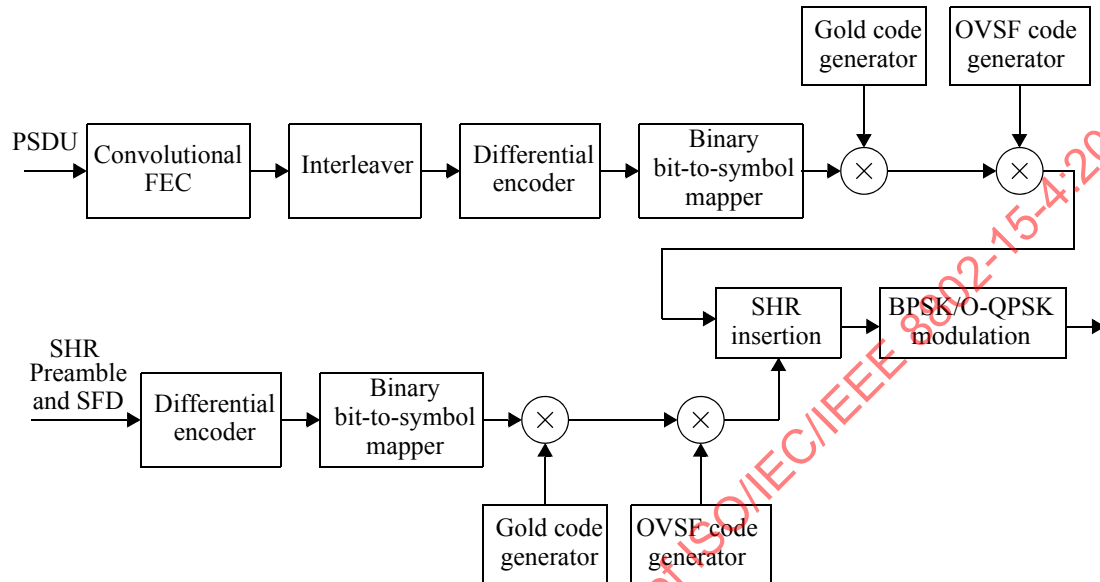


Figure 23-2—LECIM DSSS reference modulator diagram

### 23.2.3 Convolutional FEC encoding

FEC shall employ rate 1/2 convolutional coding with constraint length  $k = 7$  using the following generator polynomials:

$$G_0(x) = 1 + x^2 + x^3 + x^5 + x^6$$

$$G_1(x) = 1 + x + x^2 + x^3 + x^6$$

The encoder is shown in Figure 23-3, where  $\oplus$  denotes modulo-2 addition.

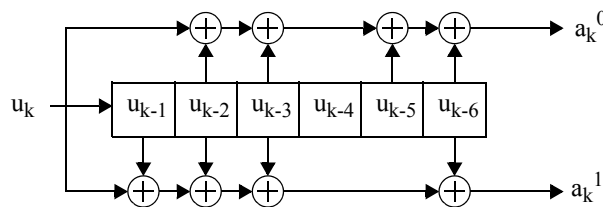


Figure 23-3—LECIM DSSS PHY convolutional encoder

Tail biting may optionally be employed. When *phyLECMFECTailBitingEnabled* is set to TRUE, the possible sizes of the PSDU are as defined in Figure 23-1 and the initial encoder state at  $k = 0$  shall be set to the last six bits of the PSDU.

When *phyLECMFECTailBitingEnabled* is set to FALSE, the initial encoder state at  $k = 0$  shall be set to  $(u_{-1}, u_{-2}, \dots, u_{-6}) = (0, 0, 0, 0, 0, 0)$ . The size of the PSDU is reduced by one octet, and the PSDU shall be extended to equal one of the possible sizes defined in Figure 23-1 by appending a termination sequence of eight bits, all zero.

#### 23.2.4 Interleaver

The output of the convolutional encoder is interleaved using a pruned bit reversal interleaving algorithm.

The text that follows contains examples of bit reversal interleavers for three PSDU sizes (256, 384, and 512 bits). PSDU sizes that are not powers of two (e.g., 384) employ pruning.

##### 23.2.4.1 256-bit fragment size

If the input sequence into the interleaver is represented by

$$[S_0 \ S_1 \dots S_{255}]$$

then the output sequence of the interleaver can be described as follows:

$$[S_0 \ S_N \dots S_{255}]$$

The value  $N$  for the  $M^{\text{th}}$  output is determined as the bit-reversal of the value  $M$ .

Representing the value  $M$  as a binary representation

$$M = [m_7 \ m_6 \dots m_0]$$

where  $m_i$  are the binary digits, then

$$N = [m_0 \ m_1 \dots m_7]$$

where  $M$  is incremented sequentially from 0 to 255.

For example, if  $M = 1 = 0000\ 0001_2$ , then  $N = 1000\ 0000_2 = 128$

##### 23.2.4.2 384-bit fragment size

If the input sequence into the interleaver is represented by

$$[S_0 \ S_1 \dots S_{383}]$$

then the output sequence of the interleaver can be described as follows:

$$[S_0 \ S_N \dots S_{383}]$$

The value  $N$  for the  $M_{th}$  output is determined as the bit-reversal of the value  $M$ .

Representing the value  $M$  as a binary representation

$$M = [m_8 m_7 \dots m_0]$$

where  $m_i$  are the binary digits, then

$$N = [m_0 m_1 \dots m_8]$$

where  $M$  is incremented sequentially from 0 to 511 and  $M'$  are the ordered set of  $M$  whose corresponding  $N$  is less than 384 (this is the pruning process).

For example:

- If  $M = 1 = 00000\ 0001_2$ , then  $N = 10000\ 0000_2 = 256$ .
- If  $M = 2 = 00000\ 0010_2$ , then  $N = 01000\ 0000_2 = 128$ .
- If  $M = 3 = 00000\ 0011_2$ , then  $N = 11000\ 0000_2 = 384$ , and since it is not less than 384, it would not be included in the ordered set  $M'$  (i.e., it is pruned from the result).
- If  $M = 4 = 00000\ 0100_2$ , then  $N = 00100\ 0000_2 = 64$ .

An example is given in “Examples of IEEE Std 802.15.4 PHY encodings” [B6].

#### 23.2.4.3 512-bit fragment size

If the input sequence into the interleaver is represented by

$$[S_0\ S_1 \dots S_{511}]$$

then the output sequence of the interleaver can be described as follows:

$$[S_0\ S_N \dots S_{511}]$$

The value  $N$  for the  $M^{\text{th}}$  output is determined as the bit-reversal of the value  $M$ .

Representing the value  $M$  as a binary representation

$$M = [m_8 m_7 \dots m_0]$$

where  $m_i$  are the binary digits, then

$$N = [m_0 m_1 \dots m_8]$$

where  $M$  is incremented sequentially from 0 to 511.

For example, if  $M = 1 = 00000\ 0001_2$ , then  $N = 10000\ 0000_2 = 256$ .

#### 23.2.5 Differential encoding

The differential encoding of the DSSS PHY is described in 13.2.3.