IEEE SA
STANDARDS ASSOCIATION

IEEE Standard for Wireless Multimedia Networks

STANDARDS

IEEE Computer Society

Developed by the
LAN/MAN Standards Committee

IEEE Std 802.15.3™-2023
(Revision of IEEE Std 802.15.3-2016)

IEEE

IEEE Std 802.15.3™-2023
(Revision of IEEE Std 802.15.3-2016)

IEEE Standard for Wireless Multimedia Networks

Developed by the

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 21 September 2023
IEEE SA Standards Board

Abstract: The protocol and compatible interconnection of data and multimedia communication equipment via 2.4 GHz, 60 GHz, and 300 GHz radio transmissions in a Wireless Personal Area Network (WPAN) using low-power and multiple modulation formats to support scalable data rates are defined in this standard. The Medium Access Control (MAC) sublayer protocol supports both isochronous and asynchronous data types.

Keywords: ad hoc network, close proximity, fast setup, fixed point-to-point, IEEE 802.15.3™, IEEE 802.15.3d™, IEEE 802.15.3e™, IEEE 802.15.3f™, mmWave, mobility, point-to-point, P2P, radio frequency, 60 GHz, submillimeter wave, terahertz, 300 GHz, wireless

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

PDF: ISBN 979-8-8557-0374-0 STD26666
Print: ISBN 979-8-8557-0375-7 STDPD26666

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA 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 makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. 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 their 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: THE NEED TO PROCURE 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 balloting 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 is 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, nor 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 the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, 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 subcommittees of the IEEE SA Board of Governors 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 evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1]

Comments on standards should be submitted using the Contact Us form.[2]

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 constitute 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.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.

[2] Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

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, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, 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; https://www.copyright.com/. 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 10 years. When a document is more than 10 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 IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.

[4] Available at: https://standards.ieee.org/standard/index.html.

Patents

IEEE standards are developed in compliance with the IEEE SA Patent Policy.[5]

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 https://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 Patents 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.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users 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.

[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

Participants

At the time this standard was completed, the IEEE 802.15 Wireless Specialty Networks Working Group had the following membership:

IEEE 802.15 Working Group

Clinton Powell, *Working Group Chair*

Philip E. Beecher, *Working Group Vice Chair*

Ann Krieger, *Working Group Vice Chair*

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

Benjamin A. Rolfe, *Working Group Treasurer*

Thomas Kürner, *Task Group 802.15.3mb Chair*

Iwao Hosako, *Task Group 802.15.3mb Vice Chair*

Monique B. Brown, Task Group *802.15.3mb Technical Editor*

Koorosh Akhavan
Carlos Aldana
Ghiath Al-kadi
Thomas Almholt
Vamsi Amalladinne
Yasuharu Amezawa
Jean-Marie Andre
Kwok Shum Au
Ankur Bansal
David Barras
Tuncer Baykas
Friedbert Berens
Ido Bettesh
Bharat Bhatia
Harry Bims
Lennert Bober
Chris Calvert
Clint Chaplin
Sangsung Choi
Junyoung Choi
Claudio da Silva
Boris Danev
Hendricus De Ruijter
Igor Dotlic
Rui Du
Ersen Ekrem
Kiyoshi Fukui
Masatoshi Fukunaga
Ming Gan
Chittabrata Ghosh
Tim Godfrey
Robert Golshan
Kristian Granhaug
Bernhard Grosswindhager
Raphael Guimond
Jianlin Guo
Ziyang Guo
Taeyoung Ha
Hiroshi Harada
Jerome Henry
Marco Hernandez
Christopher Hett
Masayuki Hirata
Jay Holcomb
Chunyu Hu
Mengshi Hu
Tetsushi Ikegami
Yeong Min Jang
Jeng-Shiann Jiang
Jinjing Jiang
Seong-Soon Joo
Volker Jungnickel
Juha Juntunen
Aniruddh Kabbinale
Akifumi Kasamatsu
Yoshio Kashiwagi
Rani Keren
Stuart J. Kerry
Minsoo Kim
Patrick Kinney
Shoichi Kitazawa
Tero Kivinen
Takumi Kobayashi
Ryuji Kohno
Jonghoe Koo
Wolfgang Kuechler
Takashi Kuramochi
Mingyu Lee
Stefan Lemsitzer
Frank Leong
Huan-Bang Li
Sang-Kyu Lim
Wei Lin
Yong Liu
Chenchen Liu
Peng Liu
Xiliang Luo
Yongsen Ma
Vinayagam Mariappan
Stephen McCann
Apurva Mody
Carl Murray
Frederic Nabki
Dries Neirynck
Jaroslaw Niewczas
Paul Nikolich
Hyunseob Oh
Pooria Pakrooh
Clark Palmer
Glenn Parsons
Riku Pirhonen
Bin Qian
Mohammad Rahmani
Enrico-Henrik Rantala
Oded Redlich
Joerg Robert
Hiroki Saito
Ruben E. Salazar Cardozo
Stephan Sand
Shigenobu Sasaki
Eren Sasoglu
Kamran Sayrafian
Jonathan Segev
Norihiko Sekine
Daoud Serang
Kunal Shah
Stephen Shellhammer
Shimi Shilo
Youngwan So
Gary Stuebing
Don Sturek
Li Sun
Takafumi Suzuki
Kiyoshi Tada
Bin Tian
Lochan Verma
Billy Verso
Yi Wang
Dag Wisland
Kuan Wu
Tetsushi Yamamoto
Shang-Te Yang
Xun Yang
Zhenzhen Ye
Su Khiong Yong
Kangjin Yoon
Larry Zakaib
Sven Zeisberg
Mingda Zhou
Qiyue Zou

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

Duschia Bodet
Christoph Herold
Josep Jornet
Akifumi Kasamatsu
Keitarou Kondou
Fumihiro Nishiyama
Hiroyo Ogawa
Vitaly Petrov
Norihiko Sekine

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

Saif Aldeen Saad Obayes AL-Kadhim
Amelia Andersdotter
Boon Chong Ang
Oscar Au
Philip E. Beecher
Steven Bezner
Harry Bims
Andreas Bluschke
Lennert Bober
Vern Brethour
Monique B. Brown
William Byrd
John Calvin
Paul Cardinal
Pin Chang
Suresh Channarasappa
Chan Chen
Zhiman Chen
C. Chou
Razvan Ciocan
Todor Cooklev
Avraham Freedman
Matthias Fritsche
James P. K. Gilb
David Goodall
Marco Hernandez
David Hess
Werner Hoelzl
Iwao Hosako
Doug Houseman
Raj Jain
Pranav Jha
Ming Jiang
Josep Miquel Jornet
Volker Jungnickel
Chintaka Kandapahala
Ruslan Karmanov
Piotr Karocki
Stuart J. Kerry
Nam Kim
Yongbum Kim
Shoichi Kitazawa
Keitarou Kondou
Vinod Kristem
Thomas Kürner
Hyeong Ho Lee
Joseph Levy
Jialing Li
Greg Luri
Jon Martens
Ronald Murias
Rajesh Murthy
N. Kishor Narang
Mfanasibili Nkonyane
Satoshi Oyama
Bansi Patel
Arumugam Paventhan
Howard Penrose
Vitaly Petrov
Clinton Powell
Venkatesha Prasad
R. K. Rannow
James Reilly
Maximilian Riegel
Benjamin A. Rolfe
Ruben E. Salazar Cardozo
John Sargent
Naotaka Sato
Cole Scott
Daniel Seidel
Diana Shi
Dorothy Stanley
Walter Struppler
Mark Sturza
Bo Sun
James Van De Ligt
Dmitri Varsanofiev
John Vergis
Peter Vouras
Alexandru Vulpe
Lisa Ward
Stephen Webb
Matthias Wendt
Scott Willy
Andreas Wolf
Yu Yuan
Qiyue Zou

When the IEEE SA Standards Board approved this standard on 21 September 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Doug Edwards
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
Joseph S. Levy
Howard Li
Gui Lin
Johnny Daozhuang Lin
Xiaohui Liu
Kevin W. Lu
Daleep C. Mohla
Andrew Myles
Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.15.3-2023, IEEE Standard for Wireless Multimedia Networks.

IEEE Std 802.15.3 was originally developed to provide superior quality of service (QoS) over relatively medium range wireless links.

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

IEEE Std 802.15.3-2003 defined an ad-hoc MAC that enabled simple and fast network formation, excellent QoS, strong security with 128-bit AES encryption in CCM mode, and methods to coexist with other wireless networks in the band. The PHY operated in the license exempt 2.4 GHz band with data rates of 11 Mb/s to 55 Mb/s at distances of greater than 70 m.

Interest in working on an amendment to fix and enhance IEEE Std 802.15.3 began at the September 2003 meeting in Singapore. A study group was formed at the November 2003 meeting in Albuquerque, which completed a project authorization request that was approved in March 2004. Work progressed quickly, and the first task group letter ballot was successfully completed in March 2005. After one recirculation, the draft began sponsor ballot in August 2005. The ballot was successful, and after a recirculation ballot to validate some minor changes, IEEE Std 802.15.3b was approved by the IEEE Standards Board on 6 December 2005, just over two years after the study group started.

IEEE Std 802.15.3b™-2005 fixed mistakes and added enhancements that improve the efficiency of the base standard. Some of the key changes/additions in this amendment are as follows:

— An improved definition of the medium access control (MAC) layer management entity (MLME) service access point (SAP).
— A new acknowledgment (ACK) policy, implied-ACK, that allows polling and a more efficient use of channel time.
— A method for relinquishing time in a channel time allocation (CTA) to allow another device (DEV) time to transmit data.
— The ability to assign device identifiers (DEVIDs) to group addresses to allow multicast connections.
— Faster recovery of network operations when the piconet coordinator (PNC) abruptly disconnects with the conditional handover and the next PNC processes.
— Multiple contention periods during a superframe.

In the July 2003 meeting in San Francisco, an interest group was formed to consider a millimeter wave (mmWave PHY) for the existing standard. A study group was formally created in the March 2004 IEEE 802 plenary meeting in Orlando and developed a project authorization request that was approved in March 2005. The first meeting as a task group was in May 2005 in Cairns, Australia, and the group worked steadily developing channel models and evaluation documents. The PHY modes were selected in November 2007 at the Atlanta meeting and draft progressed rapidly, entering Working Group letter ballot in June 2008. After three Working Group recirculation ballots, sponsor ballot started in March 2009. A total of three sponsor recirculation ballots were held, leading to approval of IEEE Std 802.15.3c-2009 by the IEEE SA Standards Board on 11 September 2009.

Some of the key features of IEEE Std 802.15.3c™-2009 and additions are as follows:

— Operation in the 60 GHz band.
— New data rates, with the highest greater than 5 Gb/s.
— Beam forming negotiation for the transmitter to increase the communication range.
— MAC packet aggregation and the acknowledgment of individual subpackets to reduce retransmission overhead.

The PHY specifies three modes and one common mode. The three PHY modes are as follows:

— Single carrier (SC) mode optimized for low power and low complexity.
— High-speed interface (HSI) mode optimized for low-latency bidirectional data transfer.
— Audio/video (AV) mode optimized for the delivery of uncompressed, high-definition video and audio.

Interest in developing a wireless communication system at THz frequencies started in 2008 with the establishment of the THz Interest Group. In May 2014, Task Group 3d was formed, covering switched point-to-point connections operating in frequencies from 60 GHz up to the lower THz bands. The Application Requirements Document (ARD), Technical Requirements Documents (TRD), Channel modeling document, and Evaluation Criteria Document for close proximity scenarios were reviewed in July 2014. Discussions began to separate the close proximity efforts at 60 GHz from the other activities at the lower THz band in September 2014; a decision was made in November 2014 to split Task Group 3d into two, one optimized for ranges under 10 centimeters (Task Group 3e) and another covering several meters or more (Task Group 3d).

To support the development of these new amendments, the base standard needed to be revised to create a single document that incorporated the prior two amendments. A PAR was generated in July 2015 at the IEEE Plenary meeting in Waikoloa Village, HI. The PAR was approved by the IEEE SASB in September 2015 and the initial draft was completed soon thereafter (by the same Technical Editor who worked on the previous three documents). After two Working Group ballots, the sponsor ballot began in December 2015 and completed in February 2016 after two Sponsor ballot recirculations. IEEE Std 802.15.3-2016 was approved by the IEEE SA Standards Board in May 2016.

The first meeting of Task Group 3e was held in March 2015 in Berlin, Germany, where the Project Authorization Request (PAR) and Criteria for Standards Development (CSD) were approved. At the May 2015 meeting, the ARD and TRD were combined as a single Technical Guidance Document (TGD) and proposals were reviewed in the July and September 2015 sessions, including the selection of two PHY modes. The group entered Working Group letter ballot in January 2016. After two Working Group recirculation ballots, sponsor ballot started in July 2016. A total of three sponsor recirculation ballots were held, leading to approval of IEEE Std 802.15.3e-2017 by the IEEE SA Standards Board on 14 February 2017.

In November 2015, the required PAR change for Task Group 3d to limit the activities to the lower THz frequency ranges was approved. During the March 2016 meeting, the supporting documents were approved and the call for proposals was issued. Proposals were reviewed during the July and September 2016 meetings. The group entered Working Group letter ballot in January 2017. After one Working Group recirculation ballot, the sponsor ballot started in March 2017. A total of two sponsor ballot recirculation ballots were held, leading to approval of IEEE Std 802.15.3d-2017 by the IEEE SA Standards Board on 28 September 2017.

IEEE Std 802.15.3e-2017 is an amendment to IEEE Std 802.15.3-2016 that defines an alternative physical layer operating in the millimeter wave band along with the necessary MAC changes to support this PHY. Some of the key features and additions are as follows:

- Operation in the 60 GHz band.
- New data rates, with the highest reaching 100 Gb/s.
- Limiting communication range to 10 centimeters or less.
- Use of a pairnet structure and Stack ACK mechanism to simplify and optimize the MAC.
- Selectable PHY modes (single carrier and OOK) to achieve either high-speed operation or system simplicity.

IEEE Std 802.15.3d-2017 is an amendment to IEEE Std 802.15.3-2016 that defines an alternative physical layer (PHY) at the lower THz frequency range, between 252 GHz and 325 GHz, for switched point-to-point links, along with the necessary MAC changes to support this PHY. The amendment builds on the concept of pairnet, introduced in IEEE Std 802.15.3e-2017, and inherits the corresponding MAC changes defined there. Some of the key features and additions are as follows:

- Operation in the THz frequency band.
- Usage of eight different bandwidths between 2.16 GHz and 69.12 GHz.
- Designed for data rates of up to 100 Gb/s.
- Use of a pairnet structure supporting wireless links for intra-device communication (e.g., board-to-board communication), close proximity communication, wireless data centers, and backhaul/fronthaul links.
- Selectable PHY modes (single carrier and on-off keying) to achieve either ultra high-speed operation or system simplicity.

IEEE Std 802.15.3f-2017 is an amendment to IEEE Std 802.15.3-2016 that updates the channelization of the millimeter wave PHY to use the expanded unlicensed millimeter wave spectrum up to 71 GHz.

The PAR for a revision project was generated in November 2021 at the IEEE Plenary virtual meeting. The PAR was approved by the IEEE SASB in December 2021 and the initial draft was completed in September 2022. Apart from several small fixes, the new draft also includes several new features for the THz PHY, such as channel plans up to 450 GHz and two new modulation schemes. After three Working Group ballots, the sponsor ballot began in March 2023 and was completed in July 2023 following two Sponsor ballot recirculations. IEEE Std 802.15.3-2023 was approved by the IEEE-SA Standards Board in September 2023.

Contents

IEEE Standard for Wireless Multimedia Networks

1. Overview

1.1 Scope

This standard defines physical layer (PHY) and medium access control (MAC) sublayer specifications for high data rate wireless connectivity (typically over 200 Mb/s) with fixed, portable, and moving devices. Data rates are high enough to satisfy a set of consumer multimedia industry needs, as well as to support emerging wireless switched point-to-point and high-rate close proximity point-to-point applications.

1.2 Purpose

The purpose of this standard is to provide for low complexity, low cost, low power consumption, and high data rate wireless connectivity among devices that support a variety of applications such as a set of consumer multimedia industry needs, wireless switched point-to-point applications in data centers, wireless backhaul/fronthaul intra-device communications, and a wide variety of additional use cases such as rapid large multimedia data downloads and file exchanges between two devices in close proximity, including between mobile devices and stationary devices (kiosks, ticket gates, etc.), and/or wireless data storage devices.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).[6, 7]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals is *able to*).

[6] The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.
[7] The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

2. Normative references

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

ANSI X3.66-1979: American National Standard for Advanced Data Communication Control Procedures (ADCCP).[8]

FIPS 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/NIST, May 9, 2023.[9]

IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture.[10, 11]

ISO/IEC 646:1991, Information technology—ISO 7-bit coded character set for information interchange.[12]

NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, Morris Dworkin, November 2007.[13]

[8] This document is available at https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub71-1980.pdf

[9] NIST publications are available from the National Institute of Standards and Technology (https://www.nist.gov/).

[10] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).

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

[12] ISO/IEC/IEEE publications are available from the International Organization for Standardization (https://www.iso.org/) and The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).

[13] This document available at https://csrc.nist.gov/pubs/sp/800/38/d/final.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause. [14]

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

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

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

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

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

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

dBm: Decibels relative to a milliwatt.

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

device address: The EUI-48 associated with device in an IEEE 802.15.3 piconet.

disassociation: The service that removes an existing association.

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

guard interval: A time period placed at the front of a symbol for inter-symbol interference avoidance.

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

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

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

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

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

[14] *IEEE Standards Dictionary Online* is available at: http://dictionary.ieee.org. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

pairnet: A network of at most two associated devices, one of which is capable of providing coordination.

pairnet associated period (PAP): Phase during which the pairnet devices are associated.

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

pairnet device: An entity that may only operate in a pairnet and that implements an IEEE 802.15.3 conformant media access control and physical layer interface to the wireless medium.

pairnet setup period: Phase during which the pairnet beacon is active but pairnet devices are not yet associated.

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

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

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

quality of service (QoS): A collective measure of the level of service (e.g., availability, latency, throughput, error rate) delivered between devices.

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

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

stack acknowledgment (Stk-ACK): An acknowledgment method where the device receiving data returns the frame number to jump back to and starts retransmission when the received frame(s) are corrupted.

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

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

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

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

3.2 Acronyms and abbreviations

AAS	asymmetric antenna system
ACK	acknowledgment
AES	advanced encryption standard

AS	application specific
ATP	association timeout period
AV	audio/visual
AWGN	additive white Gaussian noise
BBIFS	beam-forming beam-level interframe spacing
BCRD	bidirectional channel time allocation relinquish duration
BcstID	broadcast identifier
BER	bit error rate
BIFS	backoff interframe space
Blk-ACK	block acknowledgment
BPSK	binary phase-shift keying
BSID	beacon source identifier
BSIFS	beam-forming sector-level interframe spacing
BST	beam switching/steering and tracking
CAP	contention access period
CBC	cipher block chaining
CBC-MAC	cipher block chaining-message authentication code
CCA	clear channel assessment
CCM	counter mode encryption and cipher block chaining message authentication code
CES	channel estimation sequence
CID	Company ID
CMS	common mode signaling
CP	contention period
CRC	cyclic redundancy check
CSMA/CA	carrier sense multiple access with collision avoidance
CTA	channel time allocation
CTAP	channel time allocation period
CWB	continued wake beacon
DAMI	dual alternate mark inversion
DestID	destination identifier
DEV	device

DEVID	device identifier
Dly-ACK	delayed acknowledgment
DME	device management entity
DQPSK	differential quadrature phase-shift keying
DSPS	device synchronized power save
EEP	equal error protection
EIRP	equivalent isotropic radiated power
EPD	EtherType protocol discrimination
EVM	error vector magnitude
FCS	frame check sequence
FCSL	frame convergence sublayer
FEC	forward error correction
FER	frame error rate
GCM	Galois/counter mode
HCS	header check sequence
HR	high rate
HRCP	high-rate close proximity
HRP	high-rate physical layer
HRPDU	high-rate protocol data unit
HRS	high resolution
HSI	high-speed interface
ID	identifier
IE	information element
IFS	interframe space
Imm-ACK	immediate acknowledgment
Imp-ACK	implied acknowledgment
KO	key originator
LAN	local area network
LDPC	low-density parity check
LFSR	linear feedback shift register
LLC	logical link control

LLPS	low-latency power save
LQI	link quality indication
LRP	low-rate physical layer
LRPDU	low-rate protocol data unit
LSB	least significant bit
MAC	medium access control
MAN	metropolitan area network
MCDU	MAC command data unit
MCS	modulation and coding scheme
McstID	multicast identifier
MCTA	management channel time allocation
MIFS	minimum interframe space
MIMO	multiple-input, multiple-output
MLME	MAC layer management entity
mmWave	millimeter wave
MPDU	MAC protocol data unit
MPR	mandatory PHY rate
MR	medium rate
MSB	most significant bit
MSC	message sequence chart
MSDU	MAC service data unit
NAK	negative acknowledgment
NbrID	neighbor identifier
NLOS	non-line-of-sight
OFDM	orthogonal frequency-division multiplexing
OOK	on-off keying
OUI	organizationally unique identifier
PAL	protocol adaptation layer
PAN	personal area network
PAP	pairnet associated period
PCES	pilot channel estimation sequence

PCTM	pending channel time map
PDU	protocol data unit
PET	pattern estimation and tracking
PHY	physical layer
PIB	PAN information base
PLME	PHY layer management entity
PN	pseudo noise
PNC	piconet coordinator
PNCID	piconet coordinator identifier
PNID	piconet or pairnet identifier
PPRE	pilot preamble
PRBS	pseudo-random binary sequence
PRC	pairnet coordinator
PRCID	pairnet coordinator identifier
PRDEV	pairnet DEV
PRNG	pseudo-random number generator
PS	power save
PSD	power spectral density
PSP	pairnet setup period
PSPS	piconet synchronized power save
PSRC	power source
PW	pilot word
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quadrature phase-shift keying
QT	quasi-omni training
RF	radio frequency
RIFS	retransmission interframe space
RS	Reed Solomon
RSSI	received signal strength indication
RSSIR	received signal strength indication relative to sensitivity

RX	receive or receiver
SAP	service access point
SAS	symmetric antenna system
SC	single carrier
S-CAP	sub-contention access period
SDU	service data unit
SEC	security
SECID	security identifier
SFC	secure frame counter
SFD	start-of-frame delimiter
SIFS	short interframe space
SINR	signal-to-interference-plus-noise ratio
SISO	single-input, single-output
SNR	signal-to-noise ratio
SPS	synchronous power save
SrcID	source identifier
SSB	single side band
ST	sector training
STP	stream timeout period
Stk-ACK	stack acknowledgment
TCM	trellis coded modulation
TDMA	time division multiple access
TPC	transmit power control
TSD	transmit switched diversity
TU	time unit
TX	transmit or transmitter
UEP	unequal error protection
UnassocID	unassociated identifier
WAN	wide area network
WLAN	wireless local area network

4. General description

4.1 Format conventions

4.1.1 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 are either signed or unsigned integers unless explicitly defined otherwise. 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 least significant bit (LSB) leftmost and most significant bit (MSB) rightmost as shown in Figure 4-1 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 4-1—General number field format

Numbers of size greater than 1 octet occur with the octet containing the LSBs 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.

4.1.2 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 4-2.

Bits:0	**…**	**7**	**Bits 8–(k – 1)**
String bit 0	…	String bit 7	String bits 8 to k – 1

Figure 4-2—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.

4.1.3 MAC frame format

The MAC frames in the MAC sublayer are described as a sequence of fields in a specific order. Each figure in Clause 6 depicts the fields as they appear in the MAC frame and in the order in which they are transmitted in the wireless medium, from left to right where the leftmost bit is transmitted first in time.

In the figures, all bits within fields are numbered from 0 (left) to k (right) where the length of the field is k+1 bits. The octet boundaries within a field are obtained by taking the bit-numbers of the field modulo 8. Octets within numeric fields that are longer than a single octet are depicted in increasing order of significance, from lowest numbered bit on the left to the highest numbered bit on the right. The octets in fields longer than a single octet are sent to the PHY in order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits. For any text fields, the first character is in the first octet of the field with other characters following sequentially. An example of the bit and octet ordering is illustrated in Figure 4-3.

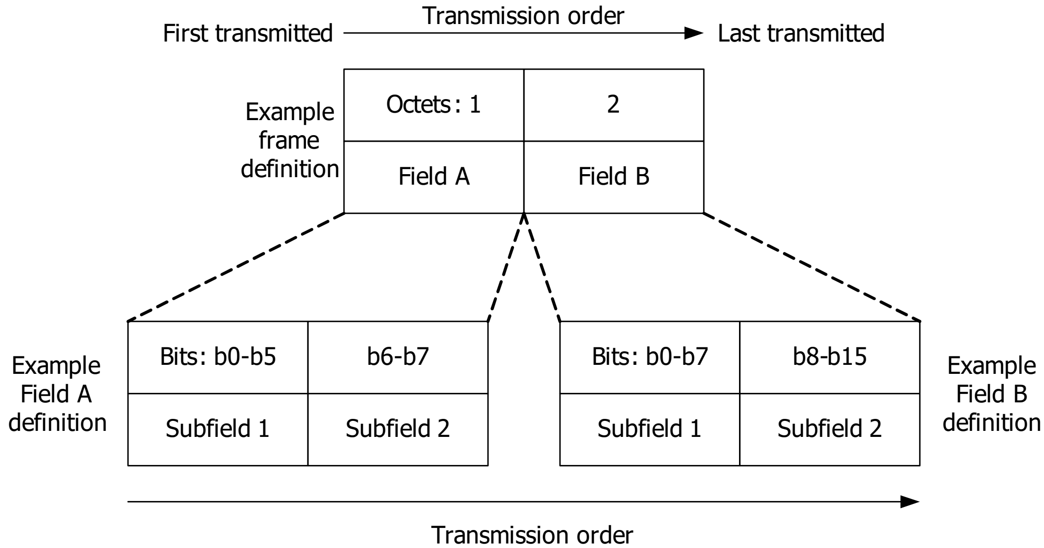

Figure 4-3—Example of bit and octet ordering

The payload in the data frame is sent with the lowest numbered octet first, LSB first, over the air, as illustrated in Figure 4-4. The frame check sequence (FCS), described in 6.2 is an exception to this convention and is transmitted with the MSB first. In the case of the header check sequence (HCS), the PHY determines the bit order in which the HCS is sent. Both the FCS and HCS are described in 6.2.

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

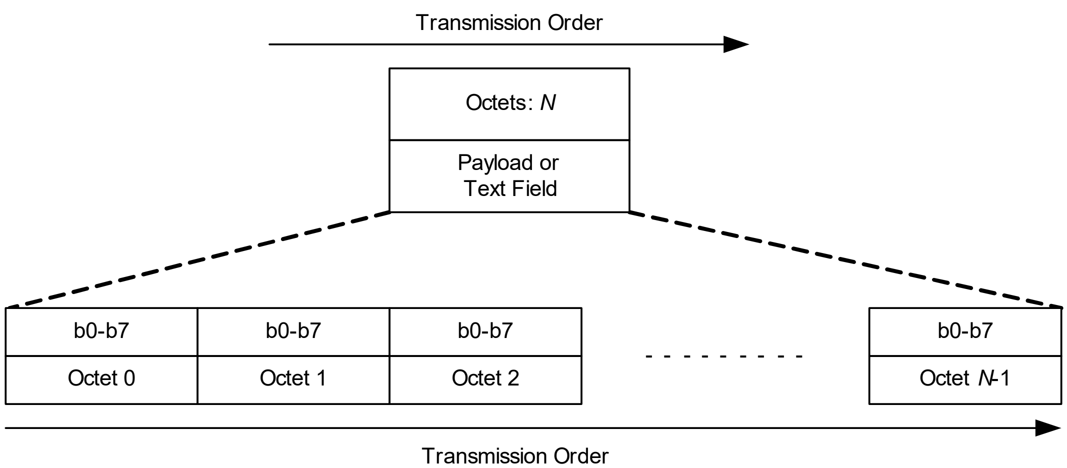

Figure 4-4—Example of payload transmission order

4.1.4 Reserved fields and values

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

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

4.2 Network types

Devices can either be operated in a piconet or in a pairnet, as defined in 4.3 and 4.5, respectively.

4.3 What is a piconet?

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

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

The piconet is not used for either the high-rate close proximity (HRCP) PHY or the terahertz (THz) PHY.

4.4 Components of an IEEE 802.15.3 piconet

An IEEE 802.15.3 piconet consists of several components, as shown in Figure 4-5. The basic component is the DEV. One DEV is required to assume the role of the piconet coordinator (PNC) of the piconet. The PNC provides the basic timing for the piconet with the beacon. Additionally, the PNC manages the quality of service (QoS) requirements, power save modes, and access control to the piconet.

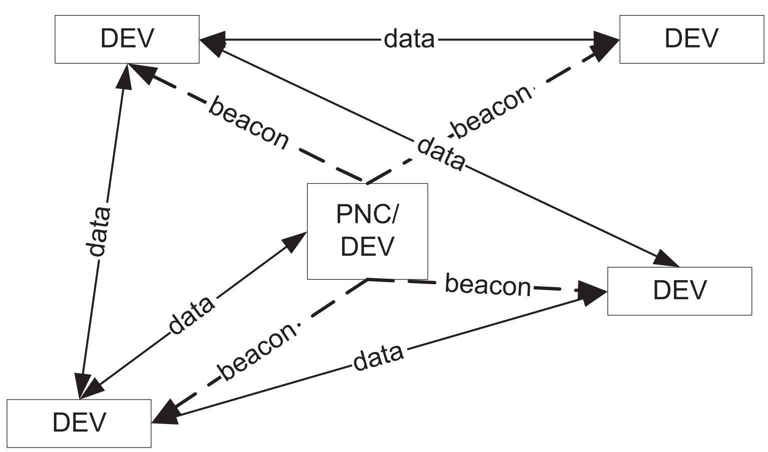

Figure 4-5—IEEE 802.15.3 piconet elements

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

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

4.5 What is a pairnet?

A pairnet consists of at most two DEVs, as shown in Figure 4-6. Typical communication distance is 10 cm or less for a pairnet DEV (PRDEV) with a HRCP PHY. For a PRDEV with a THz PHY, typical communication distance is from a few centimeters up to several hundred meters. A PRDEV always connects as a pairnet.

The pairnet is used for both the HRCP PHY and the THz PHY.

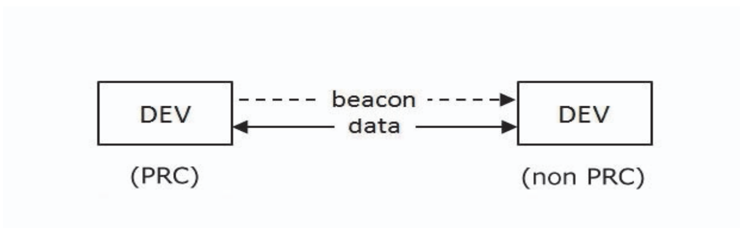

Figure 4-6—IEEE 802.15.3 pairnet elements

4.6 Components of an IEEE 802.15.3 pairnet

An IEEE 802.15.3 pairnet consists of at most two DEVs as components. These two components are called the pairnet coordinator (PRC) and the pairnet device (PRDEV). A Beacon frame is transmitted from the PRC to allow a PRDEV to connect. Once a pairnet connection is established, the Beacon frame transmissions are turned off.

4.7 MAC functionality

4.7.1 Overview

IEEE 803.15.3 MAC is designed to support the following goals:

- Fast connection time
- Ad hoc networks
- Data transport with QoS
- Security
- Dynamic membership
- Efficient data transfer

4.7.2 Coordination

4.7.2.1 General

A piconet is formed when an IEEE 802.15.3 DEV that is capable of acting as the PNC begins transmitting beacons. Thus, even if there are no associated DEVs, the PNC sending the beacon is considered to be a piconet.

PRDEVs are not capable of participating in a piconet. PRDEVs are used in a pairnet.

4.7.2.2 Starting a piconet

To start a piconet, a DEV that is capable of acting as the PNC scans the available channels to find one that is not being used, as described in 7.2.2. If it finds one that is clear, it starts the piconet by simply sending the beacon after making sure that the channel has remained empty for a specified period of time, as described in 7.2.3. If no channels are available, the DEV has the option of attempting to start a dependent piconet, as described in 7.2.8 and 7.2.9 and summarized in 4.7.2.5 and 4.7.2.6. While the process of starting a piconet does not ensure that the "most capable" PNC is initially selected based on the criteria in 7.2.4, the association and handover process does allow the "most capable" DEV to eventually become the PNC of the piconet.

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

4.7.2.3 Starting a pairnet

A DEV that is capable of acting as the PRC starts the pairnet by initializing the Sequence Number field and Last Received Sequence Number field and then sending the Beacon frame in the default channel. The default channel is defined in 13.1.5. The PRC need not scan the channels beforehand for availability.

For the THz PHY, the default channel is defined in 14.1.3.

4.7.2.4 Handing over control of the piconet

When a DEV associates with an existing piconet, as described in 7.4.2 and 4.7.6, the PNC checks the capabilities of the new DEV to see if it is more capable to be the PNC of the piconet based on the criteria defined in 7.2.4. If the new DEV is more capable and the current security policies allow it, then the PNC has the option of handing over control of the piconet to the DEV that has just joined. This handover process, as described in 7.2.4, maintains all existing time allocations so that there is no interruption in the delivery of data in the piconet. If the PNC is shutting down or wants to leave the piconet, it also uses the handover process to give control to another DEV in the piconet. The handover process also supports the handing over of a dependent PNC, as described in 7.2.7.

4.7.2.5 Creating a child piconet

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

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

The child PNC is a member of the parent piconet and thus is able to exchange data with any DEV in the parent piconet. The child PNC is also a member of the child piconet and thus is able to exchange data with any DEV in the child piconet. Only DEVs in a child piconet that are also members of the parent piconet are able to communicate directly with DEVs in the parent piconet who are not also members of that child piconet. The use of the child piconet is described in 7.2.8.

4.7.2.6 Creating a neighbor piconet

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

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

The neighbor PNC is not a member of the parent piconet and thus does not exchange information with any DEV in the parent piconet. The neighbor piconet mechanism is available to other wireless DEVs as a means of sharing the frequency spectrum. Any entity capable of initiating (i.e., requesting status as) an IEEE 802.15.3 neighbor piconet would also be capable of using this as a coexistence method. The use of the neighbor piconet is described in 7.2.9.

4.7.3 Ending a piconet

If the PNC is going to stop operation and there are no other PNC-capable DEVs in the piconet, the PNC places the PNC Shutdown information element (IE), as described in 6.4.6, into the beacon, as described in 7.2.10.2, to notify the members of the piconet.

In the case that the PNC abruptly leaves the piconet without handing over control to another PNC-capable DEV in the piconet, the piconet stops operation. After the association timeout period (ATP) expires, a PNC-capable DEV from the old piconet will be able to start a new piconet using the normal process, as described in 7.2.3.

In the case of dependent piconets, the parent PNC is able to end the dependent piconet via the Disassociation Request command, described in 6.5.2.4, for neighbor piconets, or by using the stream termination procedure, described in 7.7.2.4, for child piconets, as described in 7.2.10.3.

4.7.4 Ending a piconet with a dependent piconet involved

If the parent piconet ends operation, the parent PNC will indicate in the PNC Shutdown IE, as described in 6.4.6, a dependent piconet that will be able to continue to operate. All other dependent piconets will cease operation when the parent piconet ends operation. The dependent PNC that was selected to remain will remove the Parent Piconet IE, as described in 6.4.4, from its Beacon frame, signifying that it is no longer a dependent piconet. In the case where the parent piconet was temporarily disrupted, the parent PNC is able to attempt to join the dependent piconet and potentially receive a transfer of control via PNC handover.

A child piconet ends its piconet with the shutdown procedure and then uses the Channel Time Request command, as described in 6.5.7.2, to terminate the stream and release the resources in the parent piconet. When the child PNC shuts down its piconet, it is not required to leave the parent piconet.

The neighbor piconet uses the Disassociation Request command, as described in 6.5.2.4, to end its relationship with the parent PNC.

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

4.7.5 Ending a pairnet

If the PRC or PRDEV determines that the connected peer is gone, the pairnet is terminated. The PRC or PRDEV terminates a pairnet by sending a Disassociation Request command. The PRC can then restart sending Beacon frames in order to prepare for creating a new pairnet, as described in 8.2.7.

4.7.6 Association and disassociation in a piconet

In order to participate in a piconet, a DEV needs to join the piconet using the association process, as described in 7.4.2. Associating with the piconet provides the DEV with an identifier that is unique within that piconet, the device identifier (DEVID), as described in 6.2.5. The DEVID is used instead of the DEV address to save overhead in the system. The association process optionally provides information about the services available in the piconet as well as the services provided by the DEV, as described in 7.4.3. The association process also provides the PNC with the capabilities of the new DEV to enable the PNC to decide if it wants to hand over control of the piconet to the new DEV, as described in 7.2.4 and 4.7.2.4.

When a new DEV joins the piconet, the PNC broadcasts the information about all of the DEVs in the piconet, as described in 7.4.4, and places information in the beacon about the new DEV. This allows other DEVs in the piconet to become aware of the new DEV as well as giving information to the new DEV about the members of the piconet.

For a pairnet, a DEV connects to a PRC by sending an Association Request Command during one of the available Access Slots after a Beacon frame is received as specified in 7.5.1. The peer-to-peer data transfer begins after completion of the association process.

When a DEV wants to leave the piconet or if the PNC wants to remove a DEV from the piconet, the disassociation process, as described in 7.4.5, is used. The DEVID of the disassociated DEV is no longer valid, until reissued by the PNC. However, the PNC is not allowed to reissue the DEVID until a waiting period has expired, as described in 7.4.2.

4.7.7 Association and disassociation in a pairnet

Specific procedures for the association and disassociation of pairnet exist. The association procedure is described in 7.5.1, and the disassociation is described in 7.5.2.

4.7.8 Security overview

Security for the piconet or pairnet is defined by one of the following two modes, as described in 8.2:

a) Mode 0—Open: Security membership is not required and payload protection (either data integrity or data encryption) is not used by the MAC. The PNC is allowed to use a list of DEV addresses to admit or deny entry to the piconet. A PRC is allowed to use a list of DEV addresses to admit or deny entry to the pairnet.

b) Mode 1—Secure membership and payload protection: DEVs establish secure membership with the PNC in a piconet, or with the PRC in a pairnet, before they have access to the piconet's or the pairnet's resources. Data sent in Mode 1 is allowed to use payload protection (data integrity or data encryption with data integrity). Data integrity is required for most of the commands that are sent when in Mode 1.

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

The payload protection protocol, as described in 9.2.2 for piconets and 10.2.2 for pairnets, uses a symmetric key that is generated by the key originator and is securely distributed to DEVs that have established secure membership or a secure relationship with the key originator, as described in 8.4.3. For piconets, the symmetric key encryption algorithm used is the advanced encryption standard (AES) 128 in counter mode encryption and cipher block chaining-message authentication code (CCM). For pairnets, Galois/counter mode (GCM) of the AES-128 is used.

4.7.9 IEEE 802.15.3 superframe

4.7.9.1 IEEE 802.15.3 superframe for piconets

Timing in the IEEE 802.15.3 piconet is based on the superframe, which is illustrated in Figure 4-7. The superframe is composed of the following three parts:

— The beacon, as described in 6.3.1, which is used to set the timing allocations and to communicate management information for the piconet. The beacon consists of the Beacon frame, as described in 6.3.1, as well as any Announce commands sent by the PNC as a beacon extension, as described in 7.8.4.
— The contention access period (CAP), as described in 7.6.3, which is used to communicate commands and/or asynchronous data if it is present in the superframe.
— The channel time allocation period (CTAP), 7.6.4, which is composed of channel time allocations (CTAs), including management CTAs (MCTAs). CTAs are used for commands, isochronous streams, and asynchronous data connections.

In Figure 4-7 the MCTAs are shown first, but the PNC is allowed to place any number of them at any position in the superframe.

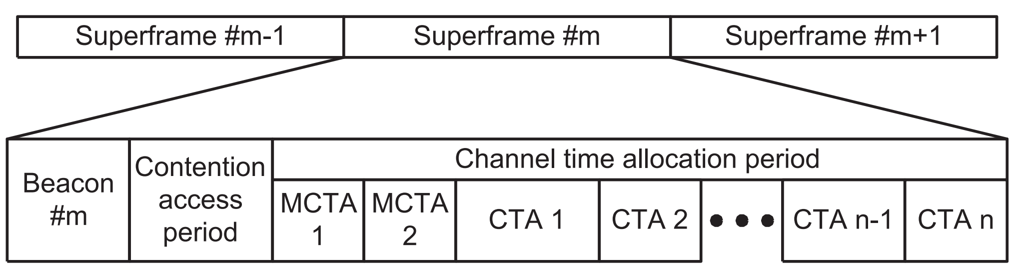

Figure 4-7—IEEE 802.15.3 piconet superframe

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

The CAP uses carrier sense multiple access/collision avoidance (CSMA/CA) for the medium access, as described in 7.6.3. The CTAP, on the other hand, uses a standard time division multiple access (TDMA) protocol where the DEVs have specified time windows, as described in 7.6.4.2. MCTAs, as described in 7.6.4.4, are either assigned to a specific source/destination pair and use TDMA for access or they are shared CTAs that are accessed using the slotted aloha protocol, as described in 7.6.4.5.

4.7.9.2 IEEE 802.15.3 superframe for pairnets

The superframe structure for pairnets is shown in Figure 4-8. Carrier sensing is not required during the pairnet setup period (PSP) and pairnet associated period (PAP). Access method during PSP is different from that during PAP.

a) Pairnet setup period (PSP)

   A PRC sends a Beacon frame periodically to initiate a point-to-point connection. A Beacon frame includes information on the number and duration of the access slots that any target DEV can use by responding with an Association Request. The number of access slots is defined in *pNAccessSlot* and the duration of each slot is defined in *pDAccessSlot*. These values are specified in the Pairnet Synchronization Parameters field set in the Beacon frames, as shown in Figure 6-56. A target DEV selects one of the access slots to send an Association Request command and sends it at the beginning of the selected access slot.

   During the PSP, all frames are transmitted using a modulation and coding scheme (MCS) from the mandatory MCS set. The superframe duration during the PSP equals the interval between the transmission of Beacon frames with the same PHY mode and is indicated by the Pairnet Synchronization Parameters field in the Beacon frame.

b) Pairnet associated period (PAP)

   All frames are sent using short interframe space (SIFS) or retransmission interframe space (RIFS). The superframe in PAP starts from the transmission start time of the Association Response command that replaces the Beacon frame. The superframe has no scheduled end time and is terminated by a Disassociation Request command or an ATP expiration.

NOTE—It is recommended that, for the case of single-carrier PHY, *pNAccessSlot* should be set to 4 in order to optimize the setup time and reduce the probability of collisions between DEVs during the association process. [15]

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

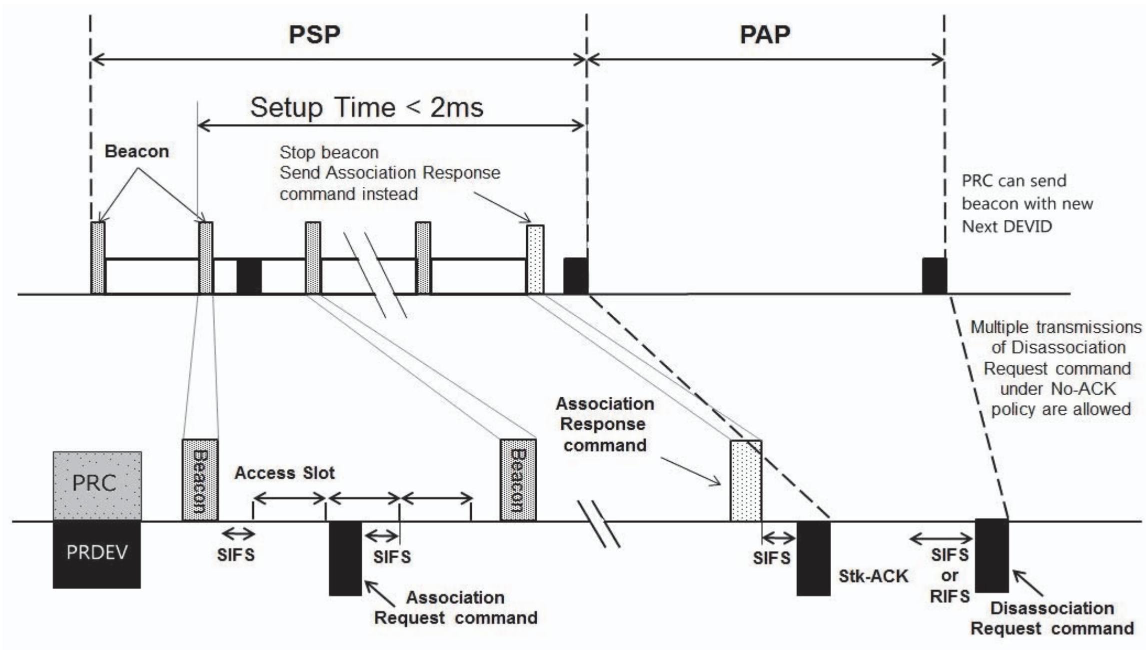

Figure 4-8—One pairnet session

4.7.10 Channel time management

4.7.10.1 Channel time management in a piconet

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

a) Sending asynchronous data in the CAP, if present, as described in 7.6.3.
b) Allocating channel time for isochronous streams in the CTAP, as described in 7.7.2.
c) Allocating asynchronous channel time in the CTAP, as described in 7.7.3.

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

If the DEV needs channel time on a regular basis, it makes a request to the PNC for isochronous channel time, as described in 7.7.2.2. If the resources are available, the PNC allocates time in a CTA for the DEV. If the requirements for the data change, then the DEV is able to request a change to the allocation, as described in 7.7.2.3. The source DEV, destination DEV, or the PNC are all allowed to terminate the stream, as described in 7.7.2.4.

For regular CTAs, the PNC is able to change their position within the superframe every superframe. If a DEV misses a beacon, it is unable to use the allocation for a regular CTA. To avoid lost throughput due to missed beacons, DEVs are allowed to request a special type of CTA called a *pseudo-static* CTA, as described in 7.6.4.2. If the DEV is allocated a pseudo-static CTA, it is allowed to use the CTA for up to *mMaxLostBeacons* missed beacons. The PNC is allowed to move the locations of these CTAs, but needs to maintain the time for the old allocation for *mMaxLostBeacons* superframes to avoid collisions, as described in 7.6.4.2.

Asynchronous allocation is slightly different. Rather than requesting recurring channel time, an asynchronous channel time request, as described in 7.7.3.1, is a request for a total amount of time to be used to transfer its data. The PNC is then able to schedule time for this request when available based on the channel time requirements. Unlike an isochronous allocation, only the source DEV or PNC are allowed to terminate an asynchronous allocation, as described in 7.7.3.2.

4.7.10.2 Channel time management in a pairnet

There is one method for transmitting data between DEVs that form a pairnet. Data exchanges are achieved during a PAP, which is described in 7.6.5.

4.7.11 Data communications between DEVs in a piconet

In order to handle large data frames from layers above the MAC sublayer, this standard supports the fragmentation and defragmentation of these data frames, as described in 7.10. The ability to fragment data frames is also useful to reduce the frame error rate (FER) of a marginal link by decreasing the frame size. The fragments are numbered with a sequence number for the upper layer frame as well as a sequence number for the fragment itself. The total number of fragments of that data frame is also sent to enable the receiving DEV to allocate the correct amount of internal memory to hold the incoming frame.

If the source DEV wishes to verify the delivery of a frame, then one of the acknowledgment (ACK) policies is used, as described in 7.12. This standard provides for three types of ACKs to enable different applications. The *no-ACK policy*, as described in 7.12.2, is appropriate for frames that do not require guaranteed delivery, where the retransmitted frame would arrive too late or where an upper layer protocol is handling the ACK and retransmission protocol. The *immediate-ACK* (Imm-ACK) policy, as described in 7.12.3, provides an ACK process in which each frame is individually ACKed following the reception of the frame. The *delayed-ACK* (Dly-ACK) policy, as described in 7.12.4, lets the source send multiple frames without the intervening ACKs. Instead, the ACKs of the individual frames are grouped into a single response frame that is sent when requested by the source DEV. The Dly-ACK process decreases the overhead in the Imm-ACK process while allowing the source DEV to verify the delivery of frames to the destination.

If the source DEV does not receive the requested ACK, then it has the option of retransmitting the frame, as described in 7.12.8, or dropping the frame. The decision to retransmit or drop the frame depends on the type of data or command that is being sent, the number of times that the source DEV has attempted sending the frame, the length of time it has spent attempting to send the frame, or other implementation-dependent factors.

4.7.12 Information discovery in the piconet

Since IEEE 802.15.3 piconets are ad hoc in nature, it is important for the DEVs in the piconet to be able to find out information about the services and capabilities of the other DEVs in the piconet at any instant in time. This standard supports four methods for discovering information about other DEVs in the piconet: the PNC Information Request command, as described in 6.5.5.2; the Probe Request command, as described in 6.5.5.6; the Announce command, as described in 6.5.6.2; and the Piconet Services command, as described in 6.5.6.1. In addition, the PNC is able to ask a DEV in the piconet to evaluate the channel conditions in either the current channel or in an alternate channel with the Remote Scan Request command, as described in 6.5.8.4. Any DEV in the piconet is able to ask another DEV about the status of the current channel with the Channel Status Request command, as described in 6.5.8.2.

The PNC Information Request command, as described in 7.13.2, is used to obtain information from the PNC about either a specific DEV in the piconet or all of the DEVs in the piconet. The PNC responds to the request with the PNC Information command, as described in 6.5.5.3, which contains information about the DEV (or DEVs) that was requested by the originator of the command.

A DEV uses the Probe Request command, 7.13.3, to find out more detailed information about other DEVs in the piconet. This command allows the originating DEV to retrieve some IEs, as defined in 7.13.3, from a target DEV in the piconet.

One of the goals of connecting DEVs in a piconet is to enable them to share services. However, to do this a DEV needs to be able to discover the services that are available in the piconet as well as to advertise its services to other DEVs in the piconet. This standard enables this with the Piconet Services IE, as described in 6.4.23, which optionally is exchanged in the association process, as described in 7.4.2.

The PNC is responsible for determining the channel in which the piconet will operate. Because the DEVs in the piconet are typically at different geographic locations, they provide additional information about interference or other piconets in the area. The Remote Scan Request command, as described in 7.13.6, is used by the PNC to request information about the channel from other DEVs. In response to the PNC's request, the DEV either scans the channels as requested or rejects the request from the PNC if it is not going to perform the scan.

Any DEV in the piconet is able to request information about the quality of the link between itself and another DEV with the Channel Status Request command, as described in 6.5.8.2. This command is used for two purposes. The first is to allow the DEVs to change transmit power, the data rate, or the requested channel time based on the quality of the data connection between the DEVs. The other use is for the PNC to determine if the DEVs in the piconet are having trouble with the channel. This information along with PNC's scanning of the channel and optional remote scan requests assists the PNC in determining if it needs to change the channel that the piconet is currently using.

4.7.13 Dynamic channel selection in the piconet

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

— Gathering information about the current channel from other DEVs in the piconet using the Channel Status Request command, as described in 7.13.5.
— Performing a passive scan of the channels, as described in 7.15.2.
— Requesting other DEVs to perform a channel scan using the Remote Scan Request command, as described in 7.13.6.

If the PNC determines that the current channel is unsuitable, it uses the dynamic channel selection procedure, as described in 7.15.2, to move the piconet to the new channel. The configuration of the piconet and the CTAs does not change with a channel change so that the services provided by the piconet are not interrupted by the change.

This operation is not applicable for pairnets.

4.7.14 Power management

This standard provides techniques to allow DEVs to reduce power usage by enabling the DEVs to turn off for one or more superframes. These techniques are as follows:

— Device synchronized power save (DSPS) mode
— Piconet synchronized power save (PSPS) mode
— Asynchronous power save (APS) mode

DSPS and PSPS are collectively known as synchronized power save (SPS) modes. In the piconet, DEVs operate in one of four power management (PM) modes; ACTIVE mode, DSPS mode, PSPS mode, or APS mode.

PSPS mode, as described in 7.17.2, allows DEVs to sleep at intervals defined by the PNC. The DEV sends a request to the PNC when it wants to enter the PSPS mode. The PNC informs the piconet by setting the DEV's bit in its power save (PS) Status IE in the beacon. The PNC then selects beacons that will be the system wake beacons and indicates the next one in the PS Status IE for the PSPS set. All DEVs in PSPS mode are required to listen to the system wake beacons.

DSPS mode, as described in 7.17.3, is designed to enable groups of DEVs to sleep for multiple superframes but still be able to wake up during the same superframe. DEVs synchronize their sleep patterns by joining a DSPS set that specifies the interval between wake periods for the DEVs and the next time the DEVs will be awake. Besides allowing the DEVs to wake up and exchange traffic at the same time, the use of DSPS sets makes it easy for other DEVs in the piconet to determine exactly when a DSPS DEV will be available to receive traffic.

APS mode, as described in 7.17.4, allows a DEV to conserve power for extended periods until the DEV chooses to listen for a beacon. The only responsibility of a DEV in APS mode is to communicate with the PNC before the end of its ATP in order to preserve its membership in the piconet.

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

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

4.7.15 Controlling transmit power in the piconet

The ability to control transmit power in the piconet enables DEVs to minimize interference with other wireless networks that share the same channel as well as to decrease the power usage in some PHY implementations. Two methods are provided by this standard for controlling transmitter power. The first method allows the PNC to set a maximum transmit power for the CAP, beacon, and MCTAs, excluding association MCTAs, as described in 7.15.3.2. Since the link between the PNC and the DEV defines the size of the piconet, controlling the power during these times allows the PNC to reduce transmit power without adversely affecting operation of the piconet.

The second method allows DEVs using a CTA to request either an increase or a decrease in the transmit power of the remote DEV, as described in 7.15.3.3. Thus if two DEVs have a "good" link in a CTA, they are able to reduce their transmitter power to decrease the power usage in some PHY implementations, and to reduce interference to other networks.

4.7.16 Superframe structure using quasi-omni mode in millimeter wave (mmWave) piconets

A millimeter wave (mmWave) piconet operates in either omni mode or quasi-omni mode. The superframe structure for omni mode is illustrated in Figure 4-7. The superframe structure for quasi-omni mode is illustrated in Figure 4-9. In quasi-omni mode, the same Beacon frame is transmitted in different quasi-omni directions in a round-robin way in order to allow mmWave DEVs located in different directional coverage to join the piconet, as described in 7.8.6.

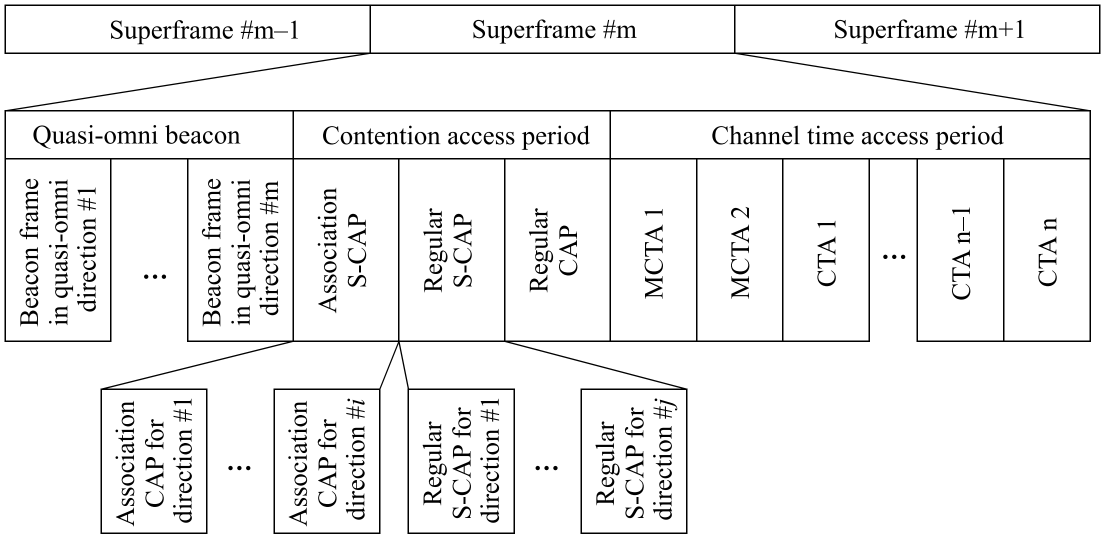

Figure 4-9—Piconet superframe structure in quasi-omni mode

4.7.17 Frame aggregation

Frame aggregation, as described in 7.11, is supported for the purpose of high throughput. Two aggregation methods are provided for mmWave DEVs, one is suitable for normal data and audio/visual (AV) streaming and the other is optimized for low-latency communications. Both use the block acknowledgment (Blk-ACK) mechanism. A third aggregation method is provided for PRDEVs, which is suitable for low-latency, high-efficiency communications and uses the stack acknowledgment (Stk-ACK) feedback mechanism.

4.7.18 Beam forming in the piconet

The beam forming procedure allows a mmWave DEV that is capable of beam forming to increase the antenna gain for supporting high data rate transmission. The beam-forming procedures are described in Clause 15.

4.7.19 Channel probing in the piconet

Channel probing, as described in 7.13.8, is an optional capability that is used by a DEV to determine the best MCS to use with the target DEV under current channel condition.

This operation is not applicable to pairnets.

4.7.20 Unequal error protection (UEP) in mmWave piconets

For flexible error protection that accounts for data that has more emphasis on the MSBs than the LSBs, e.g., video, unequal error protection (UEP) is provided as an option only for mmWave DEVs. Three types of UEP are defined, as described in 7.21, for use with the three mmWave PHYs.

4.8 Characteristics of the 2.4 GHz PHY

4.8.1 General characteristics

The 2.4 GHz PHY, Clause 11, uses the 2.4 GHz to 2.4835 GHz band that is available for unlicensed use in much of the world, as described in 11.1. Two channel plans are defined, one with 4 channels for high-density applications and one with 3 channels to enable better coexistence with IEEE 802.11b™ networks, as described in 11.2.3. The PHY also supports five data rates, ranging from 11 Mb/s to 55 Mb/s, utilizing quadrature phase-shift keying (QPSK), differential quadrature phase shift keying (DQPSK), 16-quadrature amplitude modulation (16-QAM), 32-QAM, and 64-QAM. The base rate of 22 Mb/s is uncoded, while the 11 Mb/s, 33 Mb/s, 44 Mb/s, and 55 Mb/s use trellis coded modulation, 11.3.

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

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

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

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

4.8.2 Coexistence and interoperability

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

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

- Passive scanning
- Dynamic channel selection
- The ability to request channel quality information
- Link quality and received signal strength indication (RSSI)
- A channel plan that minimizes channel overlap
- Lower transmit power
- Transmit power control
- Neighbor piconet capability

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

4.9 Characteristics of the mmWave PHY

4.9.1 mmWave PHY characteristics

The mmWave PHY, as described in Clause 12, is defined for the frequency band of 57.0 GHz to 71.0 GHz, as allocated by the regulatory agencies in Europe, Japan, Canada, and the United States as well as any other areas where the regulatory bodies have allocated this band. While four channels are defined for the PHY, regulatory requirements allow fewer in some regions.

A total of three PHYs are defined for the mmWave PHY. They are as follows:

a) Single Carrier mode in mmWave PHY (SC PHY), as described in 12.2.
b) High Speed Interface mode in mmWave PHY (HSI PHY), as described in 12.3.
c) Audio/Visual mode in mmWave PHY (AV PHY), as described in 12.4.

For DEVs that implement the mmWave PHY, at least one of the three PHYs is required. In addition, to promote coexistence and interoperability, a common mode signaling (CMS) is defined based on a low data rate SC PHY mode, as described in 12.1.13.

The SC PHY supports data rates up to 5 Gb/s. The SC PHY supports a wide range of modulations, π/2 binary phase-shift keying (BPSK), π/2 QPSK, π/2 8-phase shift keying (8-PSK), π/2 16-QAM, on-off keying (OOK), and dual alternate mark inversion (DAMI). The coding schemes included are Reed-Solomon (RS) coding with low-complexity implementation and low-density parity check (LDPC) coding with high error-correcting capability. Code spreading using either linear feedback shift register (LFSR) code or Golay sequence is also applied to increase robustness of the system.

The HSI PHY, as described in 12.3, is designed for non-line-of-sight (NLOS) operation and uses orthogonal frequency-division multiplexing (OFDM) with a forward error correction (FEC) based on LDPC.

The AV PHY, as described in 12.4, is designed for NLOS operation and the transport of uncompressed, high definition video and audio. It uses OFDM modulation with convolutional inner code and a RS outer code. The AV PHY mode supports omni-directional coverage via the low-rate PHY (LRP) for the purpose of setting up high-throughput connections using the high-rate PHY (HRP).

Different PHYs are a result of demands of different market segments, which were based on the development of the usage models for this standard. For example one usage model is for kiosk applications. This usage model requires 1.5 Gb/s at a 1 m range. The SC PHY can provide such a data rate at that short range with less complexity thus lower cost than an OFDM PHY. Another usage model required the streaming of uncompressed video. Due to the nature of uncompressed video signals a special PHY, the AV PHY, was selected to provide high throughput. A third usage model involves an ad hoc system to connect computers and devices around a conference table. In this usage model, all of the devices in the network will have bidirectional, NLOS high-speed, low-latency communication, which is provided for by the HSI PHY. Mandatory data rates of all those PHYs are selected according to specific usage models. In addition, higher data rates are provided to give options to the implementers so that they can best address the different market segments.

4.9.2 Piconets using mmWave PHY modes

When a mmWave PHY PNC-capable DEV starts a piconet, the type of piconet it starts depends on the PHY modes that are supported. For example, if the PNC-capable DEV supports only the AV PHY mode, it would start an AV piconet in which the beacon is sent with the AV PHY mode and the contention periods (CPs) would use the AV PHY mode. The AV PNC would also send the Sync frame to improve coexistence with other piconets. DEVs that support only the AV PHY mode are able to find and join the piconet using the AV PHY mode.

The same process is used for a mmWave PHY PNC-capable DEV that supports only HSI PHY mode or SC PHY mode, with the exception that an SC PHY mode PNC does not need to send a Sync frame since its beacon is sent in CMS mode already.

If a PNC-capable DEV supports more than one mmWave PHY mode, then it is able to select the type of piconet it starts, potentially starting more than one piconet, each with a different PHY mode, with the additional piconets as dependent piconets. Alternatively, the multi-mode PNC is able to have a single piconet in which DEVs use any MCS in a CTA that is supported by both DEVs.

The complete rules are described in 12.1.

4.10 Characteristics of HRCP PHY

4.10.1 HRCP PHY characteristics

The HRCP PHY, as described in Clause 13, is designed for a high data-rate application. The frequency band is 57.0 GHz to 66.0 GHz, the same as that for the mmWave PHY.

Two PHYs are defined for the HRCP PHY, as follows:

a) Single Carrier mode in HRCP (HRCP-SC) PHY, as described in 13.2
b) OOK mode in HRCP (HRCP-OOK) PHY, as described in 13.3

For DEVs that implement the HRCP PHY, at least one of the PHYs is required. The HRCP PHY supports channel bonding using up to four channels for high throughput.

The HRCP-SC PHY is designed for extremely high PHY service access point (PHY-SAP) payload bit-rate up to 13 Gb/s with a single channel and up to 157 Gb/s with a multiple input, multiple output (MIMO) channel. The SC PHY supports a wide range of modulations, π/2 BPSK, π/2 QPSK, 16-QAM, 64-QAM, and 256 QAM. The FEC scheme defines user rate-compatible LDPC codes with rates of 14/15 and 11/15. Channel aggregation is supported in HRCP-SC PHY.

The HRCP-OOK PHY is designed for cost effective DEVs that require low power, low complexity, and simple design. The HRCP-OOK PHY supports a single modulation scheme, OOK, and a single FEC scheme, RS. Channel aggregation and MIMO are not used in HRCP-OOK PHY.

4.10.2 Pairnet using HRCP PHY

When a PRC-capable DEV starts a pairnet, the type of pairnet it starts depends on the PHY modes that are supported. For example, if the PRC-capable DEV supports only the HRCP-SC mode, it would start an HRCP-SC pairnet in which the Beacon frame is sent with the HRCP-SC mode. DEVs that support only the HRCP-SC mode are able to find and connect to the pairnet in HRCP-SC mode.

The same process is used for a PRC-capable DEV that supports only HRCP-OOK mode. If a PRC-capable DEV supports more than one HRCP PHY mode, then it is able to select the type of pairnet it starts. It allows connection from each type of DEV by transmitting both the HRCP-SC mode Beacon frame and the HRCP-OOK mode Beacon frame. Figure 4-10 is an example of transmitting dual mode Beacon frames. The number and duration of the access slots and the superframe duration for each PHY mode are indicated by the Beacon frame with the corresponding PHY mode.

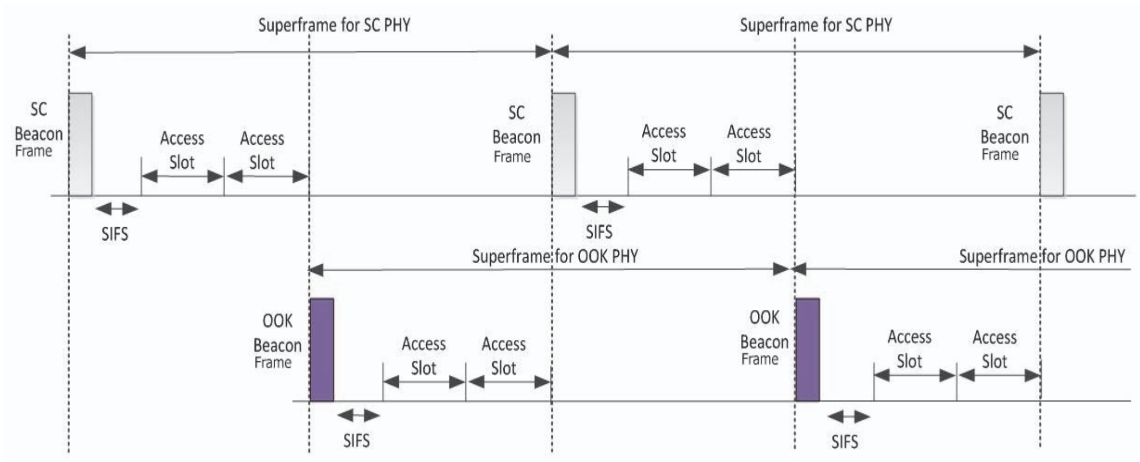

Figure 4-10—Example of transmitting dual mode beacons

4.11 Characteristics of THz PHY

4.11.1 THz PHY characteristics

The THz PHY, as described in Clause 14, is designed as an extension for wireless switched point-to-point applications, operating at a nominal PHY data rate of 100 Gb/s with fallbacks to lower data rates as needed. The frequency band is from 252 GHz to 450 GHz. The THz PHY supports a range of channel bandwidths from 2.16 GHz to 69.12 GHz.

The following two PHY modes are defined for the THz PHY:

— THz single carrier mode PHY (THz-SC PHY), as described in 14.2
— THz on-off keying mode PHY (THz-OOK PHY), as described in 14.3

For DEVs that implement the THz PHY, at least one of the PHY modes is required.

The THz-SC PHY is designed for extremely high PHY-SAP payload data rates up to 100 Gb/s, depending on the combination of modulation, bandwidth, and coding used. The THz-SC PHY supports a wide range of modulations: π/2 BPSK, π/2 QPSK, π/2 8-PSK, π/2 8-APSK, 16-APSK, 32-APSK, 16-QAM, and 64-QAM. The FEC consists of two LDPC codes with rates of 14/15 and 11/15.

The THz-OOK PHY is designed for cost effective DEVs that require low complexity and simple design. The THz-OOK PHY supports a single modulation scheme, OOK, and three FEC schemes. The RS code is mandatory and allows simple decoding without soft decision information. The LDPC codes with rates of 14/15 and 11/15 are optional and allow the use of soft-decision information.

4.11.2 Pairnet using THz PHY

When a PRC-capable DEV starts a pairnet, the type of pairnet it starts depends on the supported PHY modes. For example, if the PRC-capable DEV supports only the THz-SC mode, it will start a THz-SC pairnet in which the Beacon frame is sent with the THz-SC mode. DEVs that support only the THz-SC mode are able to find and connect to the pairnet in THz-SC mode. The same process is used for a PRC-capable DEV that supports only THz-OOK mode.

If a PRC-capable DEV supports more than one THz PHY mode, then it is able to select the type of pairnet it starts. It allows connection from each type of DEV by transmitting both the THz-SC mode Beacon frame and the THz-OOK mode Beacon frame. Figure 4-10 is an example of transmitting dual mode Beacon frames. The number and duration of the access slots and the superframe duration for each PHY mode are indicated by the Beacon frame with the corresponding PHY mode.

The switched point-to-point nature of the connection means that the pairnet connection can be terminated and a new pairnet connection can be set up with a different destination device. The process of switching and the determination of the direction to the new access point is out of the scope of this standard.

4.12 Conventions

Unless otherwise specified, the term *increment* or *incremented* means to increase a value by one.

The editorial conventions in this standard are as follows:

— Frame names (e.g., Beacon frame, MAC Command frame) are capitalized and include the word *frame.*
— Command names (e.g., Association Request command) are capitalized and include the word c*ommand*.
— Field names are capitalized and include the word field, which is not capitalized, e.g., Destination PAN Identifier field. The only exception is in a figure that shows the field, in which the word "field" is not included.
— IE names are capitalized and include the acronym IE (e.g., Capability IE).
— MAC and PHY PIB entries are preceded by a lower case prefix, *mac* or *phy*, respectively, and are capitalized at word boundaries, e.g., "*macDevServicesBroadcast*" and "*phyCurrentChannel.*"
— Constants begin with a lower case, italicized "*m*" for MAC related constants and a lower case italicized "*p*" for PHY related constants. Both use the same capitalization and use rules as PIB attributes.
— Service primitives names (e.g., MLME or MCPS) are ALL CAPS, but the .request, .indication, .response, and .confirm in the primitive names are always lower case.
— Primitive parameters are capitalized at word boundaries, e.g., NumberOfPiconets.
— In a PIB attribute, constant or service primitive, acronyms are not capitalized after the first letter, e.g., "*pMinTpcLevel,*" "*macDevServicesBroadcast,*" or "ScanForBsid."

4.13 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 4-11, showing the service hierarchy and the relationship of the two correspondent users and their associated layer (or sublayer) peer protocol entities.

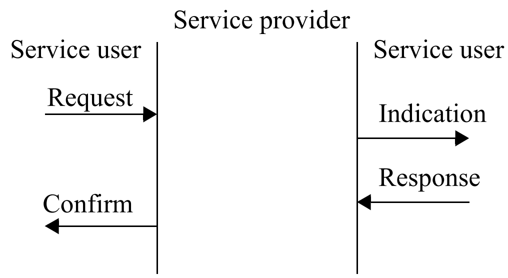

Figure 4-11—Service primitives

The services are specified by describing the information flow between the user and the 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 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 the following 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. Layer management

5.1 Overview of management model

Both MAC and PHY layers conceptually include management entities, called the MAC sublayer management entity and PHY layer management entity (MLME and PLME, respectively). These entities provide the layer management service interfaces for the layer management functions. Figure 5-1 depicts the relationship among the management entities. The protocol adaptation layer (PAL) includes both the frame convergence sublayer (FCSL) and device management entity (DME). The reference model depicts the interactions of a single PAL in a single piconet or pairnet; support for multiple piconets, and/or PALs is implementation dependent.

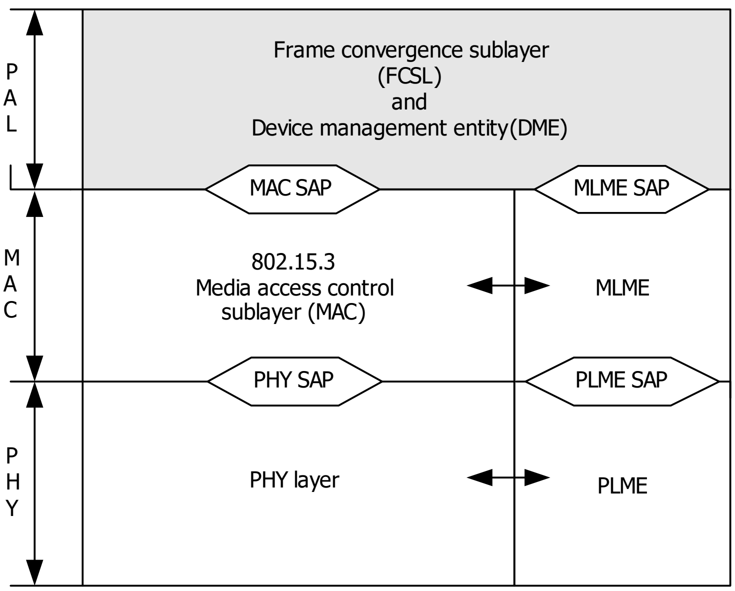

Figure 5-1—The reference model used in this standard

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

The various SAPs within this model are the following:

a) PAL SAP (not shown)
b) MAC SAP
c) MLME SAP

The PHY SAP and PLME SAP are not defined in this standard as they are rarely, if ever, exposed in a typical implementation. The PHY management objects and attributes are accessed through the MLME SAP with the generic management primitives defined in 5.2.

The MAC SAP is described further in Annex B.

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

5.2 Generic management primitives

5.2.1 Overview

The management information specific to each layer is represented as a personal area network (PAN) information base (PIB) for that layer. In a LAN/MAN the corresponding information is in the management information base (MIB) and is often associated with a management protocol such as simple network management protocol (SNMP) (see IETF RFC 1157:1990 [B7]).[16] However, piconets are not intended to be managed across a network but rather use the management information to ascertain the characteristics of the layer or sublayer.

The MLME and PLME are viewed as “containing” the PIB for that layer or sublayer. The generic model of PIB-related management primitives exchanged across the management SAPs is to allow the SAP user entity to either “GET” the value of a PIB attribute, or to “SET” the value of a PIB attribute. The invocation of a SET.request primitive may require the layer entity to perform certain defined actions. An attempt to “GET” a PIB attribute identified as a “write only” attribute is not a valid operation. An attempt to “SET” a PIB attribute identified as a “read only” attribute is not a valid operation.

The GET and SET primitives are represented as requests with associated confirm primitives. The DME uses the services provided by the MLME through the MLME SAP to access the management objects in either the MAC PIB or PHY PIB. The primitives are summarized in Table 5-1.

Table 5-1—Summary of generic management primitives

Name	Request	Confirm
MLME-GET	5.2.2	5.2.3
MLME-SET	5.2.4	5.2.5

The parameters used for these primitives are defined in Table 5-2

Other SAP-specific primitives are identified in 5.3.

[16]The numbers in brackets correspond to those of the bibliography in Annex A.

.

Table 5-2—MLME generic management primitive parameters

Name	Type	Valid range	Description
PIBattribute	Octet string	Any PIB attribute, as defined in 5.4 or 11.7	The name of the PIB attribute
PIBvalue	Variable	As defined in 5.4 or 11.7	The PIB value
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request
ReasonCode	Enumeration	INVALID_PIB_ATTRIBUTE_NAME, INVALID_PIB_ATTRIBUTE_VALUE, READ_ONLY_PIB_ATTRIBUTE, WRITE_ONLY_PIB_ATTRIBUTE	Indicates the reason for a ResultCode of FAILURE

5.2.2 MLME-GET.request

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

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

The primitive parameter is defined in Table 5-2.

5.2.3 MLME-GET.confirm

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

```
MLME-GET.confirm                    (
                                    PIBattribute,
                                    PIBvalue,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-2.

5.2.4 MLME-SET.request

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

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

The primitive parameters are defined in Table 5-2.

5.2.5 MLME-SET.confirm

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

```
MLME-SET.confirm                    (
                                    PIBattribute,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-2.

5.3 MLME SAP interface

5.3.1 Overview

The services provided by the MLME to the DME are described in an abstract way and do not imply any particular implementation. MLME SAP primitives are of the general form ACTION.request followed by ACTION.confirm. An ACTION.indication provides information to the DME that originated either in the local MAC or from another DEV. The DME optionally responds to the indication by issuing an ACTION.response. The DME uses the services provided by the MLME through the MLME SAP. The MLME interface models a single piconet environment; support for multiple piconets is implementation dependent. The primitives are summarized in Table 5-3.

Table 5-3—Summary of MLME primitives

Name	Request	Confirm	Indication	Response
MLME-RESET	5.3.2.2	5.3.2.3	—	—
MLME-SCAN	5.3.3.2	5.3.3.3	5.3.3.4	—
MLME-START	5.3.4.2	5.3.4.3	—	—
MLME-STOP	5.3.5.2	5.3.5.3	—	—
MLME-ASSOCIATE	5.3.6.2	5.3.6.3	5.3.6.4	5.3.6.5
MLME-DISASSOCIATE	5.3.7.2	5.3.7.3	5.3.7.4	—
MLME-MEMBERSHIP-UPDATE	5.3.8.2	5.3.8.3	—	—
MLME-SECURITY-ERROR	—	—	5.3.8.4	—
MLME-SECURITY-MESSAGE	5.3.8.5	5.3.8.6	5.3.8.7	—
MLME-PNC-HANDOVER	—	—	5.3.9.2	5.3.9.3
MLME-NEW-PNC	—	—	5.3.9.4	—
MLME-DEV-INFO	5.3.10.2	5.3.10.3	5.3.10.4	—
MLME-SECURITY-INFO	5.3.11.2	5.3.11.3	5.3.11.4	5.3.11.5
MLME-APPLICATION-SPECIFIC	5.3.12.2	5.3.12.3	5.3.12.4	—
MLME-ANNOUNCE-SERVICE	5.3.13.2	5.3.13.3	—	—

Table 5-3—Summary of MLME primitives (*continued*)

Name	Request	Confirm	Indication	Response
MLME-PICONET-SERVICES	5.3.13.4	5.3.13.5	5.3.13.6	—
MLME-CREATE-STREAM	5.3.14.2	5.3.14.3	5.3.14.4	—
MLME-MODIFY-STREAM	5.3.14.5	5.3.14.6	—	—
MLME-TERMINATE-STREAM	5.3.14.7	5.3.14.8	5.3.14.9	—
MLME-BSID-CHANGE	5.3.15.2	5.3.15.3	—	—
MLME-PICONET-PARM-CHANGE	—	—	5.3.15.4	—
MLME-PS-SET-INFORMATION	5.3.16.2	5.3.16.3	—	—
MLME-SPS-CONFIGURE	5.3.16.4	5.3.16.5	—	—
MLME-PM-MODE-CHANGE	5.3.16.6	5.3.16.7	5.3.16.8	—
MLME-MONITOR-PM-MODE	5.3.16.9	5.3.16.10	5.3.16.11	—
MLME-MULTICAST-CONFIGURATION	5.3.17.2	5.3.17.3	—	—
MLME-MULTICAST-RX-SETUP	5.3.17.4	5.3.17.5	—	—
MLME-BEACON-EVENT	5.3.18.2	5.3.18.3	5.3.18.4	—
MLME-TXDIV	5.3.19.2	5.3.19.3	—	—
NOTE—MLME-ASSOCIATE.Response primitive is valid only for pairnets.				

For piconets and pairnets, the MLME interface models a single piconet or pairnet environment; while support for multiple pairnets for PRDEVs is not allowed, support for multiple piconets for non-PRDEVs is implementation-dependent.

5.3.2 Resetting the MAC/MLME

5.3.2.1 Overview

These primitives support the process of resetting the MAC/MLME, which also resets the PHY/PLME. The parameters used for these primitives are defined in Table 5-4.

Table 5-4—MLME-RESET primitive parameters

Name	Type	Valid range	Description
SetDefaultPIB	Boolean	TRUE, FALSE	If TRUE, all PIB attributes are set to their default values. The default values are implementation dependent. If FALSE, the MAC is reset, but all PIB attributes retain the values that were in place prior to the generation of the MLME-RESET.request primitive.
ResultCode	Enumeration	READY, ERROR	If READY, the MLME has successfully completed initialization and is ready to receive requests. If ERROR, the MLME was unable to successfully complete initialization and is unable to receive requests.

5.3.2.2 MLME-RESET.request

This primitive requests that the MAC entity be reset to its initial conditions. The semantics of this primitive are as follows:

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

The primitive parameter is defined in Table 5-4.

The MLME restores the MAC and its internal variables (other than PIB attributes) to their initial state and values. The SetDefaultPIB parameter governs whether PIB attributes are reinitialized or left unchanged.

5.3.2.3 MLME-RESET.confirm

This primitive is generated by the MLME when it completes its initialization process. The semantics of this primitive are as follows:

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

The primitive parameter is defined in Table 5-4.

5.3.3 Scanning for piconets and pairnets

5.3.3.1 Overview

These primitives support the process of determining the presence or absence of piconets or pairnets, as described in 7.2.2. The parameters used for these primitives are defined in Table 5-5.

Table 5-5—MLME-SCAN primitive parameters

Name	Type	Valid range	Description
ScanForBsid	Boolean	TRUE, FALSE	Indicates if the scan process should search for a specific BSID, as described in 7.2.2. Not applicable for pairnets.
ScanMode	Enumeration	IMMEDIATE, TIMEOUT	Indicates the scan mode for pairnets.
NumberOfPairnets	Integer	0–255	The number of pairnets found during the scanning process.
PairnetDescriptionSet	Set of pairnet descriptions, as defined in Table 5-8	A set containing zero or more instances of a PairnetDescription	The PairnetDescriptionSet is returned to indicate the results of the scan request.
Bsid	Octet string	As defined in 6.4.3	The text string of a specific piconet for which to scan. This parameter is not used if ScanForBsid is FALSE.

Table 5-5—MLME-SCAN primitive parameters (*continued*)

Name	Type	Valid range	Description
ScanForPnid	Boolean	TRUE, FALSE	Indicates if the scan process should search for a specific PNID, as described in 7.2.2.
Pnid	Integer	0–65535	The identifier (ID) of a specific piconet for which to scan. This parameter is not used if ScanForPnid is FALSE.
ScanForPncAddress	Boolean	TRUE, FALSE	Indicates if the scan process should search for a PNC with a specific MAC address, as described in 7.2.2.
PncAddress	MAC address	Any valid individual MAC address, as described in 6.1	The MAC address of a specific PNC for which to scan. This parameter is not used if ScanForPncAddress is FALSE.
NumberOfPiconets	Integer	0–255	The number of piconets found during the scanning process.
PiconetDescriptionSet	Set of piconet descriptions, as defined in Table 5-6	A set containing zero or more instances of a PiconetDescription	The PiconetDescriptionSet is returned to indicate the results of the scan request.
NumberOfChannels	Integer	0 to the maximum number of PHY-dependent channels, as defined in 11.2.3	Indicates the number of channels scanned.
ChannelRatingList	Ordered list of integers.	0 to the maximum number of PHY-dependent channels, as defined in 11.2.3	Specifies a list of the channels scanned ordered from the best to the worst in terms of interference.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	OTHER	Indicates the reason for a ResultCode of FAILURE.

In Table 5-5, a PiconetDescriptionSet is a set of PiconetDescriptions. Each PiconetDescription consists of the elements shown in Table 5-6.

Table 5-6—Elements of PiconetDescription

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.3	The text string identifier of a discovered piconet.
Pnid	Integer	0–65535	The PNID of a discovered piconet.
PncAddress	MAC address	Any valid individual MAC address	The MAC address of the PNC of the piconet that was found.

Table 5-6—Elements of PiconetDescription (*continued*)

Name	Type	Valid range	Description
ChannelIndex	Integer	0 to the maximum number of PHY-dependent channels, as defined in 11.2.3	A PHY-dependent channel number on which the piconet was found.
SECmode	Enumeration	MODE_0, MODE_1	The security mode of the piconet that was found, as described in 6.3.1.
SignalQuality	Integer	0–15	Indicates the quality of the received frame or beacon for this piconet. The value is implementation dependent with 0 indicating the lowest quality and 15 the highest quality.
NumApplicationSpecificData	Integer	0–255	Indicates the number of ApplicationSpecificDataSets advertised in the beacon of a discovered piconet.
ApplicationSpecificDataSet	Set of application specific data, as defined in Table 5-7	A set containing zero or more instances of ApplicationSpecific Data	The ApplicationSpecificDataSet is returned to indicate the ApplicationSpecificData offered by the discovered piconet.
PhyMode	Enumeration	2.4_GHZ, SC_MMWAVE, HSI_MMWAVE, AV_MMWAVE	The PHY mode that is being used in the piconet that was found.

In Table 5-5, the ChannelRatingList is a set of *N* integer values, where *N* equals the number of channel numbers provided by the PHY. The elements of the set are channel numbers and they are ordered from best (least interference) at the lowest set index to worst (most interference) at the highest set index.

In Table 5-6, an ApplicationSpecificDataSet is a set of ApplicationSpecificData. Each ApplicationSpecificData consists of the elements shown in Table 5-7. The ApplicationSpecificDataSets are only the current values advertised in the beacon and potentially are different in subsequent beacons.

Table 5-7—Elements of ApplicationSpecificData

Name	Type	Valid range	Description
UniqueId	Octet string	Any valid organizationally unique identifier (OUI) or Company ID (CID), as defined in 6.4.8	The unique identifier for the entity that defines the format of the data, as described in 6.4.8.
ApplicationData	Octet string	Any valid octet string of length up to ApplicationDataLength	Application-dependent data, as described in 6.4.8.

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

It is not possible to obtain piconet services information, as described in 7.4.3, during the scan process.

In Table 5-5, a PairnetDescriptionSet is a set of PairnetDescriptions. Each PairnetDescription consists of the elements shown in Table 5-8.

Table 5-8—Elements of PairnetDescription

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.2	The text string identifier of a discovered pairnet.
Pnid	Integer	0–65535	The PNID of a discovered pairnet.
PrcAddress	MAC address	Any valid individual MAC address	The MAC address of the PRC of the pairnet that was found.
SECmode	Enumeration	MODE_0, MODE_1	The security mode of the pairnet that was found, as described in 6.3.1.
SignalQuality	Integer	0–15	Indicates the quality of the received frame or beacon for this pairnet. The value is implementation dependent, with 0 indicating the lowest quality and 15 the highest quality.
PrcCapability	PRC Capability, as defined in 6.4.13 for HRCP PHY and 6.4.16 for THz PHY	As defined in 6.4.13 for HRCP PHY and 6.4.16 for THz PHY	Capability of the PRC in the Beacon frame.
HigherLayer ProtocolInformation	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.45	Included 3 octets Unique ID and Higher Layer Protocol Information in variable length.
PhyMode	Enumeration	HRCP_SC_PHY, HRCP_OOK_PHY, HRCP_BOTH_PHY THZ_SC_PHY, THZ_OOK_PHY, THZ_BOTH_PHY	The PHY mode that is being used in the pairnet that was found.

5.3.3.2 MLME-SCAN.request

This primitive is used to initiate the passive scan procedure to search for either a specific piconet or pairnet if ScanForBsid is TRUE and as indicated by Bsid or any piconet or pairnet if ScanForBsid is FALSE. The semantics of this primitive are as follows:

```
MLME-SCAN.request                    (
                                     ScanForBsid,
                                     Bsid,
                                     ScanForPnid,
                                     Pnid,
                                     ScanForPncAddress,
                                     PncAddress,
                                     ScanMode,
                                     Timeout
                                     )
```

The primitive parameters are defined in Table 5-5.

5.3.3.3 MLME-SCAN.confirm

This primitive is used to report the result of the request to initiate the passive scan procedure to search for either a specific piconet or pairnet if ScanForBsid is TRUE and as indicated by Bsid, or any piconet or pairnet if ScanForBsid is FALSE. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-5. All of the piconets found during the scan will be reported in separate elements of the PiconetDescriptionSet, even if more than one piconet is found on a given channel. For a PRDEV, only pairnets will be reported.

5.3.3.4 MLME-SCAN.indication

This primitive is used to report the result of a passive scan that was initiated by the MAC. The semantics of this primitive are as follows:

```
MLME-SCAN.indication                 (
                                     NumberOfPiconets,
                                     PiconetDescriptionSet,
                                     NumberOfChannels,
                                     ChannelRatingList
                                     NumberOfPairnets,
                                     PairnetDescriptionSet
                                     )
```

The primitive parameters are defined in Table 5-5. For a PRDEV, only pairnets will be reported immediately if the ScanMode parameter is set to IMMEDIATE. If the ScanMode parameter is set to TIMEOUT, the result will be reported after the SCAN Timeout timer expires.

5.3.4 Starting a piconet or pairnet

5.3.4.1 Overview

These primitives support the process of creating a new piconet or pairnet with the DEV acting as PNC or PRC, as described in 7.2.3. The parameters used for these primitives are defined in Table 5-9.

Table 5-9—MLME-START primitive parameters

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.3	The BSID of the new piconet or pairnet.
SecMode	Enumeration	MODE_0, MODE_1	The security mode of the piconet or pairnet, as described in 6.3.1.
DevId	Integer	Any valid DEVID, as defined in 6.2.5	The assigned DEVID for the DEV that is acting as the PNC, as described in 7.2.3. This parameter is not valid for PRDEVs.
MinDepSuperframePercent	Integer	1–100	The minimum percent of the superframe requested as a CTA for the dependent piconet, as described in 7.2.8 and 7.2.9. This parameter is not valid for PRDEVs.
DesiredDepSuperframePercent	Integer	1–100	The desired percent of the superframe requested as a CTA for the dependent piconet, as described in 7.2.8 and 7.2.9. This parameter is not valid for PRDEVs.
AllocatedSuperframePercent	Integer	0–100	The percent of the superframe allocated to the new dependent piconet. If the channel time request was rejected, the value shall be set to zero. This parameter is ignored if the DEV is starting an independent piconet. This parameter is not valid for PRDEVs.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_PNC_CAPABLE, NO_CHANNELS_AVAILABLE, ALREADY_PNC, OTHER	Indicates the reason for a ResultCode of FAILURE.
PhyMode	Enumeration	2.4_GHZ, SC_MMWAVE, HSI_MMWAVE, AV_MMWAVE THZ_SC_PHY, THZ_OOK_PHY, THZ_BOTH_PHY	The PHY mode that will be used for the beacons and the CP(s) in the piconet or pairnet that will be started.

Table 5-9—MLME-START primitive parameters (*continued*)

Name	Type	Valid range	Description
PrcCapabilityIe	PRC Capability, as defined in 6.4.11a for HRCP PHY and 6.4.16 for THz PHY	As defined in 6.4.11a for HRCP PHY and 6.4.16 for THz PHY	Capability of the PRC in the Beacon frame.
HigherLayerProtocol Information	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.45	Includes 3 octets Unique ID and HigherLayerProtocol Information in variable length

5.3.4.2 MLME-START.request

This primitive is used to start a piconet or pairnet. If the DEV is not a member of the piconet or pairnet, this primitive causes the DEV to start an independent piconet or pairnet. If the DEV is a member of the piconet, this primitive causes the DEV to start a child piconet. If the DEV is associated as a neighbor member of a piconet, this primitive causes the DEV to start a neighbor piconet. The semantics of this primitive are as follows:

```
MLME-START.request                    (
                                      Bsid,
                                      SecMode,
                                      MinDepSuperframePercent,
                                      DesiredDepSuperframePercent,
                                      PhyMode
                                      PrcCapabilityIe,
                                      HigherLayerProtocolInformation
                                      )
```

The primitive parameters are defined in Table 5-9.

5.3.4.3 MLME-START.confirm

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

```
MLME-START.confirm                    (
                                      DevId,
                                      AllocatedSuperframePercent,
                                      ResultCode,
                                      ReasonCode
                                      )
```

The primitive parameters are defined in Table 5-9.

5.3.5 Stopping a piconet or a pairnet

5.3.5.1 Overview

These primitives support the process of stopping operations as a PNC or PRC. The process may result in the shutdown of piconet or pairnet operations, as described in 7.2.10, or the handover of PNC operations to another DEV in the piconet, as described in 7.2.4 and 7.2.7. The parameters used for these primitives are defined in Table 5-10.

In Table 5-10, the HandoverTargetList is a set of *N* DEVIDs, where *N* equals the number of target DEVIDs for a handover provided by the DME. The elements of the set are DEVIDs, and they are ordered from preferred target at the lowest set index to the least preferred target at the highest set index. If the HandoverTargetList contains only the BcstID, as defined in 6.2.5, the DME is not indicating any preference for the handover target.

Table 5-10—MLME-STOP primitive parameters

Name	Type	Valid range	Description
RequestType	Enumeration	SHUTDOWN, HANDOVER	If SHUTDOWN, the current piconet or pairnet operations will be stopped. If HANDOVER, an attempt to handover PNC operations will be made. Only SHUTDOWN is valid for PRDEV.
AllowedHandoverTime	Duration	0–65535	If RequestType is HANDOVER, the time in milliseconds in which a handover attempt must be completed. This parameter is not valid for PRDEV.
NumHandoverTargetDev	Integer	0–*mMaxNumValidDEVs*	The number of DEVs in the HandoverTargetList. This parameter is not valid for PRDEV.
HandoverTargetList	List of DEVIDs	0 to maximum number of DEVIDs, as defined in 6.2.5	If RequestType is HANDOVER, specifies a list of Target DEVIDs for a handover attempt. This parameter is not valid for PRDEV.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_A_PNC, HANDOVER_FAILED, HIGHER LAYER REQUESTED, OTHER	Indicates the reason for a ResultCode. NOT_A_PNC and HANDOVER_FAILED are not valid responses for PRDEVs.

5.3.5.2 MLME-STOP.request

This primitive initiates the piconet or pairnet shutdown procedure or the piconet handover procedure. The semantics of this primitive are as follows:

```
MLME-STOP.request                    (
                                     RequestType,
                                     AllowedHandoverTime,
                                     NumHandoverTargetDev,
                                     HandoverTargetList
                                     )
```

The primitive parameters are defined in Table 5-10.

5.3.5.3 MLME-STOP.confirm

This primitive reports the results of the request to stop operations as a PNC or PRC. The semantics of this primitive are as follows:

```
MLME-STOP.confirm                    (
                                     ResultCode,
                                     ReasonCode
                                     )
```

The primitive parameters are defined in Table 5-10.

5.3.6 Associating with a piconet or pairnet

5.3.6.1 Overview

The following primitives support the process of a DEV associating with a PNC or PRC, as defined in 7.4.2. The parameters used for these primitives are defined in Table 5-11.

Table 5-11—MLME-ASSOCIATE primitive parameters

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.3	The BSID of the target PNC or PRC for the association.
Pnid	Integer	0–65535	The PNID of the target PNC or PRC for the association, as defined in 6.2.3.
PncAddress	MAC address	Any valid individual MAC address	The MAC address of the target PNC or PRC for the association.
ChannelIndex	Integer	0–255	A PHY-dependent channel number to search for the target PNC for the association.
PiconetServicesInquiry	Boolean	TRUE, FALSE	Requests that the PNC or PRC send the services information about the piconet, as described in 7.4.3.

Table 5-11—MLME-ASSOCIATE primitive parameters (*continued*)

Name	Type	Valid range	Description
DevId	Integer	Any valid DEVID, as defined in 6.2.5	The DEVID assigned to a DEV as the result of an association or the DEVID of a DEV that has joined the piconet.
DevAddress	MAC address	Any valid individual MAC address	The MAC address of a DEV that has joined the piconet.
NeighborPiconetRequest	Boolean	TRUE, FALSE	Indicates that the DEV will join as a neighbor PNC rather than as a member of the piconet. Not valid for PRDEV.
PrdevCapabilityIe	PRDEV Capability as defined in 6.4.14 for HRCP PHY and 6.4.17 for THz PHY	As defined in 6.4.14 for HRCP PHY and 6.4.17 for THz PHY	Capability of the PRDEV in the Association Request command.
PairnetOperation ParametersIe	As defined in 6.4.15 for HRCP PHY and 6.4.18 for THz PHY	As defined in 6.4.15 for HRCP PHY and 6.4.18 for THz PHY	Capability of the PRDEV in the Association Response command.
VendorSpecificIE	Octet string	Any valid Vendor Defined IE, as defined in 6.4.24	The Vendor Defined IE, if present, in the Association Response command, as described in 6.5.2.3.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, PNC_NOT_FOUND, PNC_DENIED, PNC_BUSY, ALREADY_ASSOCIATED, NEIGHBOR_REFUSED, HIGHER_LAYER_DENIED, OTHER	Indicates the reason for a ResultCode of FAILURE. For PRDEVs, NEIGHBOR_REFUSED is not applicable.
HigherLayerProtocol Information	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.45	Includes 3 octets, Unique ID and Higher Layer Protocol Information in variable length.

5.3.6.2 MLME-ASSOCIATE.request

This primitive initiates the association procedure. The semantics of this primitive are as follows:

```
MLME-ASSOCIATE.request          (
                                Bsid,
                                Pnid,
                                PncAddress,
                                ChannelIndex,
                                NeighborPiconetRequest,
                                PiconetServicesInquiry,
                                PrdevCapabilityIe,
                                Timeout
                                HigherLayerProtocolInformation
                                )
```

The primitive parameters are defined in Table 5-11.

5.3.6.3 MLME-ASSOCIATE.confirm

This primitive reports the result of the association procedure. The semantics of this primitive are as follows:

```
MLME-ASSOCIATE.confirm          (
                                DevId,
                                VendorSpecificIE,
                                PairnetOperationParametersIe,
                                HigherLayerProtocolInformation,
                                ResultCode,
                                ReasonCode
                                )
```

The primitive parameters are defined in Table 5-11.

5.3.6.4 MLME-ASSOCIATE.indication

This primitive is used to indicate that a new non-PRDEV has associated with the same piconet as this DEV, or a new PRDEV has associated with this PRDEV. The semantics of this primitive are as follows:

```
MLME-ASSOCIATE.indication       (
                                DevId,
                                DevAddress
                                PairnetOperationParametersIe,
                                HigherLayerProtocolInformation,
                                )
```

The primitive parameters are defined in Table 5-11.

5.3.6.5 MLME-ASSOCIATE.response

This primitive is used to indicate the DEVID that the PRC has selected to associate with. The semantics of this primitive are as follows:

```
MLME-ASSOCIATE.response     (
                            DevId,
                            PncAddress,
                            HigherLayerProtocolInformation,
                            ResultCode,
                            ReasonCode
                            )
```

The primitive parameters are defined in Table 5-11.

5.3.7 Disassociation from a piconet or pairnet

5.3.7.1 Overview

The following primitives are used when a DEV disassociates from a PNC or PRC and when the PNC or PRC disassociates a DEV from the piconet or pairnet, as described in 7.4.5. For the pairnet, disassociation by either the DEV or the PRC invokes termination of the pairnet. The parameters used for these primitives are defined in Table 5-12.

Table 5-12—MLME-DISASSOCIATE primitive parameters

Name	Type	Valid range	Description
DevId	Integer	Any valid DEVID, as defined in 6.2.5	The DEVID of the disassociated DEV.
DevAddress	MAC address	Any valid individual MAC address	The MAC address of the disassociated DEV.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, CURRENTLY_PNC, DEV_ATP_EXPIRED, PNC_ATP_EXPIRED, DEV_DISASSOCIATED, OTHER_PNC_ACTION, HIGHER_LAYER_INITIATED, UNKNOWN	Indicates the reason for the disassociation of a DEV from a piconet or pairnet.

5.3.7.2 MLME-DISASSOCIATE.request

This primitive initiates the procedure for a DEV to disassociate from a piconet or pairnet. The semantics of this primitive are as follows:

```
MLME-DISASSOCIATE.request       (
                                DevId,
                                DEVAddress,
                                Timeout
                                )
```

The primitive parameters are defined in Table 5-12.

5.3.7.3 MLME-DISASSOCIATE.confirm

This primitive is used to confirm the result of the disassociation procedure. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-12.

5.3.7.4 MLME-DISASSOCIATE.indication

This primitive is used to indicate that either this DEV or another DEV has been disassociated from the piconet or pairnet. The semantics of this primitive are as follows:

```
MLME-DISASSOCIATE.indication       (
                                   DevId,
                                   DevAddress,
                                   ResultCode,
                                   ReasonCode
                                   )
```

The primitive parameters are defined in Table 5-12.

5.3.8 Security management

5.3.8.1 Overview

These primitives are used to initialize, update, or delete the security information as a result of a membership or key change process, as described in 8.3.5, or as the result of a security event, as described in 8.3.6 and 8.3.8. Primitives are also provided to transfer security messages. These primitives are suitable for use in an authentication process.

The parameters used for the MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitives are defined in Table 5-13.

Table 5-13—MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitive parameters

Name	Type	Valid range	Description
SECID	2 octets	As defined in 6.2.10.2	The identifier for the key.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.6 for pairnet, except for the BcstID, the McstID or the UnassocID	Either the PNCID or PRCID, if this key is for the DEV's PNC or PRC personality, or the DEV's DEVID.

Table 5-13—MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitive parameters (*continued*)

Name	Type	Valid range	Description
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.6 for pairnet, except for the BcstID, the McstID or the UnassocID	The DEVID of the target DEV for this relationship.
MembershipStatus	Enumeration	MEMBER, NON-MEMBER	Indicates the membership status for the provided SECID. If NON-MEMBER, KeyInfo is zero length.
KeyOriginator	Boolean	TRUE, FALSE	Indicates if the DEV is the key originator for this relationship. This is always true when the OrigId is the PNCID or PRCID.
KeyInfo	Octet string	Any valid symmetric key for the symmetric key security operations, as defined in 9.3 for piconet and as defined in 10.3 for pairnet	The key used for protecting frames between this DEV and the TrgtId DEV.
SrcID	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.6 for pairnet, except for the BcstID, the McstID or the UnassocID	The DEVID of the DEV that is the source of a security error.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_ASSOCIATED, TARGET_UNAVAILABLE, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, BAD_TIME_TOKEN, INVALID_SEC_VALUE, BAD_SFC, OTHER	The reason for a security error.

The parameters used for the MLME-SECURITY-MESSAGE primitive are defined in Table 5-14.

Table 5-14—MLME-SECURITY-MESSAGE primitive parameters

Name	Type	Valid range	Description
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.6 for pairnet	Specifies the DEVID of the target of the MLME request.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.6 for pairnet	Specifies the DEVID of the originator of the MLME request.

Table 5-14—MLME-SECURITY-MESSAGE primitive parameters (*continued*)

Name	Type	Valid range	Description
UniqueId	Octet string	Any valid OUI or CID, as defined in 6.4.8	A unique identifier for the entity that defines the format of the security information, as described in 6.4.8.
SecurityInformation	Octet string	Any valid octet string	Security information that will be passed from one DEV to another peer DEV in the piconet or pairnet.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_UNAVAILABLE, OTHER	The reason for a security error.

5.3.8.2 MLME-MEMBERSHIP-UPDATE.request

This primitive initiates the membership update procedure by either installing or removing a management key. Data keys are generated by the key originator of the relationship and are exchanged between peer MLMEs, as described in 8.3.5. The semantics of this primitive are as follows:

```
MLME-MEMBERSHIP-UPDATE.request      (
                                    OrigId,
                                    TrgtId,
                                    MembershipStatus,
                                    SECID,
                                    KeyOriginator,
                                    KeyInfo,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-13.

5.3.8.3 MLME-MEMBERSHIP-UPDATE.confirm

This primitive indicates the result of a membership update request. The semantics of this primitive are as follows:

```
MLME-MEMBERSHIP-UPDATE.confirm      (
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-13.

5.3.8.4 MLME-SECURITY-ERROR.indication

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

```
MLME-SECURITY-ERROR.indication       (
                                     SrcID,
                                     ReasonCode
                                     )
```

The primitive parameters are defined in Table 5-13.

5.3.8.5 MLME-SECURITY-MESSAGE.request

This primitive initiates the sending of a Security Message command, as described in 6.5.10.1, to the target DEV in the piconet or pairnet. The semantics of this primitive are as follows:

```
MLME-SECURITY-MESSAGE.request        (
                                     TrgtId,
                                     UniqueId,
                                     Timeout
                                     )
```

The primitive parameters are defined in Table 5-14.

5.3.8.6 MLME-SECURITY-MESSAGE.confirm

This primitive indicates the result of a request to send a Security Message command. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-14.

5.3.8.7 MLME-SECURITY-MESSAGE.indication

This primitive reports the reception of a Security Message command, as described in 6.5.10.1, from a DEV in the piconet or pairnet. The semantics of this primitive are as follows:

```
MLME-SECURITY-MESSAGE.indication     (
                                     OrigId,
                                     UniqueId,
                                     SecurityInformation
                                     )
```

The primitive parameters are defined in Table 5-14.

5.3.9 PNC handover

5.3.9.1 Overview

This function is not applicable for pairnets.

These primitives are used as part of the handover process, as described in 7.2.4 and 7.2.7, where the current PNC's responsibilities are transferred to another DEV in the piconet. The parameters used for these primitives are defined in Table 5-15.

Table 5-15—MLME-PNC-HANDOVER and MLME-NEW-PNC primitive parameters

Name	Type	Valid range	Description
HandoverStatus	Enumeration	STARTED, IMPLICIT, PRELIMINARY, CANCELLED	Indicates if the PNC is beginning or canceling a handover to the DEV.
NewPncDevAddress	MAC address	Any valid individual MAC address	The DEV address of the DEV assuming responsibilities as PNC.
SecMode	Enumeration	MODE_0, MODE_1	The security mode of the piconet, as defined in 6.3.1.

5.3.9.2 MLME-PNC-HANDOVER.indication

This primitive indicates the reception of a directed PNC Handover Request command, 6.5.4.2. The semantics of this primitive are as follows:

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

The primitive parameter is defined in Table 5-15.

5.3.9.3 MLME-PNC-HANDOVER.response

This primitive indicates that DME of the device targeted for PNC handover is ready to act as PNC. The semantics of this primitive are as follows:

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

There are no primitive parameters.

5.3.9.4 MLME-NEW-PNC.indication

This primitive indicates that the role of the PNC has been assumed by a different DEV in the piconet. The semantics of this primitive are as follows:

```
MLME-NEW-PNC.indication             (
                                    NewPncDevAddress,
                                    SecMode
                                    )
```

The primitive parameters are defined in Table 5-15.

5.3.10 Requesting DEV information from the PNC

5.3.10.1 Overview

This function is not applicable for pairnets.

This mechanism supports the ability for a DEV to request information from the PNC about either a specific DEV or all of the DEVs in the piconet, as described in 7.13.2. The parameters used for these primitives are defined in Table 5-16.

Table 5-16—MLME-DEV-INFO primitive parameters

Name	Type	Valid range	Description
QueriedDevId	Integer	Any valid DEVID, as defined in 6.2.5, except for the McstID or the UnassocID	The DEVID of the DEV for which information is being requested from the PNC. If it is the BcstID, the request is for information for all of the DEVs in the piconet.
NumDEVInfo	Integer	1–*mMaxNumValidDEVs*	Number of entries in the DEVInfoSet.
DEVInfoSet	A set of DEV Info fields, as defined in 6.5.5.3	A set containing 1 to *mMaxNumValidDEVs* instances of fixed-length DEV Info fields	The DEVInfoSet is returned to indicate the results of a PNC Information Request command.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_NOT_ASSOCIATED, OTHER	Indicates the reason for a ResultCode of FAILURE.

5.3.10.2 MLME-DEV-INFO.request

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

```
MLME-DEV-INFO.request            (
                                 QueriedDevId,
                                 Timeout
                                 )
```

The primitive parameters are defined in Table 5-16.

5.3.10.3 MLME-DEV-INFO.confirm

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

```
MLME-DEV-INFO.confirm          (
                               NumDEVInfo,
                               DEVInfoSet,
                               ResultCode,
                               ReasonCode
                               )
```

The primitive parameters are defined in Table 5-16.

5.3.10.4 MLME-DEV-INFO.indication

This primitive indicates the reception of DEV information that was not requested using the MLME-DEV-INFO.request primitive. The semantics of this primitive are as follows:

```
MLME-DEV-INFO.indication       (
                               NumDEVInfo,
                               DEVInfoSet
                               )
```

The primitive parameters are defined in Table 5-16.

5.3.11 Security information retrieval

5.3.11.1 Overview

These primitives are used to request security information about other DEVs in the piconet, as described in 8.4.2. The parameters used for the MLME-SECURITY-INFO primitives are defined in Table 5-17.

Table 5-17—MLME-SECURITY-INFO primitive parameters

Name	Type	Valid range	Description
QueriedDevId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.4 for pairnet, except for the McstID or the UnassocID	The DEVID of the DEV for which information is being requested. If it is set to the BcstID, then the information is being requested for all DEVs.
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.4 for pairnet	The DEVID of the DEV for which the security information request is intended.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5 for piconet and as defined in 6.2.4 for pairnet	Specifies the DEVID of the DEV that initiated the MLME request.
NumSecurityRecords	Integer	0–65535	Number of entries in the SecurityRecordSet.

Table 5-17—MLME-SECURITY-INFO primitive parameters (*continued*)

Name	Type	Valid range	Description
SecurityRecordSet	A set of Security Record fields, as defined in 6.5.5.5	A set containing 0 or more instances of variable-length Security Record field. The maximum number of instances depends on the size of the records, *pMaxFrameBodySize* and the length of the secure command security fields, as defined in 6.3.3.3 for piconet and as defined in 6.3.4 for pairnet.	The SecurityRecordSet is returned to indicate the results of a Security Information Request command.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_NOT_ ASSOCIATED, OTHER	Indicates the reason for a ResultCode of FAILURE.

5.3.11.2 MLME-SECURITY-INFO.request

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

```
MLME-SECURITY-INFO.request          (
                                    TrgtId,
                                    QueriedDevId,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-17.

5.3.11.3 MLME-SECURITY-INFO.confirm

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

```
MLME-SECURITY-INFO.confirm          (
                                    TrgtId,
                                    NumSecurityRecords,
                                    SecurityRecordSet,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-17.

5.3.11.4 MLME-SECURITY-INFO.indication

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

```
MLME-SECURITY-INFO.indication       (
                                    OrigId,
                                    QueriedDevId
                                    )
```

The primitive parameters are defined in Table 5-17.

5.3.11.5 MLME-SECURITY-INFO.response

This primitive is used by a DEV to respond to an MLME-SECURITY-INFO.indication. The semantics of the primitive are as follows:

```
MLME-SECURITY-INFO.response         (
                                    OrigId,
                                    NumSecurityRecords,
                                    SecurityRecordSet
                                    )
```

The primitive parameters are defined in Table 5-17.

5.3.12 Application specific data management

5.3.12.1 Overview

These primitives are used to request that the PNC add, modify, or remove application-specific (AS) data in the beacon and to report the reception of application specific data in a beacon, as described in 7.19. The parameters used for these primitives are defined in Table 5-18. If the RequestType is “ADD,” then the MAC ignores the AsIeIndex.

Table 5-18—MLME-APPLICATION-SPECIFIC primitive parameters

Name	Type	Valid range	Description
RequestType	Enumeration	ADD, MODIFY, REMOVE	If ADD, a request that a new AS IE be placed in the beacon. If MODIFY, a request to change the contents of an existing AS IE. If REMOVE, a request that a previously added AS IE be removed from the beacon.
AsIeIndex	Octet	0–255	An ID assigned by the PNC to each AS IE successfully added to the beacon. If RequestType is MODIFY or REMOVE, it specifies the AS IE to modify or remove.
RequestID	Octet	0–255	A unique number assigned by the requesting DEV to identify the request.

Table 5-18—MLME-APPLICATION-SPECIFIC primitive parameters (*continued*)

Name	Type	Valid range	Description
UniqueId	Octet string	Any valid OUI or CID, as defined in 6.4.8	If RequestType is ADD or MODIFY, the unique identifier of the entity that defines the format of ApplicationData.
ApplicationData	Octet string	Octet string	If the RequestType is ADD or MODIFY, the application specific data to add to the beacon, as described in 7.19.
NumApplicationSpecificData	Integer	0–255	Indicates the number of ApplicationSpecificData in the piconet beacon.
ApplicationSpecificDataSet	Set of application specific data, as defined in Table 5-7	A set containing zero or more instances of an ApplicationSpecificData	The ApplicationSpecificDataSet is returned to indicate the ApplicationSpecificData in the piconet beacon.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, UNKNOWN_ID, PNC_DENIED, OTHER	Indicates the reason for a ResultCode of FAILURE.

5.3.12.2 MLME-APPLICATION-SPECIFIC.request

This primitive is used to request to add, modify, or remove application specific data in the piconet beacon. The semantics of this primitive are as follows:

```
MLME-APPLICATION-SPECIFIC.request   (
                                     RequestType,
                                     RequestID,
                                     AsIeIndex,
                                     UniqueId,
                                     ApplicationData,
                                     Timeout
                                     )
```

The primitive parameters are defined in Table 5-18.

5.3.12.3 MLME-APPLICATION-SPECIFIC.confirm

This primitive is used to report the result of the request to add, modify, or remove application specific data in the beacon. The semantics of this primitive are as follows:

```
MLME-APPLICATION-SPECIFIC.confirm   (
                                    AsIeIndex,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-18.

5.3.12.4 MLME-APPLICATION-SPECIFIC.indication

This primitive is used to indicate that application specific data was in a successfully received beacon. The MLME passes all of the AS IEs found in every beacon to the DME. The semantics of this primitive are as follows:

```
MLME-APPLICATION-SPECIFIC.indication(
                                    NumApplicationSpecificData,
                                    ApplicationSpecificDataSet
                                    )
```

The primitive parameters are defined in Table 5-18.

5.3.13 Piconet services management

5.3.13.1 Overview

These primitives are used to transfer information regarding the services offered by DEVs in a piconet, as described in 7.4.3. The parameters used for these primitives are defined in Table 5-19.

Table 5-19—MLME-ANNOUNCE-SERVICE and MLME-PICONET-SERVICES primitive parameters

Name	Type	Valid range	Description
UniqueId	Octet string	Any valid OUI or CID, as defined in 6.4.8	The unique identifier for the entity that defines the format of the ServiceData.
ServiceData	Octet string	Octet string	The ServiceData, as described in 7.4.3.
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID of the DEV to request service information.
NumberOfServices	Integer	0–255	The number of PiconetServices in the PiconetServicesSet.
PiconetServicesSet	Set of piconet services, as defined in Table 5-20	A set containing zero or more instances of a PiconetService	The PiconetServicesSet is returned to indicate the PiconetServices offered.

Table 5-19—MLME-ANNOUNCE-SERVICE and MLME-PICONET-SERVICES primitive parameters (*continued*)

Name	Type	Valid range	Description
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_NOT_ ASSOCIATED, OTHER	Indicates the reason for a ResultCode of FAILURE.

In Table 5-19, a PiconetServicesSet is a set of PiconetServices. Each PiconetService consists of the elements shown in Table 5-20.

Table 5-20—Elements of PiconetService

Name	Type	Valid range	Description
DevId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID of the DEV offering the service.
UniqueId	Octet string	Any valid OUI or CID, as defined in 6.4.8	The unique identifier of the entity that defines the format of the ServiceData.
ServiceData	Octet string	Octet string	The ServiceData, as described in 7.4.3.

5.3.13.2 MLME-ANNOUNCE-SERVICE.request

This primitive is used to request the announcement of the availability of a service offered by this DEV. The semantics of this primitive are as follows:

```
MLME-ANNOUNCE-SERVICE.request       (
                                    UniqueId,
                                    ServiceData,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-19.

5.3.13.3 MLME-ANNOUNCE-SERVICE.confirm

This primitive is used to report the result of the request to announce the availability of a service offered by this DEV. The semantics of this primitive are as follows:

```
MLME-ANNOUNCE-SERVICE.confirm       (
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-19.

5.3.13.4 MLME-PICONET-SERVICES.request

This primitive is used to request piconet service information from a DEV in the piconet. If the target DEV is the PNC, then it is a request for all piconet service information previously announced by DEVs and stored in the PNC. The semantics of this primitive are as follows:

```
MLME-PICONET-SERVICES.request       (
                                    TrgtId,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-19.

5.3.13.5 MLME-PICONET-SERVICES.confirm

This primitive is used to report the result of the request for piconet service information from a DEV in the piconet. The semantics of this primitive are as follows:

```
MLME-PICONET-SERVICES.confirm       (
                                    NumberOfServices,
                                    PiconetServicesSet,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-19.

5.3.13.6 MLME-PICONET-SERVICES.indication

This primitive indicates the reception of piconet services information that was not requested with the MLME-PICONET-SERVICES.request primitive. The semantics of this primitive are as follows:

```
MLME-PICONET-SERVICES.indication    (
                                    NumberOfServices,
                                    PiconetServicesSet
                                    )
```

The primitive parameters are defined in Table 5-19.

5.3.14 Stream management

5.3.14.1 Overview

This function is not applicable for pairnets.

This mechanism supports the creation, modification, and termination of isochronous streams, as described in 7.7.2. The parameters used for the MLME-CREATE-STREAM, MLME-MODIFY-STREAM, and MLME-TERMINATE-STREAM primitives are defined in Table 5-21.

Table 5-21—MLME-CREATE-STREAM, MLME-MODIFY-STREAM, and MLME-TERMINATE-STREAM primitive parameters

Name	Type	Valid range	Description
RequestID	Integer	0–255	A unique value created by the originating DME to match the request primitive with the response primitive it receives from the PNC MLME.
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID of the target DEV for an isochronous stream.
SourceDataRate	Integer	1–(2^{64} – 1)	The minimum required data rate of the stream source at the MAC SAP in bits per second, as described in F.1.5.
DesiredDataRate	Integer	1–(2^{64} – 1)	The desired data rate of the stream source at the MAC SAP in bits per second, as described in F.1.5.
MaxRetries	Integer	0–255	Specifies the maximum number of retries to attempt per transmitted frame.
AckRequested	Boolean	TRUE, FALSE	Indicates if acknowledgments will be used for the stream.
MaxTransmitDelay	Duration	1–(2^{64} – 1)	Maximum allowed delay in microseconds for transmitting a MAC service data unit (MSDU) once it is presented to MAC SAP.
UserPriority	Integer	As defined in Table B.1	User priority of the stream, as described in Table B.1.
TypicalDataFrameSize	Integer	0–*pMaxFrameBodySize*	The typical size in octets of an MSDU to be presented to the MAC SAP.
SecMode	Boolean	TRUE, FALSE	Indicates if security is to be applied to the stream.
StreamGrpID	Integer	0–255	A nonzero value specifies that channel time associated with this stream may be shared by other streams associated with the same stream group ID, as described in 7.7.2.
StreamIndex	Integer	Any valid stream index, as defined in 6.2.8	The index of a stream created or the index of a stream to modify or terminate.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID of the source DEV for an isochronous stream.
AvailableDataRate	Integer	1–(2^{64} – 1)	If the request was successful, the data rate available for the stream. If the request was unsuccessful, the data rate that the PNC would have been able to allocate.

Table 5-21—MLME-CREATE-STREAM, MLME-MODIFY-STREAM, and MLME-TERMINATE-STREAM primitive parameters (*continued*)

Name	Type	Valid range	Description
ReliabilityExponent	Integer	0–31	The negative power of 10 that is the maximum frame error ratio (FER) desired including retries and frames lost due to MaxTransmitDelay. For example, a value of 4 corresponds to an FER < 10^{-4}. If the value of the parameter is zero, then the parameter is ignored.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, TARGET_UNAVAILABLE, RESOURCES_UNAVAILABLE, TERMINATED_BY_PNC, TERMINATED_BY_DEST, TRANSMIT_DELAY_UNSUPPORTED, PNC_BUSY DEV_IN_PS_MODE NOT_ASSOCIATED, UNKNOWN_STREAM, OTHER	The reason for a ResultCode of FAILURE.

5.3.14.2 MLME-CREATE-STREAM.request

This primitive is used to request the creation of an isochronous stream. The semantics of this primitive are as follows:

```
MLME-CREATE-STREAM.request          (
                                    RequestID,
                                    TrgtId,
                                    SourceDataRate,
                                    DesiredDataRate,
                                    MaxRetries,
                                    AckRequested,
                                    MaxTransmitDelay,
                                    UserPriority,
                                    TypicalDataFrameSize,
                                    SecMode,
                                    StreamGrpID,
                                    ReliabilityExponent,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-21.

5.3.14.3 MLME-CREATE-STREAM.confirm

This primitive is used to report the result of the request for the creation of an isochronous stream. The semantics of this primitive are as follows:

```
MLME-CREATE-STREAM.confirm          (
                                    RequestID,
                                    StreamIndex,
                                    AvailableDataRate,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-21.

5.3.14.4 MLME-CREATE-STREAM.indication

This primitive is used to inform a DEV that it is the target of an isochronous stream sourced from another DEV in the piconet. The semantics of this primitive are as follows:

```
MLME-CREATE-STREAM.indication       (
                                    StreamIndex,
                                    OrigId
                                    )
```

The primitive parameters are defined in Table 5-21.

5.3.14.5 MLME-MODIFY-STREAM.request

This primitive is used to request a modification to the parameters defining an existing isochronous stream. The semantics of this primitive are as follows:

```
MLME-MODIFY-STREAM.request          (
                                    StreamIndex,
                                    SourceDataRate,
                                    DesiredDataRate,
                                    MaxRetries,
                                    AckRequested,
                                    MaxTransmitDelay,
                                    UserPriority,
                                    TypicalDataFrameSize,
                                    SecMode,
                                    StreamGrpID,
                                    ReliabilityExponent,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-21.

5.3.14.6 MLME-MODIFY-STREAM.confirm

This primitive is used to report the result of the request to modify the parameters defining an existing isochronous stream. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-21.

5.3.14.7 MLME-TERMINATE-STREAM.request

This primitive is used to request the termination of an existing isochronous stream. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-21.

5.3.14.8 MLME-TERMINATE-STREAM.confirm

This primitive is used to report the result of the request to terminate an existing isochronous stream. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-21.

5.3.14.9 MLME-TERMINATE-STREAM.indication

This primitive is used to report to a DEV other than the DEV terminating the stream that an isochronous stream has been terminated. The semantics of this primitive are as follows:

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

The primitive parameter is defined in Table 5-21.

5.3.15 Piconet parameter management

5.3.15.1 Overview

These primitives allow a DEV acting as PNC to change the beacon source identifier (BSID) of the piconet and allow all DEVs to be informed of changes to piconet characteristics, as defined in 7.14 and 7.15. The parameters used for these primitives are defined in Table 5-22.

Table 5-22—MLME-BSID-CHANGE and MLME-PICONET-PARM-CHANGE primitives parameters

Name	Type	Valid range	Description
IndicationChangeType	Enumeration	BSID, PNID, CHANNEL	Indicates the parameter of the piconet that was changed.
Bsid	Octet string	As defined in 6.4.3	If a request or if the IndicationChangeType is BSID, a new BSID for the piconet, as described in 6.4.3.
Pnid	Integer	0–65535	If the IndicationChangeType is PNID, the new ID for the piconet.
ChannelIndex	Integer	0–255	If the IndicationChangeType is CHANNEL, the new PHY-dependent channel number on which the piconet is operating.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_A_PNC, OTHER	The reason for a ResultCode of FAILURE.

5.3.15.2 MLME-BSID-CHANGE.request

This primitive initiates changing the BSID. The semantics of this primitive are as follows:

```
MLME-BSID-CHANGE.request
                                        (
                                        Bsid
                                        )
```

The primitive parameter is defined in Table 5-22.

5.3.15.3 MLME-BSID-CHANGE.confirm

This primitive reports the result of a request to change the BSID. The semantics of this primitive are as follows:

```
MLME-BSID-CHANGE.confirm
                                        (
                                        ResultCode,
                                        ReasonCode
                                        )
```

The primitive parameters are defined in Table 5-22.

5.3.15.4 MLME-PICONET-PARM-CHANGE.indication

This primitive reports to a DEV a change to a piconet characteristic that was not requested by the DME. The semantics of this primitive are as follows:

```
MLME-PICONET-PARM-CHANGE.indication
                                          (
                                          IndicationChangeType,
                                          Bsid,
                                          Pnid,
                                          ChannelIndex
                                          )
```

The primitive parameters are defined in Table 5-22.

5.3.16 Power management

5.3.16.1 Overview

This mechanism supports the process of establishment and maintenance of power management (PM) modes of a DEV, as described in 7.17, and is applicable for piconets only. The parameters used for these primitives are defined in Table 5-23.

Table 5-23—MLME-PS-SET-INFORMATION, MLME-SPS-CONFIGURE, MLME-PM-MODE-CHANGE, and MLME-MONITOR-PM-MODE primitive parameters

Name	Type	Valid range	Description
PMMode	Enumeration	ACTIVE, APS, SPS	The PM mode requested by the DEV, as described in 6.5.9.6.
MaxSupportedPSSets	Integer	As defined in 6.5.9.3 and 7.17	The total number of PS sets currently supported by the PNC of this piconet.
NumCurrentPSSets	Integer	As defined in 6.5.9.3 and 7.17	Indicates the number of currently active PS sets in the piconet.
PSSetStructureSet	Set of PSSetStructures	As defined in Table 5-24	The PSSetStructureSet returns the information about the PS sets currently active in the PNC.
SetOperationType	Enumeration	CREATE, JOIN, LEAVE	The requested SPS set operation.
SPSSetIndex	Integer	As defined in 6.5.9.4	If the SetOperationType is JOIN or LEAVE, the SPS set index of the SPS set to join or leave. If the SetOperationType is CREATE, the SPS set index created by the PNC for the new SPS set.
DesiredWakeInterval	Integer	$0–(2^{32} – 1)$	If the SetOperationType is CREATE, the period in microseconds at which the requesting DEV would desire to transition from the SLEEP state to the AWAKE state.

Table 5-23—MLME-PS-SET-INFORMATION, MLME-SPS-CONFIGURE, MLME-PM-MODE-CHANGE, and MLME-MONITOR-PM-MODE primitive parameters (*continued*)

Name	Type	Valid range	Description
WakeInterval	Integer	0–(2^{32} – 1)	The time period in microseconds at which DEVs in a power save set transition from the SLEEP state to the AWAKE state.
PMActiveEvent	Enumeration	DATA_PENDING, MAX_SLEEP	If PMMode is ACTIVE, an event that causes the MLME to change the PM mode of operation to ACTIVE.
MonitorOperationType	Enumeration	ENABLE, DISABLE	The PM monitor operation requested.
TrgtId	Integer	Any valid DEVID, 6.2.5	Specifies the DEVID of the target DEV for a PM monitor operation.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_NOT_ ASSOCIATED, UNKNOWN_SET_INDEX, PNC_DENIED, PNC_BUSY, OTHER	The reason for a ResultCode of FAILURE.

In Table 5-23, PSSetStructureSet is a set of PSSetStructures. Each PSSetStructure consists of the elements shown in Table 5-24.

In Table 5-24, MemberSet is a set of DEVIDs indicating the DEVIDs that are members of the PS set.

Table 5-24—Elements of PSSetStructure

Name	Type	Valid range	Description
PSSetIndex	Integer	As defined in 6.5.9.3	The identifier for the PS set.
WakeInterval	Integer	0–(2^{32} – 1)	The period in microseconds at which DEVs in this power save set transition from the SLEEP state to the AWAKE state. The MAC divides this number by the superframe duration and rounds down to the nearest power of two.
NumMembers	Integer	0–255	The number of DEVs in this power save set.
MemberSet	Set of DEVIDs	Any valid DEVID, as defined in 6.2.5	The DEVIDs of the DEVs in this power save set.

5.3.16.2 MLME-PS-SET-INFORMATION.request

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

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

The primitive parameter is defined in Figure 5-23.

5.3.16.3 MLME-PS-SET-INFORMATION.confirm

This primitive reports the result of the request to obtain the PS set information from the PNC. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-23.

5.3.16.4 MLME-SPS-CONFIGURE.request

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

```
MLME-SPS-CONFIGURE.request
                                     (
                                     SetOperationType,
                                     SPSSetIndex,
                                     DesiredWakeInterval,
                                     Timeout
                                     )
```

The primitive parameters are defined in Table 5-23.

5.3.16.5 MLME-SPS-CONFIGURE.confirm

This primitive reports the result of a request to change the current SPS set information. The semantics of this primitive are as follows:

```
MLME-SPS-CONFIGURE.confirm
                                        (
                                        SetOperationType,
                                        SPSSetIndex,
                                        WakeInterval,
                                        ResultCode,
                                        ReasonCode
                                        )
```

The primitive parameters are defined in Table 5-23.

5.3.16.6 MLME-PM-MODE-CHANGE.request

This primitive requests a change to the DEV's PM mode of operation. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-23.

5.3.16.7 MLME-PM-MODE-CHANGE.confirm

This primitive reports the result of the request to change the DEV's PM mode of operation. The semantics of this primitive are as follows:

```
MLME-PM-MODE-CHANGE.confirm             (
                                        PMMode,
                                        ResultCode,
                                        ReasonCode
                                        )
```

The primitive parameters are defined in Table 5-23.

5.3.16.8 MLME-PM-MODE-CHANGE.indication

This primitive is used to report a change to the DEVs PM mode of operation that was not requested with an MLME-PM-MODE-CHANGE.request but rather was initiated by the MAC. One reason for the MAC to change PM modes is in response to a CTA allocated with the DEV as the destination while the DEV is in a power save mode, as described in 7.17.3.3. This primitive reports only changes in the DEV's PM mode.

Changes in the PM mode of other DEVs in the piconet are reported with the MLME-MONITOR-PM-MODE.indication primitive. The semantics of this primitive are as follows:

```
MLME-PM-MODE-CHANGE.indication
                                    (
                                    PMMode,
                                    PMActiveEvent
                                    )
```

The primitive parameter is defined in Table 5-23.

5.3.16.9 MLME-MONITOR-PM-MODE.request

This primitive requests that the MLME enable or disable the monitoring of the PM mode for a DEV in the piconet. The semantics of this primitive are as follows:

```
MLME-MONITOR-PM-MODE.request        (
                                    MonitorOperationType,
                                    TrgtId
                                    )
```

The primitive parameters are defined in Table 5-23.

5.3.16.10 MLME-MONITOR-PM-MODE.confirm

This primitive reports the result of a request that the MLME enable or disable the monitoring of the PM mode for a DEV in the piconet. The semantics of this primitive are as follows:

```
MLME-MONITOR-PM-MODE.confirm        (
                                    MonitorOperationType,
                                    TrgtId,
                                    PMMode,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-23.

5.3.16.11 MLME-MONITOR-PM-MODE.indication

This primitive reports a change in the PM mode of a DEV in the piconet, for which PM monitoring was previously enabled using an MLME-MONITOR-PM-MODE.request. This primitive only reports changes to the PM modes of other DEVs in the piconet. Changes to the DEV's PM mode are indicated with the MLME-PM-MODE-CHANGE.indication primitive. The semantics of this primitive are as follows:

```
MLME-MONITOR-PM-MODE.indication     (
                                    TrgtId,
                                    PMMode
                                    )
```

The primitive parameters are defined in Table 5-23.

5.3.17 Multicast operations

5.3.17.1 Overview

These primitives support multicast operations and thus are not applicable for pairnets that do not use multicast. The MLME-MULTICAST-CONFIGURATION primitives are used for multicast traffic that uses a Group Address and a GrpID assigned by the PNC, as described in 7.7.4. The parameters used for these primitives are defined in Table 5-25.

Table 5-25—MLME-MULTICAST-CONFIGURATION primitive parameters

Name	Type	Valid range	Description
RequestType	Enumeration	JOIN, LEAVE	Indicates if this is a request to join a multicast group or leave a multicast group.
GroupAddress	MAC address	Any valid group address, as defined in 6.4.25	A group address representing a specific multicast group.
GrpID	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID assigned by the PNC to a multicast group associated with a specific GroupAddress.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, PNC_DENIED, PNC_BUSY, OTHER	The reason for a ResultCode of FAILURE.

The MLME-MULTICAST-RX-SETUP.request primitive controls the reception of multicast traffic that uses the multicast identifier (McstID), as defined in 6.2.5. The parameters for this primitive are defined in Table 5-26.

Table 5-26—MLME-MULTICAST-RX-SETUP.request parameters

Name	Type	Valid range	Description
MulticastStatus	Enumeration	ENABLE, DISABLE, ALL, NONE	If ENABLE or DISABLE, indicates whether the MAC will pass multicast traffic defined by the stream index to the DME. If ALL, then all multicast traffic is passed to the DME. If NONE, then no multicast traffic will be passed to the DME. These restrictions apply only to frames received with the DestID set to the McstID.
SrcID	Integer	Any valid DEVID, as defined in 6.2.5	The DEVID of the source of a multicast stream.
StreamIndex	Integer	Any valid stream index, as defined in 6.2.8	The stream index of a multicast stream.

Table 5-26—MLME-MULTICAST-RX-SETUP.request parameters (*continued*)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_ASSOCIATED, UNKNOWN_STREAM, SOURCE_UNAVAILABLE, RESOURCES_UNAVAILABLE	Indicates the reason for a ResultCode of FAILURE.

5.3.17.2 MLME-MULTICAST-CONFIGURATION.request

This primitive is used by a DEV to request to join or leave a multicast group defined by a particular group address. The semantics of this primitive are as follows:

```
MLME-MULTICAST-CONFIGURATION.request(
                                    RequestType,
                                    GroupAddress,
                                    Timeout
                                    )
```

The primitive parameters are defined in Table 5-25.

5.3.17.3 MLME-MULTICAST-CONFIGURATION.confirm

This primitive is used to report the result of a request by a DEV to join or leave a multicast group defined by a particular group address. The semantics of this primitive are as follows:

```
MLME-MULTICAST-CONFIGURATION.confirm(
                                    GroupAddress,
                                    GrpID,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-25.

5.3.17.4 MLME-MULTICAST-RX-SETUP.request

This primitive allows the DME to control multicast reception for frames with the DestID set to the McstID, as defined in 6.2.5, and to allow filtering for a particular stream index associated with the McstID. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-26.

5.3.17.5 MLME-MULTICAST-RX-SETUP.confirm

This primitive indicates the result of configuring the MAC for filtering multicast traffic. The semantics of this primitive are as follows:

```
MLME-MULTICAST-RX-SETUP.confirm     (
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table 5-26.

5.3.18 Timing synchronization

5.3.18.1 Overview

This function is not applicable for pairnets.

These primitives support the synchronization of upper layer functions with the beacon timing in the piconet. The beginning of the data preamble is observed on the air by all DEVs within a piconet while the delay between the observation and the delivery of the indication is known within a MAC by design (and communicated to the application by implementation-dependent means). The parameters used for these primitives are defined in Table 5-27.

Table 5-27—MLME-BEACON-EVENT primitive parameters

Name	Type	Valid range	Description
BeaconNumber	Integer	0–65535	The beacon number of the beacon that was received.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_SUPPORTED, OTHER	The reason for a ResultCode of FAILURE.

5.3.18.2 MLME-BEACON-EVENT.request

This primitive requests activation of the MAC synchronization support facility. The semantics of this primitive are as follows:

```
MLME-BEACON-EVENT.request           ()
```

The primitive has no parameters.

5.3.18.3 MLME-BEACON-EVENT.confirm

This primitive confirms the activation of the MAC synchronization support facility. If the MAC does not support synchronization or if it encounters some other error, the response will have a ResultCode of FAILURE and the ReasonCode set to indicate the type of error. The semantics of this primitive are as follows:

```
MLME-BEACON-EVENT.confirm          (
                                   ResultCode,
                                   ReasonCode
                                   )
```

The primitive parameters are defined in Table 5-27.

5.3.18.4 MLME-BEACON-EVENT.indication

This primitive indicates the beginning of a beacon's data preamble, whether the beacon is transmitted or correctly received by the MAC. The semantics of this primitive are as follows:

```
MLME-BEACON-EVENT.indication       (
                                   BeaconNumber
                                   )
```

The primitive parameters are defined in Table 5-27.

5.3.19 Transmit switched diversity (TSD)

5.3.19.1 Overview

The transmit switched diversity (TSD) capability applies only to mmWave DEVs.

These primitives are used to exchange the TSD capability between a source and a destination, as described in 15.8. The parameters used for these primitives are defined in Table 5-28.

Table 5-28—MLME-TXDIV primitive parameters

Name	Type	Valid range	Description
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the DEVID of the target DEV for exchanging TSD information.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5	The DEVID of the DEV that initiated the MLME request.
SupportedNumAntennas	Integer	0–15	The number of antennas supported by the DEV.
Timeout	Integer	0–65535	The time in milliseconds allowed for the primitive to complete.
TXDiversityThresholdType	Enumeration	LQI type, as defined in 6.4.12	Specifies the type of LQI measurement to use.
TXDiversityTheshold	Integer	Any valid LQI value, as defined in 6.4.34	Specifies the value of the threshold for antenna switching.
AntennaIndex	Integer	0–15	Specifies antenna index to be used.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	NOT_SUPPORTED, OTHER	Indicates the reason for a ResultCode of FAILURE.

5.3.19.2 MLME-TXDIV.request

This primitive is used to request to use TSD. The semantics of this primitive are as follows:

```
MLME-TXDIV.request                    (
                                      TrgtId,
                                      TXDiversityThresholdType,
                                      TXDiversityTheshold,
                                      Timeout
                                      )
```

The primitive parameters are defined in Table 5-28.

5.3.19.3 MLME-TXDIV.confirm

This primitive is used to report the result of the request for TSD. The semantics of this primitive are as follows:

```
MLME-TXDIV.confirm                    (
                                      AntennaIndex,
                                      ResultCode,
                                      ReasonCode
                                      )
```

The primitive parameters are defined in Table 5-28.

5.4 MAC management

5.4.1 General

The MAC PIB comprises the managed objects, attributes, actions, and notifications required to manage the MAC sublayer of a DEV. The MAC PIB is divided into two groups, PNC characteristics and DEV characteristics. In the Access column of the Table 5-29 and Table 5-31, read only indicates that the parameter is only allowed to be read by the DME while read/write indicates that the DME is able to change it using the MLME-SET.request primitive.

5.4.2 MAC PIB PNC and PRC group

The MAC PIB PNC group, Table 5-29, describes both the DEV’s PNC capabilities as well as the characteristics of the current piconet.

Table 5-29—MAC PIB PNC group parameters

Managed object	Octets	Definition	Access
macCapEndTime	2	The end time of the CAP interval in the superframes, 7.8.	Read only
macSuperframeDuration	2	Duration of the superframe.	Read only
macPncCapable	1 bit	1 if the DEV has the capability to become the PNC, 0 otherwise.	Read only
macPncDesMode	1 bit	1 if it is desired that the DEV be the PNC.	Read/write

Table 5-29—MAC PIB PNC group parameters (*continued*)

Managed object	Octets	Definition	Access
macMaxPsSets	1	The maximum number of PS sets supported by the PNC.	Read only
macBsid	6–32	Identifies the piconet.	Read only
macMaxAssociatedDevs	1	As defined in 6.4.12.	Read only
macMaxChannelTimeRequests	1	As defined in 6.4.12.	Read only
macSec	1 bit	Indicates if the DEV is capable of operating a secure piconet as the PNC.	Read only
macPncServicesBroadcast	1	0x00 = PNC sends information about its services 0x01 = PNC will not send information about its services	Read/write
macAllowedChannelSet	Variable	A set of channel indices, one for each channel that the MAC is allowed to use for scanning and starting piconets.	Read/write
macAssocVendorSpecificIe	Variable	A Vendor Defined IE, as defined in 6.4.24, that is sent in the Association Response command, as described in 6.5.2.3, when the DEV is acting as the PNC.	Read/write
macDesiredAtp	2	The ATP value to send in an Association Request command.	Read/write
macPnid	2	If associated with a piconet, the PNID of that piconet.	Read only
macCapData	1	Indicates if the initial setting of the CAP Data Allowed field in the beacon, as described in 6.3.1.1.	Read/write
macCapCommand	1	Indicates if the initial setting of the CAP Commands Allowed field in the beacon, as described in 6.3.1.1.	Read/write
macCapAssociation	1	Indicates if the initial setting of the CAP Data Allowed field in the beacon, as described in 6.3.1.1.	Read/write
macPiconetMaxTxPower	1	The maximum power allowed for transmission during certain times in the superframe as described in 6.3.1.1.	Read/write
macMctaUsed	1	The initial setting of the MCTA Used field, as described in 6.3.1.1.	Read/write
macNextPncCapable	1 bit	Indicates if the DEV is capable of participating in the Next PNC procedure, as described in 7.2.6. 0 = DEV is not capable. 1 = DEV is capable.	Read only
macCapStartTime	2	The time at which the CAP starts when quasi-omni beacons are used.	Read only

The MAC PIB PRC group, given in Table 5-30, describes the DEV's PRC capabilities as well as the characteristics of the current pairnet.

Table 5-30—MAC PIB PRC group parameters

Managed Object	Octets	Definition	Access
macSuperframeDuration	2	Duration of the superframe.	Read Only
macNumAssocSlots	1	Number of association slot.	Read Only
macDurAssocSlots	1	Duration of an association slot.	Read Only
macPrcCapable	1 bit	1 if the DEV has the capability to become the PRC, 0 otherwise.	Read Only
macPrcDesMode	1 bit	1 if it is desired that the DEV be the PRC, 0 otherwise. Provided only for PRC.	Read/Write
macSec	1 bit	1 if the DEV is capable of operating a secure pairnet as the PRC, 0 otherwise.	Read Only
macAllowedChannelSet	Variable	A set of channel indices, one for each channel that the MAC is allowed to use for scanning and starting pairnet.	Read/Write
macAssocVendorSpecificIe	Variable	A Vendor Defined IE, as defined in 6.4.24, that is sent in the Association Response command, as described in 6.5.2.3, when the PRDEV is acting as the PRC.	Read/Write
macAssocHigherLayerIe	Variable	A Higher Layer Protocol Information IE, as defined in 6.4.45, that is sent in the Beacon frame and association-related commands, as described in 7.5.3, when the PRDEV is acting as the PRC. Provided only for PRC.	Read/Write
macDesiredAtp	2	The ATP value to send in an Association Request command.	Read/Write
macNextDevId	1	The Next DEVID value to send in a Beacon frame.	Read/Write

5.4.3 MAC PIB characteristic group

The MAC PIB characteristics group, Table 5-31, contains information about the capabilities and characteristics of the DEV.

Table 5-31—MAC PIB characteristic group parameters

Managed object	Octets	Definition	Access
macDevAddress	6	The MAC address of the DEV.	Read only
macDevId	1	The ID of the DEV.	Read only

Table 5-31—MAC PIB characteristic group parameters (*continued*)

Managed object	Octets	Definition	Access
macPowerManagementMode	1	The current power management mode of the DEV. 0x00 = ACTIVE 0x01 = PSPS 0x02 = DSPS 0x03 = DSPS and PSPS 0x04 = APS	Read only
macPspsSupported	1	0x00 = DEV does not support PSPS mode. 0x01 = DEV supports PSPS mode.	Read only
macDspsSupported	1	0x00 = DEV does not support DSPS mode. 0x01 = DEV supports DSPS mode.	Read only
macApsSupported	1	0x00 = DEV does not support APS mode. 0x01 = DEV supports APS mode.	Read only
macMaxStreams	1	Maximum number of streams that the DEV is able to handle.	Read only
macPowerSource	1	0x00 = Battery power. 0x01 = Mains power.	Read/write
macSecurityOptionImplemented	1	0x00 = Mode 0. 0x01 = Mode 1.	Read only
macDevServicesBroadcast	1	0x00 = DEV sends information about its services. 0x01 = DEV will not send information about its services.	Read/write
macAlwaysAwake	1 bit	As defined in 6.4.12.	Read only
macListenToSource	1 bit	As defined in 6.4.12.	Read only
macListenToMulticast	1 bit	As defined in 6.4.12.	Read only
macCtaRelinquishCapable	1 bit	Indicates if the DEV is capable of using time relinquished in a CTA by another DEV. 0x00 = DEV does not support. 0x01 = DEV does support.	Read only
macDlyAckCapable	1 bit	Indicates if the DEV is capable of using Dly-ACK as the source. 0 = DEV does not support. 1 = DEV does support.	Read only
macImpAckCapable	1 bit	Indicates if the DEV is capable of using implied ACK (Imp-ACK) as the source. 0 = DEV does not support. 1 = DEV does support.	Read only
macStpCapable	1 bit	Indicates if the DEV is capable of using the stream timeout period (STP). 0 = DEV does not support. 1 = DEV does support.	Read only

The MAC PIB PRDEV characteristics group, given in Table 5-32, contains information about the capabilities and characteristics of the PRDEV.

Table 5-32—MAC PIB PRDEV characteristic group parameters

Managed Object	Octets	Definition	Access
macDevAddress	6	The MAC address of the PRDEV.	Read Only
macDevId	1	The ID of the PRDEV.	Read Only
macPowerManagementMode	1	The current power management mode of the PRDEV. 0x00 = ACTIVE From 0x01 to 04, ignored 0x05 = LLPS	Read Only
macLlpsSupported	1	0x00 = PRDEV does not support LLPS mode. 0x01 = PRDEV supports LLPS mode.	Read Only
macPowerSource	1	0x00 = Battery power. 0x01 = Mains power.	Read/Write
macSecurityOptionImplemented	1	0x00 = Mode 0. 0x01 = Mode 1.	Read Only
macAggregationCapable	1	0x00 = PRDEV does not support aggregation. 0x01 = PRDEV supports aggregation.	Read Only

5.5 MAC SAP

5.5.1 Overview

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

The IEEE 802.15.3 MAC SAP primitives are summarized in Table 5-33. The same primitives used for the HRCP PHY are also used for the THz PHY.

Table 5-33—Summary of MAC SAP primitives

Name	Request	Confirm	Indication	Response
MAC-ASYNC-DATA	5.5.2	5.5.3	5.5.4	—
MAC-ISOCH-DATA	5.5.5	5.5.6	5.5.7	—
MAC-HRCP-DATA	5.5.10	5.5.8	5.5.9	—
MAC-HRCP-MUL-DATA	5.5.10	5.5.11	5.5.12	—

The parameters used for these primitives are defined in Table 5-34.

The parameters InterlacedFieldIndication, VideoFrameNumber, HPosition, and VPosition are used by the AV PHY in the Video Subheader field, as described in 6.2.12.2.5. These fields are sent in the MAC header because it has lower error rate than the subframes carrying video data. For uncompressed video applications, video data that is in error can still be displayed, but only if the position of the pixel data is known.

Table 5-34—MAC-ISOCH-DATA, MAC-ASYNC-DATA, MAC-HRCP-DATA, and MAC-HRCP-MUL-DATA primitive parameters

Name	Type	Valid range	Description
RequestID	Integer	0–511	An identifier, which is used to correlate a request to a response. It is unique per stream index for outstanding isochronous requests. It is unique among all outstanding asynchronous requests.
StreamIndex	Integer	Any valid stream index, as defined in 6.2.8	The stream with which the data is associated.
TransmitTimeout	Duration	1–(2^{32} – 1)	Maximum allowed delay in microseconds from when the MSDU is been presented to the MAC SAP until the frame has finished transmission and the acknowledgment, if required is successfully received.
MaxRetries	Integer	0–255	Specifies the maximum number of retries to attempt per transmitted frame with a maximum value no greater than the MaxRetries value supplied in the original stream creation request.
SecMode	Boolean	TRUE, FALSE	Indicates if security is to be applied to the MSDU or if the MSDU was received securely.
UserPriority	Integer	As defined in Table B.1	User priority of the stream, as described in Table B.1.
AckRequested	Boolean	TRUE, FALSE	Indicates if the request requires an acknowledgment, as described in 7.12, of the MSDU at the MAC layer.
ConfirmRequested	Enumeration	NEVER, ALWAYS, ON_ERROR	Indicates when a confirm primitive is required for the request.
Data	Variable number of octets	Any octet string	MSDU portion of the primitive.
TransmitDelay	Duration	1–(2^{32} – 1)	Delay in microseconds from when the MSDU is presented to the MAC SAP until the frame has finished transmission and the acknowledgment, if required, has been successfully received. If the transmission fails due to timeout, this field shall be set to the TransmitTimeout value for this frame.
TrgtId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the target DEVID of an MSDU.
OrigId	Integer	Any valid DEVID, as defined in 6.2.5	Specifies the originator DEVID of an MSDU.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the MLME request.
ReasonCode	Enumeration	TRANSMIT_TIMEOUT, MAX_RETRIES, NOT_ASSOCIATED, OTHER	The reason for a ResultCode of FAILURE.

Table 5-34—MAC-ISOCH-DATA, MAC-ASYNC-DATA, MAC-HRCP-DATA, and MAC-HRCP-MUL-DATA primitive parameters (*continued*)

Name	Type	Valid range	Description
UncompressedVideo	Boolean	TRUE, FALSE	Set to TRUE if the Data parameter contains uncompressed video data.
UepAllowed	Boolean	TRUE, FALSE	Indicates if UEP is allowed for the transmission of the data.
ErrorsFreeData	Boolean	TRUE, FALSE	Indicates if the Data parameter that was received contains errors or is free from errors.
InterlacedFieldIndication	Enumeration	TOP_FIELD, BOTTOM_FIELD, NOT_INTERLACED	If UncompressedVideo is TRUE, the parameter indicates if the pixels belong to the top or bottom field of an interlaced image or that the image is not interlaced.
VideoFrameNumber	Integer	As defined in 6.2.12.2.5	If UncompressedVideo is TRUE, the parameter contains a sequential numbering of the video frames that are being transferred.
HPosition	Integer	As defined in 6.2.12.2.5	If UncompressedVideo is TRUE, the parameter contains the horizontal position of the first pixel in the MSDU.
VPosition	Integer	As defined in 6.2.12.2.5	If UncompressedVideo is TRUE, the parameter contains the vertical position of the first pixel in the MSDU.
LogicalChannel	Enumeration	CH0, CH1	LogicalChannel value is available for use by the Higher Layer Protocol user and is therefore out of scope of this specification. This parameter is valid only for MAC-HRCP-DATA primitives.
MCSIdentifier	Enumeration	Any valid MCS identifier, as defined in Table 13-6 for HRCP-SC PHY, Table 14-9 for THz-SC PHY, or Table 14-13 for THz-OOK PHY	MCS used in the transmitted PHY frame. Only applicable to HRCP-SC PHY, THz-SC PHY, and THz-OOK PHY.
ChIdentifier	Enumeration	Any valid combination of channels, as defined in Figure 13-1 for HRCP-SC PHY, or any channel defined in Table 14-1 for THz-SC PHY and THz-OOK PHY	The frequency channel used in the transmitted PHY frame. Only applicable to HRCP-SC PHY, THz-SC PHY, and THz-OOK PHY.
DataID	Enumeration	ETHERTYPE_PROTOCOL_DISCRIMINATION, OUI_CID, ETHERTYPE	The DataID specifies Data header type
DestinationAddress	MAC Address	Any valid individual MAC address	The destination address representing the address of data destination when the DataID is “ETHERTYPE_PROTOCOL_DISCRIMINATION”

Table 5-34—MAC-ISOCH-DATA, MAC-ASYNC-DATA, MAC-HRCP-DATA, and MAC-HRCP-MUL-DATA primitive parameters (*continued*)

Name	Type	Valid range	Description
SourceAddress	MAC Address	Any valid individual MAC address	The source address representing the address of data origin when the DataID is "ETHERTYPE_PROTOCOL_ DISCRIMINATION"
EtherType	Octet string	EtherType defined in IEEE Std 802	The EtherType specified when DataID is "ETHERTYPE_PROTOCOL_ DISCRIMINATION" or "ETHERTYPE"
OUI_CID	Octet string		The OUI or CID strings specified when DataID is "OUI_CID"
LogicalChannel	Enumeration	CH0, CH1	LogicalChannel value is available for use by the Higher Layer Protocol user and is therefore out of scope of this specification. This parameter is valid only for MAC-HRCP-DATA primitives.

5.5.2 MAC-ASYNC-DATA.request

This primitive is used to initiate the transfer of an MSDU from one MAC entity to another MAC entity or entities. All asynchronous data should use Multi-protocol Data frames to avoid interoperability problems in piconets where different FCSLs are being used. The semantics of this primitive are as follows:

```
MAC-ASYNC-DATA.request          (
                                RequestID,
                                TrgtId,
                                TransmitTimeout,
                                MaxRetries,
                                UserPriority,
                                AckRequested,
                                ConfirmRequested,
                                Data
                                )
```

The primitive parameters are defined in Table 5-34.

5.5.3 MAC-ASYNC-DATA.confirm

This primitive is used to report the result of a request to transfer an asynchronous MSDU from one MAC entity to another MAC entity or entities. This primitive is only generated if the ConfirmRequested parameter in the MAC-ASYNC-DATA.request with the same RequestID value is ALWAYS or is ON_ERROR and the ResultCode is FAILURE. The semantics of this primitive are as follows:

```
MAC-ASYNC-DATA.confirm          (
                                RequestID,
                                TransmitDelay,
                                ResultCode,
                                ReasonCode
                                )
```

The primitive parameters are defined in Table 5-34.

5.5.4 MAC-ASYNC-DATA.indication

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

```
MAC-ASYNC-DATA.indication          (
                                   TrgtId,
                                   OrigId,
                                   Data
                                   )
```

The primitive parameters are defined in Table 5-34.

5.5.5 MAC-ISOCH-DATA.request

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

```
MAC-ISOCH-DATA.request             (
                                   RequestID,
                                   StreamIndex,
                                   TransmitTimeout,
                                   MaxRetries,
                                   AckRequested,
                                   ConfirmRequested,
                                   Data,
                                   DataType,
                                   UepAllowed,
                                   InterlacedFieldIndication,
                                   VideoFrameNumber,
                                   HPosition,
                                   VPosition
                                   )
```

The primitive parameters are defined in Table 5-34.

5.5.6 MAC-ISOCH-DATA.confirm

This primitive is used to report the result of a request to transfer an isochronous MSDU from one MAC entity to another MAC entity or entities. This primitive is only generated if the ConfirmRequested parameter in the MAC-ISOCH-DATA.request with the same RequestID value is ALWAYS or is ON_ERROR and the ResultCode is FAILURE. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-34.

5.5.7 MAC-ISOCH-DATA.indication

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

```
MAC-ISOCH-DATA.indication          (
                                   TrgtId,
                                   OrigId,
                                   StreamIndex,
                                   Data,
                                   DataType,
                                   ErrorFreeData,
                                   InterlacedFieldIndication,
                                   VideoFrameNumber,
                                   HPosition,
                                   VPosition
                                   )
```

Data identified as video by the source DEV may be passed by the destination DEV to the higher layers even if the data was received in error. The error status of the data passed up by this primitive is indicated in the ErrorFreeData parameter.

The primitive parameters are defined in Table 5-34.

5.5.7 MAC-HRCP-DATA.request

This primitive is used to initiate the transfer of an MSDU from one MAC entity to another MAC entity or entities using the HRCP PHY or the THz PHY. The semantics of this primitive are as follows:

```
MAC-HRCP-DATA.request         (
                              RequestID,
                              LogicalChannel,
                              ACKRequested,
                              ConfirmRequested,
                              Data
                              )
```

The primitive parameters are defined in Table 5-34.

5.5.8 MAC-HRCP-DATA.confirm

This primitive is used to report the result of a request to transfer an MSDU from one MAC entity to another MAC entity or entities using the HRCP PHY or the THz PHY. This primitive is generated only if the ConfirmRequested parameter in the MAC-HRCP-DATA.request primitive with the same RequestID value is either:

— ALWAYS, or
— ON_ERROR and the ResultCode of the MAC-THZ-DATA.confirm primitive is FAILURE.

The MCSIdentifier and ChIdentifier parameters are used to indicate the current MCS identifier and frequency channel used in the transmitted frame to the upper layer. These two parameters are applicable only for the HRCP-SC PHY or the THz PHY. The semantics of this primitive are as follows:

```
MAC-HRCP-DATA.confirm        (
                             RequestID,
                             TransmitDelay,
                             MCSIdentifier,
                             ChIdentifier,
                             ResultCode,
                             ReasonCode
                             )
```

The primitive parameters are defined in Table 5-34.

5.5.9 MAC-HRCP-DATA.indication

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

```
MAC-HRCP-DATA.indication     (
                             Data
                             )
```

The primitive parameter is defined in Table 5-34.

5.5.10 MAC-HRCP-MUL-DATA.request

This primitive is used to initiate the transfer of a Multi-protocol MSDU from one MAC entity to another MAC entity or entities using the HRCP PHY or the THz PHY. The semantics of this primitive are as follows:

```
MAC-HRCP-MUL-DATA.request    (
                             RequestID,
                             StreamIndex,
                             LogicalChannel,
                             ACKRequested,
                             ConfirmRequested,
                             DataID,
                             DestinationAddress,
                             SourceAddress,
                             EtherType,
                             OUI_CID,
                             Data
                             )
```

The primitive parameters are defined in Table 5-34.

5.5.11 MAC-HRCP-MUL-DATA.confirm

This primitive is used to report the result of a request to transfer a Multi-protocol MSDU from one MAC entity to another MAC entity or entities using either the HRCP PHY or the THz PHY. This primitive is only generated if the ConfirmRequested parameter in the MAC-HRCP-MUL-DATA.request primitive with the same RequestID value is either:

— ALWAYS, or
— ON_ERROR and the ResultCode of the MAC-THZ-MUL-DATA.confirm primitive is FAILURE.

The semantics of this primitive are as follows:

```
MAC-HRCP-MUL-DATA.confirm   (
                            RequestID,
                            TransmitDelay,
                            ResultCode,
                            ReasonCode,
                            MCSIdentifier,
                            ChIdentifier
                            )
```

The primitive parameters are defined in Table 5-34.

5.5.12 MAC-HRCP-MUL-DATA.indication

This primitive is used to indicate the reception of a Multi-Protocol MSDU while using the HRCP PHY or the THz PHY. The semantics of this primitive are as follows:

```
MAC-HRCP-MUL-DATA.indication(
                            DataID,
                            DestinationAddress,
                            SourceAddress,
                            EtherType,
                            OUI_CID,
                            LogicalChannel,
                            Data
                            )
```

The primitive parameters are defined in Table 5-34.

6. MAC frame formats

6.1 Frame format conventions

The MAC frame format conventions follow 4.1.

The MAC in all DEVs shall be able to validate the error free reception of every frame received from the PHY using the FCS. Note that the PHY only passes frames to the MAC that have passed the HCS test.

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

All DEVs shall be assigned a DEV address, which is the 48-bit extended universal identifier (EUI-48), as defined by IEEE Std 802 and assigned per the IEEE Registration Authority.[17, 18] All MAC addresses shall be sent in the canonical form defined in IEEE Std 802.

6.2 General frame format

6.2.1 Overview

The MAC frame formats, illustrated in Figure 6-1 for piconet DEVs and in Figure 6-2 for PRDEVs, are each composed of a set of fields that occur in a fixed order in all frames.The figures are a representation of the MAC Header field and MAC Frame Body field. The HCS is not shown since this is calculated and verified by the PHY. The MAC frame shall be formatted as illustrated in Figure 6-1 for piconet DEVs and in Figure 6-2 for PRDEVs. For piconet DEVs, the maximum size of the MAC Frame Body field, *pMaxFrameBodySize*, is a PHY-dependent parameter that includes the frame payload and FCS fields, but not the PHY preamble, PHY header, MAC header, MAC subheader, or MAC header validation. For PRDEVs, the maximum size of the MAC Frame Body field, *pMaxFrameBodySize,* is a PHY-dependent parameter that includes the frame payload(s), MAC subheader(s) and padding octets in the aggregated frames, and FCS field(s) but not the PHY preamble, PHY header, MAC header, or MAC header validation. The parameter *pMaxFrameBodySize* is defined in the following subclauses:

— 11.2.8.2 for the 2.4 GHz PHY
— 12.2.8.2 for the SC PHY mode
— 12.3.7.4 for the HSI PHY mode
— 12.4.2.3.2 for the AV PHY mode
— 13.2.8.2 for the HRCP-SC PHY mode
— 13.3.8.2 for the HRCP-OOK PHY mode
— 14.1.6.3 for the THz PHY

[17] Information on references can be found in Clause 2.
[18] Interested applicants should contact the IEEE Registration Authority, http://standards.ieee.org/develop/regauth/.

Octets: 2	2	1	1	3	1
Frame Control	PNID	DestID	SrcID	Fragmentation Control	Stream Index
MAC Header					

variable	0 or 4
Frame Payload	FCS
MAC Frame Body	

Figure 6-1—MAC Header field and MAC Frame Body field formats for piconet

Octets:2	2	1	1	3	1
Frame Control	Pairnet ID	DestID	SrcID	TX and ACK Information	Stream Index
MAC header					

variable
Frame Payload
MAC Frame Body

Figure 6-2—MAC Header field and MAC Frame Body field format for pairnet

The number of octets in the MAC Frame Body field shall range from zero to *pMaxFrameBodySize*, inclusive. The maximum length of the MAC Frame Body field includes the length of the security fields, if present.

The Non-secure MAC Frame Body field shall be formatted as illustrated in Figure 6-3 when the security (SEC) field is set to zero in the Frame Control field.

Octets: variable	4
Frame Payload	FCS
Non-secure MAC Frame Body	

Figure 6-3—Non-secure MAC Frame Body field format for piconet

The non-secure MAC frame body for PRDEVs shall be formatted as illustrated in Figure 6-4 when the SEC bit is set to zero in the Frame Control field.

Octets: variable
Frame Payload
MAC Frame Body

Figure 6-4—Non-secure MAC Frame Body field format for pairnet

The Secure MAC Frame Body field shall be formatted as illustrated in Figure 6-5 when the SEC field shall be set to one in the Frame Control field. The Secure Payload field in the Secure MAC Frame Body field contains the Frame Payload protected as indicated in 9.2.2.

Octets: 2	**2**	**variable**	**8**	**4**
SECID	Secure Frame Counter	Secure Payload	Integrity Code	FCS
Frame payload				
Secure MAC Frame Body				

Figure 6-5—Secure MAC Frame Body field format for piconet

The secure MAC frame body for PRDEVs shall be formatted as illustrated in Figure 6-6 when the SEC bit is set to one in the Frame Control field.

Octets: 8	**variable**
Security Header	Frame Payload
Secure MAC Frame Body	

Figure 6-6—Secure MAC Frame Body field format for pairnet

6.2.2 Frame Control field

6.2.2.1 Overview

The Frame Control field shall be formatted as illustrated in Figure 6-7.

Bits: b0–b2	**b3–b5**	**b6**	**b7–b8**	**b9**	**b10**	**b11**	**b12**	**b13**	**b14**	**b15**
Protocol Version	Frame Type	SEC	ACK Policy	Retry	More Data	Imp-ACK Request	Imp-ACK NAK	CTA Relinquish	Blk-ACK	Reserved

Figure 6-7—Frame Control field format for piconet

The Frame Control field for a pairnet shall be formatted as illustrated in Figure 6-8.

Bits: b0–b2	**b3–b5**	**b6**	**b7–b8**	**b9**	**b10–b15**
Protocol Version	Frame Type	SEC	ACK Policy	Logical Channel	Reserved

Figure 6-8—Frame control field format for pairnet

6.2.2.2 Protocol Version field

The Protocol Version field is invariant in size and placement across all revisions of IEEE Std 802.15.3. For this revision of the standard the value of the protocol version is 0b000 for piconet and 0b001 for pairnet. All other values are reserved. The revision level will be incremented only when a fundamental incompatibility exists between a new revision and the prior revision of the standard. A DEV that receives a frame with a higher revision level than it supports may discard the frame without indication to the sending DEV.

6.2.2.3 Frame Type field

The Frame Type field indicates the type of frame that is being sent. Table 6-1 lists the valid frame type values and their description. The format and the usage of each of the individual frame types is defined in 6.3.

Table 6-1—Valid frame type values for piconet

Type value b5 b4 b3	Frame type description	Subclause
0b000	Beacon frame	6.3.1
0b001	Imm-ACK frame	6.3.2.1
0b010	Delayed ACK (Dly-ACK) frame	6.3.2.2
0b011	Command frame for piconets	6.3.3
0b100	Data frame for piconets	6.3.5
0b101	Multi-protocol Data frame for piconets	6.3.7
0b110	Synchronization frame for piconets	6.3.9
0b111	Reserved	—

Table 6-2 lists the valid frame type values and their description for pairnet.

Table 6-2—Valid frame type values for pairnet

Type value b5 b4 b3	Frame type description	Subclause
0b000	Beacon frame	6.3.1
0b001	Reserved	—
0b010	Reserved	—
0b011	Command frame	6.3.4
0b100	Data frame	6.3.5
0b101	Multi-protocol Data frame	6.3.5
0b110–0b111	Reserved	—

Stk-ACK frames with no data that are sent in response to data frames are treated as data frames and those sent in response to command frames are treated as command frames.

6.2.2.4 SEC field

The SEC field shall be set to one when the Frame Payload field is protected using the key specified by the security ID (SECID). The SEC field shall be set to zero otherwise. Frames with the SEC field set to one shall use the secure frame format for that frame type, as described in 6.3.

6.2.2.5 ACK Policy field, Imp-ACK Request field, and Blk-ACK field for piconet

The ACK Policy field, Imp-ACK Request field, and Blk-ACK field are used to indicate the type of acknowledgment procedure that the addressed recipient is required or allowed to perform. The use of the ACK procedures is described in 7.12. The allowed values for the ACK Policy field, Imp-ACK Request field, and Blk-ACK field are defined in Table 6-3. The ACK policy of a frame is determined by the combination of the ACK Policy field, Imp-ACK Request field, and Blk-ACK field.

Table 6-3—Valid ACK policy field type values for piconet

Blk-ACK field	Imp-ACK Request field	ACK Policy field	ACK policy type	Description
0	0	00	No ACK	The recipient(s) does not acknowledge the transmission, and the sender treats the transmission as successful without regard for the result, as described 7.12.2.
0	0	01	Imm-ACK	The addressed recipient returns an Imm-ACK frame after successful reception, as described in 7.12.3.
0	0	10	Dly-ACK	The addressed recipient keeps track of the frames received with this policy until requested to respond with a Dly-ACK frame, as described in 7.12.4.
0	0	11	Dly-ACK request	The addressed recipient returns either an Imm-ACK or a Dly-ACK frame after successful reception, as described in 7.12.4.
0	1	01	Imp-ACK	The addressed recipient returns an Imm-ACK frame, a data frame, or a command frame after successful reception, as described in 7.12.8.
1	0	01	Blk-ACK	The addressed recipient returns a Data frame with the MAC subheader including the Blk-ACK Bitmap field after successful reception of the MAC header, as described in 7.12.6.

An ACK policy of Dly-ACK or Dly-ACK Request is valid only in the data frames of a stream that is currently employing the Dly-ACK mechanism. It is not valid for frames using the asynchronous stream index or the MCTA stream index.

6.2.2.6 ACK Policy field for pairnet

The ACK Policy field, see Table 6-4, is used to indicate the type of acknowledgment procedure that the addressed recipient is required to perform. The allowed values of the ACK Policy field are defined in 7.12..

Table 6-4—Valid ACK policy field type values for pairnet

ACK policy field b8 b7	ACK policy type	Description
0b00	No ACK	The recipient(s) does not acknowledge the transmission, and the sender treats the transmission as successful without regard for the result, as described 7.12.2.
0b01	Reserved	—
0b10	Stk-ACK	The addressed recipient uses a Stk-ACK procedure for subframe exchange described 7.12.7.2.
0b11	Reserved	—

6.2.2.7 Retry field

The Retry field shall be set to one in any data or command frame that is a retransmission of an earlier frame. It shall be set to zero in all other frames.

6.2.2.8 More Data field

The More Data field shall be set to zero if the DEV will not use the rest of the channel time in that CTA, as described in 7.6.4.2. The More Data field shall be set to zero in the last frame of an extended beacon and in a Beacon frame that is not part of an extended beacon, as described in 7.8.3. In all other cases the More Data field shall be set to one. This includes frames, other than the last one, that are part of an extended beacon.

6.2.2.9 Imp-ACK Negative Acknowledgment (NAK) field

The Imp-ACK negative acknowledgment (NAK) field shall be set to one by a DEV when all of the following conditions are true:

— The DEV is responding to a frame addressed to the DEV for which it has successfully received the MAC header with an ACK policy of Imp-ACK.
— The FCS check, 6.2.10.6, for the Frame Payload field failed.

The Imp-ACK NAK field shall be set to zero otherwise. The use of the Imp-ACK NAK field is described in 7.12.8.

6.2.2.10 CTA Relinquish field

The CTA Relinquish field shall be set to one when the DEV relinquishes CTA ownership to another DEV. It shall be set to zero otherwise. The use of the CTA Relinquish field is described in 7.6.4.9.

6.2.2.11 Logical Channel

Logical Channel is available for use by the Higher Layer Protocol User and therefore out of scope from this specification. The value of this field is set to zero for CH0 of Logical Channel, and is set to one for CH1. All MSDUs in the MAC frame shall be sent in same Logical Channel.

6.2.3 Piconet ID (PNID) field

The PNID field contains the unique identifier for the piconet, as described in 7.14.2. The PNID normally remains constant during the current instantiation of the piconet and may be persistent for multiple sequential instantiations of the piconet by the same PNC. The PNID field shall be set to the current PNID for the piconet and is used to identify frames from DEVs in the piconet.

6.2.4 PairnetID field

The PairnetID field contains the unique identifier for the pairnet. The PairnetID normally remains constant during the current instantiation of the pairnet and may be persistent for multiple sequential instantiations of the pairnet by the same PRC. The PairnetID shall be set to the current PNID for the pairnet and is used to identify frames from DEVs in the pairnet.

6.2.5 SrcID and DestID fields for piconet

There are two DEVID fields in the MAC frame format. These fields are used to indicate the source DEVID (SrcID) and destination DEVID (DestID). A DEVID for a DEV is assigned by the PNC during the association of the DEV. The DEVID is unique to an associated DEV within a piconet. The following DEVIDs are assigned or reserved:

— 0x00 is the piconet coordinator identifier (PNCID)
— 0xED–0xF6 are reserved
— 0xF7, 0xF8, 0xF9, 0xFA, 0xFB, or 0xFC are neighbor identifiers (NbrIDs)
— 0xFD is the McstID
— 0xFE is the unassociated identifier (UnassocID)
— 0xFF is the broadcast identifier (BcstID)

The maximum number of valid DEVs, *mMaxNumValidDEVs*, is the maximum number of DEVIDs that the PNC is able to allocate in a piconet. This includes all of the regular DEVIDs, the PNCID, the GrpIDs, and the NbrIDs, but not the reserved IDs, the BcstID, McstID, or the UnassocID.

6.2.6 SrcID and DestID fields for pairnet

There are two DEVID fields in the MAC frame format. These fields are used to indicate the SrcID and DestID. A DEVID for a DEV is preassigned by the PRC in the Beacon frame before the association of the DEV. The DEVID is unique to an associated DEV within a pairnet. The following DEVIDs are assigned or reserved:

— 0x00 is the PRC
— 0xED–0xFE are reserved
— 0xFF is the BcstID

The maximum number of valid DEVs, *mMaxNumValidDEVs*, is the maximum number of DEVIDs that the PRC is able to allocate in a pairnet. This includes all of the regular DEVIDs and the pairnet coordinator identifier (PRCID) but not the remaining reserved IDs.

6.2.7 Fragmentation Control field for piconet

6.2.7.1 Overview

The Fragmentation Control field is used to aid in the fragmentation and a reassembly of MSDUs and command frames. The Fragmentation Control field shall be formatted as illustrated in Figure 6-9.

Bits: b0–b8	b9–b15	b16–b22	b23
MSDU Number	Fragment Number	Last Fragment Number	Reserved

Figure 6-9—Fragmentation Control field format

The three octets that compose the Fragmentation Control field may be used for reporting PHY-dependent receive status information to the transmitting DEV in an Imm-ACK frame, a Dly-ACK frame, or a Data frame sent in response to a Blk-ACK request that has a zero length Data Payload field. If the source DEV is not reporting PHY-dependent receive status information in an Imm-ACK, Dly-ACK frame, or a Data frame sent in response Blk-ACK request that has a zero length Data Payload field, it shall set the fragmentation field of the frame to all zeros, i.e., 0x000000. All other values are PHY dependent. The receive status for the 2.4 GHz PHY is defined in 11.7. The receive status for the mmWave PHYs is defined in 12.1.9.3.

For Data frames sent in response to a Blk-ACK request that have a non-zero length Data Payload field, the Fragmentation Control field shall contain the fragmentation information.

6.2.7.2 MSDU Number field

The MSDU Number field indicates the sequence number of the current MSDU or command frame.

For data frames, each DEV shall maintain one modulo-512 counter for each of its isochronous streams, and one for its asynchronous data traffic. The MSDU numbers for all command frames shall be assigned from a single modulo-512 counter.

Each MSDU number counter shall be set to zero when the DEV is initialized. The MSDU number counter for an isochronous stream shall be set to zero when the stream index for the isochronous stream is first assigned by the PNC, as described in 7.7.2.2.

6.2.7.3 Fragment Number field

The Fragment Number field indicates the order of the current fragment within the current MSDU. The Fragment Number field shall be set to zero in all unfragmented frames.

6.2.7.4 Last Fragment Number field

The Last Fragment Number field indicates the total number of fragments within the current MSDU. The value of this field is equal to one less than the number of fragments. The Last Fragment Number field shall be the same for every fragment of a fragmented MSDU and shall be set to zero for all unfragmented MSDUs.

6.2.8 Stream Index field

The Stream Index field assigned values are as follows:

— 0x00 is assigned for asynchronous data
— 0xFB is assigned for beam forming starting from sector-level training
— 0xFC is assigned for beam forming starting from beam-level training
— 0xFD is assigned for MCTA traffic
— 0xFE is assigned for unassigned streams
— 0xFF is reserved

DEVs use other values of the stream index as dynamically assigned by the PNC during the setup of the data stream, as described in 6.5.7.2. The PNC allocates a unique stream index value for each isochronous stream in the piconet. A DEV shall support all stream index values that are not reserved.

For pairnet, the Stream Index field assigned values are as follows:

— 0x00 is assigned for data
— 0x01 to 0xFF is reserved

6.2.9 MAC header validation

When the PHY receives a frame, it validates the received frame's MAC header before passing the MAC Header field and its associated Frame Payload field to the MAC. The protection mechanism used to validate the MAC Header field is PHY dependent. In addition, the bit order and the length of the protection mechanism, *pLengthHcs*, are also PHY dependent. The MAC Header field protection mechanism is defined in the following subclauses:

— 11.2.9 for the 2.4 GHz PHY
— 12.2.4.3.3 for the SC PHY mode
— 12.3.4.5 for the HSI PHY mode
— 12.4.2.4 for the AV PHY mode
— 13.2.4.3.3 for the HRCP-SC PHY mode, the THz-SC PHY mode, and the THz-OOK PHY mode
— 13.3.4.3.3 for HRCP-OOK PHY mode

6.2.10 MAC Frame Body field

6.2.10.1 Frame Payload field

The Frame Payload field is a variable-length field that carries the information that is to be transferred to a DEV or group of DEVs in the piconet or to a DEV in the pairnet. In the case of a secure frame, it also includes the required security information and the secure payload, as illustrated in Figure 6-5 for piconet and as described in 6.2.10.9 for pairnet.

6.2.10.2 Secure session ID (SECID) field

The SECID field contains an identifier for the key that is being used to protect the frame. The SECID field shall be formatted as illustrated in Figure 6-10.

Octets: 1	**1**
Key Originator	Key ID

Figure 6-10—SECID field format

The Key Originator field for all keys except the piconet group data key or the pairnet group data key shall be set to the DEVID of the key originator in the relationship. The Key Originator field for the piconet group data key or the pairnet group data key shall be set to the BcstID.

Key ID field shall have a unique value for the key associated with the security relationship. The Key ID field value for a given key is selected by the key originator in a security relationship, as described in 8.3.9.

6.2.10.3 Secure Frame Counter (SFC) field

The Secure Frame Counter (SFC) field contains a counter that is used to ensure the uniqueness of the nonce in a secure frame. For piconets, a DEV shall not reuse a frame counter with the same time token, as described in 6.3.1.1, and key, as described in 8.3.6. For pairnets, a DEV shall not reuse a frame counter with the same key, as described in 8.3.6. For piconets, the DEV shall initialize the SFC to zero for the first frame sent and increment it for each successive secure frame sent. For pairnets, the DEV shall initialize the SFC value to zero for the first frame or subframe sent and increment it for each successive secure frame sent or each successive subframe sent in the aggregated frame. Only the SFC value of the first subframe is explicitly included in the transmitted aggregated frame. For piconets, when the time token, as described in 6.3.1, is updated, the DEV shall reset the SFC to zero. For pairnets, the SFC value shall be increased even when the time token is updated. In the case where the DEV receives a new key, the DEV shall set the SFC to zero.

6.2.10.4 Secure Payload field

The Secure Payload field is a variable-length field that contains the information, protected by the symmetric key security operations, as defined in 9.3 for piconet and as defined in Clause 10 for pairnet, that is to be transferred to a DEV or group of DEVs in the piconet or to a DEV in the pairnet. As illustrated in Figure 6-5 for piconet and as described in 6.2.10.9 for pairnet, the Secure Payload field is a part of the Frame Payload field and does not include the SECID, SFC, or Integrity Code fields.

6.2.10.5 Integrity Code field

The Integrity Code field contains an encrypted integrity code that is used to cryptographically protect the integrity of the MAC header and Frame Payload. The integrity code is computed as specified in 9.3 for piconets, and is computed as specified in 10.3 for pairnets.

6.2.10.6 Frame Check Sequence (FCS) field

The FCS field contains a 32-bit cyclic redundancy check (CRC). The CRC described here is equivalent to ANSI X3.66-1979. The MSB of the FCS is the coefficient of the highest order term, and the field is sent over the wireless medium, as indicated in 6.1. The FCS shall be calculated over the entire Frame Payload field, which is referred to here as the *Calculation field*.

If the Frame Payload field has zero length (as in an immediate ACK frame) the FCS shall not be sent.

The FCS is calculated using the following standard generator polynomial of degree 32, as shown in Equation (6-1):

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \tag{6-1}$$

The FCS is the one’s complement of the modulo 2 sum of the remainders in item a) and item b) below:

a) The remainder resulting from $((x^k \times (x^{31}+x^{30}+\ldots))$ divided (modulo 2) by $G(x)$. The value k is the number of bits in the calculation field.
b) The remainder resulting from the calculation field contents, treated as a polynomial, is multiplied by x^{32} and then divided by $G(x)$.

The FCS field shall be transmitted in the order specified in 6.1.

At the transmitter, the initial remainder of the division shall be preset to all ones and is then modified via division of the calculation fields by the generator polynomial G(x). The one's complement of this remainder is the FCS field.

At the receiver, the initial remainder shall be preset to all ones. The serial incoming bits of the calculation fields and FCS, when divided by G(x) in the absence of transmission errors, results in a unique non-zero remainder value. The unique remainder value is the following polynomial shown in Equation (6-2):

$$x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1 \tag{6-2}$$

6.2.10.7 MAC frame body for pairnet

The MAC frame body for pairnet is described in 6.3.1.2, 6.3.4.1, 6.3.6.1, and 6.3.8.2.

6.2.10.8 Security header for pairnet

The Security header for pairnet shall be formatted as illustrated in Figure 6-11.

Octets: 2	6
SECID	SFC

Figure 6-11—Security header

The SECID field is used to identify the key set that is used to encrypt and/or authenticate the data in the frame, as defined in 6.2.10.2.

The SFC field contains a counter that is used to ensure the uniqueness of the nonce of a secure frame, as defined in 6.2.10.3.

6.2.10.9 Secure MAC frame body for pairnet

The Secure MAC frame body for pairnet is described in 6.3.1.4, 6.3.4.2, 6.3.6.2, and 6.3.8.2.

6.2.11 SC and HSI aggregated frame format for piconet

6.2.11.1 Overview

Figure 6-12 illustrates the SC and HSI aggregated frame format.

MAC Subheader	MAC Frame Body

Figure 6-12—SC and HSI aggregated frame format

6.2.11.2 Standard aggregation format

6.2.11.2.1 Non-secure standard aggregation format

To use standard aggregation, the Aggregation field and Low-latency Mode field in PHY header shall be set as described in 12.2.4.3.2.

The MAC Subheader field for standard aggregation shall be formatted as illustrated in Figure 6-13.

Octets: 2	**5**	**…**	**5**	**1**
Blk-ACK Bitmap	Subheader 1	…	Subheader 8	RX Buffer Size

Figure 6-13—MAC Subheader field format for standard aggregation

The Blk-ACK Bitmap field shall be formatted as illustrated in Figure 6-14.

Octets: 1	**1**
ACK or MSB ACK	ACK or LSB ACK

Figure 6-14—Blk-ACK bitmap field format

In equal error protection (EEP) mode, the ACK or MSB ACK field is a bitmap that indicates if a subframe was correctly received. The bit position zero, which is the first bit from right in Figure 6-14, corresponds to the first subframe of the frame that is being ACKed. If a subframe was correctly received, then the bit for that subframe shall be set to one and shall be set to zero otherwise. The ACK or LSB ACK field shall be set to the same value as the ACK or MSB ACK field.

To use UEP mode, the UEP field in PHY header shall be set as described in 12.2.4.3.2. In UEP mode, the ACK or MSB ACK field is a bitmap that indicates if the MSBs of a subframe were correctly received. The bit position zero, which is the first bit from right in Figure 6-14, corresponds to the first subframe of the frame that is being ACKed. If the MSBs of the subframe were correctly received, then the bit for that subframe shall be set to one and shall be set to zero otherwise. The ACK or LSB ACK field is a bitmap that indicates if the LSBs of a subframe were correctly received. The bit position zero of the ACK or LSB ACK field corresponds to the first subframe of the frame that is being ACKed. If the LSBs of the subframe were correctly received, then the bit for that subframe is set to one and shall be set to zero otherwise.

The Subheader field shall be formatted as illustrated in Figure 6-15.

Bits: b0–b4	**b5**	**b6**	**b7**	**b8–b18**	**b19–b20**	**b21**	**b22–b30**	**b31–b37**	**b38**	**b39**
MCS information	FCS present	Retry	Resolution indication	Subframe length	Subframe information	Skewed constellation	MSDU number	Fragment number	Last fragment	Reserved

Figure 6-15—Subheader field format

In UEP mode, the MCS Information field shall be set to the UEP MCS (as defined in 12.2.3.9 for SC PHY and 12.3.3.1 for HSI PHY) that is used for the subframe.

In EEP mode, the MCS Information field shall be set to zero. All the subframes use the MCS indicated in the PHY header.

The FCS Present field shall be set to one if the subframe uses an FCS and shall be set to zero otherwise. The FCS is optional for subframes for the cases in which the upper layer will handle checking the data integrity.

If the FCS present field is set to zero, then the subframe is considered correctly received only if the PHY header, MAC header, and MAC subheader were all correctly received.

The Retry field shall be set to one if the subframe is a retransmission and shall be set to zero otherwise.

The Resolution Indication field shall be set to zero if the resolution of the Subframe Length field is 1 octet and shall be set to one if the resolution is 512 octets.

The Subframe Length field is used to determine the length of the subframe before coding, not including the FCS. If the resolution of the Subframe Length field is 1 octet, then this field contains the length of the subframe in octets. If the resolution of the Subframe Length field is 512 octets, then the length of the subframe in octets is 512 times the value of this field. If the Subframe length field is set to zero, the corresponding subframe is not present in the aggregated data frame.

The Subframe Information field indicates if the subframe contains MSB, LSB, or MSB and LSB combined data. Valid values for the Subframe Information field are as follows:

0 → The subframe contains only MSB data
1 → The subframe contains only LSB data
2 → The subframe contains MSB and LSB data
3 → Reserved

The Skewed Constellation field shall be set to one if a skewed constellation is used in the subframe and shall be set to zero otherwise.

The MSDU Number field indicates the MSDU number of the subframe. Each subframe that contains a fragment from the same MSDU shall have the same MSDU number.

The Fragment Number field indicates the fragment sequence number of the subframe within the current MSDU, if the subframe contains a MSDU fragment. This field shall be set to zero if the corresponding subframe contains an unfragmented MSDU.

The Last Fragment field shall be set to one if the subframe contains the last fragment of the MSDU and shall be set to zero otherwise.

The receiver (RX) Buffer Size field indicates the free buffer space at the target DEV as a multiple of the preferred fragment size, as defined in 6.4.12.

The MAC Frame Body field for standard aggregation shall be formatted as illustrated in Figure 6-16.

Octets: variable	**4 or 8**	**…**	**variable**	**4 or 8**
Subframe Payload 1	FCS or Combined FCS	…	Subframe Payload *n*	FCS or Combined FCS

Figure 6-16—MAC Frame Body field format for standard aggregation

The maximum number of subframes that are aggregated in one frame shall be *mMaxSubframeSize*, as defined in 7.20.

In EEP mode, the subframe contains the FCS field, as defined in 6.2.10.6. In UEP mode, the subframe contains the Combined FCS field, as defined in Figure 6-44. The Subframe Payload field is formatted as defined in 6.3.7 for the Multi-protocol Data frame.

6.2.11.2.2 Secure standard aggregation format

The secure MAC Subheader field for standard aggregation shall be formatted as illustrated in Figure 6-17.

Octets: 5	2	5	…	5	1
Security Header	Blk-ACK bitmap	Subheader 1	…	Subheader 8	RX Buffer Size

Figure 6-17—Secure MAC Subheader field format for standard aggregation

The Security Header field shall be formatted as illustrated in Figure 6-18.

Octets: 2	2	1
SECID	SFC	Subframe Security

Figure 6-18—Security Header field format

The SECID field is used to identify the key set that is used to encrypt and/or authenticate the data in the frame, as defined in 6.2.10.2.

The SFC field contains a counter that is used to ensure the uniqueness of the nonce of a secure frame, as defined in 6.2.10.3.

The Subframe Security field contains a bitmap that indicate if a subframe applies security or not. The bit position zero, which is the first bit from right in Figure 6-18, corresponds to the first subframe. The bit shall be set to one if the subframe applies security and shall be set to zero otherwise.

The secure MAC Frame Body field for standard aggregation shall be formatted as illustrated in Figure 6-19.

Octets: variable	8	4 or 8	…	variable	8	4 or 8
Subframe payload 1	Integrity code	FCS or Combined FCS	…	Subframe payload *n*	Integrity code	FCS or Combined FCS

Figure 6-19—Secure MAC Frame Body field format for standard aggregation

The Integrity Code field is used to cryptographically protect the integrity of the header and payload, as defined in 6.2.10.5.

6.2.11.3 Low-latency aggregation formats

6.2.11.3.1 Non-secure low-latency aggregation format

To use low-latency aggregation mode, the Aggregation field and the Low-latency Mode field in PHY header shall be set as described in 12.2.4.3.2.

The MAC Subheader field for low-latency aggregation shall be formatted as illustrated in Figure 6-20.

Octets: 1	1	2	2	64
UEP	RX Buffer Size	MSDU Request Number	MSDU Response Number	Blk-ACK bitmap

Figure 6-20—MAC Subheader field format for low-latency aggregation

The UEP field shall be formatted as illustrated in Figure 6-21.

Bits: b0–b4	b5–b7
UEP MCS	Reserved

Figure 6-21—UEP field format

In EEP mode, the UEP MCS field shall be set to zero and all subframes use the MCS indicated in the PHY header.

In UEP mode, the UEP MCS field indicates the MCS used for UEP mode for the subframes, as defined in 12.2.3.9.

The RX Buffer size field indicates the free buffer space at the destination DEV as a multiple of the preferred fragment size, as defined in 6.4.12.

The MSDU Request Number field and the MSDU Response Number field shall be formatted as illustrated in Figure 6-22.

Bits: b0–b8	b9–b15
MSDU Sequence Number	Reserved

Figure 6-22—MSDU Request/Response Number field format

The MSDU Sequence Number field is used as the basis for the Blk-ACK bitmap.

The MSDU Request Number field indicates the most recent MSDU sequence number acknowledged at the transmitter.

The MSDU Response Number field indicates the first MSDU sequence number of the transmitted Blk-ACK bitmap-field. The MSDU Response Number field is used to generate the offset for the Blk-ACK bitmap field. The MSDU Response Number field shall be a copy from the received MSDU Request Number, upon reception of a valid MAC Subheader.

For EEP mode the Blk-ACK Bitmap field shall be formatted as illustrated in Figure 6-23.

Bits: b0	b1	…	b255	b256–b511
Subframe ACK #1	Subframe ACK #2	…	Subframe ACK #256	Reserved

Figure 6-23—EEP mode Blk-ACK Bitmap field format

The Subframe ACK field is an offset from the MSDU sequence number of the received MSDU Response Number field. It shall be set to one if the corresponding subframe was correctly received and shall be set to zero otherwise.

For UEP mode, the Blk-ACK Bitmap field shall be formatted as illustrated in Figure 6-24.

Bits: b0	b1	b2	b3	…	b510	b511
Subframe MSB ACK #1	Subframe LSB ACK #1	Subframe MSB ACK #2	Subframe LSB ACK #2	…	Subframe MSB ACK #256	Subframe LSB ACK #256

Figure 6-24—UEP mode Blk-ACK Bitmap field format

The Subframe MSB ACK field shall be set to one if the MSBs of the corresponding subframe were correctly received and shall be set to zero otherwise.

The Subframe LSB ACK field shall be set to one if the LSBs of the corresponding subframe were correctly received and shall be set to zero otherwise.

The MAC Frame Body field for low-latency aggregation shall be formatted as illustrated in Figure 6-25.

Octets: 3	variable	4 or 8	…	3	variable	4 or 8
MSDU Subheader *n*	Subframe Payload 1	FCS or Combined FCS	…	MSDU Subheader *n*	Subframe Payload *n*	FCS or Combined FCS

Figure 6-25—MAC Frame Body field format for low-latency aggregation

The maximum number of subframes that are aggregated in one frame shall be *mMaxSubframeSize*, as defined in 7.20.

The MSDU Subheader field shall be formatted as illustrated in Figure 6-26.

Bits: b0–b8	b9–b14	b15	b16–b23
MSDU Number	Subframe Length	Reserved	MSDU Subheader HCS

Figure 6-26—MSDU Subheader field format

The MSDU Number field indicates the sequence number of the MSDU that is aggregated in the subframe. The MSDU number is unique in a window of 256 MSDUs, allowing retransmission of an out-of-order MSDU over several frames.

Subframe Length field contains the length of each subframe in unit of four octets. This field may be set to zero to allow idle transmission of data if data is not present at the FCSL. The HCS field shall be inverted in the case of a zero-length MSDU, so that the MSDU HCS check fails.

The MSDU Subheader HCS field contain a CRC defined as the one's complement of the remainder of the division of the first 16 bits of the header by the polynomial $x^8 + x^2 + x + 1$. A serial implementation is illustrated in Figure 6-27.

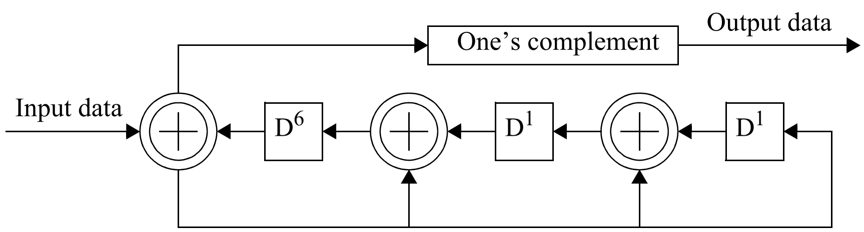

Figure 6-27—HCS implementation example for MSDU subheader

In EEP mode, the subframe contains the FCS field, as defined in 6.2.10.6. In UEP mode, the subframe contains the Combined FCS field, as defined in Figure 6-44.

6.2.11.3.2 Secure low-latency aggregation format

The Secure MAC Subheader field for low-latency aggregation shall be formatted as shown in Figure 6-28.

Octets: 2	2	1	1	2	2	32/64
SECID	SFC	UEP	RX Buffer Size	MSDU Request Number	MSDU Response Number	Blk-ACK Bitmap

Figure 6-28—Secure MAC Subheader field format for low-latency aggregation

The SECID field is used to identify the key set that is used to encrypt and/or authenticate the data in the frame, as defined in 6.2.10.2.

The SFC field contains a counter that is used to ensure the uniqueness of the nonce of a secure frame, as defined in 6.2.10.3.

The secure MAC Frame Body field for low-latency aggregation shall be formatted as shown in Figure 6-29.

Octets: 3	variable	8	4 or 8	…	3	variable	8	4 or 8
Secure MSDU Subheader 1	Subframe Payload 1	Integrity Code	FCS or Combined FCS	…	Secure MSDU Subheader *n*	Subframe Payload *n*	Integrity Code	FCS or Combined FCS

Figure 6-29—Secure MAC Frame Body field format for low-latency aggregation

The Secure MSDU Subheader field shall be formatted as illustrated in Figure 6-30.

Bits: b0–b8	b9–b14	b15	b16–b23
MSDU Number	Subframe Length	Subframe Security	MSDU Subheader HCS

Figure 6-30—Secure MSDU Subheader field format

The Subframe Security field shall be set to one if the subframe applies security and shall be set to zero otherwise.

The Integrity Code field is used to cryptographically protect the integrity of the header and payload, as defined in 6.2.10.5.

6.2.12 AV aggregated frame format for piconet

6.2.12.1 Overview

The AV aggregated frame format is optimized to carry uncompressed audio and video in an efficient manner. The AV aggregated frame format is used instead of the standard aggregation or low-latency aggregation formats.

Figure 6-31 illustrates the AV aggregated MAC Frame Body field format.

MAC Frame Body				
Subframe 1	Subframe 2	…	Subframe 6	Subframe 7

Figure 6-31—AV aggregated MAC Frame Body field format

The collection of subframes in a single frame is referred to as the MAC Frame Body field. The MAC Frame Body field of a MAC frame may have one to seven, inclusive, subframes of varying sizes.

6.2.12.2 Extended MAC header format

6.2.12.2.1 Overview

The HRP Extended MAC Header field is used to describe the contents of an AV aggregated frame, which typically is used to transport uncompressed audio and video.

The HRP Extended MAC Header field shall be formatted as illustrated in Figure 6-32.

Octets: 10	2	5	6	24	16
MAC header	Extended Control Header	MAC Extension Header	Security Header	Video Header	Reserved

Figure 6-32—HRP Extended MAC Header field format

The LRP Extended MAC Header field shall be formatted as illustrated in Figure 6-33.

Octets: 10	2	5	5
MAC Header	Extended Control Header	MAC Extension Header	Security Header

Figure 6-33—LRP Extended MAC Header field format

The MAC Extension Header field and Security Header field are present in LRP frames only if their corresponding fields are set to one in the Extended Control Header.

6.2.12.2.2 Extended Control Header field

The Extended Control Header field shall be formatted as illustrated in Figure 6-34.

Bits: b0–b2	b3	b4	b5	b6–b8	b9–b15
Frame Class	MAC Extension Header Present	Security Header Present	Video Header Present	Frame Type	Reserved

Figure 6-34—Extended Control Header field format

Valid values for the Frame Class field are given in Table 6-5.

Table 6-5—Frame Class field values

Field value	Frame class
0b000	Regular
0b0001	AV aggregated
0b010	Omni-ACK
0b011	Beacon
0b100–0b111	Reserved

Regular frames have a single payload field without MAC level aggregation while AV aggregated frames have one or more subframes as a part of the frame.

The MAC Extension Header Present field shall be set to one if the MAC Extension Header field is in the Extended MAC Header field and shall be set to zero otherwise.

The Security Header Present field shall be set to one if the Security Header field is in the Extended MAC Header field and shall be set to zero otherwise.

The Video Header Present field shall be set to one if the Video Header field is in the Extended MAC Header field and shall be set to zero otherwise.

For HRP frames, the MAC Extension Header Present field, Security Header Present field, and Video Header Present field indicate the validity of the associated field. Because the HRP header is a fixed length, these fields are always present in the Extended MAC Header field. If a header in the HRP MAC header is present but not valid, it may be set to any value and shall be ignored upon reception. For LRP frames, unused headers, as indicated by the appropriate header present field, are not present in the MAC header.

The Frame Type field is only defined for regular frame class; it shall be set to zero in other frames. AV aggregated frames use the Type field in the MAC Extension Header field while the Omni-ACK and Beacon frames do not require the Frame Type field. Valid values for the Frame Type field are as follows:

0 → MAC commands
1 → Data
2 → Audio
3–7 → Reserved

6.2.12.2.3 MAC Extension Header field

The MAC Extension Header field is illustrated in Figure 6-35.

Bits: b0–b3	**…**	**b24–b27**	**b28–b31**	**b32–b39**
Type 1	…	Type 7	Reserved	ACK Groups

Figure 6-35—MAC Extension Header field format

The Type field indicates the type of data that is contained in the subframe. Valid values for the Type field are as follows:

0x0 → MAC commands
0x1 → Data
0x2 → Audio
0x3 → Video
0x4–0xF → Reserved

The ACK Groups field shall be formatted as illustrated in Figure 6-36.

Bits: b0	**b1**	**…**	**b6**	**b7**
Subframe 1	Subframe 2	…	Subframe 7	LSB FCS

Figure 6-36—ACK Groups field format

The bit for a subframe shall be set to one if the subframe is in the same ACK group as the previous (i.e., lower numbered subframe). Otherwise, it is the first subframe in an ACK group and its bit shall be set to zero. The first bit, corresponding to subframe 1, shall always be set to zero as it is always the start of an ACK group. No more than 5 ACK groups shall be defined; therefore, the number of bits set to zero among the subframe bits shall not exceed five.

The LSB FCS field shall be set to one if the LSB FCS is part of the calculation to determine if a subframe was correctly received, as defined in 7.1. It shall be set to zero if the LSB FCS is ignored in determining if a subframe was correctly received. The setting of this field applies only to those subframes that are sent with a UEP HRP mode. For all other subframes, the LSB FCS field shall be set to zero.

6.2.12.2.4 Security Header field

The Security Header field shall be formatted as illustrated in Figure 6-37.

Octets: 4	2
Security Control	SFC

Figure 6-37—Security Header field format

The Security Control field shall be formatted as illustrated in Figure 6-38.

Bits: b0–b15	b16–b17	b18–b19	…	b30–b31
SECID	Reserved	Subframe 1 Security	…	Subframe 7 Security

Figure 6-38—Security Control field format

The SECID field is used to identify the key set that is used to encrypt and/or authenticate the data in the frame, as defined in 6.2.10.2.

The Subframe Security field indicates the type of security that is applied to a subframe. Valid values are as follows:

0b00 → No security applied
0b01 → Encryption and integrity code
0b10–0b11 → Reserved

The SFC field is defined in 6.2.10.3. The SFC shall be incremented for each subframe in the frame, even if security is not applied to the subframe.

6.2.12.2.5 Video Header field

The Video Header field shall be formatted as illustrated in Figure 6-39.

Octets: 5	5	5	5	4
Video Control 1	Video Control 2	Video Control 3	Video Control 4	Reserved

Figure 6-39—Video Header field format

Unless otherwise stated, the numbers for all of the fields begin with zero, e.g., the first video frame is number zero, then number one, and so on. One of the first video subframes in the first frame sent in a stream should have the Video Frame Number field, the H-Position field, and V-Position field set to zero.

The Video Subheader field is included as part of the MAC header because the error rate for the MAC header is lower than that of the subframes.

The Video Control field shall be formatted as illustrated in Figure 6-40.

Bits: b0–b3	b4–b6	b7	b8–b23	b24–b39
Reserved	Video Frame Number	Interlaced Field Indication	H-Position	V-Position

Figure 6-40—Video Control field format

The Interlaced Field Indication field shall be set to one if the video subframes carry pixels for the bottom field. It shall be set to zero if the video subframes carry pixels for the top field or if the video subframes carry pixels for non-interlaced video modes.

The Video Frame Number field contains a counter that keeps track of the video frame to which the pixels in the subframe belong. The video frame number is calculated follows:

- For progressive video, the Video Frame Number field shall be incremented sequentially. After reaching the max value of 0x7, the next value shall be zero. All frames belonging to the same video frame have identical Video Frame Number values.
- For interlaced video, the Video Frame Number field shall be incremented in a step of two. Thus, each video frame has two frame numbers. All frames belonging to the first field have even Video Frame Numbers and all frames belonging to the second field have odd Video Frame Numbers. For example, for the first uncompressed video frame, the frames belonging to the first field have a Video Frame Number set to zero, and the frames belonging to the second field have a Video Frame Number set to one. Therefore, the same video frame has two Video Frame Numbers.

The H-Position field contains the horizontal position of the first pixel in the subframe where zero is on the left side of the screen.

The V-Position field contains the vertical position of the first pixel in the subframe where zero corresponds to the top of the screen. For interlaced formats, the V-positions of the lines range from 0 to 539 independent of status as even or odd frame number.

6.2.12.3 Subframe format

The subframes with Type Video in the MAC frame shall be formatted as illustrated in Figure 6-41.

Octets: variable	8
Subframe Payload	Combined FCS

Figure 6-41—Subframe field format for type video

Subframes other than those with Type Video shall be formatted as illustrated in Figure 6-42.

Octets: variable	4
Subframe Payload	FCS

Figure 6-42—Subframe field format for type other than video

The Subframe Payload field shall be formatted as defined in 6.3.7 for the Multi-protocol Data frame. If the subframe has an integrity code, as indicated by the Subframe Security field, then Subframe Payload field shall be formatted as illustrated in Figure 6-43, where the Payload Data field shall be formatted as defined in 6.3.7 for the Multi-protocol Data frame.

Octets: 8	**variable**
Integrity Code	Payload Data

Figure 6-43—Subframe Payload with integrity code

The Integrity Code field is defined in 6.2.10.5.

The Combined FCS field shall be formatted as illustrated in Figure 6-44.

Bits:b0–b3	**b4–b7**	**b8–1b1**	**…**	**b28–b31**	**b32–b35**	**b36–b39**	**b40–b43**	**…**	**b52–b55**	**b56–b59**	**b60–b63**
LSB FCS 5	MSB FCS 5	LSB FCS 6	…	MSB FCS 8	LSB FCS 1	MSB FCS 1	LSB FCS 2	…	MSB FCS 3	LSB FCS 4	MSB FCS 4

Figure 6-44—Combined FCS field format

The LSB FCS field contains the FCS, as defined in 6.2.10.6, calculated over the four LSBs of each octet in the Subframe Payload field. The LSB FCS field occupies the LSBs of the Combined FCS field.

The MSB FCS field contains the FCS, as defined in 6.2.10.6, calculated over the four MSBs of each, if present, and Subframe Payload field. The MSB FCS field occupies the MSBs of the Combined FCS field.

The FCS field contains the FCS, as defined in 6.2.10.6, calculated over Subframe Payload field.

Subframes with frame type other than data do not support aggregation or fragmentation.

Subframes with frame type data support the aggregation of MSDUs or the fragmentation of MSDUs in their payloads. The format of a Subframe Payload field for a subframe with Frame Type data is illustrated in Figure 6-45.

Octets: variable	**variable**	**…**	**variable**
Sub-payload 1	Sub-payload 2	…	Sub-payload *n*

Figure 6-45—Subframe Payload field with Frame Type data

Each Sub-payload field shall be formatted as shown in Figure 6-46.

Bits: b0–b19	**b20–b29**	**b30**	**b31**	**variable**
Length	Sequence Number	Last Fragment	First Fragment	MSDU

Figure 6-46—Sub-payload field format

The Length field indicate the length of the MSDU field in octets.

The Sequence Number field is incremented for each fragment, regardless if it is of the same or different MSDU.

The Last Fragment field and First Fragment field shall be set as indicated in Table 6-6.

Table 6-6—Fragment field settings

Fragment type	Last Fragment field	First Fragment field
First fragment	0	1
Last fragment	1	0
Middle fragment	0	0
Complete MSDU	1	1

The MSDU field for subframes with frame type data shall be formatted, as defined in 6.3.7.

6.2.12.4 AV aggregated

The AV aggregated frame shall be formatted as illustrated in Figure 6-47. The AV aggregated frame shall not be used for LRP frames.

Octets: 66	variable	variable	…	variable
Extended MAC Header	Subframe 1	Subframe 1	…	Subframe *n*

Figure 6-47—AV aggregated frame format

The AV aggregated frame has one to seven, inclusive, subframes. The subframe format is defined in 6.2.12.3.

6.2.12.5 Regular

The HRP regular frame shall be formatted as illustrated in Figure 6-48.

Octets: 66	variable
Extended MAC Header	Subframe

Figure 6-48—HRP regular frame format

The LRP regular frame shall be formatted as illustrated in Figure 6-49.

Octets: variable	variable
Extended MAC header	Subframe

Figure 6-49—LRP regular frame format

The subframe format is defined in 6.2.12.3.

6.2.12.6 Directional ACK

The Directional ACK frame format is defined in 12.4.4.9. The Directional ACK frame is used to acknowledge HRP frames and beam formed LRP frames. The ACK policy for a directional ACK shall be Imm-ACK. The directional ACK shall be only used with the AV PHY HRP frames.

6.2.13 TX and ACK Information field for pairnets

The transmit (TX) and ACK Information field shall be formatted as illustrated in Figure 6-51.

Bits: b0–b8	b9	b10–b19	b20	b21	b22	b23
Number of Subframes	Last Received Frame Type	Last Received Sequence Number	Buffer Full	Buffer Empty	DEV Sleep	Reserved

Figure 6-50—TX and ACK Information field format for pairnet

The Number of Subframes field indicates the number of subframes included in the current frame. Up to 256 subframes can be aggregated into a single frame. The valid range of this field is [0–256] and other values outside this range are reserved.

The Last Received Frame Type field indicates the type of frame that was sent with the sequence number indicated by the Last Received Sequence Number field. A value of one indicates a data frame, and a value of zero indicates a command frame.

The Last Received Sequence Number field indicates the most recent contiguous sequence number of subframes that was successfully received by the DEV. The details are illustrated in 7.12.7. The initial value of the Last Received Sequence Number field for both command and data frames is 0x3FF.

The Buffer Full field indicates that the reception buffer of the sender is full. A value of one indicates that the buffer is full and a value of zero indicates that the buffer is not full.

The Buffer empty field shall be set to one when the reception buffer becomes empty at any point in time between the previous ACK replying frame and the current ACK transmission. Otherwise this field shall be set to zero.

The DEV Sleep field indicates if the sender will transition to sleep state. A value of one indicates the sender is going to sleep and shall be set to zero otherwise.

6.3 Format of individual frame types

6.3.1 Beacon frame

6.3.1.1 Non-secure Beacon frame for piconets

The Non-secure Beacon frame Frame Payload field shall be formatted as illustrated in Figure 6-51.

Octets: 19	**variable**	**…**	**variable**	**4**
Piconet synchronization parameters	IE-1	…	IE-*n*	FCS

Figure 6-51—Non-secure Beacon frame Frame Payload field format

The individual IEs in the Beacon frame Frame Payload field are listed in Table 6-23. These IEs are encoded in type, length, value format and are defined in 6.4. The IEs in the beacon payload may appear in any order except for the CTA IEs, which shall be the first IEs of the beacon payload following the Piconet Synchronization Parameters field.

The Piconet Synchronization Parameter field shall be formatted as illustrated in Figure 6-52.

Octets: 6	**2**	**2**	**1**	**1**	**1**	**6**
Time Token	Superframe Duration	CAP End Time	Max TX Power Level	Piconet Mode	PNC Response	PNC Address

Figure 6-52—Piconet Synchronization Parameters field format

The Time Token field contains a strictly increasing counter, which is incremented in each beacon. The Time Token counter shall be set to zero by the PNC when a piconet is started, as defined in 7.2.3. The beacon number is defined to be the 16 LSBs of the time token.

NOTE—For a 1 ms superframe, the time token would roll over in just under 9000 years. At that point the PNC would shutdown the piconet and start a new one.

The Superframe Duration field contains the duration of the current superframe. The resolution of this field is 1 µs and therefore has a range of [0–65535] µs. However, the valid range of this field is [*mMinSuperframeDuration*, *mMaxSuperframeDuration*].

The CAP End Time field specifies the end of the CAP interval for the current superframe, as defined in 7.8. The resolution of this field is 1 µs, which gives a range of [0–65535] µs. The CAP begins a SIFS after the end of the beacon and continues until the CAP end time.

The Max TX Power Level field is used to indicate the maximum TX power level allowed in the current superframe by the PNC in the piconet, as described in 7.15.3.2. The value is in dBm encoded in two's complement format. For example, a +2 dBm TX power level is encoded as 0x02, while a –2 dBm TX power level is encoded as 0xFE. However, if the PNC does not want to limit the TX power, then it shall set the field to 0x7F.

The Piconet Mode field defines certain characteristics about the piconet and the superframe. The encoding of this octet shall be formatted as illustrated in Figure 6-53.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
CAP Data	CAP Commands	CAP Association	MCTA Used	SEC Mode	Reserved	Other IEs Unchanged	CTA IEs Unchanged

Figure 6-53—Piconet Mode field

If the CAP Data field, CAP Commands field, or CAP Association field value is set to one, then that type of data or command is allowed to be sent in the CAP of the current superframe. The CAP Commands field applies to all commands except for the Association Request command, which is covered by the CAP Association field. Otherwise, that type of frame is not allowed to be sent in the CAP. The use of these fields is described in 7.6.3 and 11.2.10.

The MCTA Used field shall be set to one if the PNC will be using open or association MCTAs in the superframe.

The SEC Mode field indicates the current security settings in the piconet, as defined in 8.2. The field is encoded as illustrated in Table 6-7.

Table 6-7—SEC Mode field encodings

Type value	SEC mode
0	Mode 0
1	Mode 1

The Other IEs Unchanged field may be set to one if all the IEs (other than CTA IEs) in the beacon payload are identical to the IEs (other than CTA IEs) contained in the previous beacon, as described in 7.8.3.

The CTA IEs Unchanged field may be set to one if the CTA IEs in the beacon are identical to the CTA IEs contained in the previous beacon, as described in 7.8.3.

The PNC Response field shall be formatted as illustrated in Figure 6-54.

Bits: b0–b3	b4–b7
MCTA Allocation Rate	Reserved

Figure 6-54—PNC Response field format

The MCTA Allocation Rate field indicates the frequency with which the PNC will be allocating either open MCTAs or directed uplink MCTAs for each DEV. For example, if the MCTA Allocation Rate field is set to a value of 8, the PNC is indicating that it will be providing either an open MCTA or a directed uplink MCTA for each DEV in the piconet at least once out of every 8 superframes. A value of 15 means the PNC is not giving any guarantees about when it will allocate MCTAs. A value of 0 indicates that the PNC is using only the CAP to provide access to the PNC.

The PNC Address field contains the DEV address of the PNC, as described in 6.1.

The MAC header settings for a Non-secure Beacon frame shall be set and interpreted as described in Table 6-8.

Table 6-8—MAC header settings for a Non-secure Beacon frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-1	Decoded
SEC	0	Decoded
ACK policy	No-ACK value in Table 6-3	May be ignored
Retry	0	May be ignored
More data	As required, 7.8.3	Decoded
DestID	BcstID	Decoded
SrcID	PNCID	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

6.3.1.2 Non-secure Beacon frame for pairnets

The Beacon frame shall be formatted as illustrated in Figure 6-55.

Octets:15	variable	…	variable	4
Pairnet Synchronization Parameters	Information element-1		Information element-n	FCS

Figure 6-55—Non-secure Beacon frame Frame Payload field format for pairnets

The individual information elements (IEs) in the Beacon frame body are listed in Table 6-24. These IEs are defined in 6.4. The IEs in the Beacon frame’s payload may appear in any order. A Beacon frame sent by a PRC shall contain a PRC Capability IE.

The Pairnet Synchronization Parameters field shall be formatted as illustrated in Figure 6-56.

Octets:1	1	2	2	1	1	1	6
Number of Association Slots	Duration of an Association Slot	Superframe Duration	Recommended ATP	Next DEVID	Pairnet Mode	Expected RSSI	PRC address

Figure 6-56—Pairnet Synchronization Parameters field format

Number of Association Slots field indicates the number of slots available for the DEVs to send Association Request commands. This value is the same as *pNAccessSlot*.

Duration of an Association Slot field indicates the time length of a slot. This value is the same as *pDAccessSlot*.

The Superframe Duration field contains the duration of the current superframe in the PSP. The resolution of this field is 1 µs and has a range of [0–65535] µs. However, the valid range of this field lies between *mMinSuperframeDuration* and *mMaxSuperframeDuration*. Note that the superframe duration may be longer than *pNAccessSlot*pDAccessSlot* when dual Beacon frames are transmitted.

The Recommended ATP field indicates ATP length value that is recommended by PRC. The resolution of this field is 1 ms and therefore has a range of [0–65535] ms.

NOTE 1—It is recommended that a PRDEV should use short ATP length value less than or equal to 500 ms.

The Next DEVID field indicates the DEVID for the subsequent DEV. The DEV that wishes to associate with the PRC shall set this value as its own DEVID. The value of the Next DEVID field shall be different from the current DEVID and selected randomly.

The Expected RSSI field indicates the RSSI value of received signal at the antenna input of DEV located at specific distance from the PRC. Equivalent isotropic radiated power (EIRP) and path loss between the PRC and the PRDEV can be used to determine the Expected RSSI value at the PRC. The DEV shall only send an Association Request command to the PRC when the actual measured RSSI level of the received Beacon frame exceeds this value. The resolution of this field is 1 dB and has a range of [+30 to –226] dBm.

NOTE 2—Expected RSSI can be calculated at the PRC as follows: Expected RSSI = EIRP – Path Loss [this value is equal to: TX Power at RF of the PRC + (antenna gain – cable loss at the PRC) – Path Loss].

When the DEV measures the RSSI level of the received Beacon frame, antenna gain and cable loss at the DEV should be considered in the decision on sending an Association Request. That is, the DEV transmits an Association Request when the following condition holds: Measured RSSI at the DEV ≥ Expected RSSI value indicated in the Beacon frame + (antenna gain – cable loss at the DEV).

The Pairnet Mode field defines certain characteristics about the pairnet and the superframe. The encoding of this octet shall be formatted as illustrated in Figure 6-57.

Bits: b0–b3	b4	b5–b7
Reserved	SEC Mode	Reserved

Figure 6-57—Pairnet Mode field

The SEC Mode field indicates the current security settings in the pairnet as defined in 8.2. The field is encoded as illustrated in Table 6-7.

The PRC Address field contains the DEV address of the PRC, as described in 6.1.

The MAC header settings for a Beacon frame shall be set and interpreted as described in Table 6-9.

Table 6-9—MAC header settings for a Beacon frame for pairnets

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-2	Decoded
SEC	0	Decoded
ACK policy	No-ACK value in Table 6-4	May be ignored

Table 6-9—MAC header settings for a Beacon frame for pairnets (*continued*)

Header field	Setting on transmission	Interpretation on reception
DestID	BcstID	Decoded
SrcID	PRCID	Decoded
TX and ACK Information	0x000000	May be ignored
Stream Index	0x00	May be ignored

6.3.1.3 Secure Beacon frame for piconets

The Secure Beacon frame Secure Payload field shall be formatted as illustrated in Figure 6-58. The Secure Beacon frame format is used when the piconet is operating in a secure mode.

Octets: 2	2	19	variable	…	variable	8	4
SECID	SFC	Piconet synchronization parameters	Information element-1	…	Information element-*n*	Integrity code	FCS

Figure 6-58—Secure Beacon frame Secure Payload field format

The SECID field is defined in 6.2.10.2.

The SFC field is used by the DEV for this frame to ensure uniqueness of the nonce, as defined in 6.2.10.3.

The Piconet Synchronization Parameters field is defined in 6.3.1.1.

The Integrity Code is defined in 6.2.10.5.

The MAC header settings for a Secure Beacon frame shall be set and interpreted as described in Table 6-10.

Table 6-10—MAC header settings for a Secure Beacon frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-1	Decoded
SEC	1	Decoded
ACK policy	No-ACK value in Table 6-3	May be ignored
Retry	0	May be ignored
More data	As required, 7.8.3	Decoded
DestID	BcstID	Decoded
SrcID	PNCID	Decoded
Fragmentation control	0x000000	May be ignored
Stream index	0x00	May be ignored

6.3.1.4 Secure Beacon frame for pairnets

The Secure Beacon frame shall be formatted as illustrated in Figure 6-59. The Secure Beacon frame format is used when the pairnet is operating in a secure mode.

Octets: 2	6	6	15	variable	…	variable	16	4
SECID	SFC	Time Token	Pairnet Synchronization Parameters	Information element-1	…	Information element-*n*	Integrity Code	FCS

Figure 6-59—Secure Beacon frame format for pairnets

The SECID field is defined in 6.2.10.2.

The SFC field is used by the DEV for this frame to ensure the uniqueness of the nonce, as defined in 6.2.10.3.

The Time Token field contains a strictly increasing counter, which shall be incremented in each Beacon frame. The time token counter shall be set to zero by the PRC when the PRC starts a pairnet for the first time. The time token counter value is used as the CurrentTimeToken of the DEV. The beacon number is defined to be the 16 LSBs of the time token.

The Pairnet Synchronization Parameters field is defined in 6.3.1.2.

The Integrity Code is defined in 6.2.10.5.

The MAC header settings for a Secure Beacon frame shall be set and interpreted as described in Table 6-11.

Table 6-11—MAC header settings for a Secure Beacon frame for pairnets

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-2	Decoded
SEC	1	Decoded
ACK policy	No-ACK value in Table 6-4	May be ignored
DestID	BcstID	Decoded
SrcID	PRCID	Decoded
TX and ACK Information	0x000000	May be ignored
Stream Index	0x00	May be ignored

6.3.2 Acknowledgment frames for piconets

6.3.2.1 Imm-ACK frame

The Imm-ACK frame has neither a Frame Payload field nor an FCS field.

The MAC header settings for an Imm-ACK frame shall be set and interpreted as described in Table 6-12.

Table 6-12—MAC header settings of an Imm-ACK frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Imm-ACK value in Table 6-1	Decoded
SEC	0	May be ignored
ACK policy	No-ACK value in Table 6-3	May be ignored
Retry	0	May be ignored
More data	0	May be ignored
DestID	SrcID of the received frame	Decoded
SrcID	DestID of the received frame	Decoded
Fragmentation control	0x000000 or Receive Status value, 6.2.7	May be ignored
Stream index	0x00	May be ignored

6.3.2.2 Delayed ACK (Dly-ACK) frame

The Dly-ACK frame is only used in response to an isochronous stream data frame with the ACK Policy field set to Dly-ACK Request. The Dly-ACK frame Frame Payload field shall be formatted as illustrated in Figure 6-60.

Octets: 1	1	1	2	…	2	4
Max Burst	Max Frames	MPDUs ACKed	MPDU ID Block-1	…	MPDU ID Block-*n*	FCS

Figure 6-60—Dly-ACK frame Frame Payload field format

The Max Burst field indicates the number of frames of *pMaxFrameBodySize* that may be sent in one burst. A burst is the collection of the frames that are pending acknowledgment via a Dly-ACK frame.

The Max Frames field indicates the maximum number of frames, regardless of size, that may be sent before requesting a Dly-ACK from the DEV receiving the frames.

Any burst shall meet the restrictions of both the Max Frames field and the Max Burst field, as described in 7.12.4.

The MAC protocol data units (MPDUs) ACKed field shall contain the number of MPDUs that are being ACKed with this frame.

The MPDU ID Block fields shall be sent in the same order as the data frames were received. The MPDU ID Block field shall be formatted as illustrated in Figure 6-61.

Bits: b0–b8	b9–b15
MSDU number	Fragment number

Figure 6-61—MPDU ID Block field format

The MAC header settings for a Dly-ACK frame shall be set and interpreted as described in Table 6-13.

Table 6-13—MAC header settings of a Dly-ACK frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Dly-ACK value in Table 6-1	Decoded
SEC	0	May be ignored
ACK policy	No-ACK value in Table 6-3	May be ignored
Retry	0	May be ignored
More data	0	May be ignored
DestID	SrcID of the received frame	Decoded
SrcID	DestID of the received frame	Decoded
Fragmentation control	0x000000 or Receive Status value, 6.2.7	May be ignored
Stream index	0x00	May be ignored

6.3.3 Command frame for piconets

6.3.3.1 General

When sending command frames the ACK Policy field in the Frame Control field shall be set to either Imm-ACK or no-ACK.

Only certain commands may be fragmented, as indicated in 6.5. For commands that are not allowed to be fragmented, the Fragmentation Control field shall be set to 0x000000.

6.3.3.2 Non-secure Command frame

The Non-secure Command frame Frame Payload field shall be formatted as shown in Figure 6-62. The command types are described in 6.5.

Octets: variable	4
Command block	FCS

Figure 6-62—Non-secure Command frame Frame Payload field format

The Command Block field shall be formatted as shown in Figure 6-63.

Octets: 2	2	variable
Command Type	Length	Payload

Figure 6-63—Command Block field format

The Command Type field indicates the type of command and is defined in Table 6-47.

The Length field contains the length of the Payload field in octets.

The Payload field contains information specific to the MAC command. The Payload field for each of the MAC commands are defined in 6.5.

The MAC header settings for a Non-secure Command frame shall be set and interpreted as described in Table 6-14.

Table 6-14—MAC header settings of a Non-secure Command frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-1	Decoded
SEC	0	Decoded
ACK policy	As required for command protocol	Decoded
Retry	As appropriate	Decoded
More data	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
Stream index	0x00 or 0xFD	May be ignored

6.3.3.3 Secure Command frame

The Secure Command frame Secure Payload field shall be formatted as illustrated in Figure 6-64.

Octets: 2	2	variable	8	4
SECID	SFC	Command Block	Integrity Code	FCS

Figure 6-64—Secure Command frame Secure Payload field format

The SECID field is defined in 6.2.10.2.

The SFC field is defined in 6.2.10.3.

The Command Block field shall be formatted as illustrated in Figure 6-63.

The Integrity Code field is defined in 6.2.10.5.

This frame format is used when the piconet is operating in a secure mode.

The MAC header settings for a Secure Command frame shall be set and interpreted, as described in Table 6-15.

Table 6-15—MAC header settings of a Secure Command frame for piconet

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-1	Decoded
SEC	1	Decoded
ACK policy	As required for command protocol	Decoded
Retry	As appropriate	Decoded
More data	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
Stream index	0x00 or 0xFD	May be ignored

6.3.4 Command frame for pairnets

6.3.4.1 Non-secure Command frame

The Non-secure command frame format shall be structured as illustrated in Figure 6-65.

Octets: 4	variable	4
MAC Subheader	Command Block	FCS

Figure 6-65—Non-secure Command frame format for pairnets

The MAC Subheader field is defined in 6.3.6.1.

The Command Block field shall be formatted as shown in Figure 6-66.

Octets: 2	2	variable
Command Type	Length	Payload

Figure 6-66—Command block format for pairnet

The Command Type field indicates the type of command and is defined in Table 6-48.

The Length field contains the length of the Payload field in octets.

The Payload field contains information specific to the MAC command. The Payload field for each of the MAC commands is defined in 6.5.

The MAC subheader settings for a command frame shall be set and interpreted as described in Table 6-16.

Table 6-16—MAC subheader settings for a non-secure command frame for pairnet

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-2	Decoded
SEC	0	Decoded
ACK policy	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	0x00	May be ignored

6.3.4.2 Secure command frame

The Secure command frame format shall be formatted as illustrated in Figure 6-67. This frame format is used when the pairnet is operating in a secure mode.

Octet: 2	6	4	variable	16	4
SECID	SFC	MAC Subheader	Command Block	Integrity Code	FCS

Figure 6-67—Secure command frame format

The SECID is defined in 6.2.10.2.

The SFC is defined in 6.2.10.3.

The Integrity code field is defined in 6.2.10.5.

The FCS field is defined in 6.2.10.6. The Calculation field shall include SECID field, SFC field, Command Block field, and Integrity Code field.

The command block shall be formatted as illustrated in Figure 6-66.

The MAC subheader settings for a secure command frame shall be set and interpreted as described in Table 6-17.

Table 6-17—MAC subheader settings for a Secure Command frame for pairnet

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-2	Decoded
SEC	1	Decoded
ACK policy	As appropriate	Decoded
DestID	As appropriate	Decoded

Table 6-17—MAC subheader settings for a Secure Command frame for pairnet (*continued*)

Header field	Setting on transmission	Interpretation on reception
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	0x00	May be ignored

6.3.5 Data frame for piconets

6.3.5.1 Non-secure Data frame

The Non-secure Data frame Frame Payload field shall be formatted as shown in Figure 6-68.

Octets: variable	0 or 4
Data Payload	FCS

Figure 6-68—Non-secure Data frame Frame Payload format

The Frame Type field shall be set to the Data frame value in Table 6-1, and the SEC field shall be set to zero. The other fields in the MAC header take on values that are appropriate for that particular Non-secure Data frame. All fields in the MAC header of a Non-secure Data frame shall be decoded on reception.

Null data frames, which are non-secure data frames with no Data ID field, no Data Header field, and a zero-length Data Payload field, are allowed. For example, a null data frame may be used with Dly-ACK negotiation, as described in 7.12.4.

The FCS is not sent for zero-length frames, as described in 6.2.10.6.

6.3.5.2 Secure Data frame

The Secure Data frame Secure Payload field shall be formatted as illustrated in Figure 6-69.

Octets: 2	2	variable	8	4
SECID	SFC	Data Payload	Integrity Code	FCS

Figure 6-69—Secure Data frame Secure Payload field format

The Frame Type field shall be set to the Data frame value in Table 6-1, and the SEC field shall be set to one. The other fields in the MAC header take on values that are appropriate for that particular Secure Data frame. All fields in the MAC header of a Secure Multi-protocol Data frame shall be decoded on reception.

The SECID field is defined in 6.2.10.2.

The SFC field is defined in 6.2.10.3.

The Data ID field is defined in 6.3.5.1.

The Data Header field is defined in 6.3.5.1.

6.3.6 Data frame for pairnets

6.3.6.1 Non-Secure Pairnet Aggregated Data frame

Figure 6-70 illustrates the Frame Payload field format of the Non-Secure Pairnet Aggregated Data frame.

Octets: 4	**variable**	**…**	**4**	**variable**
MAC Subheader 1	MAC Subframe Body 1	…	MAC Subheader *n*	MAC Subframe body *n*

Figure 6-70—Frame Payload field format for Non-Secure Pairnet Aggregated Data frame

The MAC Subheader field shall be formatted as illustrated in Figure 6-71.

Octets: 3	**1**
Subframe Information	Subheader HCS

Figure 6-71—MAC Subheader field format

The Subframe Information field shall be formatted as illustrated in Figure 6-72.

Bits: b0	**b1–b10**	**b11–b23**
Last Fragment	Sequence Number	Payload length

Figure 6-72—Subframe Information field format

The Last Fragment field shall be set to one if the subframe contains the payload that is the last fragment of the MSDU and shall be set to zero otherwise.

The Sequence Number field indicates the sequence number of this subframe. The initial value of both command and data sequence numbers is 0x000.

The Payload Length field is used to determine the length of the payload before coding, not including the FCS and padding octets. This field contains the value of one less than the actual length of the payload in octets. For example, a value of zero in the Payload length field indicates a payload of one octet.

The Subheader HCS field is a CRC defined as the one's-complement of the remainder of the division of the 24 bits of the header by the polynomial $x^8 + x^2 + x + 1$. A serial implementation is illustrated in Figure 6-73

.

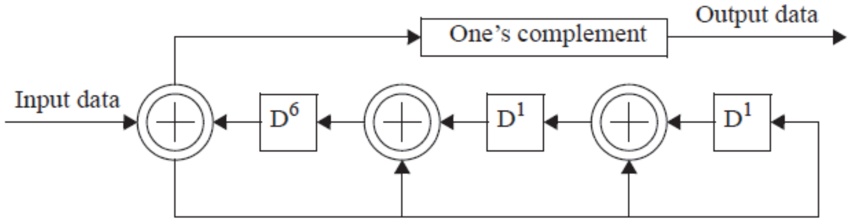

Figure 6-73—HCS implementation example for Subheader HCS

The Subheader HCS is an exception to the data ordering convention in 6.1 and is transmitted with MSB first.

The maximum number of subframes that are aggregated in one frame shall be *mMaxSubframeSize*, as defined in 7.20.

The format of the MAC Subframe Body field for aggregation shall be formatted as illustrated in Figure 6-74.

Octets: variable	4	variable
Payload	FCS	Padding

Figure 6-74—Format of the MAC Subframe Body field for aggregation

The Payload field is a variable length field that carries the information that is to be transferred to a DEV.

The FCS field is defined in 6.2.10.6.

The Padding is defined in 7.11.3.

The MAC header settings for a Non-Secure Pairnet Aggregated Data frame shall be set and interpreted as described in Table 6-18.

Table 6-18—MAC header settings for a Non-Secure Pairnet Aggregated Data frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Data value in Table 6-2	Decoded
SEC	0	Decoded
ACK policy	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	0x00	May be ignored

6.3.6.2 Secure Pairnet Aggregated Data frame

Figure 6-75 and Figure 6-76 illustrate the Secure Pairnet Aggregated Data frame.

Octets: 2	6	variable
SECID	SFC	Secure Pairnet Aggregated Data Frame Payload

Figure 6-75—Frame Payload field format for Secure Aggregated Data Frame

The SECID is defined in 6.2.10.2.

The SFC is defined in 6.2.10.3. Only the SFC of the first subframe is included in the Secure Pairnet Aggregated Data frame.

The Secure Pairnet Aggregated Data Frame Payload field shall be formatted as illustrated in Figure 6-76.

<table>
<tr><th>Octets: 4</th><th>variable</th><th>16</th><th>4</th><th>vari-able</th><th>…</th><th>4</th><th>variable</th><th>16</th><th>4</th><th>var-iable</th></tr>
<tr><td rowspan="2">MAC Sub-header 1</td><td>Secure Payload 1</td><td>Integrity Code</td><td>FCS</td><td>Pad-ding</td><td></td><td rowspan="2">MAC Sub-header n</td><td>Secure Payload n</td><td>Integrity Code</td><td>FCS</td><td>Pad-ding</td></tr>
<tr><td colspan="4">Secure MAC Subframe Body 1</td><td></td><td colspan="4">Secure MAC Subframe Body n</td></tr>
</table>

Figure 6-76—Secure Pairnet Aggregated Data Frame Payload field format

The MAC Subheader field is defined in 6.3.6.1. The Payload Length field in the MAC Subheader includes the length of the secure payload, not including the Integrity Code, FCS and padding octets.

The Secure Payload field is a variable-length field that contains the information, protected by the symmetric key security operations as defined in Clause 10, that is to be transferred to a DEV.

The Integrity Code field is defined in 6.2.10.5.

The Padding is defined in 7.11.3.

The FCS field is defined in 6.2.10.6. In the first subframe, the Calculation field shall include the SECID field, SFC field, Payload 1 field, and Integrity Code field. In the second and subsequent subframes, the Calculation field shall include the Payload *n* field and Integrity Code field.

The MAC header settings for a Secure Pairnet Aggregated Data frame shall be set and interpreted as described in Table 6-19.

Table 6-19—MAC header settings for a Secure Pairnet Aggregated Data frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Data value in Table 6-2	Decoded
SEC	1	Decoded
ACK policy	As appropriate	Decoded

Table 6-19—MAC header settings for a Secure Pairnet Aggregated Data frame (*continued*)

Header field	Setting on transmission	Interpretation on reception
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	0x00	May be ignored

6.3.7 Multi-protocol Data frame for piconets

6.3.7.1 Non-secure Multi-protocol Data frame

The Non-secure Multi-protocol Data frame is used to support multiple protocols simultaneously operating in the piconet. Frame Payload field shall be formatted as shown in Figure 6-77.

Octets: 1	variable	variable	0 or 4
Data ID	Data Header	Data Payload	FCS

Figure 6-77—Non-secure Multi-protocol Data frame Frame Payload format

The Frame Type field shall be set to the Multi-protocol Data frame value in Table 6-1, and the SEC field shall be set to zero. The other fields in the MAC header take on values that are appropriate for that particular data frame. All fields in the MAC header of a Non-secure Multi-protocol Data frame shall be decoded on reception.

The Data ID field indicates the contents of the Data Header field. If the Data Payload field is zero length, i.e., a null data frame, the Data ID field and Data Header field are omitted. Valid values of the Data ID field are given in Table 6-20.

Table 6-20—Data ID values

Data ID field value	Data Header field size	Data Header field contents
0x00	As defined in Figure 6-78	As defined in Figure 6-78
0x01	3 octets	OUI or CID as defined in 6.4.8
0x02	2 octets	An EtherType
0x03–0xFF	—	Reserved

The Data ID value of 0x00 is used to support EtherType protocol discrimination (EPD), as defined in IEEE Std 802. If the Data ID field is set to 0x00, then the Data Header field shall be formatted as illustrated in Figure 6-78.

Octets: 6	6	2
Destination address	Source address	EtherType

Figure 6-78—Data Header field format when Data ID field is 0x00

The Destination Address field and Source Address field contain the destination and source MAC addresses for the data payload. Note that these addresses do not necessarily correspond to the DestID and SrcID of the frame in a bridged network.

The EtherType field contains an EtherType, as defined in IEEE Std 802.

NOTE—Additional fields related to the EtherType will follow the Data Header field in the Data Payload field based on the value of the EtherType, e.g., virtual LAN tags.

If the Data ID field is set 0x01, then the format of the remainder of the Data Payload field is determined by the entity that has been assigned the OUI or CID.

If the Data ID field is set to 0x02, then the format of the remainder of the Data Payload field is determined by the value of the EtherType.

6.3.7.2 Secure Multi-protocol Data frame

The Secure Multi-protocol Data frame Secure Payload field shall be formatted as illustrated in Figure 6-79.

Octets: 2	2	1	variable	variable	8	4
SECID	SFC	Data ID	Data Header	Data Payload	Integrity Code	FCS

Figure 6-79—Secure Multi-protocol Data frame Secure Payload field format

The Frame Type field shall be set to the Multi-protocol Data frame value in Table 6-1, and the SEC field shall be set to one. The other fields in the MAC header take on values that are appropriate for that particular Secure Multi-protocol Data frame. All fields in the MAC header of a Secure Data frame shall be decoded on reception.

The SECID field is defined in 6.2.10.2.

The SFC field is defined in 6.2.10.3.

The Data ID field is defined in 6.3.7.1.

The Data Header field is defined in 6.3.7.1.

The Integrity Code field is defined in 6.2.10.5.

6.3.8 Multi-protocol Data frame for pairnets

6.3.8.1 Non-Secure Pairnet Aggregated Multi-protocol Data frame

The Non-Secure Pairnet Aggregated Multi-protocol Data frame uses the same frame format as Figure 6-74 but the Payload field is replaced by the Pairnet Multi-protocol Data Payload format illustrated in Figure 6-80.

Octets: 1	variable	variable
Data ID	Data Header	Data Payload

Figure 6-80—Pairnet Multi-protocol Data Payload format

The Frame Type field shall be set to the Multi-protocol Data frame value in Table 6-2, and the SEC field shall be set to zero. The MAC header settings for a Non-Secure Pairnet Multi-protocol Data frame shall be set and interpreted as described in Table 6-21.

Table 6-21—MAC header settings for a Non-Secure Pairnet Aggregated Multi-protocol Data frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Data value in Table 6-2	Decoded
SEC	0	Decoded
ACK policy	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	As appropriate	Decoded

The Data ID field and the Data Header field are defined in 6.3.7.1.

6.3.8.2 Secure Pairnet Aggregated Multi-protocol Data frame

The Secure Pairnet Aggregated Multi-protocol Data frame uses the same frame format as Figure 6-76 but the Payload 1 through *n* fields are replaced by the Pairnet Multi-protocol Data Payload format illustrated in Figure 6-80. Figure 6-81 illustrates the Secure Pairnet Aggregated Multi-protocol Data Frame Payload field format.

<table>
<tr><th>Octets: 4</th><th>1</th><th>variable</th><th>variable</th><th>16</th><th>4</th><th>variable</th><th>…</th><th>4</th><th>1</th><th>variable</th><th>variable</th><th>16</th><th>4</th><th>variable</th></tr>
<tr><td rowspan="2">MAC Subheader 1</td><td>Data ID</td><td>Data Header</td><td>Data Payload 1</td><td>Integrity Code</td><td>FCS</td><td>Padding</td><td></td><td rowspan="2">MAC Subheader n</td><td>Data ID</td><td>Data Header</td><td>Data Payload n</td><td>Integrity Code</td><td>FCS</td><td>Padding</td></tr>
<tr><td colspan="6">Secure MAC Subframe Body 1</td><td></td><td colspan="6">Secure MAC Subframe Body n</td></tr>
</table>

Figure 6-81—Secure Pairnet Aggregated Multi-protocol Data Frame Payload field format

The Frame Type field shall be set to the Secure Pairnet Aggregated Multi-protocol Data frame value in Table 6-2 and the SEC field shall be set to one. The MAC header settings for a Secure Pairnet Aggregated Multi-protocol Data frame shall be set and interpreted as described in Table 6-22.

Table 6-22—MAC header settings for a Secure Pairnet Aggregated Multi-protocol Data frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Data value in Table 6-2	Decoded
SEC	1	Decoded
ACK policy	As appropriate	Decoded
DestID	As appropriate	Decoded
SrcID	As appropriate	Decoded
TX and ACK Information	As appropriate	Decoded
Stream Index	As appropriate	Decoded

The SECID field is defined in 6.2.10.2.

The SFC field is defined in 6.2.10.3.

The MAC Subheader field is defined in 6.3.6.1. The Payload Length field in the MAC Subheader includes the sum of the lengths of the Data ID field, Data Header field, and Data Payload field, not including the Integrity Code, FCS and padding octets.

The Data ID field and the Data Header field are defined in 6.3.7.1.

The Data Payload field is a variable-length field that contains the information that is to be transferred to a DEV. The Data ID field, Data Header field, and Data Payload field are encrypted by the symmetric key security operations as defined in Clause 10.

The Integrity Code field is defined in 6.2.10.5.

The FCS field is defined in 6.2.10.6. In the first subframe, the Calculation field shall include the SECID field, SFC field, DataID field, Data Header field, Data Payload 1 field, and Integrity Code field. In the second and subsequent subframes, the Calculation field shall include the Data ID field, Data Header field, Payload *n* field, and Integrity Code field.

The Padding is defined in 7.11.3.

6.3.9 Synchronization frame for piconets

The Sync frame Frame Payload field shall be formatted as illustrated in Figure 6-82.

Octets: 4	4	…	4	4
Synchronization Parameters	CTA Block-1	…	CTA Block-*n*	FCS

Figure 6-82—Sync frame format

The Synchronization Parameters field shall be formatted as illustrated in Figure 6-83.

Octets: 2	2
Superframe Duration	Frame Start Time

Figure 6-83—Synchronization Parameters field format

The Superframe Duration field indicates the duration of the current superframe, as described in 6.3.1.1.

The Frame Start Time field indicates the time stamp for the Sync frame, which is the start time of the preamble of a Sync frame, measured from the start of the superframe, in units of microseconds.

The CTA Block field shall be formatted as illustrated in Figure 6-84.

Octets: 2	2
CTA Location	CTA Duration

Figure 6-84—CTA Block field format

The CTA Location field indicates the start time of the allocation, measured from the start of the superframe, as described in 6.3.1.1.

The CTA Duration field specifies the duration of the CTA, as described in 6.3.1.1.

6.4 Information elements (IEs)

6.4.1 Overview

The IEs for piconets are listed in Table 6-23. The required column indicates the type of DEVs that are required to be able to support the IE.

Table 6-23—IEs for piconet

Element ID hex value	Element	Subclause	Present in beacon	Required
0x00	CTA IE for piconet	6.4.2	As needed	All DEVs
0x01	BSID IE for piconet	6.4.3	In every beacon	All DEVs
0x02	Parent Piconet IE	6.4.4	As needed	Optional
0x03	DEV Association IE for piconet	6.4.5	As needed	All DEVs
0x04	PNC Shutdown IE for piconet	6.4.6	As needed	Optional
0x05	Piconet Parameter Change IE	6.4.7	As needed	All DEVs
0x06	AS IE for piconet	6.4.8	As needed	Optional
0x07	Pending channel time map (PCTM) IE for piconet	6.4.9	As needed	All DEVs
0x08	PNC Handover IE for piconet	6.4.10	As needed	All DEVs
0x09	CTA Status IE for piconet	6.4.11	As needed	All DEVs
0x0A	Capability IE for piconet	6.4.12	Non-beacon IE	All DEVs
0x0B	Transmit Power Parameters IE	6.4.19	Non-beacon IE	All DEVs
0x0C	PS Status IE for piconet	6.4.20	As needed	All DEVs
0x0D	CWB IE for piconet	6.4.21	As needed	All DEVs
0x0E	Overlapping PNID IE for piconet	6.4.22	Non-beacon IE	All DEVs
0x0F	Piconet Services IE	6.4.23	Non-beacon IE	Optional
0x10	Group ID IE for piconet	6.4.25	Non-beacon IE	Optional
0x11	Stream Renew IE for piconet	6.4.26	Non-beacon IE	All DEVs
0x12	Next PNC IE for piconet	6.4.27	As needed	All DEVs
0x13	Piconet Channel Status IE	6.4.28	Non-beacon IE	Optional
0x14	Synchronization IE for piconet	6.4.29	As needed	mmWave DEVs
0x15	TSD IE for piconet	6.4.30	Non-beacon IE	mmWave DEVs
0x16	UEP Specific IE for piconet	6.4.31	Non-beacon IE	Optional
0x17	IFS IE for piconet	6.4.32	As needed	SC PHY DEVs
0x18	CTA Relinquish Duration IE for piconet	6.4.33	Non-beacon IE	Optional
0x19	Feedback IE for piconet	6.4.34	Non-beacon IE	mmWave DEVs
0x1A	Mapping IE for piconet	6.4.35	Non-beacon IE	mmWave DEVs
0x1B	BST Clustering IE for piconet	6.4.36	Non-beacon IE	mmWave DEVs
0x1C	PET Clustering IE for piconet	6.4.37	Non-beacon IE	mmWave DEVs
0x1D	Beam PET IE for piconet	6.4.38	Non-beacon IE	mmWave DEVs

Table 6-23—IEs for piconet (*continued*)

Element ID hex value	Element	Subclause	Present in beacon	Required
0x1E	HRS Beam PET IE for piconet	6.4.39	Non-beacon IE	Optional
0x1F	PET Amplitude IE for piconet	6.4.40	Non-beacon IE	Optional
0x20	PET Phase IE for piconet	6.4.41	Non-beacon IE	Optional
0x21	Sync Frame Frequency IE for piconet	6.4.42	Non-beacon IE	Optional
0x22	Directional Peer IE for piconet	6.4.43	As needed	Optional
0x23–0x7F	Reserved			
0x80–0xFF	Vendor Defined IE	6.4.24	As needed	Optional

The IEs for PRDEVs are listed in Table 6-24.

Table 6-24—IEs for pairnet

Element ID hex value	Element	Subclause	Present in Beacon frame
0x00	Reserved	—	—
0x01	BSID	6.4.3	In every Beacon frame
0x02–0x09	Reserved	—	—
0x0a	PRC Capability	6.4.13	In every Beacon frame
0x0b–0x22	Reserved	—	—
0x23	MIMO Information	6.4.44	As needed
0x24	PRDEV Capability	6.4.14	Non-Beacon frame IE
0x25	Pairnet Operation Parameters	6.4.15	Non-Beacon frame IE
0x26	Higher Layer Protocol Information	6.4.45	As needed
0x27–0x7F	Reserved	—	—
0x80–0xFF	Vendor Defined IE	6.4.24	As needed

The requirements for supporting an IE are listed in Table 6-25 for HRCP PHY DEVs and THz PHY DEVs. IEs not shown in the table are not used by either of the aforementioned PHYs.

Table 6-25—Requirements for supporting an IE

Element	HRCP PHY	THz PHY
BSID IE	Mandatory	Mandatory
PRC Capability IE	Mandatory	Mandatory

Table 6-25—Requirements for supporting an IE (*continued*)

Element	HRCP PHY	THz PHY
MIMO Information IE	Optional	Not used
PRDEV Capability IE	Mandatory	Mandatory
Pairnet Operation Parameters IE	Mandatory	Mandatory
Higher Layer Protocol Information IE	Optional	Optional
Vendor Defined IE	Optional	Optional

The format of an individual IE is shown in Figure 6-85. Unless otherwise specified, these elements may appear in any order in the frames that are allowed to include more than one of these elements.

Octets: 1	1	variable
Element ID	Length	Content

Figure 6-85—IE format

The Element ID field contains the value from either Table 6-23 or Table 6-24, depending on whether a piconet or pairnet is implemented, that identifies the IE.

The Length field contains the length of the Content field in octets.

The Content field contains information specific to the IE.

6.4.2 CTA IE for piconet

The CTA IE Content field shall be formatted as illustrated in Figure 6-86. Because the length parameter supports only 255 octets of payload in an IE, the PNC may split the CTA information into more than one CTA IE entry in the beacon. The CTA blocks shall be ordered by increasing value of the CTA location with the highest value being the last.

Octets: 7	7	…	7
CTA Block-1	CTA Block-2	…	CTA Block-*n*

Figure 6-86—CTA IE Content field format

The CTA blocks shall be formatted as illustrated in Figure 6-87.

Octets: 1	1	1	2	2
DestID	SrcID	Stream Index	CTA Location	CTA Duration

Figure 6-87—CTA block

The DestID indicates the DEV to which the source DEV may send the frames.

The SrcID indicates the DEV to which the channel time is being allocated.

If the CTA is for a child piconet, the DestID and SrcID shall both be the DEVID of the DEV that is the child piconet's PNC.

If the CTA is for a neighbor piconet, the DestID and SrcID shall both be the DEVID assigned by the PNC for the neighbor piconet and shall be one of the reserved neighbor piconet IDs, as described in 6.2.5.

The Stream Index field, as described in 6.2.8, indicates the stream corresponding to the channel time allocation.

The CTA Location field indicates the start time of the allocation in microseconds measured from the start of the beacon, as described in 7.8.

The Duration field specifies the duration of the CTA in microseconds.

The end time of each allocation is the start time contained in the CTA Location field plus the CTA duration.

6.4.3 BSID IE for piconet

The BSID IE is used to provide a text string to identify the piconet. The BSID IE Content field shall be formatted as illustrated in Figure 6-88.

Octets: 6–32
Piconet BSID

Figure 6-88—BSID IE Content field format

The Piconet BSID field is a set of ISO/IEC 646 encoded characters that is used to identify the piconet. The setting of the piconet BSID is described in 7.14.4. The BSID field shall not be null terminated. The length of the field is determined from the Length field of the IE.

6.4.4 Parent Piconet IE

The Parent Piconet IE is used to provide a text string to identify the parent piconet and the DEV address of the parent PNC. The Parent Piconet IE Content field shall be formatted as illustrated in Figure 6-89.

Octets: 6	**6–32**
Parent PNC Address	Parent Piconet BSID

Figure 6-89—Parent Piconet IE Content field format

The Parent PNC Address field contains the DEV address, as described in 6.1, of the parent PNC for the piconet.

The Parent Piconet BSID field contains the piconet BSID from the BSID IE, as described in 6.4.3, in the parent PNC's beacon.

6.4.5 DEV Association IE for piconet

The DEV Association IE is used to notify current members in the piconet about one or more DEVs that have either just associated or disassociated from the piconet. The DEV Association IE Content field shall be formatted as illustrated in Figure 6-90.

Octets: 11	**11**	…	**11**
DEV-1 Association Info	DEV-2 Association Info	…	DEV-*n* Association Info

Figure 6-90—DEV Association IE Content field format

The DEV Association Info fields shall be formatted as illustrated in Figure 6-91.

Octets: 6	**1**	**1**	**3**
DEV Address	DEVID	DEV status	DEV Capabilities

Figure 6-91—DEV Association Info fields

The DEV Address field contains the address of the DEV, as described in 6.1, that corresponds to the DEVID.

The DEVID is the identifier assigned by the PNC to a DEV.

The DEV Status field shall be formatted as illustrated in Figure 6-92.

Bits: b0	**b1–b7**
Association Status	Reserved

Figure 6-92—DEV Status field format

The Association Status field shall be encoded as follows:

0 → Disassociated
1 → Associated

The DEV Capabilities field is defined in 6.4.12.

6.4.6 PNC Shutdown IE for piconet

The PNC Shutdown IE is used to indicate that the PNC is shutting down. The PNC Shutdown IE Content field shall be formatted as illustrated in Figure 6-93.

Octets: 1
Remaining DEVID

Figure 6-93—PNC Shutdown IE Content field format

The Remaining DEVID field indicates which dependent piconet PNC is allowed to continue operation, as described in 7.2.10.2. It shall be set to the PNCID if there are no dependent piconets in the current piconet.

6.4.7 Piconet Parameter Change IE

The Piconet Parameter Change IE Content field shall be formatted as illustrated in Figure 6-94.

Octets: 1	2	1	2	2	6–32
Change Type	Change Beacon Number	New Channel Index	Superframe Timing	PNID	BSID

Figure 6-94—Piconet Parameter Change IE Content field format

The Change Type field shall be set as given in Table 6-26.

The New Channel Index, Superframe Timing, PNID and BSID fields are defined in Table 6-26.

Table 6-26—Description of field contents for change type values

Change Type field value	Interpretation	Field to decode	Description of field contents
0	PNID	PNID	The new PNID, 6.2.3, that will take effect beginning with the superframe that has the beacon number equal to the Change Beacon Number field.
1	BSID	BSID	The new BSID, 6.4.3, that will take effect beginning with the superframe that has the beacon number equal to the Change Beacon Number field.
2	MOVE	Superframe timing	The offset in microseconds between the beacon’s expected transmission time and the time that it will be sent by the PNC, 7.14.2. The change occurs with the beacon that has the beacon number equal to the Change Beacon Number field.
3	SIZE	Superframe timing	The new superframe duration, 7.14.3, that will be used for the superframe that has the beacon number equal to the Change Beacon Number field.
4	CHANNEL	New channel index	The channel index of the PHY channel that the piconet will begin using the beacon that has the beacon number equal to the Change Beacon Number field. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY, the mapping is defined in 11.2.3. For the SC PHY and HSI PHY, the mapping is defined in 12.1.6. For the AV PHY, the mapping is defined in 12.4.2.1.
5–255	Reserved	None	

The Change Beacon Number field is the beacon number of the superframe when the change will take effect. The difference between the beacon number of the beacon, which first includes this IE and the Change Beacon Number field, is defined to be the NbrOfChangeBeacons. For a piconet without pseudo-static CTAs, NbrOfChangeBeacons shall be at least two. For a piconet that has pseudo-static CTAs, NbrOfChangeBeacons shall be at least *mMaxLostBeacons*. For a piconet that has child or neighbor piconets, NbrOfChangeBeacons shall be at least eight. However, a child or neighbor PNC may set the

NbrOfChangeBeacons to a different number based on the Change Beacon Number field in the parent PNC's beacon, as defined in 7.15.2.

6.4.8 AS IE for piconet

The purpose of the application specific (AS) IE is to provide information specific to the application for enhanced operation that is outside of the scope of this standard. The AS IE Content field shall be formatted as illustrated in Figure 6-95.

Octets: 3	variable
Unique ID	Application Specific Data

Figure 6-95—AS IE Content field format

The Unique ID field contains an OUI or CID assigned by the IEEE Registration Authority that refers to an entity that defines the format of the field that follows. If a value of the Unique Id field is not understood by the receiving DEV, that DEV shall ignore the remainder of the associated AS IE.

The Application Specific Data field is provided by the PNC. Its use by an application specific capable DEV is outside of the scope of this standard.

More than one AS IE may be placed in any beacon. The negotiation of the application specific capability between the DEV and the PNC is outside of the scope of this standard.

6.4.9 Pending channel time map (PCTM) IE for piconet

The pending channel time map (PCTM) IE is used to request that a DSPS or APS DEV switch to ACTIVE mode. The PCTM IE Content field shall be formatted as illustrated in Figure 6-96.

Octets: 1	1–32
Start DEVID	PCTM

Figure 6-96—PCTM IE Content field format

The Start DEVID field indicates the DEVID that corresponds to the first bit in the PCTM.

The PCTM field contains a bitmap of 1 to 32 octets in length. Each bit of the PCTM field when set to one indicates the PNC is requesting that the DEV whose DEVID is equal to the start DEVID plus the bit position in the PCTM bitmap listen to the next beacon for a CTA, as described in 7.17.3. The bit position 0, i.e., the first bit or LSB of the bitmap, corresponds to the start DEVID.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs, and the reserved DEVIDs, as described in 6.2.5, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

6.4.10 PNC Handover IE for piconet

The PNC Handover IE is included in the last beacons sent by the old PNC just prior to the old PNC relinquishing control of the piconet. The PNC Handover IE Content field shall be formatted as illustrated in Figure 6-97.

Octets: 6	1	2
New PNC Address	New PNC DEVID	Handover Beacon Number

Figure 6-97—PNC Handover IE Content field format

The New PNC Address field contains the DEV address, 6.1, of the DEV that will be taking over as PNC.

The New PNC DEVID field contains the current DEVID of the DEV that will be taking over as PNC.

The Handover Beacon Number field contains the beacon number of the first beacon that will be sent by the new PNC. The last beacon sent by the old PNC will have a beacon number one less than the Handover Beacon Number field.

6.4.11 CTA Status IE for piconet

The CTA Status IE is used by the PNC to inform the DEVs of certain characteristics of a CTA. The CTA Status IE Content field shall be formatted as illustrated in Figure 6-98.

Octets: 1	1	1	1	2	2
DestID	SrcID	Stream Index	Channel Time Request Info	CTA Sub-Rate	Start Beacon Number

Figure 6-98—CTA Status IE Content field format

The DestID field contains the DEVID of the destination for this CTA.

The SrcID field contains the DEVID of the source for this CTA.

The Stream Index field indicates the stream for which the PNC is providing information, as described in 6.5.7.2.

The Channel Time Request Info field shall be formatted as illustrated in Figure 6-99.

Bits: b0–b2	b3	b4	b5	b6	b7
Priority	Terminate	PM Channel Time Request Type	CTA Type	CTA Rate Type	Reserved

Figure 6-99—Channel Time Request Info field format

The Priority field is defined in B.1.

The Terminate field shall be set to one if the stream has been terminated. Otherwise it shall be set to zero.

The PM Channel Time Request Type field is defined in 6.5.7.2.

The CTA Type field is defined in 6.5.7.2.

The CTA Rate Type field is defined in 6.5.7.2.

The CTA Sub-Rate field is set to the number of beacons between every CTA, as described in 6.5.7.2. If one or more CTAs are allocated per superframe, this value shall be set to zero.

The Start Beacon Number field is set to the beacon number, as described in 6.3.1.1, of the first beacon where the CTA of the new or modified stream will first appear.

6.4.12 Capability IE for piconet

The Capability IE Content field shall be formatted as illustrated in Figure 6-100.

Octets: 7/13
Overall Capabilities

Figure 6-100—Capability IE format

For the 2.4 GHz PHYs, the Overall Capabilities contains the 2.4 GHz PHY Capabilities field. The 2.4 GHz PHY Capability field shall be formatted as illustrated in Figure 6-101.

Octets: 4	**3**
PNC Capabilities	2.4 GHz DEV Capabilities

Figure 6-101—2.4 GHz PHY Capability field format

For mmWave PHYs, the Overall Capabilities field contains the mmWave PHY Capabilities field. The mmWave PHY Capability field shall be formatted as illustrated in Figure 6-102.

Octets: 4	**6**	**3**
PNC Capabilities	mmWave DEV Capabilities	Beam-Forming Capabilities

Figure 6-102—mmWave PHY Capability field format

The PNC Capabilities field shall be formatted as illustrated in Figure 6-103.

Octets: 1	**1**	**1**	**1**
Max Associated DEVs	Max Channel Time Requests	Max TX Power	PNC Rating

Figure 6-103—PNC Capabilities field format

The Max Associated DEVs field indicates the number of associated DEVs this DEV is able to manage if it is PNC-capable and becomes the PNC. Non-PNC-capable DEVs shall set this field to zero.

The Max Channel Time Requests field indicates the number of channel time requests the DEV is capable of handling as a PNC. This field shall be set to zero in a non-PNC-capable DEV.

The Max TX Power Level field indicates the maximum transmit power that is possible for the DEV. The power level is in dBm, encoded in two's complement notation. For example, if a DEV was capable of 14 dBm TX power, the field would take on the value 0x0E; while if the DEV was capable of –4 dBm TX power, the field would take on the value 0xFC.

The PNC Rating field shall be formatted as illustrated in Figure 6-104. Bits b4–b7 are arranged in order of preference for PNC selection, with the highest preference (PNC-capable) corresponding to the MSB.

Bits: b0–b2	b3	b4	b5	b6	b7
Reserved	Next PNC Capable	PSRC	SEC	PNC Des-Mode	PNC Capable

Figure 6-104—PNC Rating field format

The Next PNC Capable field shall be set to one if the DEV is capable of performing the next PNC functionality, as described in 7.2.6. Otherwise, the field shall be set to zero.

The power source (PSRC) field shall be set to one if the DEV is receiving power from the alternating current mains and shall be set to zero otherwise.

The SEC field shall be set to one if the DEV is capable of acting as a key originator (see 8.4). Otherwise, the SEC field shall be set to zero.

The PNC Des-Mode field is the desired mode of the DEV. This field shall be set to one if it is desired that the DEV be the PNC of the piconet and the PNC Capable field is set to one. Otherwise, this field shall be set to zero.

The PNC Capable field shall be set to one if the DEV is capable of being a PNC in the piconet. Otherwise, the PNC Capable field shall be set to zero.

The 2.4 GHz DEV Capabilities field shall be formatted as illustrated in Figure 6-105.

The mmWave DEV Capabilities field shall be formatted as illustrated in Figure 6-106.

The Supported Data Rates or Supported MCSs field is a PHY-dependent mapping that indicates the data rates that the DEV is capable of using. For the 2.4 GHz PHY, this field is the Supported Data Rates field, it shall be formatted as illustrated in Figure 6-105, and the mapping of a field value to a set of data rates is defined in Table 11-21. For the mmWave PHYs, this field is the Supported MCSs field, it shall be formatted as illustrated in Figure 6-106, and the mapping of a field value to a set of MCSs is defined in 12.1.9.1.

The Preferred Fragment Size field is a PHY-dependent mapping that indicates the maximum MAC frame size preferred to be received by the DEV when fragmentation is used. For the 2.4 GHz PHY, this field shall be formatted as illustrated in Figure 6-105, and the mapping of a field value to a preferred fragment size is defined in Table 11-22. For the mmWave PHYs, this field shall be formatted as illustrated in Figure 6-106, and the mapping of a field value to a preferred fragment size is defined in 12.1.9.2.

Bits: b0	b1	b2	b3	b4	b5	b6	7
Supported Data Rates					Preferred Fragment Size		

Bits: b8	b9	b10	b11	b12	b13	b14	b15
Always AWAKE	Listen to Source	Listen to Multicast	Dly-ACK	Imp-ACK	CTA Relinquish	STP	Reserved

Bits: b16	b17	b18	b19	b20	b21	b22	b23
Reserved							

Figure 6-105—2.4 GHz DEV Capabilities field format

The Always AWAKE field shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all CTAs, regardless of the DestID or SrcID. Otherwise, the field shall be set to zero.

The Listen to Source field shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all CTAs where the SrcID is equal to the DEVID of a DEV that is currently the source of a stream to that DEV regardless of the DestID of those CTAs. Otherwise, the field shall be set to zero.

The Listen to Multicast field shall be set to one to indicate the DEV is in ACTIVE mode and that it will listen to all multicast CTAs regardless of the SrcID or the Stream Index. Otherwise, the field shall be set to zero.

The Dly-ACK field shall be set to one if the DEV is capable of performing the Dly-ACK procedure, as defined in 7.12.4. Otherwise, the field shall be set to zero.

The Imp-ACK field shall be set to one indicate that the DEV is capable of performing the Imp-ACK procedure, as defined in 7.12.8. Otherwise, the field shall be set to zero.

The CTA Relinquish field shall be set to one if the DEV is capable of participating in the CTA relinquish procedure, as defined in 7.6.4.9. Otherwise, the field shall be set to zero.

The stream timeout period (STP) field shall be set to one to indicate that the DEV is capable of renewing its streams within the stream timeout period, as defined in 7.4.5. Otherwise, the field shall be set to zero.

The values of the fields in the PNC Capabilities field and the DEV Capabilities field shall not change while a DEV is associated in a piconet.

For the mmWave PHYs, the SC Capable field shall be set to one if the DEV supports the SC PHY, as defined in 12.2. It shall be set to zero otherwise.

The HSI Capable field shall be set to one if the DEV supports the HSI PHY, as defined in 12.3, and shall be set to zero otherwise.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
Supported MCSs							

Bits:b8	b9	b10	b11	b12	b13	b14	b15
Preferred Fragment Size			Always AWAKE	Listen to Source	Listen to Multicast	Dly-ACK	Imp-ACK

Bits: b16	b17	b18	b19	b20	b21	b22	b23
CTA Relinquish	STP	SC Capable	HSI Capable	AV Capable	OOK Capable	DAMI Capable	Blk-ACK

Bits: b24	b25	b26	b27	b28	b29	b30	b31
UEP Capable		UEP Type			HRP RX Capable	HRP TX Capable	TSD Support

Bits: b32	b33	b34	b35	b36	b37	b38	b39
Supported IFS				Supported Aggregation		Pilot Word Capability	PCES Capability

Bits: b40	b41	b42	b43	b44	b45	b46	b47
Sync Frame Capable	Reserved						

Figure 6-106—mmWave DEV Capabilities field format for mmWave PHYs

The AV Capable field shall be set to one if the DEV supports the AV PHY, as defined in 15.4, and shall be set to zero otherwise.

The OOK Capable field shall be set to one if the DEV supports the OOK mode, as defined in Annex G, and shall be set to zero otherwise.

The DAMI Capable field shall be set to one if the DEV supports the DAMI mode, as defined in Annex G, and shall be set to zero otherwise.

The Blk-ACK field shall be set to one if the DEV is capable of performing Blk-ACK, as defined in 7.12.6. Otherwise, the field shall be set to zero.

The UEP Capable field indicates if the DEV supports UEP in the PHY. The valid values of the UEP Capable field are given in Table 6-27.

The UEP Type field indicates the type of UEP that is supported based on the value of the UEP Capable field. The valid values of the UEP Type field are given in Table 6-28.

Table 6-27—UEP Capable field values

Value	UEP support
0	No UEP
1	SC UEP
2	HSI UEP
3	AV UEP

Table 6-28—UEP Type field values

Value	PHY UEP type of mmWave PHYs
0	No UEP support
1	UEP type 1 using different FECs (SC PHY)
2	UEP type 2 using different MCSs (SC PHY)
3	UEP type 3 using different MCSs (All PHYs)
4	UEP type 3 using skewed constellation (All PHYs)
5	UEP type 3 using different MCSs and skewed constellation (SC and HSI PHYs)
6	All UEP types (SC PHY)
7	Reserved

The HRP RX Capable field shall be set to one if the DEV supports HRP RX for the AV PHY, as defined in 12.4, and shall be set to zero otherwise.

The HRP TX Capable field shall be set to one if the DEV supports HRP TX for the AV PHY, as defined in 12.4, and shall be set to zero otherwise.

The TSD Support field shall be set to one if the DEV is capable of TSD, as defined in 15.8; it shall be set to zero otherwise.

The Supported Interframe Spacing (IFS) field indicates the minimum value of the IFS supported by a DEV. The valid values of the field are given in Table 6-29.

Table 6-29—Supported IFS encoding

Field value	SIFS and MIFS duration
0	0.2 µs
1	0.4 µs
2	0.6 µs
3	0.8 µs
4	1.0 µs

Table 6-29—Supported IFS encoding (*continued*)

Field value	SIFS and MIFS duration
5	2.0 µs
6	2.5 µs
7–15	Reserved

The minimum allowed IFS for either SIFS or minimum interframe space (MIFS) may also be constrained by the PHY mode in use. A source DEV may use the shortest SIFS or MIFS supported by the destination DEV. The SIFS and MIFS used may be different.

The Supported Aggregation field indicates the type of aggregation that is supported by the DEV. The valid values for the Supported Aggregation field are as follows:

0 → No aggregation support
1 → Standard aggregation support
2 → Low-latency aggregation support
3 → Standard and low-latency aggregation support

The Pilot Word Capability field shall be set to one if the DEV supports a pilot word of length 8. It shall be set to zero otherwise.

The pilot channel estimation sequence (PCES) Capability field shall be set to one if the DEV supports the use of PCES and shall be set to zero otherwise.

The Sync Frame Capable field shall be set to one if the DEV supports Sync frame transmission, as defined in 7.11, and shall be set to zero otherwise.

The Beam-Forming Capabilities field shall be formatted as illustrated in Figure 6-107.

Bits: b0–b1	b2–b5	b6–b9	b10–b12	b13	b14–b15	b16–b17	b18–b23
LQI Type	Number TX Sectors	Number RX Sectors	Antenna Type	PET	Number TX Quasi-Omni Directions	Number RX Quasi-Omni Directions	Reserved

Figure 6-107—Beam-Forming Capabilities field format

The link quality indication (LQI) Type field indicate the type of LQI used in beam-forming procedure. Valid values of the LQI type fields are as follows:

0 → RSSIR
1 → SNR
2 → SINR
3 → Reserved

The Number TX Sectors field indicates the number of TX sectors supported by the DEV.

The Number RX Sectors field indicates the number of RX sectors supported by the DEV.

The Antenna Type field indicate the supported antenna type of the DEV. Valid values of the Antenna Type field are as follows:

> 0 → No beam-forming capability
> 1 → Beam-forming antenna capable
> 2–7 → Reserved

The pattern estimation and tracking (PET) field shall be set to one if the DEV supports pattern estimation and tracking and shall be set to zero otherwise.

The Number TX Quasi-omni Directions field indicates the number of TX quasi-omni directions supported by the DEV.

The Number RX Quasi-omni Directions field indicates the number of RX quasi-omni directions supported by the DEV.

6.4.13 PRC Capability IE for pairnet

The PRC Capability IE Content field shall be formatted as illustrated in Figure 6-108. The PRC Capability IE shall be included in each Beacon frame.

Octets: 7
PRC Capability

Figure 6-108—PRC Capability IE Content field format

The PRC Capability field shall be formatted as illustrated in Figure 6-109.

The SC Capable field shall be set to one if the DEV supports the SC PHY, as defined in 13.2, and shall be set to zero otherwise.

The OOK Capable field shall be set to one if the DEV supports the OOK PHY, as defined in 13.3, and shall be set to zero otherwise.

The supported SIFS field contains the value of the shortest SIFS supported by the DEV in units of 0.1 µs encoded as an unsigned integer. For example, a value of 0b01001 indicates that the shortest SIFS supported by the DEV is 0.9 µs. Values greater than 2.5 µs are reserved.

The Multi-protocol Support field shall be set to one if the SC PHY DEV supports the Pairnet Multi-protocol Data Frame, as defined in 6.3.8.1 and 6.3.8.2, and shall be set to zero otherwise.

.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
SC Capable	OOK Capable	Supported SIFS					Multi-protocol Support

Bits: b8	b9	b10	b11	b12	b13	b14	b15
LLPS Control							

Bits: b16	b17	b18	b19	b20	b21	b22	b23
Preferred Payload Size		Preferred Total Aggregation Size			Supported Unit of Sub frame Padding		Pilot Symbol Capable

Bits: b24	b25	b26	b27	b28	b29	b30	b31
SC Supported MCS			Reserved	Reserved	Long RIFS Support	SC Supported Channel Bonding	

Bits: b32	b33	b34	b35	b36	b37	b38	b39
SC Supported Channel Bonding							

Bits: b40	b41	b42	b43	b44	b45	b46	b47
SC Channel Aggregation	SC Supported Channel Aggregation Pattern						

Bits: b48	b49	b50	b51	b52	b53	b54	b55
Reserved	Reserved	Reserved	OOK Spreading	OOK Supported Channel Bonding			Reserved

Figure 6-109—PRC Capability field format

The low-latency power save (LLPS) Control field contains LLPS related parameters, as defined in Figure 6-110.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
LLPS Allow	LLPS Interval			LLPS Start		LLPS Extend	

Figure 6-110—LLPS Control field format

The LLPS Allow field shall be set to one if the PRC allows the PRDEVs to use power save mode after association is completed, otherwise it is set to zero.

The LLPS Interval field indicates the value of ACK sending interval when the DEV is in DEV Sleep mode. The field is defined in Table 6-30.

Table 6-30—LLPS Interval field values

Bits: b0	b1	b2	LLPS interval
0	0	0	1 ms
0	0	1	5 ms
0	1	0	10 ms
0	1	1	50 ms
1	0	0	100 ms
1	0	1	Reserved
…	…	...	
1	1	1	

The LLPS Start field indicates the value of consecutive ACKs duration to start LLPS. The valid values of the LLPS Start field are given in Table 6-31.

Table 6-31—LLPS Start field values

Bits: b0	b1	LLPS Interval
0	0	0.1 ms
0	1	1 ms
1	0	10 ms
1	1	Reserved

The LLPS Extend field indicates the value of consecutive ACKs duration to extend LLPS. The valid values of the LLPS Extend field are given in Table 6-32.

Table 6-32—LLPS Extend field values

Bits: b0	b1	LLPS Extend
0	0	0.1 ms
0	1	1 ms
1	0	10 ms
1	1	Reserved

The Preferred Payload Size field indicates the maximum preferred data size of a single subframe payload to be received by the DEV. This field shall be formatted as illustrated in Table 6-33.

Table 6-33—Preferred Payload Size field values

Bits: b0	b1	Preferred Payload Size
0	0	2048 octets
0	1	4096 octets
1	0	8192 octets
1	1	Reserved

The Preferred Total Aggregation Size field, shown in Figure 6-109, indicates the maximum preferred total data size in a single frame to be received by the DEV when fragmentation is used. This field shall be formatted as illustrated in Table 6-34.

Table 6-34—Preferred Total Aggregation Size field values

Bits: b0	b1	b2	Preferred Total Aggregation Size
0	0	0	16448 octets
0	0	1	32896 octets
0	1	0	65792 octets
0	1	1	131584 octets
1	0	0	263168 octets
1	0	1	526336 octets
1	1	0	1050624 octets
1	1	1	2099200 octets

The Supported Unit of Subframe Padding field indicates the unit of the subframe padding that can be received by the DEV as defined in Figure 6-111. Each field shall be set to one for supported capability, and otherwise set to zero.

Bits: b0	b1
64-bit unit of padding supported	12- bit unit of padding supported

Figure 6-111—Supported Unit of Subframe Padding field format

The Pilot Symbol capable field shall be set to one if the DEV is capable of decoding the frame with pilot symbols, and shall be set to zero otherwise.

The SC Supported MCS field shall be formatted as illustrated in Figure 6-112.

Bits: b0	b1	b2
SC 16-QAM supported	SC 64-QAM supported	SC 256-QAM supported

Figure 6-112—SC Supported MCS field format

The SC 16-QAM field shall be set to one if 16-QAM modulation is supported by the SC PHY DEV and shall be set to zero otherwise. The SC 64-QAM field shall be set to one if 64-QAM modulation is supported by the SC PHY DEV and shall be set to zero otherwise. The SC 256-QAM field shall be set to one if 256-QAM modulation is supported by the SC PHY DEV and shall be set to zero otherwise.

The Long RIFS Support field shall be set to one if the DEV supports the longer RIFS duration defined in Table 13-20. It shall be set to zero otherwise.

The SC Supported Channel Bonding field indicates the bonded channels supported by the SC PHY DEV. The SC Supported Channel bonding field shall be formatted as illustrated in Figure 6-113.

Bits: b0	b1	b2	b3	b4	b5	b6	b7	b8	b9
CHNL_ID 7 is supported	CHNL_ID 8 is supported	CHNL_ID 9 is supported	CHNL_ID 10 is supported	CHNL_ID 11 is supported	CHNL_ID 12 is supported	CHNL_ID 13 is supported	CHNL_ID 14 is supported	CHNL_ID 15 is supported	CHNL_ID 16 is supported

Figure 6-113—SC Supported Channel Bonding field format

The SC Channel Aggregation field only applies when the CHNL_ID is between 7 and 16 inclusive. In that case, the field shall be set to one if channel aggregation is supported or zero if channel bonding is supported.

The SC Supported Channel Aggregation pattern field indicates the supported combinations of CHNL_IDs used for channel aggregation by the SC PHY DEV. The SC Supported Channel Aggregation field shall be formatted as illustrated in the Figure 6-114 and Table 6-35. Each field shall be set to one for supported combinations, and shall be set to zero otherwise. Hence, if all bits set to be zero, the SC PHY DEV does not support any channel aggregation pattern. Check mark in the Table 6-35 means the allowable aggregation channel for each pattern.

Bits: b0	b1	b2	b3	b4	b5	b6
pattern0	pattern1	pattern2	pattern3	pattern4	pattern5	pattern6

Figure 6-114—SC Supported Channel Aggregation pattern field format

Table 6-35—Channel Aggregation Patterns for SC PHY in Figure 6-114

CHNL_ID	2	4	6	7	8	10	11	12	14
pattern0	✓	✓							
pattern1	✓	✓	✓						
pattern2				✓	✓				
pattern3					✓	✓			
pattern4				✓		✓			
pattern5					✓		✓		
pattern6								✓	✓

The OOK spreading field shall be set to one if spreading factor 2 is supported by the HRCP-OOK PHY DEV and shall be set to zero if spreading is not supported by the DEV.

The OOK Supported Channel Bonding field indicates the number of bonded channels supported by the OOK-PHY DEV. The Supported OOK Channel bonding field shall be formatted as illustrated in Figure 6-115.

Bits: b0	b1	b2
OOK 2 channel bonding is supported	OOK 3 channel bonding is supported	OOK 4 channel bonding is supported

Figure 6-115—OOK Supported Channel Bonding field format

The CHNL_ID used for OOK channel bonding is specified in 13.3.2.2.

6.4.14 PRDEV Capability IE for pairnet

The PRDEV Capability IE Content field shall be formatted as illustrated in Figure 6-116. The PRDEV Capability shall be included in each association request command frame.

Octets: 7
PRDEV Capability

Figure 6-116—PRDEV Capability IE Content field format

The PRDEV Capability field shall be formatted as illustrated in Figure 6-109, with the LLPS control bits changed to reserved.

The SC Capable field is defined in 6.4.13.

The OOK capable field is defined in 6.4.13.

The Supported SIFS field is defined in 6.4.13.

The Multi-protocol Support field is defined in 6.4.13.

The Preferred Payload Size field is defined in 6.4.13.

The Preferred Total Aggregation Size field is defined in 6.4.13.

The Supported Unit of Subframe Padding field is defined in 6.4.13.

The Pilot Symbol Capable field is defined in 6.4.13.

The SC Supported MCS field is defined in 6.4.13.

The Long RIFS Support field is defined in 6.4.13.

The SC Supported Channel Bonding Capability field is defined in 6.4.13.

The SC Supported Channel Aggregation Pattern field is defined in 6.4.13.

The OOK Spreading field is defined in 6.4.13.

The OOK Supported Channel Bonding field is defined in 6.4.13.

6.4.15 Pairnet Operation Parameters IE

The Pairnet Operation Parameters IE Content field shall be formatted as illustrated in Figure 6-117. The Pairnet Operation Parameters IE shall be included in each Association Response command frame. This capability indicates the communication parameter to be used in the current session that was decided by the PRC to satisfy both PRC and PRDEV capabilities.

Octets: 7
Operation Parameters

Figure 6-117—Pairnet Operation Parameters IE Content field format

The Operation Parameters field shall be formatted as illustrated in Figure 6-118.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
PHY Mode		Supported SIFS					Multi-protocol Support

Bits: b8	b9	b10	b11	b12	b13	b14	b15
Reserved							

Bits: b16	b17	b18	b19	b20	b21	b22	b23
Preferred Payload Size		Preferred Total Aggregation Size			Reserved		Reserved

Bits: b24	b25	b26	b27	b28	b29	b30	b31
SC Supported MCS			Reserved	Reserved	Long RIFS Support	SC Channel Bonding	

Bits: b32	b33	b34	b35	b36	b37	b38	b39
SC Channel Bonding							

Bits: b40	b41	b42	b43	b44	b45	b46	b47
SC Channel Aggregation	SC Channel Aggregation Pattern						

Bits: b48	b49	b50	b51	b52	b53	b54	b55
Reserved	Reserved	Reserved	Reserved	OOK Channel Bonding			Reserved

Figure 6-118—Operation Parameters field format

The PHY Mode field indicates which PHY mode is used in the session as defined in Table 6-36.

Table 6-36—PHY Mode field values

Bits: b0	b1	PHY Mode
1	0	SC
0	1	OOK
0	0	Reserved
1	1	

The Supported SIFS field is defined in 6.4.13. The larger value of SIFS in PRC and DEV capability shall be encoded in this field.

The Multi-protocol Support field is defined in 6.4.13. It shall be set to one if both the Multi-protocol Support field in the PRC capability IE and the Multi-protocol Support field in the PRDEV capability IE are set to one and shall be set to zero otherwise.

The Preferred Payload Size field is defined in 6.4.13. The smaller value of Preferred Payload Size in PRC and DEV capability shall be encoded in this field.

The Preferred Total Aggregation Size field is defined in 6.4.13. The smaller value of Preferred Total Aggregation Size in PRC and DEV capability shall be encoded in this field.

The SC Supported MCS field is defined in 6.4.13. Each bit in this field shall be set to one if both of the bits in the SC Supported MCS field in the PRC Capability IE and PRDEV capability IE are set to one and shall be set to zero otherwise.

The Long RIFS Support field is defined in 6.4.13.

The SC Channel Aggregation field only applies when the CHNL_ID is between 7 and 16 inclusive. In that case, the field shall be set to one if channel aggregation is used or zero if channel bonding is used.

The SC Channel Bonding field indicates the number of bonded channels that shall be used in the current session as defined in the Figure 6-119 and only one bit in the field shall be set to one. All bits of the field shall be set to zero if the channel bonding is not used.

Bits: b0	b1	b2	b3	b4	b5	b6	b7	b8	b9
CHNL_ID 7 is used	CHNL_ID 8 is used	CHNL_ID 9 is used	CHNL_ID 10 is used	CHNL_ID 11 is used	CHNL_ID 12 is used	CHNL_ID 13 is used	CHNL_ID 14 is used	CHNL_ID 15 is used	CHNL_ID 16 is used

Figure 6-119—SC Channel Bonding field format

The SC Channel Aggregation Pattern field indicates the supported combinations of CHNL_IDs used for channel aggregation in the current session. The Supported SC Channel Aggregation pattern field shall be formatted as illustrated in the Figure 6-113 and only one bit in the field shall be set to one.

The OOK Channel Bonding field indicates the number of bonded channels that shall be used in the current session as defined in the Figure 6-120 and only one bit in the field shall be set to one. All bits of the field shall be set to zero if the channel bonding is not used.

Bits: b0	b1	b2
OOK 2 channel bonding is used	OOK 3 channel bonding is used	OOK4 channel bonding is used

Figure 6-120—OOK Channel Bonding field format

6.4.16 THz PRC Capability IE for pairnet

The THz PRC Capability IE shall be included in each Beacon frame. The THz PRC Capability IE Content field shall be formatted as illustrated in Figure 6-120.

Octets: 17
PRC Capability

Figure 6-121—THz PRC Capability IE Content field format

The THz PRC Capability field shall be formatted as illustrated in Figure 6-122.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
SC Capable	OOK Capable	Supported SIFS					Multi-protocol Support
Bits: b8	**b9**	**b10**	**b11**	**b12**	**b13**	**b14**	**b15**
LLPS Control							
Bits: b16	**b17**	**b18**	**b19**	**b20**	**b21**	**b22**	**b23**
Preferred Payload Size		Preferred Total Aggregation Size			Supported Unit of Subframe Padding		Pilot Symbol Capable
Bits: b24	**b25**	**b26**	**b27**	**b28**	**b29**	**b30**	**b31**
SC Supported Modulations						OOK Supported FEC	
Bits: b32	**b33–b42**		**b43–b133**				**b134–b135**
Long RIFS Support	Supported Bandwidths		Spectrum Part Supported				Reserved

Figure 6-122—THz PRC Capability field format

The SC Capable field shall be set to one if the DEV supports the THz-SC PHY, as defined in 14.2, and shall be set to zero otherwise.

The OOK Capable field shall be set to one if the DEV supports the THz-OOK PHY, as defined in 14.3, and shall be set to zero otherwise.

The Supported SIFS field shall be set to the value of the shortest SIFS supported by the DEV in units of 0.1 µs encoded as an unsigned integer. For example, a value of 0b01001 indicates that the shortest SIFS supported by the DEV is 0.9 µs. Values greater than 0b11001 (2.5 µs) are reserved.

The Multi-protocol Support field shall be set to one if the THz-SC PHY DEV supports the Pairnet Multi-protocol Data Frame, as defined in 6.3.8.1 and 6.3.8.2, and shall be set to zero otherwise.

The LLPS Control field contains LLPS related parameters, as defined in 6.4.13.

The LLPS Allow field shall be set to one if the PRC allows the PRDEVs to use power save mode after association is completed; otherwise, it shall be set to zero.

The LLPS Interval field indicates the value of the ACK sending interval when the DEV is in DEV Sleep mode. The field is defined in Table 6-30.

The LLPS Start field tells the PRDEV when to enter LLPS mode. The field value is equal to a period of time during which consecutive ACKs are sent but no payload is sent. The valid values of the LLPS Start field are given in Table 6-31.

The LLPS Extend field tells the PRDEV when to extend LLPS mode. The field value is equal to a period of time during which consecutive ACKs are sent but no payload is sent. The valid values of the LLPS Extend field are given in Table 6-32.

The Preferred Payload Size field indicates the maximum preferred data size of a single subframe payload to be received by the DEV. This field shall be set to a non-reserved value defined in Table 6-33.

The Preferred Total Aggregation Size field indicates the maximum preferred total data size in a single frame to be received by the DEV when fragmentation is used. This field shall be set to a value defined in Table 6-34.

The Supported Unit of Subframe Padding field indicates the unit of the subframe padding that can be received by the DEV, as defined in Figure 6-111. Each field shall be set to one for supported capability and shall be set to zero otherwise.

The Pilot Symbol Capable field shall be set to one if the DEV is capable of decoding the frame with pilot symbols and shall be set to zero otherwise.

The SC Supported Modulations field includes only those modulation schemes that are optional, as described in 14.2.3.1. The field is encoded as a bitmap and is given in Table 6-37. A bit shall be set to one if the modulation is supported and shall be set to zero otherwise.

Table 6-37—SC Supported Modulations field format

Bit	Description
0	$\pi/2$ 8-PSK
1	$\pi/2$ 8-APSK
2	16-QAM
3	64-QAM
4	16-APSK
5	32-APSK

The OOK Supported FEC field is encoded as a bitmap, as given in Table 6-38. A bit shall be set to one if the LDPC code is supported and shall be set to zero otherwise.

The Long RIFS Support field is defined in 6.4.13.

Table 6-38—OOK Supported FEC field format

Bit	Description
0	LDPC (1440,1344)
1	LDPC (1440,1056)

A maximum of eight different bandwidths are supported for the THz PHY. The Supported Bandwidths field is encoded as a bitmap, as given in Table 6-39. A bit shall be set to one if the bandwidth is supported and shall be set to zero otherwise.

Table 6-39—Supported Bandwidths field format

Bit	Description
0	Bandwidth 2.16 GHz
1	Bandwidth 4.32 GHz
2	Bandwidth 8.64 GHz
3	Bandwidth 12.96 GHz
4	Bandwidth 17.28 GHz
5	Bandwidth 25.92 GHz
6	Bandwidth 51.84 GHz
7	Bandwidth 69.12 GHz
8	Bandwidth 34.56 GHz

Spectrum parts are given in the smallest granularity of 2.16 GHz. The spectrum parts correspond to the frequency ranges of CHNL_ID 1 to 32, as defined in Table 14-1. The Spectrum Parts Supported field is encoded as a bitmap. The bit i–1 shall be set to one if the spectrum part corresponding to $(250.56 + 2.16 \times i)$ GHz $< f <$ $(252.72 + 2.16 \times i)$ GHz is supported and shall be set to zero otherwise. For the different frequency bands, the available default channels described in Table 6-40 shall be used. For at least one of these default channels, the bits corresponding to the spectrum parts shall be set to one; bit 1, defined in Table 6-39, shall also be set to one.

Table 6-40—THz default channels

Spectrum range	Default CHNL_ID	Bit number in Figure 6-122
252–325 GHz	41	17, 18
356–450 GHz (lower part)	137	86, 87
356–450 GHz (lower part)	149	110, 111

6.4.17 THz PRDEV Capability IE for pairnet

The THz PRDEV Capability IE shall be included in each Beacon frame. The THz PRDEV Capability IE Content field shall be formatted as illustrated in Figure 6-123.

Octets: 9
PRDEV Capability

Figure 6-123—THz PRDEV Capability IE Content field format

The PRDEV Capability field shall be formatted as illustrated in Figure 6-122, with the exception that the LLPS Control field is replaced by a reserved field.

6.4.18 THz Pairnet Operation Parameter IE

The THz Pairnet Operation Parameters IE shall be included in each Association Response command frame.

The THz Pairnet Operation Parameters IE Content field shall be formatted as illustrated in Figure 6-124. The Operation Parameters field indicates the communication parameters to be used in the current session, as decided by the PRC to satisfy both PRC and PRDEV capabilities.

Octets: 17
Operation Parameters

Figure 6-124—THz Pairnet Operation Parameters IE Content field format

The Operation Parameters field shall be formatted as illustrated in Figure 6-125.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
PHY Mode		Supported SIFS					Multi-protocol Support

Bits: b8	b9	b10	b11	b12	b13	b14	b15
Reserved							

Bits: b16	b17	b18	b19	b20	b21	b22	b23
Preferred Payload Size		Preferred Total Aggregation Size			Reserved		Reserved

Bits: b24	b25	b26	b27	b28	b29	b30	b31
SC Supported Modulations						OOK Supported FEC	

Bits: b32	b33–b42	b43–b133	b134–135
Long RIFS Support	Supported Bandwidths	Spectrum Part Supported	Reserved

Figure 6-125—Operation Parameters field format

The PHY Mode field indicates which PHY mode is used in the session and shall be set to a non-reserved value defined in Table 6-41.

Table 6-41—PHY Mode field values

Bits: b0	b1	PHY Mode
1	0	SC
0	1	OOK
0	0	Reserved
1	1	Reserved

The Supported SIFS field is defined in 6.4.16. The SIFS values in the PRC Capability field and in the PRDEV Capability field are compared, and the larger of the two values shall be encoded in this field.

The Multi-protocol Support field is defined in 6.4.16. It shall be set to one if both the Multi-protocol Support field in the PRC Capability IE and the Multi-protocol Support field in the PRDEV Capability IE are set to one and shall be set to zero otherwise.

The Preferred Payload Size field is defined in 6.4.16. The value of the Preferred Payload Size field in the PRC Capability field is compared with the Preferred Payload Size field in the PRDEV Capability IE, and the smaller of these two values shall be encoded in this field.

The Preferred Total Aggregation Size field is defined in 6.4.16. The Preferred Total Aggregation Size fields in the PRC Capability field and the PRDEV Capability field are compared, and the smaller of the two values shall be encoded in this field.

The SC Supported Modulations field is defined in 6.4.16. Each bit in this field shall be set to one if both of the bits in the SC Supported MCS fields in the PRC Capability IE and the PRDEV Capability IE are set to one and shall be set to zero otherwise.

The OOK Supported FEC field is defined in 6.4.16. Each bit in this field shall be set to one if both of the bits in the OOK Supported FEC fields in the PRC Capability IE and the PRDEV Capability IE are set to one and shall be set to zero otherwise.

The Long RIFS Support field is defined in 6.4.13.

The Supported Bandwidths field is defined in 6.4.16. Each bit in this field shall be set to one if both of the bits in the Supported Bandwidths fields in the PRC Capability IE and the PRDEV Capability IE are set to one and shall be set to zero otherwise.

The Supported Spectrum Part field is defined in 6.4.16. Each bit in this field shall be set to one if both of the bits in the Supported Spectrum Part fields in the PRC Capability IE and the PRDEV Capability IE are set to one and shall be set to zero otherwise.

6.4.19 Transmit Power Parameters IE

The Transmit Power Parameters IE is used to communicate the transmit power control (TPC) capabilities of a DEV. The Transmit Power Parameters IE Content field shall be formatted as illustrated in Figure 6-126.

Octets: 1	**1**	**1**
TX Power Levels	TX Power Step Size	Current TX Power

Figure 6-126—Transmit Power Parameters IE Content field format

The TX Power Levels field indicates the number of levels supported by a DEV.

The TX Step Size field indicates the TX power level step size in 1 dB resolution, e.g., a number 4 in this field means that the DEV has nominally 4 dB steps.

If a DEV does not support TPC, it shall set the TX Power Levels and TX Power Step Size fields to zero.

The Current TX Power field is the DEV's estimate of its transmitter power measured at the antenna interface. The value is in dBm encoded in two's complement format. For example, a +2 dBm TX power level is encoded as 0x02 while a –2 dBm TX power level is encoded as 0xFE.

6.4.20 PS Status IE for piconet

The PS Status IE Content field shall be formatted as illustrated in Figure 6-127.

Octets: 2	1	1	1–32
Next Wake Beacon	PS Set Index	Start DEVID	DEVID Bitmap

Figure 6-127—PS Status IE Content field format

The Next Wake Beacon field shall be set to the beacon number, as described in 6.3.1.1, of the next wake beacon for the set indicated in the PS Set Index field, as described in 7.17.2. It shall be set to zero when the PS Set Index field is zero, i.e., for the set of DEVs in APS mode.

The PS Set Index field is set to the index of the power save set as follows:

0 → APS
1 → PSPS
2–253 → DSPS

The Start DEVID field indicates the DEVID that corresponds to the first bit in the DEVID bitmap.

The DEVID Bitmap field is 1 to 32 octets in length. Each bit of the DEVID Bitmap field corresponds to a DEVID that is equal to the start DEVID plus the bit position in the bitmap. The bit position zero, i.e., the first bit or LSB of the bitmap, corresponds to the start DEVID. The bit corresponding to a DEVID shall be set to one when a DEV that is a member of this PS set is in a power save mode. It shall be set to zero otherwise.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs, and the reserved IDs, as described in 6.2.5, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

6.4.21 CWB IE for piconet

The continued wake beacon (CWB) IE Content field shall be formatted as illustrated in Figure 6-128.

Octets: 1	1–32
Start DEVID	DEVID bitmap

Figure 6-128—CWB IE Content field format

The Start DEVID field indicates the DEVID that corresponds to the first bit in the DEVID bitmap.

The DEVID Bitmap field is 1 to 32 octets in length. Each bit of the DEVID bitmap corresponds to a DEVID that is equal to the start DEVID plus the bit position in the bitmap. The bit position zero, i.e., the first bit or LSB of the bitmap, corresponds to the start DEVID. The bit corresponding to a DEVID shall be set to one when the PNC is requesting that the DEV listen to the next beacon for a CTA block, as described in 7.17.3.4. It shall be set to zero otherwise.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs, and the reserved DEVIDs, as described in 6.2.5, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

6.4.22 Overlapping PNID IE for piconet

The Overlapping PNID IE is used to communicate the PNIDs that a DEV has detected either in its channel or in other channels. The Overlapping PNID IE Content field shall be formatted as illustrated in Figure 6-129.

Octets: 2	**1**	**…**	**2**	**1**
PNID 1	Channel Index 1	…	PNID *n*	Channel Index *n*

Figure 6-129—Overlapping PNID IE Content field format

The PNID field contains the PNID from a frame that a DEV has received since the last time this IE was sent by the DEV to the PNC.

If the DEV has received a beacon from a different piconet on the current channel with the same PNID but a different PNC address, it will add that PNID and channel index to this IE. Otherwise, the IE will not contain the same PNID/channel index pair as for the current piconet. Thus a DEV will report piconets with the same PNID in other channels but will not erroneously report frames from the current piconet as being from an overlapping piconet.

The Channel Index field contains the channel on which the PNID was found.

6.4.23 Piconet Services IE

The Piconet Services IE is used to provide information about the application layer capabilities of an individual DEV. The Piconet Services IE Content field shall be formatted as illustrated in Figure 6-130.

Octets: 0/1	**0/1/3**	**0–127**
DEVID	Services ID	Piconet Services

Figure 6-130—Piconet Services IE Content field format

The DEVID field identifies the DEV corresponding to the Piconet Services field. If the PNC is sending the IE as the aggregate capabilities of the piconet, the DEVID field shall be set to the BcstID.

If the value of the DEVID field is the PNCID, then the Piconet Services field is not present and the Services ID field is 1 octet in length. In this case, the Services ID field contains the following information:

0 → Reserved
1 → Broadcast of Piconet Services IE not allowed
2 → Piconet Services IE not supported
3–254 → Reserved
255 → Other failure

If the value of the DEVID field is not the PNCID, then the Services ID field contains a Unique ID field, as defined in 6.4.8. In this case, the format of the Piconet Services field is defined by the entity indicated in the Unique ID field.

The Piconet Services field is used to indicate the application layer capabilities of the DEV indicated by the DEVID. The content of the Piconet Services field is outside of the scope of this standard.

6.4.24 Vendor Defined IE

The Vendor Defined IE Content field shall be formatted as illustrated in Figure 6-131.

Octets: 3	variable
Unique ID	Vendor Defined information

Figure 6-131—Vendor Defined IE Content field format

The Unique ID field is defined in 6.4.8.

The format of the Vendor Defined Information field is defined by the entity identified in the Unique ID field. Its use by a DEV is outside of the scope of this standard.

6.4.25 Group ID IE for piconet

The Group ID IE is used to list the DEVs that are members of a multicast group. The Group ID IE Content field shall be formatted as illustrated in Figure 6-132.

Octets: 6	1	1	1–32
Group Address	GrpID	Start DEVID	Group IDs

Figure 6-132—Group ID IE Content field format

The Group Address field is a 48-bit MAC address with the I/G bit set to indicate a group MAC address, as defined in IEEE Std 802.

The GrpID field contains the DEVID that has been assigned by the PNC for the address in the Group Address field.

The Start DEVID field indicates the DEVID that corresponds to the first bit in the Group IDs field.

The Group IDs field contains a bitmap of 1 to 32 octets in length. Each bit of the Group IDs field when set to one indicates the DEV whose DEVID is equal to the start DEVID plus the bit position in the Group ID bitmap is a member of the multicast group identified by the Group Address field and GrpID field. The bits in the Group IDs field are set to zero otherwise. The bit position 0, i.e., the first bit or LSB of the bitmap, corresponds to the start DEVID.

The bits corresponding to the PNCID, UnassocID, BcstID, McstID, NbrIDs, and the reserved DEVIDs, 6.2.5, shall be set to zero upon transmission by the PNC and shall be ignored upon reception.

6.4.26 Stream Renew IE for piconet

The Stream Renew IE is used by a DEV to renew the STP for streams for which it is the source. The Stream Renew IE Content field shall be formatted as illustrated in Figure 6-133.

Octets: 1	**1**	**…**	**1**
Stream Index 1	Stream Index 2	…	Stream Index *n*

Figure 6-133—Stream Renew IE Content field format

The Stream Index fields contain the stream indices that the DEV wants to maintain.

6.4.27 Next PNC IE for piconet

The Next PNC IE is used to inform the members of the piconet which DEV will become the PNC after an implicit handover, as described in 7.2.6. The Next PNC IE Content field shall be formatted as illustrated in Figure 6-134.

Octets: 1	**2**
Next PNC	Next PNID

Figure 6-134—Next PNC IE Content field format

The Next PNC field contains the DEVID of the DEV that the PNC has chosen to be the Next PNC.

The Next PNID field shall contain the PNID to be used by the next PNC if it takes over as PNC, as described in 7.2.6.

6.4.28 Piconet Channel Status IE

The Piconet Channel Status IE is used to report those DEVs from which the source DEV has correctly received a MAC header. The Piconet Channel Status IE Content field shall be formatted as illustrated in Figure 6-135.

Octets: 32	**1**
Bitmap of DEVs in Range	Channel Status Details

Figure 6-135—Piconet Channel Status IE Content field format

The Bitmap of DEVs in Range field is a listing of all of the DEVs in the piconet that were heard by the DEV sending the command. The bit position zero, i.e., the first bit or LSB of the bitmap, corresponds to DEVID zero. The bit corresponding to a DEVID is set to one if a MAC header was correctly received from a DEV with that DEVID since the last Piconet Channel Status IE was sent. Otherwise, the bit shall be set to zero.

The Channels Status Details field shall be set to one if the DEV scanned for power save DEVs. Otherwise, this value shall be set to zero.

6.4.29 Synchronization IE for piconet

The Synchronization IE Content field shall be formatted as illustrated in Figure 6-136. The Synchronization IE shall be supported by DEVs that support the SC PHY or HSI PHY.

Octets: 4	1	5	5
Quasi-omni Beacon Info	Section Indications	Association S-CAP Info	Regular S-CAP Info

Figure 6-136—Synchronization IE Content field format

The Quasi-omni Beacon Info field shall be formatted as illustrated in Figure 6-137.

Bits: b0–b15	b16–b18	b19–b21	b22–b25	b26–b31
Beacon Offset Time	Beacon Index	Number Beacon Frames	PNC Directional Antenna Capabilities	Reserved

Figure 6-137—Quasi-omni Beacon Info field format

The Beacon Offset Time field contains the time in microseconds that is the start of this Beacon frame delayed from the start of the superframe.

The Beacon Index field indicates the index of the current Beacon frame.

The Number Beacon Frames field indicates the number of Beacon frames that will be sent as part of the quasi-omni beacon.

The PNC Directional Antenna Capabilities field shall be formatted as illustrated in Figure 6-138.

Bits: b0	b1	b2	b3
Antenna Symmetry	PNC Beam-Forming Capable	Sectors Only	BST Support

Figure 6-138—PNC Directional Capabilities field format

The Antenna Symmetry field shall be set to one if the PNC has a symmetric antenna system (SAS) and shall be set to zero otherwise, i.e., the PNC has an asymmetric antenna system (AAS).

The PNC Beam-Forming Capable field shall be set to one if the PNC is capable of beam forming and shall be set to zero otherwise.

The Sectors Only field shall be set to one if the PNC supports only sectorized antenna and shall be set to zero otherwise.

The Beam Switching/Steering and Tracking (BST) Support field shall be set to one if the PNC supports beam switching/steering and tracking, as described in Clause 15, and shall be set to zero otherwise.

The Section Indication field shall be formatted as illustrated in Figure 6-139.

Bits: b0	b1	b2	b3	b4–b7
PNC Quasi-omni Tracking Present	Sector Training Present	Association S-CAP Present	Regular S-CAP Present	Reserved

Figure 6-139—Section Indication field format

The PNC Quasi-omni Tracking Present field shall be set to one if the PNC quasi-omni tracking section exists in the beacon and shall be set to zero otherwise.

The Sector Training Present field shall be set to one if the sector training section exists in the beacon and shall be set to zero otherwise.

The Association Sub-contention Access Field (S-CAP) Present field shall be set to one if association S-CAP sections exist in the CAP and shall be set to zero otherwise.

The Regular S-CAP Present field shall be set to one if regular S-CAP sections exist in the CAP and shall be set to zero otherwise.

The Association S-CAP info field shall be formatted as illustrated in Figure 6-140.

Bits: b0–b15	b16–b31	b32–b33	b34–b35	b36–b37	b38–b39
Association S-CAP Start Time	S-CAP Duration	Total Number Association S-CAPs	Number Association S-CAPs in Superframe	First PNC Quasi-omni RX Index	Reserved

Figure 6-140—Association S-CAP Info field format

The Association S-CAP Start Time field specifies the start time offset of the association S-CAPs from the start of the superframe in microseconds.

The S-CAP Duration field specifies the duration of the association S-CAP, as described in 7.8.6.3, in microseconds.

The Total Number Association S-CAPs is the same as the total number of PNC quasi-omni RX directions.

The Number Association S-CAPs in Superframe field specifies the number of association S-CAPs in the current superframe.

The First PNC Quasi-omni RX Index field specifies the PNC's quasi-omni RX index that will be used in the first association S-CAP in the current superframe.

The Regular S-CAP info field shall be formatted as illustrated in Figure 6-141.

Bits: b0–b15	b16–b31	b32–b33	b34–b35	b36–b37	b38–b39
Regular S-CAP Start Time	S-CAP Duration	Total Number Regular S-CAPs	Number Regular S-CAPs in Superframe	First PNC Quasi-omni RX Index	Reserved

Figure 6-141—Regular S-CAP Info field format

The Regular S-CAP Start Time field specifies the start time offset of the regular S-CAP from the start of the superframe in microseconds.

The S-CAP Duration field specifies the duration of the regular S-CAPs in microseconds.

The Total Number Regular S-CAPs field is set to be the same as the total number of PNC quasi-omni RX directions.

The Number Regular S-CAPs field in Superframe field specifies the number of regular S-CAPs in the current superframe.

The First PNC Quasi-omni RX Index field specifies the PNC's quasi-omni RX index that will be used in the first regular S-CAP in the current superframe.

6.4.30 TSD IE for piconet

The TSD IE is used for information exchange in TSD procedure as described 15.8. The TSD IE Content field shall be formatted as illustrated in Figure 6-142.

Octets: 1	1	1	1
Mode	Number Of Transmit Directions	TSD Feedback Period	Transmit Direction Index

Figure 6-142—TSD IE Content field format

The Mode field is encoded as indicated in Table 6-42.

Table 6-42—Mode field encoding

Field value	Mode	Description
0	Announce	Announce TSD IE sent from PNC to DEV
1	Request to switch	DEV requests PNC to switch to the next transmit direction
2	Request to stay	DEV requests PNC to stay at the current transmit direction
3	Response	In response of a request
4–255	Reserved	

The Number Of Transmit Directions field indicates the number of TX directions supported by the PNC.

The TSD Feedback Period field indicates the time interval between the TSD IEs sent from the DEV to the PNC for feedback.

The Transmit Direction Index field indicates the index of the transmit direction of the PNC. A field value of zero indicates that the PNC shall select one of the transmit directions that was not used previously.

6.4.31 UEP Specific IE for piconet

The UEP Specific IE is used to indicate the separating position of MSB and LSB data. For example, it is used to handle different components of a video signal, e.g., RGB, YCbCr. This IE shall be only used in UEP type 2. The UEP Specific IE Content field shall be formatted as illustrated in Figure 6-143.

Octets: 4
MSB/LSB Separation

Figure 6-143—UEP Specific IE Content field format

The MSB/LSB Separation field shall be formatted as illustrated in Figure 6-144.

Octets: 1	1	1	1
Component 1	Component 2	Component 3	Reserved

Figure 6-144—MSB/LSB Separation field format

The Component fields contain the information to indicate the separating position of MSB and LSB data. Valid values of the Component 1, 2, and 3 fields are as follows:

0 → All bits are MSB
1 → b_0 is LSB, b_1–b_7 are MSB
2 → b_0–b_1 is LSB, b_2–b_7 are MSB
3 → b_0–b_2 is LSB, b_3–b_7 are MSB
4 → b_0–b_3 is LSB, b_4–b_7 are MSB
5 → b_0–b_4 is LSB, b_5–b_7 are MSB
6 → b_0–b_5 is LSB, b_6–b_7 are MSB
7 → b_0–b_6 is LSB, b_7 is MSB
8 → All bits are LSB
9–255 → Reserved

6.4.32 IFS IE for piconet

The IFS IE Content field shall be formatted as illustrated in Figure 6-145.

Octets: 1
IFS Value

Figure 6-145—IFS IE Content field format

The IFS Value field shall be formatted as illustrated in Figure 6-146.

Bits: b0–b3	b4–b7
SIFS	MIFS

Figure 6-146—IFS value field format

The mapping of a field value to a specific IFS value is defined in Table 6-29.

6.4.33 CTA Relinquish Duration IE for piconet

The CTA Relinquish Duration IE Content field shall be formatted as illustrated in Figure 6-147.

Octets: 2
CTA Relinquish Duration

Figure 6-147—CTA Relinquish Duration IE Content field format

The CTA Relinquish Duration field contains the time in microseconds for which the time in the CTA is relinquished.

6.4.34 Feedback IE for piconet

The Feedback IE shall be formatted as illustrated in Figure 6-148.

Bits: b0–b3	b4–b9	b10–b13	b14–b19	b20	b21–b23	b24–b27	b28–b31
Best Sector	Best Sector LQI	Second Best Sector	Second Best Sector LQI	End Of Sector Training	Reserved	Total Number Of Feedbacks	Index of Current Feedback

Figure 6-148—Feedback IE Content field format

The Best Sector field shall be set to the index of the best transmit sector for the originating DEV. In the SAS case, this also represents the index of the best receiver sectors for the target DEV.

The Best Sector LQI field shall be set to the value of LQI measured by the Target DEV for the best sector. The Best Sector LQI field is encoded in dB. The corresponding LQI values in the case of signal-to-noise ratio (SNR)/signal-to-interference-plus-noise ratio (SINR) and receive signal strength indication relative to sensitivity (RSSIR) are given in Table 6-43 and Table 6-44. The LQI can be measured using the Channel Estimation Sequence (CES) field in the preamble.

The Second Best Sector field shall be set to the index of the second best transmit sector for the originating DEV. In the SAS case, this also represents the index of the second best receiver sectors for the target DEV. If the second best sector is not supported, then this field shall be set to zero.

The Second Best Sector LQI field shall be set to the value of LQI measured by the Target DEV for the second best sector. The Second Best Sector LQI is encoded in the same manner as the Best Sector LQI. If the second best sector is not supported, then this field shall be set to zero.

The encoding of LQI for SNR/SINR is defined in Table 6-43. The SNR value is equal to E_c/N_0 (dB) + 20 dB.

Table 6-43—SNR/SINR value for LQI field

Field value	SNR/SINR value
0b000000	≤ 0 dB
0b000001	0.5 dB
0b000010	1 dB
…	…
0b111110	31 dB
0b111111	> 31.5 dB

The encoding of LQI for RSSIR is defined in Table 6-44 where 0 dB indicates that the received signal level is equal to receiver sensitivity for the respective base rate.

Table 6-44—RSSIR value for LQI field

Field value	RSSIR value
0b000000	≤ 0 dB
0b000001	1 dB
0b000010	2 dB
…	…
0b111110	62 dB
0b111111	> 63 dB

The End Of Sector Training field is used to indicate that a DEV wants to end the training stage. Any DEV who wants to terminate the training stage at the end of sector-level training may set this field to one during sector-level training period. However, if one of DEVs, DEV1, sets the field to one but the other DEV, DEV2, does not, then DEV1 shall help DEV2 to finish beam-level training by using DEV1's best transmit and receive sectors.

The Total Number Of Feedbacks field shall be set to the number of Announce commands that will be sent as part of the feedback process.

The Index Of Current Feedback field shall be set to number of the current Feedback IE in the series that is being sent. The first Feedback IE that is sent shall have the Index of the Current Feedback field set to zero.

6.4.35 Mapping IE for piconet

The Mapping IE Content field shall be formatted as illustrated in Figure 6-149.

Bits: b0–b5	b6–b7	b8–b13	b14–b17	b18–b21	b22–b23
Number TX Beams	Sync Mode	Number Of RX Beams	Best Cluster Tracking Period	Second Best Cluster Tracking Period	Reserved

Figure 6-149—Mapping IE Content field format

When sent at the end of sector-level training, the Number of TX Beams field and Number of RX Beams field shall be set to one less than the number of beams that the sending device will be using during the beam-level training. When sent at the end of the beam-level training, these fields shall be set to the number of high resolution (HRS) beams to be used during tracking.

If the system is SAS, the Number of RX Beams field shall be set to zero.

The Sync Mode field shall encode the length of the sync sequence to be used for subsequent training sequences. The value of the field shall be the same as that used in the PHY header, as described in 12.2.4.2.2 and 12.3.4.3.

The Best Cluster Tracking Period and Second Best Cluster Tracking Period fields shall be set to the tracking period of the best cluster and the second best cluster, respectively. The Tracking Period fields shall be encoded as defined in Table 6-45.

Table 6-45—Tracking period encoding

Field value	Tracking period value
0b0000	Reserved
0b0001	0.004 ms
0b0010	0.008 ms
0b0011	0.016 ms
0b0100	0.032 ms
0b0101	0.064 ms
0b0110	0.128 ms
0b0111	0.256 ms
0b1000	0.512 ms
0b1001	1.024 ms
0b1010	2.048 ms
0b1011	4.096 ms
0b1100	8.192 ms
0b1101	16.384 ms

Table 6-45—Tracking period encoding (*continued*)

Field value	Tracking period value
0b1110	32.768 ms
0b1111	Reserved

6.4.36 BST Clustering IE for piconet

The BST Clustering IE is used for AAS sector- and beam-level training, as defined in 15.5.2.2.2 and 15.5.2.3.2, respectively. It is also used for SAS sector- and beam-level training, as defined in 15.5.2.2.3 and 15.5.2.3.3, respectively. The BST Clustering IE Content field shall be formatted as illustrated in Figure 6-150.

Octets: variable	variable
BST TX Cluster Mapping	BST RX Cluster Mapping

Figure 6-150—BST Clustering IE Content field format

The BST TX Cluster Mapping field and BST RX Cluster mapping field shall be formatted as illustrated in Figure 6-151. In the case of a SAS system, the TX and RX clusters are identical, and only one BST Cluster Mapping field shall be included in the IE.

Bits: b0–b3	b4–b7	b8–b13	…	b(4+N*4)–b(7+N*4)	b(8+N*4)–b(13+N*4)/0
Number Of Clusters	Number Of Beams In Cluster 0	Number Of Beams In Cluster 1	…	Number Of Beams In Cluster N	Reserved

Figure 6-151—BST TX Cluster Mapping field and BST RX Cluster Mapping field format

The Number Of Clusters field contains one less than the number of TX or RX clusters, C_T and C_R, respectively, as defined in 15.2.5.

The Number Of Beams in Cluster field shall be set to one less than the number of beams that make up that cluster.

The Reserved field shall be included if there is an even number of clusters and shall be omitted otherwise.

6.4.37 PET Clustering IE for piconet

The PET Clustering IE is used for AAS sector- and beam-level training, as defined in 15.5.2.2.2 and 15.5.2.3.2, respectively. It is also used for SAS sector- and beam-level training, as defined in 15.5.2.2.3 and 15.5.2.3.3, respectively. The PET Clustering IE Content field shall be formatted as illustrated in Figure 6-152.

Octets: variable	variable
PET TX Cluster Mapping	PET RX Cluster Mapping

Figure 6-152—PET Clustering IE Content field format

The PET TX Cluster Mapping field and PET RX Cluster Mapping field shall be formatted as illustrated in Figure 6-153. In the case of a SAS system, the transmit and receive clusters are identical, and only one Cluster Mapping field shall be included in the IE.

Octets: 1	2	2	…	2
Number Of Clusters	Cluster 0 Descriptor	Cluster 1 Descriptor	…	Cluster *N* Descriptor

Figure 6-153—PET TX Cluster Mapping field and PET RX Cluster Mapping field format

The Number of Clusters field shall be set to one less than the number of TX or RX clusters, C_T and C_R, respectively, as defined in 15.2.5.

The Cluster Descriptor fields shall be for the TX or RX arrays as identified by the position of the field in the PET Clustering IE. The Cluster Descriptor field shall be formatted as illustrated in Figure 6-154.

Octets: 1	1
Center Beam Index	Cluster Encoding

Figure 6-154—Cluster Descriptor field format

The Cluster Encoding field describes the geometry of the cluster and shall be as described in 15.2.5.

The Center Beam Index field shall be the index of the beam or HRS beam around which the cluster is formed.

6.4.38 Beam PET IE for piconet

The Beam PET IE Content field shall be formatted as illustrated in Figure 6-155.

Octets: 4	4
PET TX Configuration	PET RX Configuration

Figure 6-155—Beam PET IE Content field format

For the SAS case, the transmit and receive PET configurations are the same, so the PET RX Configuration field shall be omitted.

The PET TX Configuration field and PET RX Configuration field shall be formatted as shown in Figure 6-156.

Bits: b0–b3	b4–b7	b8–b15	b16–b23	b24–b26	b27–b29	b30–b31
Number Of Antennas, *z*-axis	Number Of Antennas, *x*-axis	Beam Codebook ID, *z*-axis	Beam Codebook ID, *x*-axis	Amplitude Resolution	Phase Resolution	Reserved

Figure 6-156—PET Configuration field format

The Number of Antennas, *z*-axis and Number of Antennas, *x*-axis fields shall be set to one less than the number of antennas along the *z*-axis and *x*-axis, respectively, as described in 15.2.4. The *z*-axis and *x*-axis are only applied here as an example to define an antenna pattern for a linear two-dimensional (2-D) antenna array, as described in 15.2.4. The reference to *z*-axis and *x*-axis are changeable according to the implementation requirement. These values shall be for the TX or RX arrays as identified by the position of the field in the PET Clustering IE.

The Beam Codebook ID fields shall identify the codebooks, as described in 15.3, to be used for the respective axes.

The Amplitude Resolution field shall indicate the number of discrete values for amplitude that can result from pattern estimation. The field shall be coded as N, where there are 2^N possible resulting amplitude values. The value of this field shall be limited to the range 0 to 4.

Similarly, the Phase Resolution field shall indicate the number of discrete phase values that can result from pattern estimation. This field shall also be coded as N, where there are 2^N possible resulting phase values. The value of this field shall be limited to the range 0 to 4.

6.4.39 HRS Beam PET IE for piconet

The HRS Beam PET IE Content field shall be formatted as illustrated in Figure 6-157.

Octets: 2	2
PET TX HRS Configuration	PET RX HRS Configuration

Figure 6-157—HRS Beam PET IE Content field format

For the SAS case, the PET TX HRS Configuration field and PET RX HRS Configuration field are the same, so the PET RX HRS Configuration field shall be omitted.

The PET TX HRS Configuration field and PET TX HRS Configuration field shall be formatted as illustrated in Figure 6-158.

Octets: 1	1
HRS Beam Codebook ID, *z*-axis	HRS Beam Codebook ID, *x*-axis

Figure 6-158—PET HRS Configuration field format

The HRS Beam Codebook ID fields shall identify the codebooks, as described in 12.3.2, to be used for their respective axes.

6.4.40 PET Amplitude IE for piconet

The PET Amplitude IE Content field shall be formatted as illustrated in Figure 6-159.

Bits: b0–b3	b4–b7	…	b(4+*M**4)–b(7+*M**4)	b(8+*M**4)–b(13+*M**4)/0
Element 0 Amplitude	Element 1 Amplitude	…	Element *M* Amplitude	Reserved

Figure 6-159—PET Amplitude IE Content field format

The Element Amplitude fields are sent in the predefined antenna element order, as defined in 15.2.4. If *F* represents the amplitude relative to the element with the highest amplitude, the value of the Element Amplitude field shall be the one less than the numerator, *N*, of a fraction $N/D = F$, where *D* is the number of possible values as specified by the Amplitude Resolution field of the Beam PET IE. For example, if the Amplitude Resolution field value is 3, then there are $2^3 = 8$ possible values and $D = 8$. In this case, one of the antenna element amplitudes will be $N = 8$, and the others will have values in the range of $N = 1$ to $N = 8$. For this case, one Element Amplitude value shall be seven while the others will have values in the range zero to seven.

The Reserved field shall be present if there is an odd number of antenna elements; it shall be omitted otherwise.

6.4.41 PET Phase IE for piconet

The PET Phase IE Content field shall be formatted as illustrated in Figure 6-160.

Bits: b0–b3	b4–b7	…	b(4+*M**4)–b(7+*M**4)	b(8+*M**4)–b(13+*M**4)/0
Element 0 Phase	Element 1 Phase	…	Element *M* Phase	Reserved

Figure 6-160—PET Phase IE Content field format

The Element Phase fields are sent in the predefined antenna element order, as defined in 13.2.4. If *F* represents the phase as a fraction of a full circle, the Element Phase field shall set to the numerator, *N*, of a fraction $N/D = F$, where *D* is the number of possible values as specified by the Phase Resolution field of the Beam PET IE. For example, if the phase resolution field value is 4, then there are $2^4 = 16$ possible values and $D = 16$. In this case, each LSB of the phase value is equal to 22.5°, and the values shall be in the range of $N = 0$ to $N = 15$, which corresponds to phases of 0° to 342.5°.

The Reserved field shall be present if there is an odd number of antenna elements; it shall be omitted otherwise.

6.4.42 Sync Frame Frequency IE for piconet

The Sync Frame Frequency IE Content field shall be formatted as illustrated in Figure 6-161.

Bits: b0–b4	b5	b6–b7
Sync Frame Frequency	Sync Frame Direction	Reserved

Figure 6-161—Sync Frame Frequency IE Content field format

The Sync Frame Frequency field is set to the number of superframes between transmission of two Sync frames, as defined in 6.3.9, requested by the PNC. If the Sync Frame Frequency field is set to zero, then the PNC is requesting the DEV cease sending Sync frames.

The Sync Frame Direction field shall be set to zero if the PNC is requesting omni-directional transmission of the Sync frame and shall be set to one if the PNC is requesting directional transmission of the Sync frame. If the request is for directional transmission, then the Sync frame direction is a round robin of the DEV's available directions.

6.4.43 Directional Peer IE for piconet

The Directional Peer IE Content field shall be formatted as illustrated in Figure 6-162.

Octets: 1	1	1	2
SrcID	DestID	Configuration	Allocated Superframes

Figure 6-162—Directional Peer IE Content field format

The SrcID field contains the DEVID of the source for the directional peer communication in the regular CAP and/or regular S-CAP.

The DestID field contains the DEVID of the destination for the directional peer communication in the regular CAP and/or regular S-CAP.

The Configuration field shall be formatted as illustrated in Figure 6-163.

Bits: b0	b1	b2–b7
Request/Release	Unidirectional/Bidirectional Allowance	Reserved

Figure 6-163—Configuration field format

The Request/Release field shall be set to one to request directional communication and shall be set to zero to release the directional communication.

The Unidirectional/Bidirectional Allowance field shall be set to one for unidirectional communication and shall be set to zero otherwise.

The Allocated Superframes field indicates the number of superframes allocated for directional communication between the source and destination.

6.4.44 MIMO Information IE for pairnet

The MIMO Information IE Content field shall be formatted as illustrated in Figure 6-164.

Octets: 2
MIMO Parameter

Figure 6-164—MIMO Information IE Content field format

The MIMO Parameter field indicates the parameter of the MIMO configuration of PRDEV. This field shall be interpreted by the DEV if the DEV supports MIMO. The MIMO Parameter field value is illustrated in Figure 6-165.

Bits: b0–b3	b4–b12	b13–b14	b15
SC Supported MIMO	Number of MIMO Array Training	Array Training Interval	MIMO CES Type

Figure 6-165—MIMO Parameter field format

In the Beacon frame, the SC Supported MIMO field indicates the number of MIMO branches supported by the SC PHY PNC. However, in the Association Request command frame, this field indicates the number of MIMO branches to be used in the data transmission phase. The SC Supported MIMO field shall be formatted as illustrated in Figure 6-166.

Bits: b0	b1	b2	b3
2×2 MIMO supported	4×4 MIMO supported	9×9 MIMO supported	16×16 MIMO supported

Figure 6-166—SC Supported MIMO field format

A bit in the SC Supported MIMO field shall be set to one if the SC PHY DEV supports that configuration and shall be set to zero otherwise.

The Number of MIMO Array Training field indicates the required number of MIMO Array Training commands from the DEV for MIMO negotiation training encoded as an unsigned integer.

The Array Training Interval field indicates the interval time of transmission of Array Training command. The valid values of the Array Training Interval field are given in Table 6-46.

Table 6-46—Array Training Interval field values

Bits: b13–b14	Array Training Interval
0b00	10 ms
0b01	20 ms
0b10	40 ms
0b11	80 ms

The MIMO CES Type field shall be set to one when MIMO CES for frequency domain channel estimation is used as described in 13.2.9.5.4, and shall be set to zero when MIMO CES for time domain channel estimation is used as described in 13.2.9.5.5.

6.4.45 Higher Layer Protocol Information IE for pairnet

The Higher Layer Protocol Information IE Content field shall be formatted as illustrated in Figure 6-167.

Octets: 3	variable
Unique ID	Higher Layer Protocol Information

Figure 6-167—Higher Layer Protocol Information IE Content field format

The Unique ID is defined in 6.4.7.

The Higher Layer Protocol Information field contains higher layer information whose format and content are determined by the entity identified in the Unique Identifier field. The use of the information in this IE is outside of the scope of this standard. Details are described in 7.5.3.

NOTE—It is recommended to process the Higher Layer Protocol Information IE prior to other IEs to achieve fast connection setup.

6.5 MAC commands

6.5.1 Overview

The MAC commands are listed in Table 6-47 and Table 6-48. If the column labeled “Associated” in Table 6-47 and Table 6-48 is marked with an “X,” then that MAC command shall only be sent by a DEV that is associated in the piconet or a pairnet. If the column labeled “Secure membership (if required)” in Table 6-47 and Table 6-48 is marked with an “X” and secure membership is required for the piconet or the pairnet, then that MAC command shall only be sent by a DEV that has established secure membership with the PNC in the piconet or with the PRC in the pairnet. Because a neighbor PNC is not a member of the piconet, it sends only non-secure MAC commands. The PNC or PRC or destination DEV shall ignore any MAC command from a DEV that is not allowed to be sent, as indicated in Table 6-47 and Table 6-48. The “Required” column indicates the type of DEVs that are required to support the command.

For peer-to-peer communications, if the DEV has established a secure relationship with a peer DEV, and the "Secure membership (if required)" column is marked with an "X," that command shall be sent to the peer DEV with a secure command using the key specified in Table 8-1.

Table 6-47—Command types for piconet

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)	Required
0x0000	Association Request command	6.5.2.2			All DEVs
0x0001	Association Response command	6.5.2.3	X		All DEVs
0x0002	Disassociation Request command	6.5.2.4	X		All DEVs
0x0003	Request Key command	6.5.3.2	X	X	Optional
0x0004	Request Key Response command	6.5.3.3	X	X	Optional
0x0005	Distribute Key Request command	6.5.3.4	X	X	Optional
0x0006	Distribute Key Response command	6.5.3.5	X	X	Optional
0x0007	PNC Handover Request command	6.5.4.2	X	X	All DEVs
0x0008	PNC Handover Response command	6.5.4.3	X	X	All DEVs
0x0009	PNC Handover Information command	6.5.4.4	X	X	All DEVs
0x000A	PNC Information Request command for piconet	6.5.5.2	X	X	All DEVs
0x000B	PNC Information command for piconet	6.5.5.3	X	X	All DEVs
0x000C	Security Information Request command	6.5.5.4	X	X	Optional
0x000D	Security Information command	6.5.5.5	X	X	Optional
0x000E	Probe Request command	6.5.5.6	X		All DEVs
0x000F	Probe Response command	6.5.5.7	X		All DEVs
0x0010	Piconet Services command	6.5.6.1	X		All DEVs
0x0011	Announce command for piconet	6.5.6.2	X		All DEVs
0x0012	Channel Time Request command	6.5.7.2	X	X	All DEVs
0x0013	Channel Time Response command	6.5.7.3	X	X	All DEVs
0x0014	Channel Status Request command	6.5.8.2	X	X	All DEVs
0x0015	Channel status response	6.5.8.3	X	X	All DEVs
0x0016	Remote Scan Request command	6.5.8.4	X	X	All DEVs
0x0017	Remote Scan Response command	6.5.8.5	X	X	All DEVs

Table 6-47—Command types for piconet (*continued*)

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)	Required
0x0018	Transmit Power Change command	6.5.8.6	X	X	All DEVs
0x0019	PS Set Information Request command	6.5.9.2	X	X	All DEVs
0x001A	PS Set Information Response command	6.5.9.3	X	X	All DEVs
0x001B	SPS Configuration Request command	6.5.9.4	X	X	All DEVs
0x001C	SPS Configuration Response command	6.5.9.5	X	X	All DEVs
0x001D	PM Mode Change command	6.5.9.6	X	X	All DEVs
0x001E	Security Message command	6.5.10.1	X		Optional
0x001F	Announce Response command for piconet	6.5.6.3	X		All DEVs
0x0020	PM Mode Change Response command	6.5.9.7	X	X	All DEVs
0x0021	AS IE Request command for piconets	6.5.10.3	X	X	Optional
0x0022	AS IE Response command for piconets	6.5.10.4	X	X	Optional
0x0023	Multicast Configuration Request command	6.5.11.1	X	X	Optional
0x0024	Multicast Configuration Response command	6.5.11.2	X	X	Optional
0x0025–0x00FF	Reserved				
0x0100–0xFFFF	Vendor Defined command	6.5.10.2	X		Optional

The command types for PRDEVs are given in Table 6-48.

Table 6-48—Command types for pairnet

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)
0x0000	Association Request command	6.5.2.2	—	
0x0001	Association Response command	6.5.2.3	X	
0x0002	Disassociation Request command	6.5.2.4	X	
0x0003	Request Key command	6.5.2.1	X	X

Table 6-48—Command types for pairnet (*continued*)

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)
0x0004	Request Key Response command	6.5.2.2	X	X
0x0005	Distribute Key Request command	6.5.3.4	X	X
0x0006	Distribute Key Response command	6.5.2.4	X	X
0x0007-0x000B	Reserved	—	—	
0x000C	Security Information Request command	6.5.4.3	X	X
0x000D	Security Information command	6.5.5.5	X	X
0x000E	Probe Request command	6.5.5.6	X	
0x000F	Probe Response command	6.5.5.7	X	
0x0010–0x0017	Reserved	—	—	
0x0018	Transmit Power Change command	6.5.7.5	X	
0x0019	Array Training command	6.5.10.5	X	
0x001A	Array Training feedback	6.5.10.6	X	
0x001B–0x001D	Reserved	—	—	
0x001E	Security Message command	6.5.10.1	X	
0x001F–0x00FF	Reserved	—	—	
0x0100–0xFFFF	Vendor Defined	6.5.9.2	X	

The requirements for MAC commands based on the PHY type are listed in Table 6-49. The requirements for MAC commands based on the PHY type are listed in Table 6-49. MAC commands not shown in the table are not used by those PHYs listed.

Table 6-49—MAC command usage requirements

Command name	HRCP PHY	THz PHY
Association Request command	Mandatory	Mandatory
Association Response command	Mandatory	Mandatory
Disassociation Request command	Mandatory	Mandatory
Request Key command	Optional	Optional
Request Key Response command	Optional	Optional
Distribute Key Request command	Optional	Optional
Distribute Key Response command	Optional	Optional
Security Information Request command	Optional	Optional

Table 6-49—MAC command usage requirements (*continued*)

Command name	HRCP PHY	THz PHY
Security Information command	Optional	Optional
Probe Request command	Optional	Optional
Probe Response command	Optional	Optional
Transmit Power Change command	Optional	Optional
Array Training command	Optional	Not used
Array Training Feedback command	Optional	Not used
Security Message command	Optional	Optional
Vendor Defined	Optional	Optional

Unless otherwise stated in the command descriptions, in all commands sent between the PNC and a DEV, the SEC field shall be set to zero when the piconet is operating in security mode 0. When the piconet is operating in security mode 1, unless otherwise stated in the command description, the SEC field shall be set to one and the DEV shall be a secure member of the piconet in order to send that command. The ACK Policy field shall be set to Imm-ACK for all commands unless otherwise stated in the command description.

The only commands that may be fragmented are as follows:

- PNC Information command for piconet, as described in 6.5.5.3
- PNC Handover Information command, as described in 6.5.4.4
- PS Set Information Response command, as described in 6.5.9.3

The fragmentation and defragmentation of these commands employ the same method as that used for data frames, as described in 7.10.

6.5.2 Association and disassociation commands

6.5.2.1 General

These commands are used by a DEV to join a piconet or a pairnet and by a DEV, the PRC, or the PNC to end a DEV's membership in the piconet or a pairnet.

6.5.2.2 Association Request command

The Association Request command Payload field shall be formatted as illustrated in Figure 6-168 for piconet and Figure 6-169 for pairnet. The ACK Policy shall be set to No-ACK for pairnet. The SEC field in the Frame Control field shall be set to zero. The DestID shall be set to the PNCID. For piconet operation, the SrcID shall be set to either the UnassocID, as described in 6.2.5, or the DEV's newly allocated DEVID, as described in 7.4.2. For pairnet operation, the SrcID shall be set to the DEVID obtained from the Next DEVID field in the Beacon frame, as described in 7.5.1.

Octets: 6	As defined in 6.4.12	2	1	variable
DEV Address	Overall Capabilities	ATP	DEV Utility	IEs

Figure 6-168—Association Request command Payload field format for piconets

Octets: 6	As defined in 6.4.11b	2	variable
DEV Address	PRDEV Capability	ATP	IEs

Figure 6-169—Association Request command Payload field format for pairnets

The DEV Address field is the address of the DEV, as described in 6.1, requesting association.

The Overall Capabilities field is defined in 6.4.12.

The ATP field is the maximum amount of time in milliseconds that the association relationship will be maintained in the absence of communication between the PNC or PRC and DEV, as described in 7.4.5.

NOTE—It is recommended that a PRDEV should use a short ATP length value less than or equal to 500 ms.

The DEV Utility field shall be formatted as illustrated in Figure 6-170.

Bits: b0	b1	b2–b4	b5–b7
Piconet Services Inquiry	Neighbor PNC	Best PNC TX Quasi-omni Pattern	Reserved

Figure 6-170—DEV Utility field format

The Piconet Services Inquiry field shall be set to one if the associating DEV is requesting that the PNC send the Piconet Services command, as described in 6.5.6.1, and shall be set to zero otherwise.

The Neighbor PNC field shall be set to one if the DEV intends to be a neighbor PNC, as described in 7.2.9, in the current piconet and shall be set to zero otherwise.

The Best PNC TX Quasi-omni Pattern field contains the PNC TX quasi-omni pattern, as determined by the DEV from receiving beacons, that the PNC should use for further quasi-omni transmissions to the DEV.

If the PNC uses AAS, as determined from the received beacon, the DEV may set the Response TX sector field to the index of the beacon that was initially heard by the DEV. It shall be set to zero otherwise.

The IEs field contains zero or more IEs.

6.5.2.3 Association Response command

The Association Response command Payload field shall be formatted as illustrated in Figure 6-171 for a piconet and in Figure 6-172 for a pairnet. The ACK Policy field shall be set to no-ACK for non-PRDEVs and Stk-ACK for PRDEVs. The SEC field in the Frame Control field shall be set to zero. The DestID shall be set to the UnassocID, as described in 6.2.5, for piconet DEVs or the DEVID that has been announced in the Beacon frame, as described in 6.2.6, for PRDEVs. The SrcID shall be set to the PNCID.

Octets: 6	1	2	1	variable
DEV Address	DEVID	ATP	Reason Code	IEs

Figure 6-171—Association Response command Payload field format for piconets

Octets: 6	1	2	1	variable/0
DEV address	DEVID	ATP	Reason code	IEs

Figure 6-172—Association Response command Payload field format for pairnets

The DEV Address field is the address of the DEV, as described in 6.1, requesting association.

The DEVID field is the identifier allocated to the DEV if the association is successful. If this field contains the UnassocID, the DEV is not allowed to associate for the reason indicated in the reason code. For the successful association of a neighbor PNC, the DEVID shall be one of the reserved NbrIDs, as described in 6.2.5.

The ATP field contains the finalized value for the ATP in milliseconds. This value may be different from that requested by the DEV in its Association Request command if the PNC is not able to support the value requested.

NOTE—It is recommended that a PRDEV should use a short ATP length value less than or equal to 500 ms.

The valid values of the Reason Code are as follows:

0 → Success
1 → Already serving maximum number of DEVs (not allowed for PRDEVs)
2 → Lack of available channel time to serve the DEV
3 → Channel too severe to serve the DEV
4 → PNC turning off with no PNC-capable DEV in the piconet (not allowed for PRDEVs)
5 → Neighbor piconet not allowed (not allowed for PRDEVs)
6 → Channel change in progress (not allowed for PRDEVs)
7 → PNC handover in progress (not allowed for PRDEVs)
8 → Association denied
9 → Higher layer denied
10–254 → Reserved
225 → Other failure

The IEs field contains zero or more IEs.

6.5.2.4 Disassociation Request command

The Disassociation Request command Payload field shall be formatted as illustrated in Figure 6-173.

Octets: 1
Reason Code

Figure 6-173—Disassociation Request command Payload field format

The valid reason codes are as follows:

0 → ATP expired
1 → Channel too severe to serve the DEV
2 → PNC or PRC unable to service DEV
3 → PNC turning off with no PNC-capable DEV in the piconet (not allowed for PRDEVs)
4 → DEV leaving piconet or pairnet
5 → Data communication session finished (valid only for PRDEVs)
6 → Higher layer initiated disassociation
7–254 → Reserved
225 → Other failure

The Disassociation Request command shall use the secure command frame format if the DEV is a secure member of the piconet.

6.5.3 Security commands

6.5.3.1 General

This set of commands is used to establish the security and privacy functions between a DEV and the PNC in a piconet, between DEVs in the piconet, and between a DEV and the PRC in a pairnet.

6.5.3.2 Request Key command

The Request Key command is used to request a payload protection key from the key originator. The SEC field in the Frame Control field shall be set to one.

This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Request Key command has no Payload field.

6.5.3.3 Request Key Response command

The Request Key Response command is used by a key originator in a security relationship to send the requested key in an encrypted format to the requesting DEV. The SEC field in the Frame Control field shall be set to one. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The integrity code is generated using the management key that is shared between the requesting DEV and the key originator. The Request Key Response command Payload field shall be formatted as illustrated in Figure 6-174.

Octets: 2	variable
SECID	Encrypted Key

Figure 6-174—Request Key Response command Payload field format

The SECID field is the unique identifier for the security relationship with which the key is associated. The SECID field is used to identify the type of key and the key originator and is defined in 6.2.10.2.

The Encrypted Key field is defined in the symmetric key security operations, as described in 9.3.1.

6.5.3.4 Distribute Key Request command

The Distribute Key Request command is used to transmit a key to another DEV. The SEC field in the Frame Control field shall be set to one. For a piconet, this command may have the ACK Policy field set to no-ACK only if the source ID is the PNCID. For pairnet, this command shall have the ACK Policy field set to Stk-ACK. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Distribute Key Request command Payload filed shall be formatted as illustrated in Figure 6-175.

Octets: 2	variable
SECID	Encrypted Key

Figure 6-175—Distribute Key Request command Payload field format

The SECID field is the unique identifier for the security relationship with which the key is associated. The SECID field is used to identify the type of key and the key originator and is defined in 6.2.10.2.

The Encrypted Key field is defined in the symmetric key security operations, as described in 9.3.1.

6.5.3.5 Distribute Key Response command

The Distribute Key Response command is used in a distribute key protocol to inform the key originator whether or not the key was properly received. The SEC field in the Frame Control field shall be set to one. This command shall be protected using the management key that is shared between the requesting DEV and the key originator. The Distribute Key Response command Payload field shall be formatted as illustrated in Figure 6-176.

Octets: 2
SECID

Figure 6-176—Distribute Key Response command Payload field format

The SECID field is the unique identifier for the security relationship with which the key is associated. The SECID field is used to identify the type of key and the key originator and is defined in 6.2.10.2.

6.5.4 PNC handover commands for piconet

6.5.4.1 General

These commands are used to handover PNC responsibilities.

6.5.4.2 PNC Handover Request command

This command is used by the PNC to hand over its responsibility to another DEV in the piconet that is capable of being a PNC. The PNC Handover Request command Payload field shall be formatted as illustrated in Figure 6-177.

Octets: 1	**1**	**1**	**1**
Number Of DEVs	Number Of Channel Time Request Blocks	Number Of PS Sets	Handover Status

Figure 6-177—PNC Handover Request command Payload field format

The Number Of DEVs field indicates the total number of DEVs that are currently members of the piconet. In addition, this field indicates the number of DEV Information records that will be transfered from the old PNC to the new PNC via the PNC Information command, as described in 6.5.5.3.

The Number Of Channel Time Request Blocks field is the number of Channel Time Request Block fields, excluding requests for asynchronous channel time, currently being served by the PNC that will be transferred from the old PNC to the new PNC via the PNC Handover Information command, as described in 6.5.4.4.

The Number Of PS Sets field indicates the total number of PS sets that will be transferred from the old PNC to the new PNC via the PS Set Information Response command, as described in 6.5.9.3.

The allowed values of the Handover Status field are as follows:

0 → The PNC is starting the PNC handover process with the destination DEV.
1 → The PNC is canceling the handover process with the destination DEV.
2 → The PNC is only transferring information and is not beginning the PNC handover process.
3 → The PNC is starting the PNC handover process with information that was sent previously, as described in 7.2.5.
4–255 → Reserved.

6.5.4.3 PNC Handover Response command

The format of the PNC Handover Response command Payload field shall be as illustrated in Figure 6-178.

Octets: 1
Reason Code

Figure 6-178—PNC Handover Response command Payload field format

The Reason Code field indicates that the new PNC is either ready to take over as the new PNC or that it will be unable to become the PNC. The valid Reason Code values are as follows:

0x00 –› Success, ready for handover
0x01–0xEC –› Success, member of parent piconet with DEVID equal to Reason Code value
0xED–0xF6 –› Reserved
0xF7–0xFC –› Success, associated in parent piconet with NbrID equal to Reason Code value
0xFE –› Handover refused, unable to join parent piconet
0xFF –› Handover refused, unable to act as PNC for more than one piconet

6.5.4.4 PNC Handover Information command

The PNC Handover Information command Payload field shall be formatted as illustrated in Figure 6-179.

Octets: 1	**12**	**2**	**…**	**1**	**12**	**2**
DEVID	Channel Time Request Block 1	Next Beacon	…	DEVID	Channel Time Request Block *n*	Next Beacon

Figure 6-179—PNC Handover Information command Payload field format

The DEVID field contains the identifier of the source of the Channel Time Request Block field that follows in the command.

The Channel Time Request Block field is defined in 6.5.7.2. Note that asynchronous Channel Time Request Block fields are not passed in this command, thus the Num Targets field in the Channel Time Request Block field is always one. Consequently, the Channel Time Request Block fields will all be a fixed length.

The Next Beacon field indicates the beacon number, as described in 6.3.1.1, of the next superframe when this channel time request will be allocated.

6.5.5 Information request commands

6.5.5.1 General

This set of commands is used to obtain information about another DEV in the piconet. The PNC Information Request and PNC Information commands are used to retrieve data about any or all of the currently associated DEVs in the piconet. The Security Information Request and Security Information commands are used to retrieve security information about any or all of the currently associated DEVs in the piconet. The Probe Request and Probe Response commands are used to retrieve IEs from a specific DEV in the piconet.

6.5.5.2 PNC Information Request command for piconet

The DestID for the PNC Information Request command Payload field shall be the PNCID. The PNC Information Request command shall be formatted as illustrated in Figure 6-180.

Octets: 1
Queried DEVID

Figure 6-180—PNC Information Request command Payload field format

The Queried DEVID field contains the DEVID of the DEV whose information is being requested from the PNC. If the value of this field is BcstID, then the DEV is requesting information regarding the entire list of associated DEVs from the PNC.

6.5.5.3 PNC Information command for piconet

This command may be sent either as a response to the PNC Information Request command by a DEV or it may be sent unsolicited. In either case the SrcID shall be the PNCID. This command may be sent either in a directed command frame to a DEV or it may be sent in a broadcast command frame meant for all DEVs in the piconet. If the DestID is BcstID, then the ACK Policy field shall be no-ACK. The PNC Information command Payload field shall be formatted as illustrated in Figure 6-181.

Octets: 18/24	**18/24**	**---**	**18/24**
DEV-1 Info	DEV-2 Info	…	DEV-*m* Info

Figure 6-181—PNC Information command Payload field format

The DEV Info field shall be formatted as illustrated in Figure 6-182.

Octets: 6	**1**	**1**	**As defined in 6.4.12**	**2**	**1**
DEV Address	DEVID	Membership Status	Capability	ATP	System Wake Beacon Interval

Figure 6-182—Format of a DEV Info field in a PNC Information command

The DEV Address field contains the address of the DEV, as described in 6.1, corresponding to the DEVID.

The DEVID field contains the ID assigned to the DEV by the PNC. This field shall not contain the BcstID, the UnassocID, the McstID, or the reserved IDs, as described in 6.2.5.

The DEV Info Utility field shall be formatted as illustrated in Figure 6-183.

Bits: b0	**b1–b7**
Membership Status	Reserved

Figure 6-183—DEV Info Utility field format

The Membership Status field shall be set to zero if the DEV is associated but is not a secure member of the piconet and shall be set to one if the DEV is associated and a secure member of the piconet.

The Overall Capabilities field shall be formatted as illustrated in Figure 6-101 and is defined in 6.4.12.

The ATP field is defined in 6.5.2.2.

The System Wake Beacon Interval field, as described in 6.5.9.4, is the value that the DEV sent to the PNC via the SPS Configuration Request command, as described in 6.5.9.4.

6.5.5.4 Security Information Request command

The Security Information Request command enables a DEV to request security information regarding a single DEV or all DEVs. The Security Information Request command Payload field shall be formatted as illustrated in Figure 6-184.

Octets: 1
Queried DEVID

Figure 6-184—Security Information Request command Payload field format

The Queried DEVID field indicates the DEV whose security information is being requested. If the value of this field is the BcstID, then the DEV is requesting all of the security information maintained by the target DEV.

6.5.5.5 Security Information command

The Security Information command Payload field shall be formatted as illustrated in Figure 6-185.

Octets: 1	**1**	**variable**	**variable**	**…**	**variable**
Total Number Of Frames	Sequence Number	DEV-1 Security Record	DEV-2 Security Record	…	DEV-*m* Security Record

Figure 6-185—Security Information command Payload field format

The Total Number Of Frames field indicates the number of frames that will be sent to complete this request.

The Sequence Number field specifies the number of frames that have been sent prior to this frame by this DEV in response to the initial request. Thus the first frame has a Sequence Number of 0 while the last frame has a Sequence Number equal to the Total Number Of Frames minus one.

A given Security Record field shall be formatted as illustrated in Figure 6-186.

Octets: 2	**6**	**1**	**2**	**variable**
Length	DEV Address	DEVID	Verification Info Length	Verification Info

Figure 6-186—Format of an Security Record field in an Security Information command

The DEV Address field contains the address of the DEV, as described in 6.1, corresponding to the DEVID.

The DEVID field contains the ID assigned to the DEV by the PNC or PRC. If the DEV is not currently associated in this piconet or pairnet, the field shall be set to the UnassocID. This field shall not contain the broadcast or multicast DEVIDs.

The Verification Info Length field indicates the length of the verification information that is included in the Security Record field. If this length is zero, no Verification Info field shall be included.

The Verification Info field specifies the security information that may be used to verify the identity of that particular DEV.

6.5.5.6 Probe Request command

The Probe Request command is used either to request information about a DEV or to see if a DEV is still present in the piconet or pairnet. This command may be exchanged between any two DEVs in the piconet according to the rules outlined in Table 6-50 and Table 6-52. The individual IEs used in this frame are described in 6.4. The Probe Request command Payload field shall be formatted as illustrated in Figure 6-187.

Octets: 4	2
Information Requested	Request Index

Figure 6-187—Probe Request command Payload field format

The Information Requested field shall be formatted as illustrated in Figure 6-188.

Bits: b0	b1–b31
IE Request Type	IEs Requested

Figure 6-188—Information Requested field format

The IE Request Type field indicates the format of the IEs requested field. This field shall be set to zero if the IEs Requested field is a bitmap and shall be set to one if the IEs Requested field is a binary encoding of the IE's ID.

If the IE Request Type field indicates that the IEs Requested field is a bitmap, then the sender shall set a value of one in a bit to request the IE that corresponds to the bit position. Otherwise, the sender shall set the bit to zero. The bit position for an IE is same as the value of the element-ID for that IE. That is, the bit position of n in the information request field corresponds with the IE whose element ID, Table 6-23 for non-PRDEVs and Table 6-24 for PRDEVs, is n.

If the IE Request Type field indicates that the rest of the bits are binary coded, then the IEs Requested field contains the element ID of the IE that is being requested by the sender of this command from its intended recipient.

Both the IE Request Type field and the IEs Requested field shall be set to zero when the source DEV is not requesting any information from the destination DEV.

If the IEs Requested field indicates that the CTA Status IE, as described in 6.4.11, is being requested from the destination DEV, the first octet of the Request Index field is set to the stream index of the stream for which CTA information is requested. If the Request Index field is set to zero, the DEV is requesting information about all isochronous streams directed to the requesting DEV and to the BcstID and McstID. If the Information Requested field indicates that the CTA Status IE is not being requested from the destination DEV, the Request Index field has no meaning and shall be set to zero.

Table 6-50 lists the rules that shall apply to requesting IEs from another non-PRDEV based on the identity of the originator of the request.

Table 6-50—Rules for requesting IEs in a Probe Request command for non-PRDEVs

IE	Subclause	PNC allowed to request?	DEV allowed to request?
CTA IE for piconet	6.4.2	Shall not request	Shall not request
BSID IE for piconet	6.4.3	Shall not request	May request
Parent Piconet IE	6.4.4	Shall not request	May request
DEV Association IE for piconet	6.4.5	Shall not request	Shall not request
PNC Shutdown IE for piconet	6.4.6	Shall not request	Shall not request
Piconet Parameter Change IE	6.4.7	Shall not request	Shall not request
AS IE for piconet	6.4.8	May request	May request
Pending channel time map (PCTM) IE for piconet	6.4.9	Shall not request	May request
PNC Handover IE for piconet	6.4.10	Shall not request	Shall not request
CTA Status IE for piconet	6.4.11	Shall not request	May request
Capability IE for piconet	6.4.12	May request	May request
Transmit Power Parameters IE	6.4.19	May request	May request
PS Status IE for piconet	6.4.20	Shall not request	Shall not request
CWB IE for piconet	6.4.21	Shall not request	Shall not request
Overlapping PNID IE for piconet	6.4.22	May request	Shall not request
Piconet Services IE	6.4.23	May request	May request
Vendor Defined IE or reserved	6.4.24	May request	May request
Group ID IE for piconet	6.4.25	Shall not request	May request
Stream Renew IE for piconet	6.4.26	Shall not request	Shall not request
Next PNC IE for piconet	6.4.27	Shall not request	Shall not request
Piconet Channel Status IE	6.4.28	May request	May request
Synchronization IE for piconet	6.4.29	Shall not request	Shall not request
TSD IE for piconet	6.4.30	Shall not request	Shall not request
UEP Specific IE for piconet	6.4.31	May request	May request
IFS IE for piconet	6.4.32	Shall not request	Shall not request
CTA Relinquish Duration IE for piconet	6.4.33	Shall not request	Shall not request
Feedback IE for piconet	6.4.34	Shall not request	Shall not request
Mapping IE for piconet	6.4.35	Shall not request	Shall not request
BST Clustering IE for piconet	6.4.36	Shall not request	Shall not request

Table 6-50—Rules for requesting IEs in a Probe Request command for non-PRDEVs (*continued*)

IE	Subclause	PNC allowed to request?	DEV allowed to request?
PET Clustering IE for piconet	6.4.37	Shall not request	Shall not request
Beam PET IE for piconet	6.4.38	Shall not request	Shall not request
HRS Beam PET IE for piconet	6.4.39	Shall not request	Shall not request
PET Amplitude IE for piconet	6.4.40	Shall not request	Shall not request
PET Phase IE for piconet	6.4.41	Shall not request	Shall not request
Sync Frame Frequency IE for piconet	6.4.42	Shall not request	Shall not request
Directional Peer IE for piconet	6.4.43	Shall not request	Shall not request

Table 6-51 lists the rules that shall apply to requesting IEs from another PRDEV based on the identity of the originator of the request.

Table 6-51—Rules for requesting IEs in a Probe Request command for PRDEVs

Element	Subclause	PRC allowed to request?	DEV allowed to request?
BSID	6.4.2	Shall not request	May request
PRC Capability	6.4.13	Shall not request	May request
MIMO Information	6.4.44	Shall not request	May request
PRDEV Capability	6.4.14	May request	Shall not request
Pairnet Operation Parameters	6.4.15	Shall not request	May request

6.5.5.7 Probe Response command

The Probe Response command is used to return information about a DEV to a requesting DEV. The individual IEs used in this frame are described in 6.4. The Probe Response command Payload field shall be formatted as illustrated in Figure 6-189.

Octets: variable
IEs Provided

Figure 6-189—Probe Response command Payload field format

The IEs Provided field contains the IEs, as described in 6.4, that the source DEV of this command is providing to the destination. The elements themselves may be placed in any order.

Table 6-52 lists the rules that shall apply to a non-PRDEV responding to a request for an IE based on the sender of the request.

Table 6-52—Rules for responding to requests in Probe commands for non-PRDEVs

IE	Subclause	DEV receives request from DEV	DEV receives request from PNC	PNC receives request from DEV
CTA IE for piconet	6.4.2	Shall ignore	Shall ignore	Shall ignore
BSID IE for piconet	6.4.3	Shall ignore	Shall ignore	Shall respond
Parent Piconet IE	6.4.4	Shall ignore	Shall ignore	Shall respond
DEV Association IE for piconet	6.4.5	Shall ignore	Shall ignore	Shall ignore
PNC Shutdown IE for piconet	6.4.6	Shall ignore	Shall ignore	Shall ignore
Piconet Parameter Change IE	6.4.7	Shall ignore	Shall ignore	Shall ignore
AS IE for piconet	6.4.8	May respond	May respond	May respond
Pending channel time map (PCTM) IE for piconet	6.4.9	Shall ignore	Shall ignore	Shall respond
PNC Handover IE for piconet	6.4.10	Shall ignore	Shall ignore	Shall ignore
CTA Status IE for piconet	6.4.11	Shall ignore	Shall ignore	Shall respond
Capability IE for piconet	6.4.12	Shall respond	Shall respond	Shall respond
Transmit Power Parameters IE	6.4.19	Shall respond	Shall respond	Shall respond
PS Status IE for piconet	6.4.20	Shall ignore	Shall ignore	Shall ignore
CWB IE for piconet	6.4.21	Shall ignore	Shall ignore	Shall ignore
Overlapping PNID IE for piconet	6.4.22	Shall ignore	May respond	Shall ignore
Piconet Services IE	6.4.23	May respond	May respond	May respond
Vendor Defined IE or reserved	6.4.24	May respond	May respond	May respond
Group ID IE for piconet	6.4.25	Shall ignore	Shall ignore	Shall respond
Stream Renew IE for piconet	6.4.26	Shall ignore	Shall ignore	Shall ignore
Next PNC IE for piconet	6.4.27	Shall ignore	Shall ignore	Shall ignore
Piconet Channel Status IE	6.4.28	May respond	May respond	May respond
Synchronization IE for piconet	6.4.29	Shall ignore	Shall ignore	Shall ignore
TSD IE for piconet	6.4.30	Shall ignore	Shall ignore	Shall ignore
UEP Specific IE for piconet	6.4.31	May respond	May respond	May respond
IFS IE for piconet	6.4.32	Shall ignore	Shall ignore	Shall ignore
CTA Relinquish Duration IE for piconet	6.4.33	Shall ignore	Shall ignore	Shall ignore
Feedback IE for piconet	6.4.34	Shall ignore	Shall ignore	Shall ignore
Mapping IE for piconet	6.4.35	Shall ignore	Shall ignore	Shall ignore
BST Clustering IE for piconet	6.4.36	Shall ignore	Shall ignore	Shall ignore
PET Clustering IE for piconet	6.4.37	Shall ignore	Shall ignore	Shall ignore

Table 6-52—Rules for responding to requests in Probe commands for non-PRDEVs (*continued*)

IE	Subclause	DEV receives request from DEV	DEV receives request from PNC	PNC receives request from DEV
Beam PET IE for piconet	6.4.38	Shall ignore	Shall ignore	Shall ignore
HRS Beam PET IE for piconet	6.4.39	Shall ignore	Shall ignore	Shall ignore
PET Amplitude IE for piconet	6.4.40	Shall ignore	Shall ignore	Shall ignore
PET Phase IE for piconet	6.4.41	Shall ignore	Shall ignore	Shall ignore
Sync Frame Frequency IE for piconet	6.4.42	Shall ignore	Shall ignore	Shall ignore
Directional Peer IE for piconet	6.4.43	Shall ignore	Shall ignore	Shall ignore

Table 6-53 lists the rules that shall apply to PRDEVs responding to a request for an IE based on the sender of the request.

Table 6-53—Rules for responding to requests in Probe Request commands for PRDEVs

IE	Subclause	DEV receives request from PRC	PRC receives request from DEV
BSID	6.4.2	Shall ignore	Shall respond
PRC Capability	6.4.13	Shall ignore	Shall respond
MIMO Information	6.4.44	Shall ignore	May ignore
PRDEV Capability	6.4.14	Shall respond	Shall ignore
Pairnet Operation Parameters	6.4.15	May respond	Shall respond

6.5.6 Information announcement commands for piconet

6.5.6.1 Piconet Services command

The Piconet Services command is sent by the PNC to provide information about the application layer capabilities of all of the DEVs in a piconet. The Piconet Services command Payload field shall be formatted as illustrated in Figure 6-190.

Octets: variable	variable	…	variable
Piconet Services IE-1	Piconet Services IE-2	…	Piconet Services IE-*n*

Figure 6-190—Piconet Services command Payload field format

The Piconet Services IE is defined in 6.4.23.

6.5.6.2 Announce command for piconet

The Announce command is used to send unrequested information about a DEV to one or more DEVs in the piconet. The individual IEs used in this frame are described in 6.4. The Announce command Payload field shall be formatted as illustrated in Figure 6-191.

Octets: variable
IEs Provided

Figure 6-191—Announce command Payload field format

The IEs Provided field contains the IEs, as described in 6.4, that the source DEV of this command is providing to the destination. The elements themselves may be placed in any order.

Table 6-54 lists the rules that shall apply to sending an unrequested IE based on the sender of the request.

Table 6-54—Rules for sending IEs in an Announce command

IE	Subclause	PNC allowed to send?	DEV allowed to send?
CTA IE for piconet	6.4.2	Shall not send	Shall not send
BSID IE for piconet	6.4.3	Shall not send	Shall not send
Parent Piconet IE	6.4.4	Shall not send	Shall not send
DEV Association IE for piconet	6.4.5	May send	Shall not send
PNC Shutdown IE for piconet	6.4.6	May send	Shall not send
Piconet Parameter Change IE	6.4.7	May send	Shall not send
AS IE for piconet	6.4.8	May send	May send
Pending channel time map (PCTM) IE for piconet	6.4.9	May send	Shall not send
PNC Handover IE for piconet	6.4.10	May send	Shall not send
CTA Status IE for piconet	6.4.11	May send	Shall not send
Capability IE for piconet	6.4.12	May send	May send
Transmit Power Parameters IE	6.4.19	May send	May send
PS Status IE for piconet	6.4.20	May send	Shall not send
CWB IE for piconet	6.4.21	Shall not send	Shall not send
Overlapping PNID IE for piconet	6.4.22	Shall not send	May send
Piconet Services IE	6.4.23	May send	May send
Vendor Defined IE or reserved	6.4.24	May send	May send
Group ID IE for piconet	6.4.25	May send	Shall not send
Stream Renew IE for piconet	6.4.26	Shall not send	May send
Next PNC IE for piconet	6.4.27	May send	Shall not send

Table 6-54—Rules for sending IEs in an Announce command (*continued*)

IE	Subclause	PNC allowed to send?	DEV allowed to send?
Piconet Channel Status IE	6.4.28	May send	May send
Synchronization IE for piconet	6.4.29	May send	Shall not send
TSD IE for piconet	6.4.30	May send	May send
UEP Specific IE for piconet	6.4.31	May send	May send
IFS IE for piconet	6.4.32	May send	Shall not send
CTA Relinquish Duration IE for piconet	6.4.33	May send	May send
Feedback IE for piconet	6.4.34	May send	May send
Mapping IE for piconet	6.4.35	May send	May send
BST Clustering IE for piconet	6.4.36	May send	May send
PET Clustering IE for piconet	6.4.37	May send	May send
Beam PET IE for piconet	6.4.38	May send	May send
HRS Beam PET IE for piconet	6.4.39	May send	May send
PET Amplitude IE for piconet	6.4.40	May send	May send
PET Phase IE for piconet	6.4.41	May send	May send
Sync Frame Frequency IE for piconet	6.4.42	May send	Shall not send
Directional Peer IE for piconet	6.4.43	May send	May send

6.5.6.3 Announce Response command for piconet

The Announce Response command is used to confirm the receipt of IEs from another DEV. The individual IEs used in this frame are described in 6.4. The Announce command Payload field shall be formatted as illustrated in Figure 6-192.

Octets: 2	2	…	2
Announce Response Block 1	Announce Response Block-2	…	Announce Response Block-*n*

Figure 6-192—Announce Response command Payload field format

The Announce Response Block fields shall be formatted as illustrated in Figure 6-193.

Octets: 1	1
IE Element ID	Reason Code

Figure 6-193—Announce Response Block field format

The IE Element ID field contains the element ID of the IE for which the response is being sent.

The Reason Code field indicates the result of the Announce command for the IE specified in the IE Element ID field. The valid values of the Reason Code field are as follows:

0 –> Success
1 –> Unsupported IE
2–254 –> Reserved
255 –> Other failure

6.5.7 Channel time allocation request, modification, and termination commands for piconets

6.5.7.1 General

This group of commands is used for the request, modification, termination, and grant of channel time within the CTAP.

6.5.7.2 Channel Time Request command

The Channel Time Request command may be used to request, modify, or terminate CTAs corresponding with either isochronous streams or asynchronous data traffic. The Channel Time Request command structure shall be formatted as illustrated in Figure 6-194. The DEV that sends this command is the originator and is seeking from the PNC channel time allocations during which to communicate with a target DEV or DEVs.

Octets: 12–138	12–138	…	12–138
Channel Time Request Block 1	Channel Time Request Block-2	…	Channel Time Request Block-*n*

Figure 6-194—Channel Time Request command Payload field format

Each Channel Time Request Block field corresponds to a channel time request. If the DEV is making a request for asynchronous channel time where the destinations share CTAs, then there shall be only one asynchronous Channel Time Request Block field in the command, and it shall be the last Channel Time Request Block field in the Channel Time Request command. The Channel Time Request Block field for a given Channel Time Request command shall be formatted as illustrated in Figure 6-195.

Octets: 1	1–127	1	1	1	1	2	2	1	1
Num Targets	Target ID List	DSPS Set Index	Stream Request ID	Stream index	Channel Time Request control	CTA rate factor	Channel Time Request TU	Minimum number of TUs	Desired Number Of TUs

Figure 6-195—Channel Time Request Block field format

The Num Targets field indicates the number of target DEVIDs in the target ID list. For isochronous requests, i.e., stream index not equal to the asynchronous stream index, the num targets field shall be set to one. For asynchronous requests, the num targets fields shall take on values from 1 to 127.

The Target ID List field is a series of DEVIDs with which the originating DEV seeks to establish communications by requesting channel time allocations from the PNC.

The DSPS Set Index field is used to identify the DSPS set with which the channel time request corresponds, if the channel time request is for a DSPS allocation. Only valid DSPS set indices, as described in 6.5.9.4, are allowed for a DSPS allocation request. Otherwise, the field shall be set to zero and shall be ignored on reception.

The Stream Request ID field is used to uniquely identify the DEV's request before it receives a stream index from the PNC. If the channel time request is for a new isochronous stream, then the stream request ID is a non-zero identifier generated by the originating DEV that is unique among the DEV's channel time requests. The stream request ID shall remain constant during the entire frame exchange sequence for establishing a new stream. If the channel time request is to modify or terminate an existing stream or the request is for an asynchronous allocation, the stream request ID shall be set to zero and shall be ignored on reception.

The Stream Index field is defined in 6.2.8. In the case where the DEV is requesting the creation of an isochronous stream, it is set to the unassigned stream value, as described in 6.2.8, by the originating DEV. In the case where the DEV is requesting the reservation or termination of an asynchronous channel time, it is set to the asynchronous stream value, as described in 6.2.8. When the stream index is other than the unassigned stream index or asynchronous stream index value, this channel time request is a request to modify or terminate an existing CTA. In the case where the DEV is requesting a specific MCTA interval, as described in 7.6.4.4, the stream index shall be set to the MCTA stream value, as described in 6.2.8, and the Target ID List field shall contain only the PNCID. In the case where the DEV is requesting an allocation for beam forming, the stream index shall be set to beam-forming stream value, as described in 6.2.8.

The Channel Time Request Control field shall be formatted as illustrated in Figure 6-196.

Bits: b0–b2	**b3**	**b4**	**b5**	**b6**	**b7**
User Priority	Reserved	PM Channel Time Request Type	CTA Type	CTA Rate Type	Target ID List Type

Figure 6-196—Channel Time Request Control field format

The User Priority field is defined in Table D.1.

The PM Channel Time Request Type field indicates the type of request. It shall be set to zero to request an ACTIVE channel time allocation and shall be set to one to request a DSPS channel time allocation. For sub-rate allocations, an ACTIVE allocation puts no restriction on the superframe of the first CTA. A DSPS allocation synchronizes all CTAs with the DSPS set awake superframes of the DSPS set specified by the DSPS index. The value of the CTA Rate Factor shall be no smaller than the DSPS set's wake beacon interval.

The CTA Type field indicates whether a pseudo-static CTA is being requested. The CTA Type field shall be set to one if the channel time request is for a pseudo-static CTA and shall be set to zero otherwise.

The CTA Rate Type field indicates whether a super-rate CTA or a sub-rate CTA is being requested. The CTA Rate Type field shall be set to one for a sub-rate CTA and zero for a super-rate CTA.

The CTA Rate Factor field in conjunction with the CTA Rate Type field specifies the frequency at which the requesting DEV would like the PNC to allocate channel time. In the case of a super-rate request, the PNC will interpret the CTA Rate Factor as the maximum spacing allowed between CTAs allocated in a superframe, as described in F.1.3.

For instance, in the case where the CTA Rate Type field is set to zero, a value indicating a super-rate CTA request, and the CTA Rate Factor field contains a value N greater than zero, the requesting DEV is requesting super-rate CTAs from the PNC. If these super-rate CTAs are allocated by the PNC, they will appear *N* times per superframe. A PNC shall support at least 8 CTAs per stream in the same superframe. The CTA Rate Type field set to zero and the CTA Rate Factor field set to zero shall be reserved.

NOTE—A sub-rate request always has a CTA Rate Factor greater than one. Thus a CTA Rate Type equal to one and a CTA Rate Factor equal to one is not allowed.

In the case where the CTA Rate Type field is set to one, a value indicating a sub-rate CTA request, and the CTA Rate Factor field contains a non-zero value *N*, the requesting DEV is requesting sub-rate CTAs from the PNC. If these sub-rate CTAs, are allocated by the PNC, they will appear in the beacon once every *N* superframes. The CTA Rate Factor in this case shall be limited to powers of 2 (i.e., 2, 4, 8, …), up to and including the value of 65 536, which shall be represented by a CTA Rate Factor equal to zero.

If the Channel Time Request Block field is a request for an MCTA interval, only the CTA Rate Factor field and stream index shall be interpreted by the PNC. All other fields except the stream index and num targets fields shall be set to zero.

The Target ID List Type field shall be set to zero for asynchronous group channel time requests and shall be set to one for individual asynchronous channel time requests, as described in 7.7.3.1.

The Channel Time Request Time Unit (TU) field indicates the unit of time that the DEV is using for the CTA(s) it is requesting. This allows the PNC to know the units of channel time the DEV is able to make use of so that the PNC will efficiently allocate channel time. The resolution of this field is 1 μs and therefore has a range of [0–65535] μs.

For an isochronous request, the Minimum Number of TUs field indicates the minimum number of Channel Time Request TUs required by the originating DEV to support the stream.

For an isochronous request, the Desired Number of TUs field indicates the number of Channel Time Request TUs that is desired by the requesting DEV. The Desired Number of TUs field shall be greater than or equal to the Minimum Number of TUs field.

In the case of an isochronous super-rate allocation, the Minimum Number of TUs and the Desired Number of TUs are the number of TUs requested in each superframe. In the case of an isochronous sub-rate allocation, the fields contain the number of TUs requested in each of the superframes containing the sub-rate CTA. For example, a request for a Minimum Number of TUs of 4 with a sub-rate CTA Rate Factor of 4 indicates that the DEV is requesting 4 TUs every fourth superframe. Likewise, a request for a Minimum Number of TUs of 11 with a super-rate CTA Rate Factor of 4 indicates that the DEV is requesting at least 11 TUs per superframe, spread into 4 allocations that are evenly spaced in the superframe.

For an asynchronous request, the concatenation of the Minimum Number of TUs field and the Desired Number of TUs field indicates the total number of TUs that are requested for this allocation, i.e., it is interpreted as a single, 2-octet field. Note that this is a request for a total amount of time rather than a recurring use of time in the superframe. The use of this field is defined in 7.7.3.

6.5.7.3 Channel Time Response command

The Channel Time Response command Payload field shall be formatted as illustrated in Figure 6-197.

Octets: 1	**1**	**1**	**1**
Stream Request ID	Stream Index	Available Number Of TUs	Reason Code

Figure 6-197—Channel Time Response command Payload field format

The Stream Request ID field is defined in 6.5.7.2.

The Stream Index field is defined in 6.5.7.2.

The Available Number of TUs field is used by the PNC to indicate to the requesting DEV the number of TUs per CTA Rate Factor it has assigned to the requested isochronous stream. In the case of a super-rate allocation, it is the number of TUs assigned in each superframe. In the case of a sub-rate allocation it is the number of TUs assigned in each of the sub-rate superframes.

For isochronous channel time requests, if the Available Number of TUs is greater than or equal to the Minimum Number of TUs requested and less than or equal to the Desired Number of TUs requested, then the requesting DEV is informed that there is channel time available. If, however, the Available Number of TUs field is less than the Minimum Number of TUs requested, then the requesting DEV is informed that the PNC is unable to fulfill the DEV's request for channel time. In this case, the Available Number of TUs will be set by the PNC to the number of TUs that the PNC would have been able to allocate for this request, as described in 7.7.2.2.

For asynchronous stream requests, the response frame is sent only if the PNC is unable to fulfill the request, in which case the available number of TUs field is set to zero.

The Reason Code field indicates whether a channel time request was successful or unsuccessful. The codes assignable to this field are as follows:

0 → Success
1 → Success, DEV in PS mode
2 → Target DEV unassociated
3 → Target DEV not a member
4 → Priority unsupported
5 → Stream terminated by PNC
6 → Stream terminated by target DEV
7 → Channel time unavailable
8 → Destination DEV in power save mode
9 → Unable to allocate as pseudo-static CTA
10 → Superframe overloading
11 → Requested super-rate or sub-rate unsupported
12 → Request denied
13 → PNC handover in progress
14 → STP expired
15–254 → Reserved
255 → Other failure

6.5.8 Channel status commands for piconets

6.5.8.1 General

This group of commands is used to request and provide information about the remote DEV's view of the channel and to change the transmitter power based on the current channel conditions.

6.5.8.2 Channel Status Request command

The Channel Status Request command may be sent by any DEV in the piconet to any other DEV in the piconet, including the PNC, to request the current channel condition as experienced at the target DEV. This command may also be sent by the PNC as a broadcast frame, i.e., the DestID set to the BcstID. The Channel Status Request command has no Payload field.

6.5.8.3 Channel status response

The Channel Status Response command is sent by the target DEV in response to the originating DEV's request to let the originating DEV know the current channel condition at the target DEV. When the DestID of this command is the PNCID, the values in the command shall correspond to all frames exchanged by the DEV with other DEVs in the piconet. When the DestID of this command is a non-PNC DEVID, the values in the command shall correspond to the frames exchanged between the requesting DEV and the target DEV. The Channel Status Response command Payload field shall be formatted as illustrated in Figure 6-198.

Octets: 2	2	2	2	2
Measurement Window Size	TX Frame Count	RX Frame Count	RX Frame Error Count	RX Frame Loss Count

Figure 6-198—Channel Status Response command Payload field format

The Measurement Window Size field is the number of superframes during which the measurements were taken. The minimum Measurement Window Size for a valid measurement for this command shall be 2 superframes. A Measurement Window Size of zero indicates that the responding DEV does not provide channel status statistics.

The TX Frame Count field contains the total number of frames, not including Imm-ACK frames, that were transmitted by the sender of this command to the destination of this command. This count includes all transmission attempts, including retransmissions of the same frame.

The RX Frame Count field contains the total number of frames, not including Imm-ACK frames, that were correctly received by the sender of this command, as described in 7.1. Only the directed frames transmitted by the destination of this command intended for the sender of this command are included.

The RX Frame Error Count field contains the total number of frames, not including Imm-ACK frames, that were received in error by the sender of this command from the destination of this command. A frame is considered to have been received in error if the header is correctly received but the frame is not correctly received, as described in 7.1.

The RX Frame Loss Count field contains the number of frames in streams with the ACK Policy field set to no-ACK, not including Imm-ACK frames, that were determined by the originator of the command to have been lost. The originating DEV determines this for a particular stream index by observing gaps in the Fragmentation Control field of received frames. These numbers are accumulated for all streams between the originating DEV and the target DEV.

6.5.8.4 Remote Scan Request command

The SrcID for Remote Scan Request command shall be the PNCID. The Remote Scan Request command Payload field shall be formatted as illustrated in Figure 6-199.

Octets: 1	…	1
Channel 1	…	Channel *n*

Figure 6-199—Remote Scan Request command Payload field format

The Channel Number field indicates the channels that are to be scanned. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY, the mapping is defined in 11.2.3.

6.5.8.5 Remote Scan Response command

The DestID for the Remote Scan Response command shall be the PNCID. The Remote Scan Response command Payload field shall be formatted as illustrated in Figure 6-200.

Octets: 1	1	variable	1	variable	variable
Reason Code	Number Channels	Channel Rating List	Number Piconets	Remote Piconet Description Set	IE

Figure 6-200—Remote Scan Response command Payload field format

The allowed Reason Code field values are as follows:

0 → Success
1 → Request denied
2 → Invalid channel requested
3–254 → Reserved
255 → Other failure

If the request is denied, then the Remote Scan Response command shall include only the Command Type, Length, and Reason Code fields.

The Number Channels field indicates the number of channels that were scanned by the remote DEV.

The Channel Rating List field contains a list of channel indices ordered from best (least interference) to worst (most interference). The Channel Rating List field shall be formatted as shown in Figure 6-201.

Octets: 1	…	1
Best Channel Index	…	Worst Channel Index

Figure 6-201—Channel Rating List field format

The Number Piconets field indicates the number of piconets that were found. If the DEV did not find any piconets, then the number shall be set to zero and there shall be no remote piconet description sets in the command.

The Remote Piconet Description Set is a collection of one or more Remote Piconet Description fields. Each Remote Piconet Description field shall be formatted as shown in Figure 6-202.

Octets: 2	1	1	1	6	2–34	2–40
PNID	Scanned Frame Type	Channel Index	Piconet Type	PNC Address	BSID IE	Parent Piconet IE

Figure 6-202—Remote Piconet Description field format

The PNID is the identifier of the piconet that was found by the DEV.

The Scanned Frame Type field indicates the type of frame that was received by the DEV with the piconet information. The allowed values are as follows:

0 → The DEV found the PNID in a beacon.
1 → The DEV found the PNID only in a non-Beacon frame.
2–255 → Reserved.

The Channel Index field indicates the PHY channel where the information was found. The mapping of the channel number is PHY dependent. For the 2.4 GHz PHY, the mapping is defined in 11.2.3.

The Piconet Type field indicates the type of piconet that was found. The allowed values are as follows:

0 → Non-dependent piconet
1 → Dependent piconet
2–255 → Reserved

The PNC Address field is defined in 6.3.1. If the DEV found a beacon, it shall put the PNC Address from the beacon into the command. Otherwise, the PNC Address field shall be set to zero.

The BSID IE is defined in 6.4.3. If the DEV found a beacon, it shall put this IE into the command. Otherwise, it shall include a BSID IE with zero length, i.e., only the element ID and length fields.

The Parent Piconet IE is defined in 6.4.4. If the DEV found a beacon from a dependent piconet, it shall put this IE into the command. Otherwise, it shall include a Parent Piconet IE with zero length, i.e., only the element ID and length fields.

The IE field is provided for future expansion of the standard. The IE is not defined in this revision of the standard, and so the source DEV may omit the IE in the Remote Scan Response command. The IE shall be ignored upon reception by the destination DEV of this command.

6.5.8.6 Transmit Power Change command

The Transmit Power Change command is used to request a change in the transmit power of a DEV. The Transmit Power Change command Payload field shall be formatted as illustrated in Figure 6-203.

Octets: 1
TX Power Change

Figure 6-203—Transmit Power Change command Payload field format

The TX Power Change field contains the requested TX power level change in dB at the destination DEV in two's complement format. For example, a +2 dB change in the TX power level is 0x02 while a –2 dB TX power level change is encoded as 0xFE.

6.5.9 Power management commands for piconets

6.5.9.1 General

These commands are used to enable DEVs to conserve power as well as by other DEVs that want to know when the DEVs using power management will be available for communication.

6.5.9.2 PS Set Information Request command

The PS Set Information Request command is used to acquire information from the PNC regarding the number of PS sets and their structure. The PS Set Information Request command has no Payload field.

6.5.9.3 PS Set Information Response command

The PS Set Information Response command Payload field shall be formatted as illustrated in Figure 6-204.

Octets: 1	**1**	**8–39**	**…**	**8–39**
Max Supported PS Sets	Number Current PS Sets	PS Set Structure 1	…	PS Set Structure *n*

Figure 6-204—PS Set Information Response command Payload field format

The Max Supported PS Sets field indicates the number of PS sets supported by the PNC of this piconet.

The Number Current PS Sets field is a count of the number of PS Set Structures in this command as well as the number of currently active PS sets in the piconet.

Each PS Set Structure shall be formatted as illustrated in Figure 6-205.

Octets: 1	**2**	**2**	**1**	**1**	**1–32**
PS Set Index	Wake Beacon Interval	Next Wake Beacon	Bitmap Length	Start DEVID	DEVID Bitmap

Figure 6-205—PS Set Structure field format

When the PS Set Index field is zero, the DEVID Bitmap field lists the DEVs currently in APS mode, if any. When the PS Set Index field is one, the DEVID Bitmap field indicates the DEVs currently in PSPS mode, if any. When the PS Set Index field is any value between 0x02 and 0xFD, inclusive, the DEVID Bitmap field indicates the DEVs currently in this particular DSPS set. The PS set indices are defined as follows:

0x00 → APS set
0x01 → PSPS set
0x02–0xFD → DSPS sets
0xFE → Unallocated DSPS set
0xFF → Reserved

The Wake Beacon Interval field is defined in 6.5.9.4. This field is set to the system wake beacon interval for PS sets 0 and 1. For all other PS sets it is set to the wake beacon interval of that DSPS set. Note that the wake beacon interval has no interpretation for PS set 0, as described in 7.17.4.

The Next Wake Beacon field is defined in 6.5.9.5. This field is set to the next system wake beacon for PS sets 0 and 1. For all other PS sets it is set to the next wake beacon of that DSPS set. Note that the Next Wake Beacon field has no interpretation for PS set 0, as described in 7.17.4.

The Bitmap Length field contains the number of octets in the DEVID bitmap. This field shall take on values from 1 to 32, inclusive.

The Start DEVID field indicates the DEVID corresponding to the LSB in the DEVID bitmap.

The DEVID Bitmap field is a bitmap of the DEVIDs in a specific PS set. A value of 0 in a bitmap position indicates that the DEV corresponding to that DEVID is not part of the PS set. A value of 1 in a bitmap position indicates that the DEV corresponding to that DEVID is in the PS set.

6.5.9.4 SPS Configuration Request command

The SPS Configuration Request command is used to set up and manage SPS set memberships for SPS DEVs currently participating or requesting to participate in one or more SPS modes. The SPS Configuration Request command Payload field shall be formatted as illustrated in Figure 6-206.

Octets: 1	**1**	**2**
Operation Type	SPS Set Index	Wake Beacon Interval

Figure 6-206—SPS Configuration Request command Payload field format

The Operation Type field indicates whether a DEV is requesting either to join or to leave an existing SPS set. The valid operation types are as follows:

0 → Join
1 → Leave
2–255 → Reserved

The SPS Set Index field is used to identify the SPS set the requesting DEV wants to create/join, join, configure, or leave. The SPS Set Index field shall not be set to zero (APS) in this command.

Table 6-55 lists the interpretation of the fields for various combinations of the Operation Type field and SPS Set Index field.

The Wake Beacon Interval field contains the number of superframes, including the current one, between wake beacons, as described in 7.17. For example, a wake beacon interval of 8 indicates that the DEV is requesting a wake beacon every 8th beacon, as shown in Figure 7-72. Valid values for the wake beacon interval for either the PSPS or DSPS ranges are in powers of 2 (e.g., 2, 4, 8, …). Furthermore, the wake beacon interval shall have a value between 2 and 256 for PSPS and between 2 and 65536 for DSPS. Because the value 65536 cannot be represented with 2 octets, a wake beacon interval of 0 shall represent the interval value 65536.

Table 6-55—SPS Configuration Request command parameter entries

Operation type	SPS set index value	Wake beacon interval	Comments
0 (create/join)	Unallocated DSPS set (0xFE)	Any valid DSPS wake beacon interval value. This value is decoded by the PNC upon reception.	The unallocated DSPS Set Index (0xFE) shall be used to request the PNC to establish a new DSPS set.
0 (join)	0x01	Any valid PSPS system wake beacon interval value. This value is decoded by the PNC upon reception.	The PSPS mode is permanently associated with PS set index 0x01.
0 (join)	0x02–0xFD	Shall be set to zero and ignored by the PNC upon reception.	The DEV is requesting to join an existing DSPS set.
1 (leave)	0x01–0xFD	Shall be set to zero and ignored by the PNC upon reception.	The DEV is requesting to leave the indicated SPS set.

6.5.9.5 SPS Configuration Response command

The SPS Configuration Response command is sent by the PNC as a response to an SPS Configuration Request command received from a DEV. The SPS Configuration Response command Payload field shall be formatted as illustrated in Figure 6-207.

Octets: 1	1	2
Reason Code	SPS Set Index	Next Wake Beacon

Figure 6-207—SPS Configuration Response command Payload field format

The Reason Code field contains the result of the SPS Configuration Request command. The valid reason codes are as follows:

0 → Success
1 → Already member
2 → Invalid SPS set (i.e., attempting either to configure PS set 0 or to join a non-existing set)
3 → Set creation failed
4 → Unique Wake Beacon Interval required
5 → DSPS set deleted by PNC
6 → PNC handover in progress
7–254 → Reserved
255 → Other failure

The Reason Code field is set to zero (Success), if the create or join operation is successful. If the SPS Set Index field has been set to zero or any value not representing an SPS set, the reason code shall be set to "Invalid SPS Set." If a DEV requests to join a DSPS set where it is already a member, the reason code shall be set to "Already member." However, a DEV is allowed to make multiple requests to join the PSPS set. This has the effect of updating the DEV's desired system wake beacon interval value. If a DSPS set creation fails for any other reason than listed above, the reason code shall be set to "Set creation failed."

The SPS Set Index is defined in 6.5.9.4.

The Next Wake Beacon field indicates the beacon number, as described in 6.3.1.1, of either the next system wake beacon for the PSPS mode when the SPS set index is set to one, or the next DSPS wake beacon when the SPS set index is greater than or equal to 0x02 and less than or equal to 0xFD.

6.5.9.6 PM Mode Change command

The Power Management (PM) Mode Change command Payload field shall be formatted as illustrated in Figure 6-208.

Octets: 1
PM Mode

Figure 6-208—PM Mode Change command Payload field format

The PM Mode field shall be set as follows:

0 → ACTIVE mode
1 → APS mode
2 → SPS mode
3–255 → Reserved

6.5.9.7 PM Mode Change Response command

The PM Mode Change Response command is sent by the PNC as a response to a PM Mode Change Request command received from a DEV. The PM Mode Change Response command Payload field shall be formatted as illustrated in Figure 6-209.

Octets: 1	**1**
PM mode	Reason code

Figure 6-209—PM Mode Change Response command Payload field format

The PM Mode field is defined in 6.5.9.6 and is set by the PNC to indicate current PM mode of the DEV.

The Reason Code field indicates the result of the attempt by the DEV to change PM modes. The valid reason codes are as follows:

0 → Success
1 → Not a member of any existing SPS sets
2–254 → Reserved
255 → Other failure

6.5.10 Special commands

6.5.10.1 Security Message command

The Security Message command is used to send security-related information to another DEV in the piconet or pairnet. The SEC field in the Frame Control field shall be set to zero. The Security Message command Payload field shall be formatted as illustrated in Figure 6-210.

Octets: 3	variable
Unique ID	Security Information

Figure 6-210—Security Message command Payload field format

The Unique ID is defined in 6.4.8.

The Security Information field contains security-related information whose format is defined by the entity indicated in the Unique ID field. Its use by a DEV is outside of the scope of this standard.

6.5.10.2 Vendor Defined command

The Vendor Defined command Payload field shall be formatted as illustrated in Figure 6-211.

Octets: 3	variable
Unique ID	Vendor Defined Data

Figure 6-211—Vendor Defined command Payload field format

The Unique ID field is defined in 6.4.8.

The format of the Vendor Defined Data field is defined by the entity indicated in the Unique ID field. Its use by a DEV is outside of the scope of this standard.

6.5.10.3 AS IE Request command for piconets

The AS IE Request command is used to send an AS IE to the PNC to be put in the beacon. The DestID shall be set to the PNCID. The AS IE Request command Payload field shall be formatted as illustrated in Figure 6-212.

Octets: 1	1	1	variable
Request ID	Request type	AS IE Index	AS IE

Figure 6-212—AS IE Request command Payload field format

The Request ID field is an identifier generated by the originating DEV that is unique among the DEV's AS IE requests.

The Request Type field indicates the type request. The valid values are as follows:

0 → Add
1 → Modify
2 → Remove
3–255 → Reserved

The AS IE Index field is assigned by the PNC and is used to identify the AS IE for the request. If the Request Type field is “add,” then the AS IE Index field shall be ignored by the PNC.

The AS IE field is defined in 6.4.8. If the Request Type field is “remove,” then the length of Application Specific Data field in the AS IE shall be zero.

6.5.10.4 AS IE Response command for piconets

The AS IE Request command is used to respond to a request to put an AS IE in the beacon. The SrcID shall be set to the PNCID. The AS IE Request command Payload field shall be formatted as illustrated in Figure 6-213.

Octets: 1	**1**	**1**
Request ID	AS IE Index	Reason Code

Figure 6-213—AS IE Response command Payload field format

The Request ID field is defined in 6.5.10.3.

The AS IE Index field is defined in 6.5.10.3.

The valid values of the Reason Code field are as follows:

0 → Success
1 → Request rejected
2 → Unknown AS IE index
2–254 → Reserved
255 → Other failure

6.5.10.5 Array Training command for piconets

The Array Training command is used to select a set of antenna elements used in MIMO communication after association is established, as described in 13.2.9.3. The Array Training command is sent repeatedly, so it shall be sent with the ACK Policy field set to No-ACK policy. The Array Training command Payload field shall be formatted as illustrated in Figure 6-214.

3
MIMO Parameter

Figure 6-214—Array Training command Payload field format

MIMO Parameter field shall be formatted as illustrated in Figure 6-215.

Bits:b0–b8	b9–b17	b18–23
Number of Array Training from DEV	Number of Array Training Remained	Reserved

Figure 6-215—MIMO Parameter field format

The Number of Array Training from DEV field contains the total number of Array Training commands to be sent by the DEV in the training sequence encoded as an unsigned integer.

The Number of Array Training Remained field contains the number of Array Training commands remaining to be sent in the training sequence encoded as an unsigned integer.

6.5.10.6 Array Training Feedback for piconets

Array Training Feedback command is used to notify the completion status of Array Training commands. This is sent from a PRC to a DEV. The Array Training Feedback command shall be formatted as illustrated in Figure 6-216.

The list of successfully received training commands field indicates what numbers of Array Training commands are successfully received by the PRC.

The RSSI report field indicates the RSSI value of each received Array Training command signal at the PRC.

If the Resend all Array Training commands field is set to one, the DEV shall resend all Array Training commands.

L1	L2	1
List of successfully received training commands	RSSI report	Resend all Array Training commands

Figure 6-216—Array Training Feedback command Payload field format

Here L1 is equal to ceil(N_{ar}/8). L2 is equal to N_{ar}.

The list of successfully received training commands field is shown in Figure 6-217.

Bits: b0	b1	…	b(N_{ar})-1	0–7
Reception status for Array Training command #1	Reception status for Array Training command #2		Reception status for Array Training command #N_{ar}	0 padding

Figure 6-217—List of successfully received training commands field

Each reception status for Array Training command field is set to one if that command is successfully received otherwise zero.

This field length is an integral multiplication of octets, padding the final block with zeros if necessary.

RSSI report is optional and is as shown in Figure 6-218.

Octets: 1	1	…	1
RSSI of Array Training command #1	RSSI of Array Training command #2	…	RSSI of Array Training command #N_{ar}

Figure 6-218—RSSI report field

Values in the RSSI of Array Training command field are shown in Table 6-56. The resolution of this field is 1 dB and has a range of –71 dBm to –10 dBm.

Table 6-56—Valid Number of RSSI of Array Training command field value

Value	RSSI of the Array Training command [dBm] or reception status
0x00	Not received
0x01	–71
…	…
0x3E	–10
0x3F—0xFF	Reserved

6.5.11 Piconet Multicast configuration commands

6.5.11.1 Multicast Configuration Request command

The Multicast Configuration Request command is used to request a GrpID, 7.7.4. The DestID shall be set to the PNCID. The Multicast Configuration Request command Payload field shall be formatted as illustrated in Figure 6-168.

Octets: 1	6
Action	Group Address

Figure 6-219—Multicast Configuration Request command Payload field format

The valid values of the Action field are as follows:

0 → Join
1 → Leave
2–255 → Reserved

The Group Address field is defined in 6.4.24.

6.5.11.2 Multicast Configuration Response command

The Multicast Configuration Response command is used by the PNC to respond to a request for a GrpID, as described in 7.7.4. The SrcID shall be set to the PNCID. The Multicast Configuration Response command Payload field shall be formatted as illustrated in Figure 6-220.

Octets: 6	1	1
Group Address	GrpID	Reason code

Figure 6-220—Multicast Configuration Response command Payload field format

The Group Address field is defined in 6.4.24.

If the request for a GrpID was successful, the GrpID field is the DEVID, as defined in 6.2.5, that has been assigned by the PNC for the address in the Group Address field. Otherwise, the GrpID field shall be set to zero.

The valid values of the Reason Code are as follows:

0 → Success
1 → Failure, lack of DEVIDs
2 → Failure, handover in progress
3 → Failure, resources unavailable
4 → Failure, not a valid group address
5–254 → Reserved
255 → Other failure

7. MAC functional description

7.1 Introduction

This clause provides a description of the MAC functionality.

In this clause, unless otherwise indicated, receiving a frame means that the PHY has successfully received a frame over the medium and both the FCS and HCS calculations match their respective data, as defined in 6.2.9 and 11.2.9.

Asynchronous MSDUs shall be delivered to the FCSL in the order of reception.

An example message sequence chart (MSC) is shown in Figure 7-1, which illustrates three MLME requests and the associated timeouts. In the first case, the request completes before the timeout expires, and so the confirm returns with the ResultCode set equal to SUCCESS. In the second case, the requested action completes unsuccessfully before the timer expires; therefore, the confirm primitive is returned with the ResultCode set equal to FAILURE, and the ReasonCode indicates the reason for the failure, if known. In the third case, the requested action does not complete before the timeout expires, and so the confirm primitive is returned with the ResultCode set equal to FAILURE and the ReasonCode set equal to TIMEOUT.

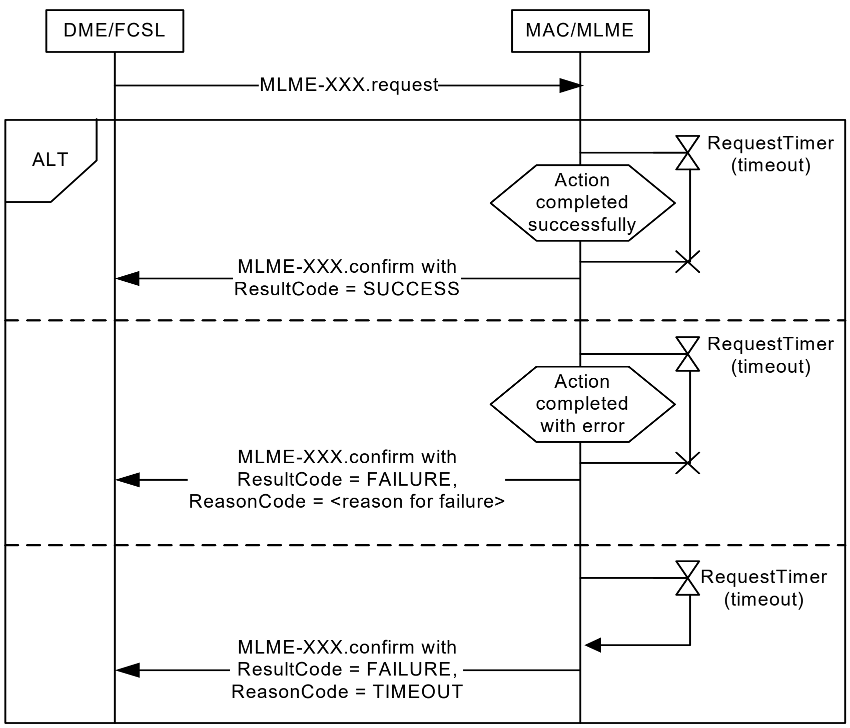

Figure 7-1—MSC showing examples of primitive timers

Throughout this clause, some of the procedures and MSCs are written as though the optional MLME SAP, 5.3, is exposed and is supported by the MLME. Unless stated otherwise, the procedures initiated by an MLME primitive may also be initiated internally in the MAC. For procedures not shown to be initiated by an MLME primitive, the procedure is executed by the MAC without interaction with the DME.

For a MAC header to be correctly received by the MAC, the PHY first verifies the HCS. In addition, the MAC header shall have a protocol revision supported by the MAC; have a DestID equal to a DEVID, BcstID, McstID, or when applicable the PNCID or UnassocID; and have a PNID equal to the PNID of the piconet with which the DEV is synchronized. Because the FCS validation is not required, it is possible for the MAC header to be correctly received even if the frame is not correctly received.

For a frame to be correctly received by the MAC, the MAC header shall be correctly received and the frame payload shall pass the FCS validation. If a DEV correctly receives a frame from an unassociated DEV, it may ignore the frame and may choose not to respond to the frame. If secure membership is required in the piconet and a DEV correctly receives a frame from a DEV that is not a member of the piconet, it shall ignore the frame and shall not respond to the frame, except for the ACK, if the ACK Policy field is set to Imm-ACK, Imp-ACK, or Dly-ACK Request.

7.2 Starting, maintaining, and stopping piconets

7.2.1 General

An IEEE 802.15.3 piconet begins when a PNC-capable DEV takes on the responsibility of being the PNC. The types of piconets defined in this standard are as follows:

— Independent piconet: A piconet with no dependent piconets and no parent piconets.
— Parent piconet: A piconet that has one or more dependent piconets.
— Dependent piconet: A piconet that requires a time allocation in another piconet, called the *parent piconet,* and is synchronized with the parent piconet's timing.

There are two types of dependent piconets. They are as follows:

— Child piconet: A dependent piconet where the PNC is a member of the parent piconet.
— Neighbor piconet: A dependent piconet where the PNC is not a member of the parent piconet.

7.2.2 Scanning through channels

All DEVs shall use passive scanning to detect an active piconet. That is, DEVs shall be in receive mode for a period of time in a channel no less than *mMinChannelScan* to look for Beacon frames or, if supported, Sync frames from a PNC. If a particular BSID, PNID, or PNC address to scan for is not specified with an MLME-SCAN.request, the DEV searches for any Beacon frame or, if supported, a Sync frame. If a particular BSID, PNID, or PNC address to scan for is specified, the DEV shall ignore all received frames not matching the parameter or parameters contained in the request.

In addition, the searching DEV shall collect statistics on each channel scanned and save them in the ChannelRatingList, as described in 5.3.3.

DEVs search for piconets by traversing through all available PHY channels. A DEV may search the channels in any order as long as all valid channels are contained in the search pattern. The result of a piconet scan shall include information on any parent, child, as described in 7.2.8, or IEEE 802.15.3 neighbor, as described in 7.2.9, piconets that were detected. This provides a complete inventory of each channel.

While searching, if any frame type other than a Beacon frame is received, the searching DEV shall stay in the channel for a minimum of *mMinChannelScan* from the time of reception of first frame and look for a beacon from the PNC. The DEV shall scan all indicated channels to find piconets before returning the scan information via the MLME-SCAN.confirm primitive. The DEV shall report only piconets found due to the reception of a Beacon frame or, if supported, a Sync frame as a part of the MLME-SCAN.confirm primitive.

Figure 7-2 illustrates the message flow for a successful scan operation.

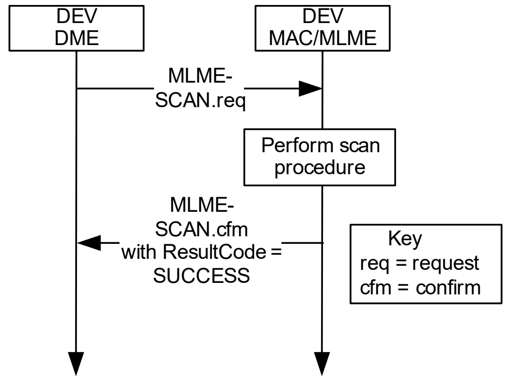

Figure 7-2—MSC for scan operation

7.2.3 Starting a piconet

A DEV that is instructed to start a piconet through MLME-START.request, as described in 5.3.4.2, shall try only to start its own piconet and shall not attempt to associate with an existing piconet.

The MAC should choose the channel with the least amount of interference to start the piconet based on the results of a recent scan (either DME initiated or MAC initiated).

Once the MAC has received the MLME-START.request primitive, it shall listen to the channel for *mMinChannelScan* duration to determine if the channel is still clear. If, at the end of this listening period, the MAC determines that the channel is clear, the DEV, now the PNC, shall commence broadcasting its beacon once every superframe duration. When the piconet starts, the PNC allocates an additional DEVID to itself for the purposes of exchanging data with other DEVs that become members of the established piconet.

Once a PNC has established a piconet, the PNC should periodically allocate channel time in the CTAP so that there is quiet time for it to scan the channel for other piconets. If the PNC detects another piconet in the same channel that is not one of its own child or neighbor piconets, it may take action to improve coexistence with the other piconet. Some of the actions the PNC may take include the following:

— Changing to a different channel, as described in 7.15.2.
— Becoming a child or neighbor piconet of the other piconet, as described in 7.2.8 and 7.2.9.
— Reducing the piconet’s transmit power, as described in 7.15.3.

The PNC shall continue in its role for as long as it desires or until the PNC determines that an associating DEV is more capable, at which time the PNC may decide to initiate the PNC handover procedure, as described in 7.2.4.

If the MAC determines that no channels are available, it will respond with an MLME-START.confirm with a ResultCode of FAILURE and ReasonCode of NO_CHANNELS_AVAILABLE.

Figure 7-3 illustrates the message flow for a successful start operation.

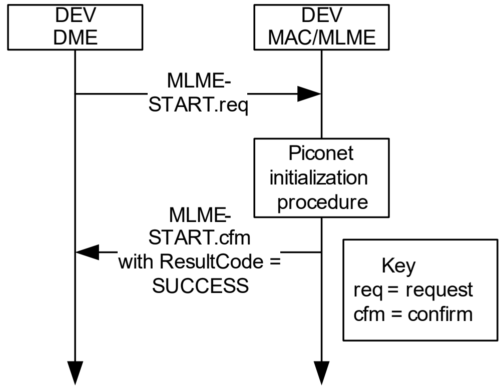

Figure 7-3—MSC for starting a piconet

7.2.4 PNC handover

When the PNC leaves the piconet or when it transfers its PNC functionality to another DEV, the PNC may choose a DEV that is capable of being a PNC as its successor. PNC handover is optional for a PNC using a mmWave PHY. The PNC Capable field in the PNC Capabilities field, as described in 6.4.2, is used to indicate that a DEV is capable of being a PNC. The PNC shall use the information in the PNC Capabilities field of the other DEVs in the piconet with the evaluation criteria defined in Table 7-1 to select the most qualified PNC-capable DEV that is currently a member of the piconet to be the new PNC. The PNC shall send a PNC Handover Request command, as described in 6.5.4.2, to its chosen DEV with the parameters specified in 6.5.4.2. If the piconet is not a dependent piconet, the DEV shall accept the nomination and be prepared to receive the piconet information records. If the DEV is currently the PNC of a dependent piconet, it may refuse the request by sending a PNC Handover Response command to the PNC with the Reason Code field set to “Handover refused, unable to act as PNC for more than one piconet.” If both the current and the new PNC are members of the same dependent piconet, then the DEV shall accept the handover request unless it is unable to join the parent piconet as either a regular DEV or a neighbor PNC. In the case where the DEV is unable to join the parent piconet, the DEV sends the PNC Handover Response command to the PNC with the Reason Code field set to “Handover refused, unable to join parent piconet.”

Table 7-1—Comparison order of fields for PNC handover

Order	Information	Note
1	PNC Des-mode field in PNC capabilities field	PNC Des-mode=1 is preferred
2	SEC field in PNC capabilities field	SEC=1 is preferred
3	PSRC field in PNC capabilities field	PSRC=1 is preferred
4	Maximum associated DEVs	Higher value is preferred
5	Maximum channel time requests	Higher value is preferred
6	Transmitter power level (PHY dependent)	Higher value is preferred
7	Maximum PHY rate (PHY dependent)	Higher value is preferred
8	DEV address	Higher value is preferred

When the handover is initiated, the HandoverStatus is STARTED. If the handover timer expires, a PNC Handover Request command shall be sent to the DEV with a HandoverStatus of CANCELLED. In addition, if the DEV sees a PNC Shutdown IE from the PNC during the handover process, it knows that the handover was canceled.

The PNC may allocate channel time with the chosen PNC-capable DEV as the destination for the purpose of transferring information about the DEVs in the piconet and their current channel time requests. The PNC shall first send a PNC Information command, as described in 6.5.5.3, to the chosen PNC-capable DEV. In the PNC Information command, the PNC shall include all DEVs that are associated in the piconet, including any associated neighbor PNCs, the DEV personality of the PNC, and an entry for the PNCID. Once the PNC has successfully sent this command, it shall then begin sending all of the current channel time requests to the chosen PNC-capable DEV using a PNC Handover Information command, as described in 6.5.4.4. Once the PNC has successfully sent the PNC Handover Information command, it shall send a PS Set Information Response command, as described in 6.5.9.3, to the new PNC. The PNC may fragment the PNC Information, PNC Handover Information, and PS Set Information Response commands using the process described in 7.10.

The PNC Handover Information command shall not be sent if the PNC has indicated in the PNC Handover Request command that it does not have any Channel Time Request Block fields to transfer. The PS Set Information Response command shall not be sent if the PNC has indicated in the PNC Handover Request command that it does not have any PS sets to transfer.

The handover procedure will transfer all information necessary for the new PNC to take over except for the following:

— Asynchronous Channel Time Request Block fields will not be transferred. All DEVs with asynchronous data to send need to send a new Channel Time Request command, as described in 6.5.7.2, to the new PNC after it has sent its first beacon.
— CTA locations are not transferred, except in the preceding beacons.

The current PNC shall not accept any new commands that would change any of the information that will be transferred to the new PNC once the PNC handover process has been initiated. The PNC shall refuse these requests with the Reason Code field set to “Handover in progress” for commands that have this reason code defined.

Once the chosen PNC-capable DEV has received the required information from the current PNC, it shall respond to the current PNC with a PNC Handover Response command, as described in 6.5.4.3. This will signal to the current PNC that the chosen PNC-capable DEV is ready to take over as the new PNC.

After the PNC receives the PNC Handover Response command with Reason Code field indicating success, it shall place a PNC Handover IE, as described in 6.4.10, in the beacon with the Handover Beacon Number field set to the beacon number of the superframe in which the new PNC will send its first beacon. Upon receiving the PNC Handover IE, the chosen PNC-capable DEV will prepare to broadcast its first beacon as the new PNC. After sending the last beacon, the old PNC relinquishes control of the piconet and stops generating beacons. The new PNC shall broadcast its first beacon at the time the beacon would have been sent by the old PNC. This time may vary from the actual time due to clock inaccuracies of the old and new PNCs. The new PNC shall start sending beacons with the time token counter set to one more than the time token of the last beacon that will be sent by the old PNC. The new PNC shall begin using the PNCID as the SrcID for all beacon or command frames transmitted. The new PNC shall use the PNCID or its previously assigned DEVID as the SrcID for all data frames transmitted. When the PNC handover is successful, the association of the remaining DEVs with the piconet is unaffected and hence they are not required to reassociate with the new PNC.

The PNC shall ensure that the handover announcement complies with the rules for beacon announcements in 7.8.5.

Figure 7-4 illustrates the message sequence of a PNC handover to a PNC-capable DEV that is currently a member of the piconet.

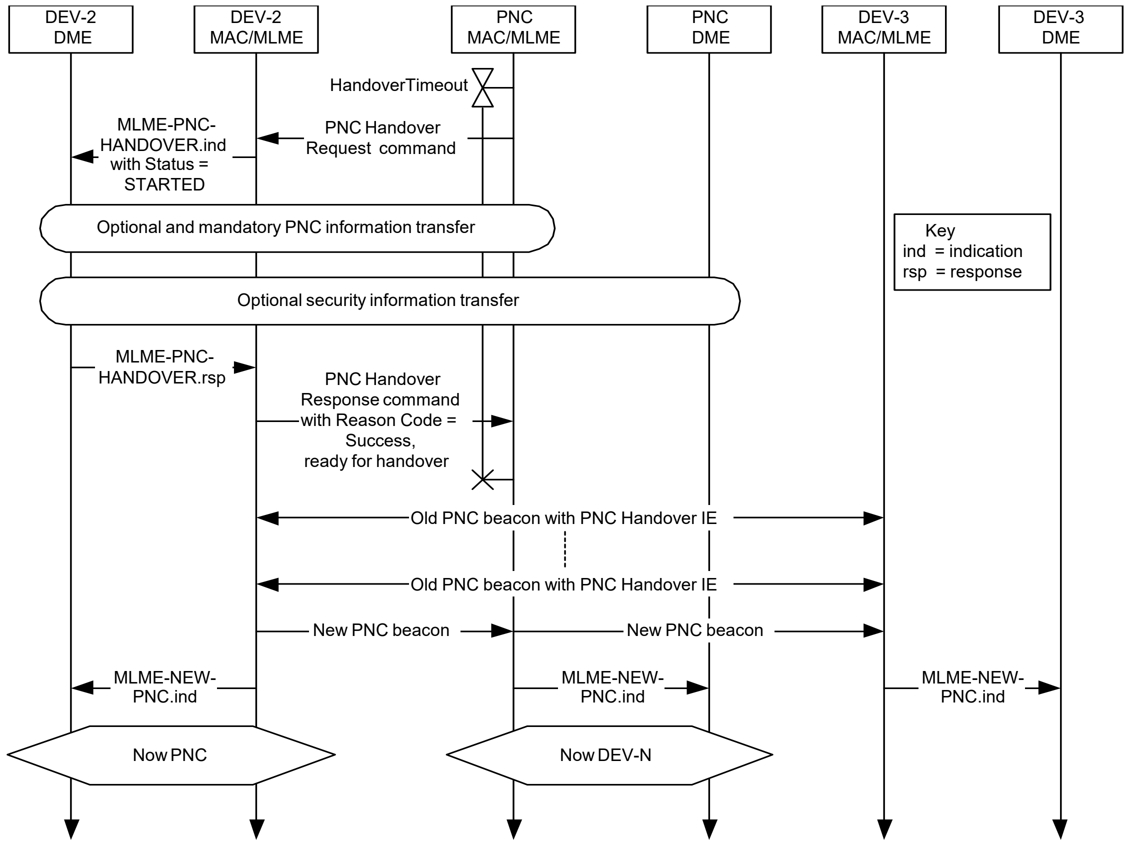

Figure 7-4—PNC handover MSC

The mandatory and optional information transfer for PNC handover is illustrated in Figure 7-5.

The optional security information handover referenced in Figure 7-4 is shown in Figure 8-2.

The DME initiates the handover process using MLME-STOP.request with RequestType set to HANDOVER. This process is illustrated in Figure 7-6. The DME may choose the target DEV or DEVs for the handover or allow the PNC to determine the target DEV. If the handover completes successfully, the MLME-STOP.confirm primitive is generated with a ResultCode set to SUCCESS. If the handover does not successfully complete within the time period specified by the DME, the PNC shall perform the PNC shutdown process defined by 7.2.10. After completion of the shutdown process, the MLME-STOP.confirm primitive is generated with the ResultCode set to FAILURE and the ReasonCode set to HANDOVER_FAILED.

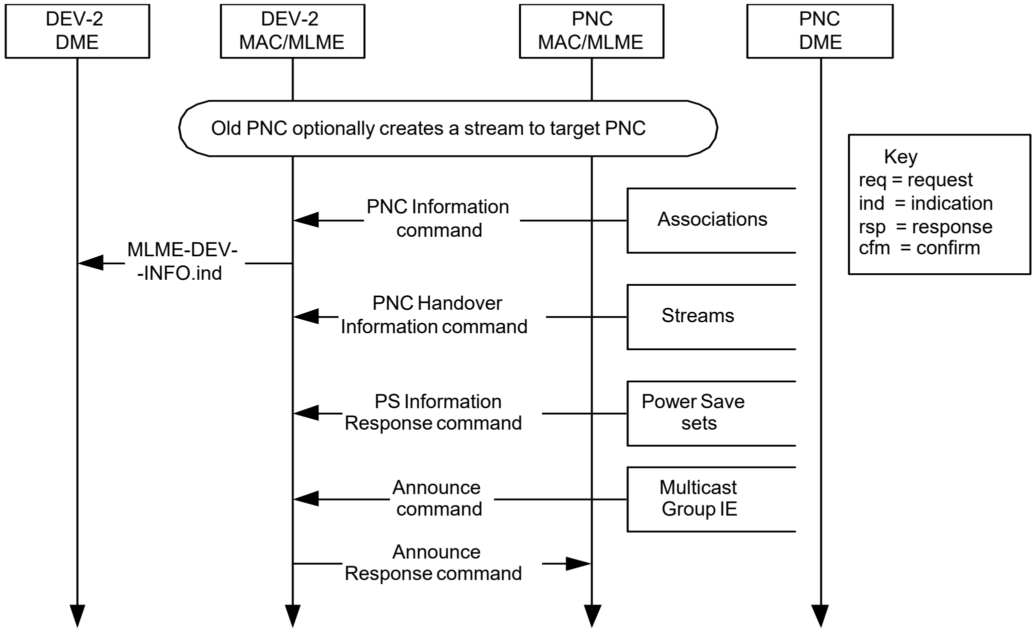

Figure 7-5—Handover information transfer MSC

Note that the PNC handover process should not stop any of the isochronous data connections. Figure 7-4 illustrates only the handover process and hence does not show other traffic. However, since the PNC needs to allocate sufficient channel time to transfer the DEV and channel time request data, some of the data traffic may be affected depending on the traffic conditions within the piconet.

When a DEV joins a piconet, the PNC shall compare the PNC Capabilities field of the new DEV to its own. If the PNC Des-Mode field is set in the new DEV but not in the current PNC, the current PNC shall perform PNC handover. If the new DEV is more qualified to be the PNC, based on the PNC selection criteria in Table 7-1, the PNC may perform PNC handover.

As Table 7-1 shows, PNC Des-Mode is the top priority field in the PNC selection criteria. Since the PNC Des-Mode is the highest priority, a DEV with this field set is more likely to become the PNC of the piconet. Thus, this field should be set if it is desirable for the DEV to be the PNC of the piconet. If only one DEV has the PNC Des-Mode field set, then that DEV would become the PNC.

If the piconet is using mode 1 security, then the new PNC and DEVs need to follow the security procedures in 8.3.2 in addition to the handover process described here.

A dependent PNC receiving a parent beacon with a PNC Handover IE may immediately insert the Piconet Parameter Change IE into its beacons with the Change Type field set to MOVE, as described in 6.4.7, and the Superframe Timing field set to zero. A member of a piconet that receives this Piconet Parameter Change IE in the beacon from the PNC shall not transmit after the superframe that has a beacon number equal one less than the Change Beacon Number field in the Piconet Parameter Change IE until it has correctly received a beacon from its PNC, as described in 7.1. This requirement applies to DEVs in independent as well as dependent piconets.

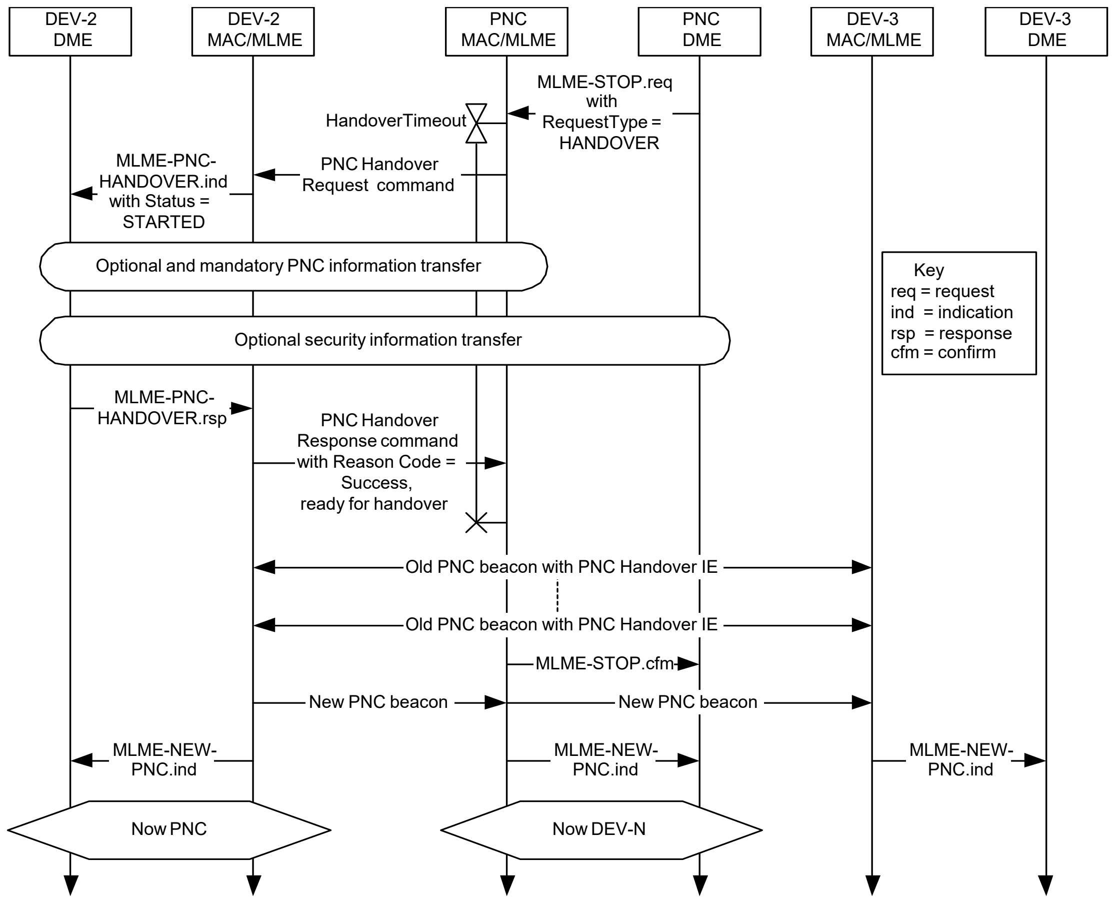

Figure 7-6—DME initiated PNC handover

7.2.5 Preliminary handover

The Preliminary Handover procedure shall not be used by a dependent PNC.

If a DEV has been chosen as next PNC, the PNC shall periodically send handover information to that DEV so that it has the information it needs to take over as PNC. This sending of handover information is known as preliminary handover. In order to send the handover information, the PNC shall send the Handover Request command with the Handover Status field set to Preliminary Handover. The DEV receiving this frame shall store the information for use at a later time, but not take over as PNC at this time.

Information transfer shall take place as shown in Figure 7-5. In addition, any security information may be transferred, as defined in 8.4.2.

When all the information has been received, the next PNC shall send the Handover Response command with Reason Code set to "Success, Ready for Handover."

The current PNC may issue a handover based on previously transferred information by sending the Handover Request command with the Handover Status field set to "Handover Based on Previous Information." Once the chosen PNC-capable DEV is prepared to start operating as PNC, it shall respond to the current PNC with a PNC Handover Response command with Reason Code set to "Success, Ready for Handover."

The handover process now proceeds, as described in 7.2.4, from the point where the Handover Response command is sent.

7.2.6 Next PNC

The Next PNC procedure shall not be used for dependent piconets.

Although the PNC will normally attempt to perform a handover when it shuts down or leaves the piconet, this is not always possible. The PNC could be powered off in such a way that handover is not performed. The PNC could also move out of range of the rest of the DEVs in the piconet, or vice versa. In order to avoid the interruption in communication that would occur when the PNC disappears without handover, the PNC may choose a DEV in the piconet to be the next PNC.

The PNC should periodically evaluate the PNC capabilities of the members of the piconet and other factors to select a DEV to be the next PNC. In addition to the PNC capabilities field, the PNC should use information provided by the DEVs in the Piconet Channel Status IE, as described in 6.4.28, to determine which potential PNC could best be heard by the other DEVs in the piconet. In the context of the Piconet Channel Status IE, "heard" means correctly receiving a MAC header.

A DEV should continuously determine the DEVs that it can hear while it is in the AWAKE state, as defined in 7.17. If MCTAs are used in the piconet, DEVs that are not extremely power sensitive should listen to all of the uplink MCTAs.

The Next PNC IE, as described in 6.4.27, should be periodically announced in the beacon according to the beacon information announcement procedure, as defined in 7.8.5.

The PNC also selects the Next PNID for the Next PNC IE. The PNC shall select a Next PNID such that it is different from any other PNID that has been detected, as described in 7.14.4.

If no other DEV in the piconet is PNC capable, the PNC shall set the Next PNC field to zero and the Next PNID field to the current PNID in the Next PNC IE.

If a DEV has been chosen as next PNC, the PNC shall periodically perform preliminary handover, as defined in 7.2.5.

When the DEV identified as the next PNC fails to detect *mMaxLostBeacons*+1 consecutive beacons, it shall scan the channel for any frames. Since no frames are transmitted in dynamic CTAs or the CAP if the beacon was not correctly received, the next PNC shall use the presence of any frames from the current piconet as an indication that the beacon is still being transmitted but the next PNC has lost contact with the PNC. In this case the next PNC shall not take over the role of PNC for that piconet. If no MAC headers are correctly received during the 2 superframe duration scan, the next PNC shall begin sending out beacons using the same superframe duration and channel. The process of the next PNC taking over as PNC is known as implicit handover. There is a possibility that the next PNC has left the range of all of the other DEVs in the piconet as opposed to the original PNC going out of range or powering off.

The next PNC shall start its beacon after the time it would have expected the beacon from the previous PNC so that if they are both in range of some DEVs, their beacons will not collide. The next PNC should choose a position for the beacon that was unused based on the previous CTAs, to minimize the probability of collision. In order to reduce the possibility of two piconets in close range using the same PNID, the next PNC shall use the announced Next PNID as the PNID of the piconet.

DEVs that are not identified as the next PNC should store the Next PNID from the beacon. When the DEVs that are not identified as the next PNC fail to detect *mMaxLostBeacons*, they should start scanning for beacons. If they do not see beacons with the current PNID, but see beacons with the PNID field set to the

Next PNID and the PNC Address field set to the DEV address that corresponds to the DEVID of the Next PNC, they are automatically associated with that piconet.

7.2.7 Dependent PNC handover

The dependent PNC handover process begins in the same manner as a regular PNC handover, as described in 7.2.4, with the current PNC sending a PNC Handover Request command to the target DEV that it has selected to become the new PNC, as shown in Figure 7-7. In this and the two subsequent figures, the identities of PNC, DEV-2 and DEV-3 are all relative to the dependent piconet and not the parent piconet. If the target DEV is not a member of the parent piconet, then that DEV shall begin the association process to join the parent piconet and, if required, become a secure member of the parent piconet. The target DEV may request to associate with the parent piconet as either a neighbor PNC or a member of the piconet. While the target DEV is attempting to join the parent piconet, the current dependent PNC shall send the target DEV the information about all of the DEVs with a PNC Information command, all of the current channel time requests with a PNC Handover Information command and the power save information, if any, using a PS Set Information Response command. The target DEV may also request the transfer of any security information at this point using a Security Information Request command. Note that the transfer of this information will not interfere with the target DEV's association because the former occurs only during the time reserved for the dependent piconet while the latter occurs only during the time reserved for the parent piconet.

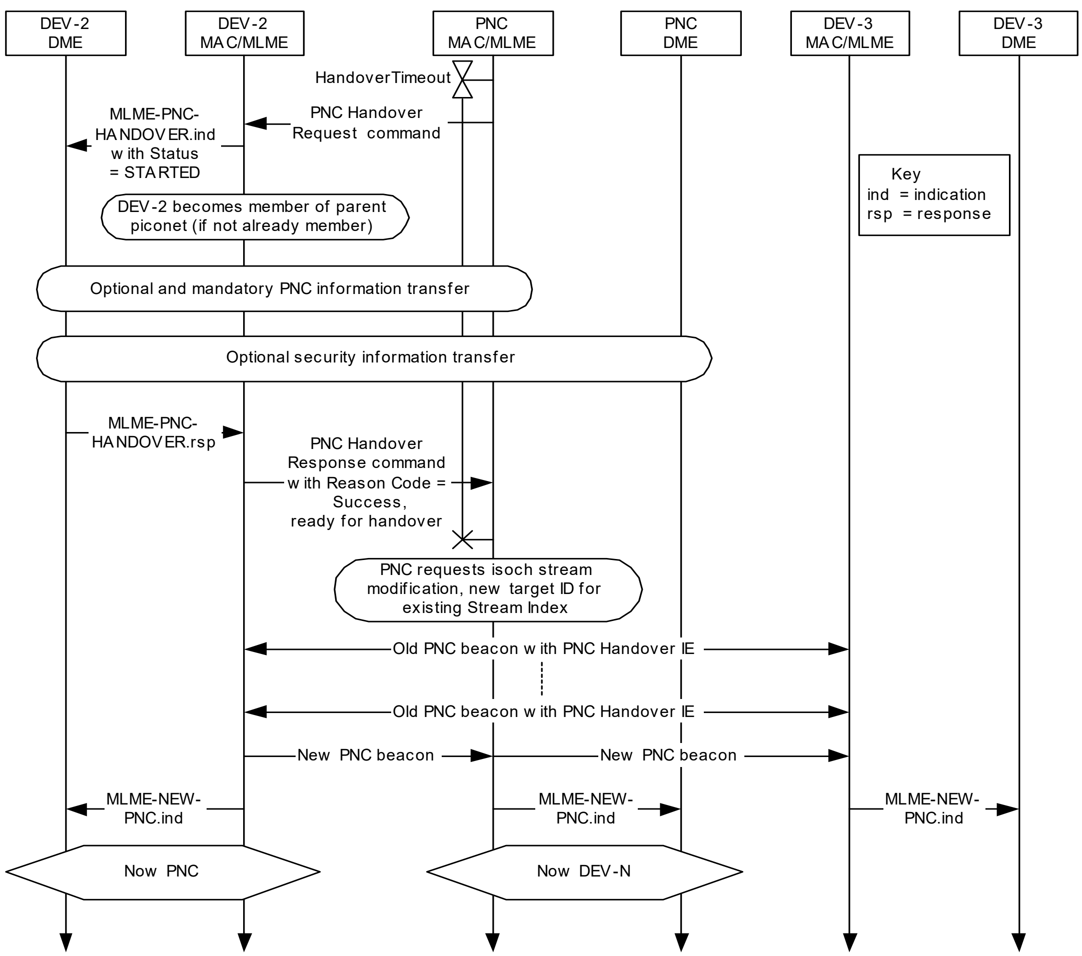

Figure 7-7—Successful PNC handover in a dependent piconet

Figure 7-7 references three processes not included in the figure. The MSC for the optional security information handover is shown in Figure 8-2. The association process that is used to become member of the parent piconet is shown in Figure 7-15, while the stream modification request to hand over control of a CTA is shown in Figure 7-38.

Once the transfer of the information is complete and the target DEV has joined the parent piconet, the target DEV shall send a PNC Handover Response command to the dependent PNC with a Result Code set to the DEVID that was assigned to it by the parent PNC. This informs the dependent PNC that the target DEV is ready to take over control of the piconet. At this point, the dependent PNC shall send a Channel Time Request command to the parent PNC to hand over the control of the dependent piconet CTA to the new dependent PNC, as described in 7.7.2.3. Once the parent PNC changes the SrcID and DestID of the dependent piconet CTA, the current dependent PNC shall either complete the handover process to the new PNC or it shall shutdown the dependent piconet because it will not be able to regain control of the CTA.

After the dependent PNC receives a beacon from the parent PNC with the change in the SrcID of the dependent piconet CTA, the current dependent PNC shall begin placing the PNC Handover IE in its beacon, using the procedure indicated in 7.2.4, with the Handover Beacon Number field set to indicate the first beacon that will be sent by the new PNC. The last superframe controlled by the current dependent PNC will be the one in which the beacon number is one less than the Handover Beacon Number field. The following superframe will begin when the target DEV, now the new dependent PNC, sends its first beacon.

There are multiple points in the handover process where it is possible for the handover to fail. The current dependent PNC may cancel the handover process up until the time when it requests that the parent PNC hand over control of the dependent piconet CTA to the new dependent PNC. The dependent PNC cancels the process by sending a PNC Handover Request command to the target DEV with the Handover Status field set to one to indicate that the process has been canceled.

The handover process will also fail if the target DEV fails to join the parent piconet. If the target DEV attempts to join as a neighbor PNC but the parent PNC does not support neighbor PNCs or does not wish to allow any more neighbor PNCs, then the association request by the new dependent PNC will be rejected. In that case, the DEV may also try to join as a regular DEV, in which case the dependent piconet would become a child piconet after the handover process.

If the target DEV fails to join the parent piconet as either a regular DEV or a neighbor PNC, it shall send a PNC Handover Response command to the dependent PNC with the Reason Code set to “Handover refused, unable to join parent piconet” as illustrated in Figure 7-8. The target DEV may refuse the handover at any time while the dependent PNC is sending the information about the piconet.

If the parent PNC rejects the request to hand over control of the CTA to the new dependent PNC, the dependent PNC shall send a PNC Handover Request command to the target DEV with the Handover Status field set to one to indicate that the handover process is being canceled, as illustrated in Figure 7-9.

If the dependent PNC cancels the handover process, the target DEV may disassociate from the parent piconet. If DEV-2 joined the parent piconet as a neighbor PNC, it should disassociate from the parent piconet if the handover process is canceled to free up that resource for other DEVs that need to form a neighbor piconet.

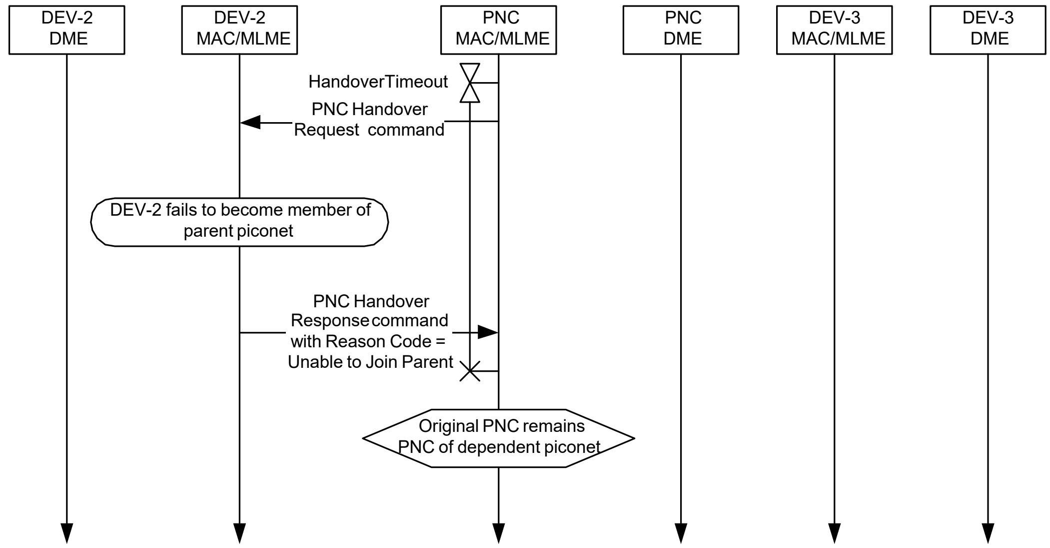

Figure 7-8—Failed dependent PNC handover when target DEV fails to join parent piconet

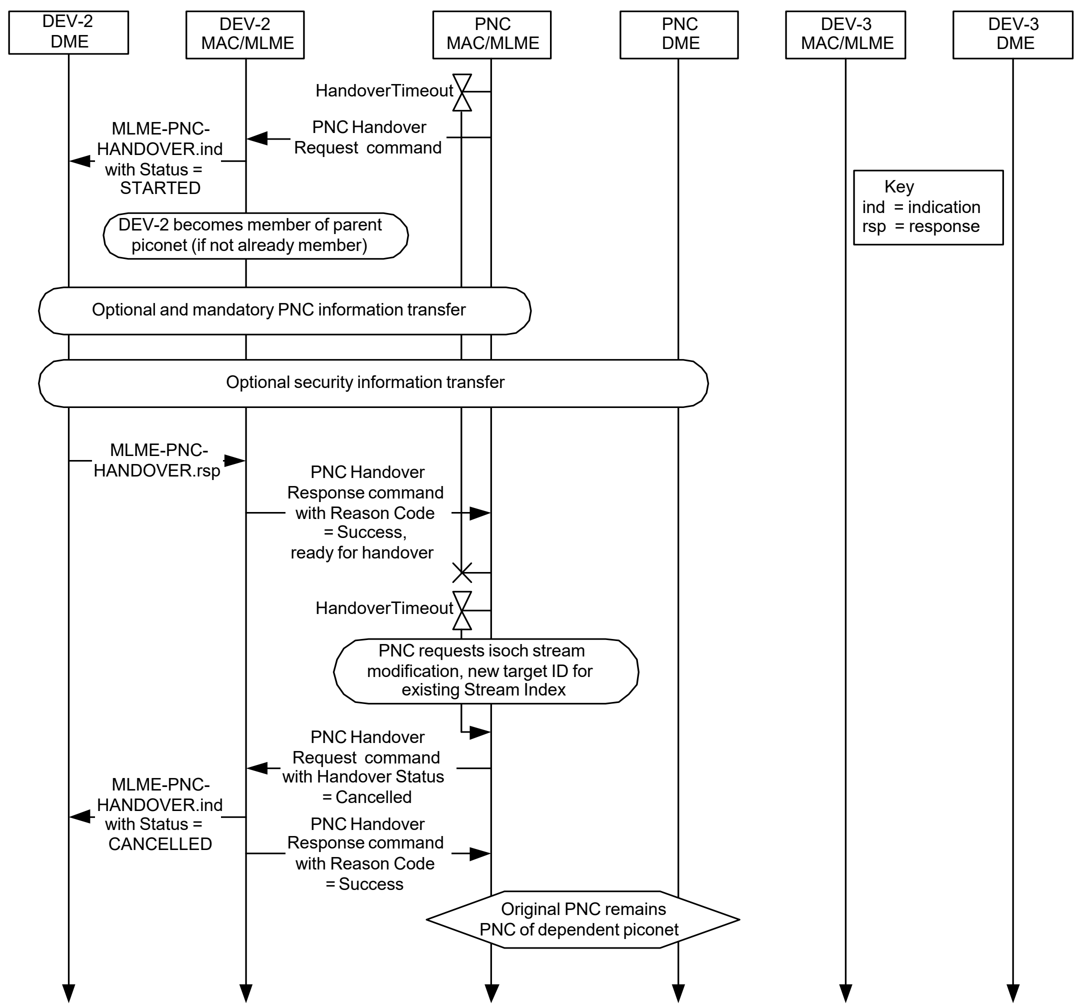

Figure 7-9—Failed dependent PNC handover when control for the dependent piconet CTA is handed over in the parent piconet

7.2.8 Child piconet

When a PNC-capable DEV that is a member of an existing piconet wants to form a child piconet, the DEV shall use the Channel Time Request command, defined in 6.5.7.2 to request a pseudo-static private CTA. A private CTA is a CTA for which the SrcID and DestID are identical. The DEV shall set the Target ID List field in the Channel Time Request command to contain only the DEVID of the originating DEV, the Stream Index field to zero, and the PM Channel Time Request Type field to ACTIVE. The PNC will recognize this as a request for a child piconet. The PNC may allocate a private CTA for the child piconet depending on the availability of network resources, its capabilities, and security policy.

If the PNC rejects the formation of a child PNC for any other reason than insufficient channel time or unable to allocate as pseudo-static, it shall send a Channel Time Response command with the Reason Code field set to "request denied."

The DEV, now the child PNC, shall start sending its beacon in its allocated private CTA. The child PNC shall use a PNID that is distinct from the parent PNID. The child piconet beacon contains a Parent Piconet IE, as described in 6.4.4. Also included in the child piconet beacon is a private CTA for the parent piconet, using the PNCID for both the SrcID and DestID. This is provided to reserve the time, not to convey any information to the parent PNC.

It is possible for more than a single child piconet to be created from a common parent piconet. It is also possible for another dependent piconet to be formed in a child or neighbor piconet. There is no restriction in this standard on the number of levels that may be created. However, there is a practical limitation to the number of dependent piconets and the levels that are able to be supported.

The standard does not provide for the direct frame transfer between a member of a child piconet and a member of a parent piconet. Furthermore, this includes any other child piconets that are dependents of the parent. However, the child PNC DEV is a member of the parent piconet and thus may exchange data with any DEV in the parent piconet. The child PNC DEV is also a member of the child piconet and thus may exchange data with any DEV in the child piconet.

If the child PNC misses *mMaxLostBeacons* parent PNC beacons, the child PNC shall stop transmitting beacons to its piconet. When the child PNC hears the parent's beacons again, it shall resume sending its beacon as long as its ATP has not expired.

The FCSL initiates the formation of a child piconet using an MLME-START.request primitive while a DEV is currently a member of a piconet.

Figure 7-10 illustrates the relationship between the parent piconet superframe and the child piconet superframe. In the figure the superframe periodicity is the same for both the child and the parent piconets. Note that the CTA positions CTA1, CTA2, etc., are not to scale in Figure 7-10 and so are illustrative only.

The MSC for creating a child piconet is illustrated in Figure 7-11. The association and channel time request processes are defined in 7.4.2 and 7.7.2.2, respectively. The MSC for the association process is shown in Figure 7-15 while the MSC for a channel time request is shown in Figure 7-34.

The child piconet is an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association, security, etc. shall be handled within the child piconet and do not involve the parent PNC.

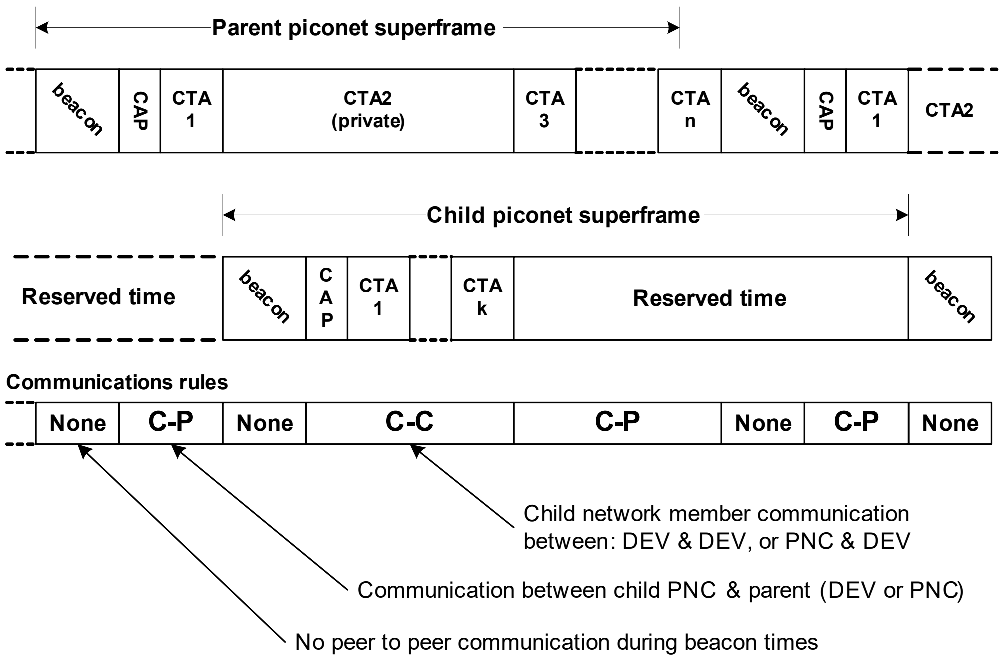

Figure 7-10—Parent piconet and child piconet superframe relationship

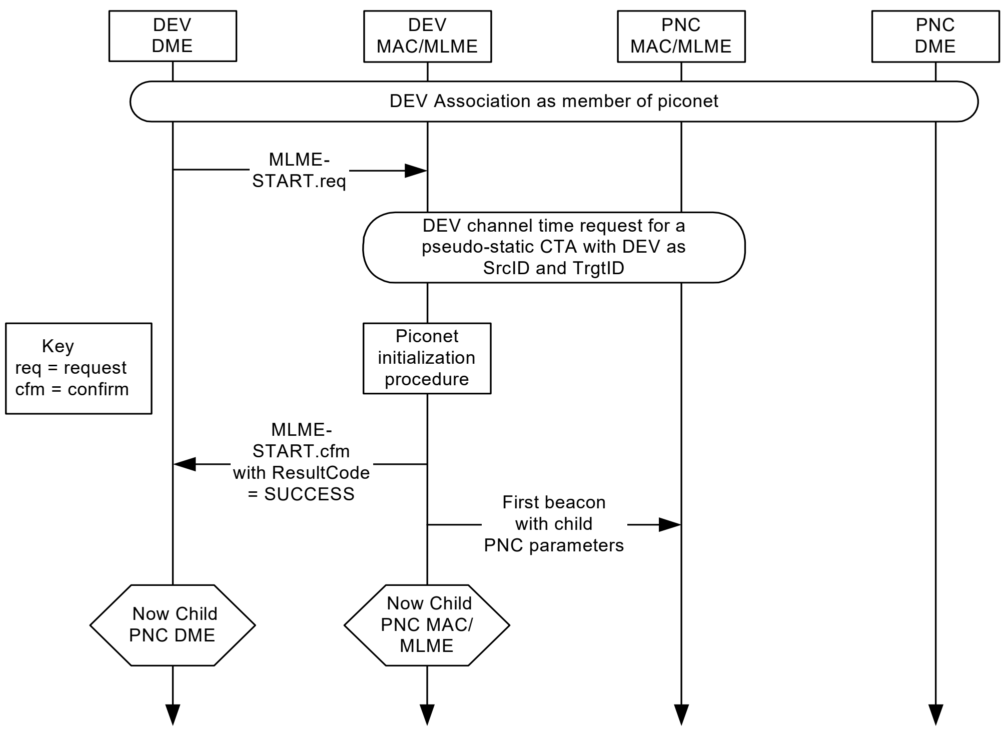

Figure 7-11—MSC for creating a child piconet

7.2.9 Neighbor piconet

If after following the scan procedure in 7.2.2, no free channels are available, then a neighbor PNC-capable DEV (i.e., a PNC-capable DEV from a different system) may attempt to start a neighbor piconet on the same channel as the existing piconet. To start a neighbor piconet, the neighbor PNC-capable DEV shall send an Association Request command, as described in 6.5.2.2, to the PNC. The Neighbor PNC field in the DEV Utility field shall be set as indicated in 6.5.2.2 when the Association Request command is sent. A neighbor PNC is not required to establish a secure relationship with the parent PNC, and so a PNC operating in mode 1 may reject the request for the neighbor piconet.

If the neighbor association request is accepted, then the PNC shall set the DEVID in the Association Response command to be one of the unused NbrIDs, as described in 6.2.5. If the request was rejected, as described in 6.5.2.3, depending on the reason code, the neighbor PNC-capable DEV may retry the request at a later time. If the reason code in the rejection indicates that neighbor piconets are not supported, then the neighbor PNC-capable DEV should not retry the request while that DEV is the PNC of the parent piconet.

After the association request is accepted, the neighbor PNC-capable DEV then sends a Channel Time Request command, as described in 6.5.7.2, to obtain a private pseudo-static CTA for the neighbor piconet. The Channel Time Request command shall have the Target ID List field to contain only the NbrID that was assigned to the neighbor PNC-capable DEV by the PNC in the Association Response command.

If the PNC permits the formation of a neighbor piconet and there is sufficient channel time available, the PNC shall allocate a private CTA using the NbrID as both the source and destination DEVID.

The neighbor PNC-capable DEV, now the neighbor PNC, shall start sending its beacon in its private CTA. The neighbor PNC shall use a PNID that is distinct from the parent PNID. If the neighbor piconet is operating an IEEE 802.15.3 piconet, its beacon shall contain a Parent Piconet IE, as described in 6.4.4.

If the neighbor PNC is operating an IEEE 802.15.3 piconet, a private CTA for the parent piconet is included in its beacon, using the PNCID for both the SrcID and DestID. This is provided to reserve the time for the parent piconet, not to convey any information to the parent PNC.

If the network operated by the neighbor PNC is not an IEEE 802.15.3 piconet, the neighbor PNC shall allow communications in its network only during the time allocated by the parent piconet using methods appropriate to its protocol. It shall ensure that its network does not have transmission outside of its allocated CTA.

If the neighbor PNC misses *mMaxLostBeacons* parent PNC beacons, the neighbor PNC shall stop its own transmissions and the transmissions of the DEVs in its piconet. When the neighbor PNC receives the parent PNC's beacon again, it may return to normal operation as long as its ATP has not expired.

The DME initiates the formation of a neighbor piconet by first using an MLME-ASSOCIATE.request with the NeighborPiconetRequest field set to TRUE. After successfully associating as a neighbor device, the DME uses an MLME-START.request to start neighbor PNC operations.

Figure 7-12 illustrates the relationship between the parent piconet superframe and the neighbor piconet superframe.

The MSC for the initiation of the neighbor piconet is illustrated in Figure 7-13. The association and channel time request processes are defined in 7.4.2 and 7.7.2.2, respectively. The MSC for the association process is shown in Figure 7-15 while the MSC for a channel time request is shown in Figure 7-34.

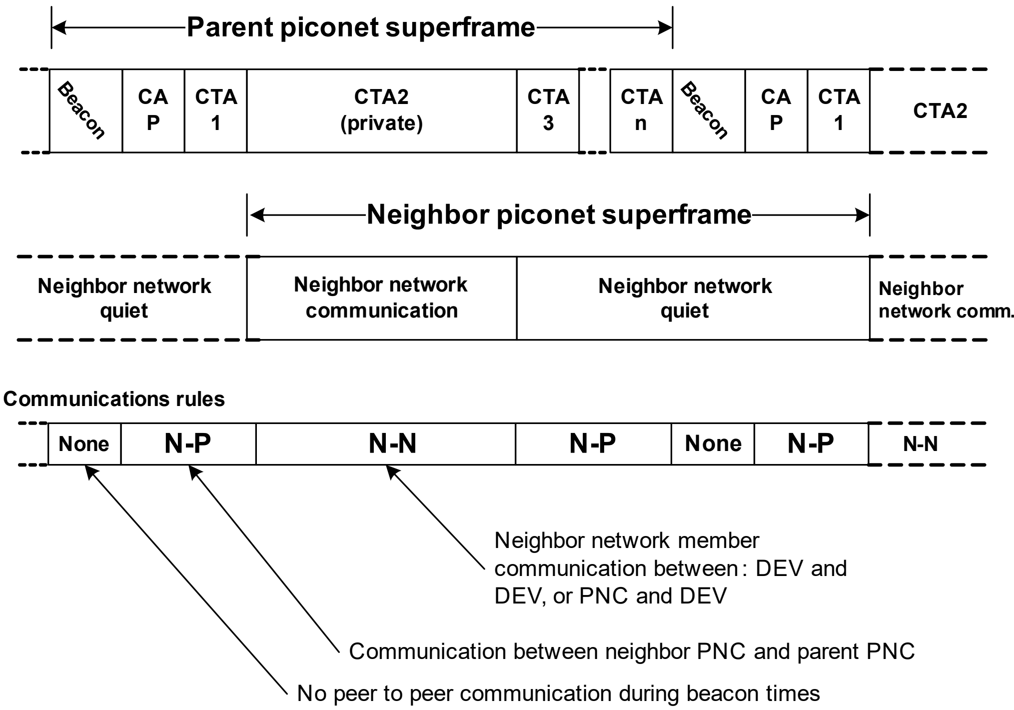

Figure 7-12—Parent piconet and neighbor piconet superframe relationship

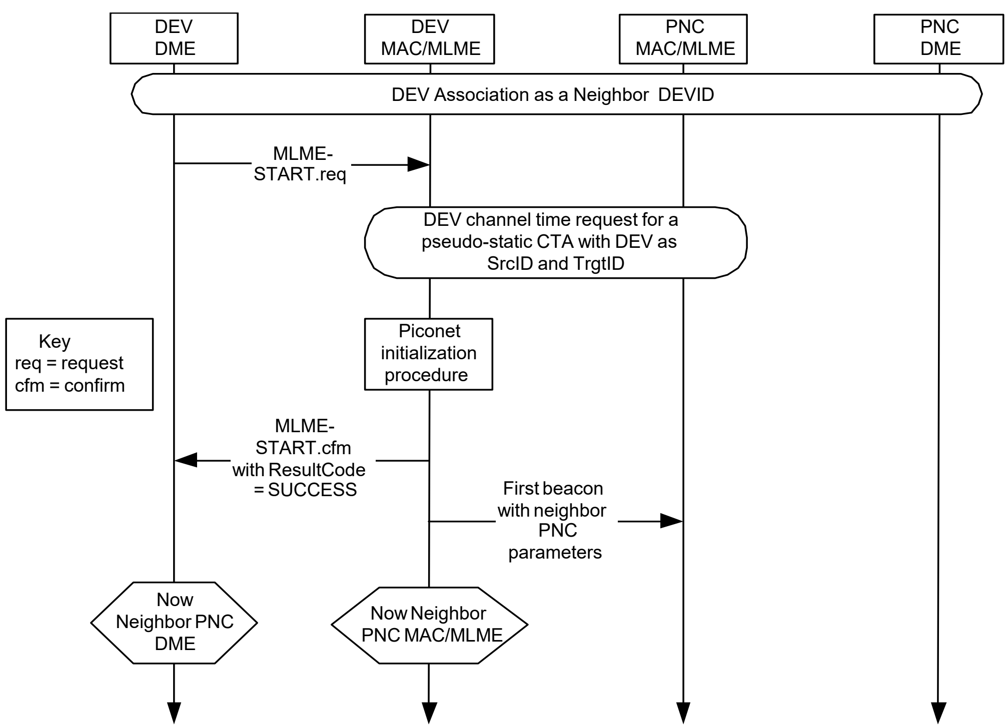

Figure 7-13—MSC for initiating a neighbor piconet

The neighbor piconet is an autonomous piconet except that it is dependent on a private CTA from the parent piconet. Association, security, etc., shall be handled within the neighbor piconet and do not involve the parent PNC.

The neighbor PNC is not a member of the parent piconet and shall only send the following commands to the parent PNC:

- Association Request command
- Disassociation Request command
- Channel Time Request command
- Vendor Defined commands
- Security Message command
- Any Probe Request, Probe Response or Announce commands
- Any required Imm-ACK frames

The parent PNC is not a member of the neighbor piconet.

7.2.10 Stopping piconet operations

7.2.10.1 General

If the PNC is going to leave the piconet and there is no DEV capable of taking over as PNC or there is not sufficient time for a handover, the PNC will shut down the piconet operations. The DME may initiate the shutdown piconet operations using an MLME-STOP.request primitive with the RequestType set to SHUTDOWN.

7.2.10.2 Stopping an independent or parent piconet

If the PNC is going to remove itself from the piconet and no other DEVs are capable of taking over as the PNC, the PNC shall place the PNC Shutdown IE, as described in 6.4.6, in the beacon. The PNC shall ensure that the shutdown announcement complies with the rules for beacon announcements in 7.8.5. The only exception to this requirement is if the PNC will be shutting down and does not have enough time to wait for the next system wake beacon to complete the handover process. The process of stopping a piconet without handing over is illustrated in Figure 7-14.

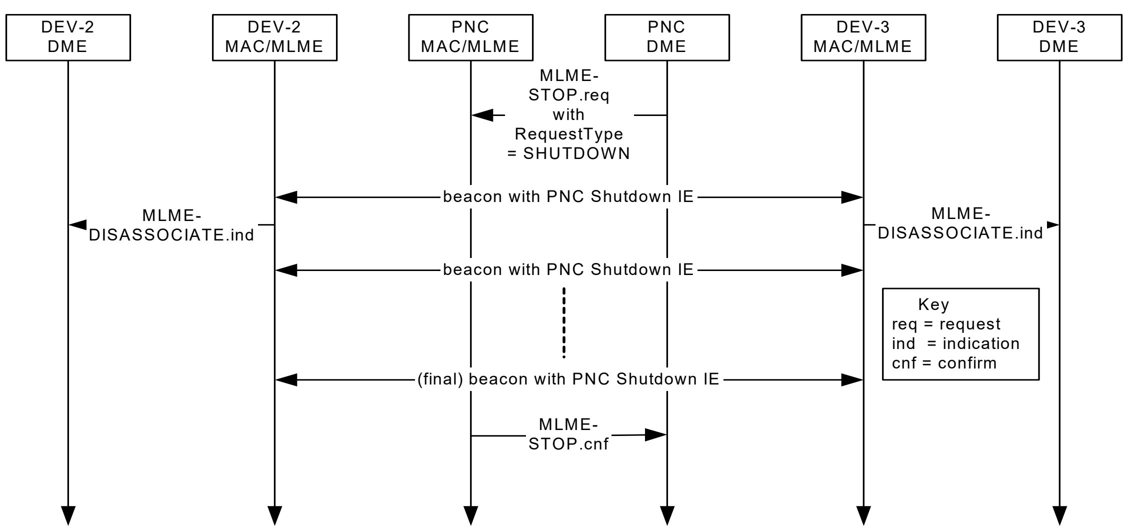

Figure 7-14—MSC for stopping a piconet

If the parent PNC is not itself a dependent PNC and it is currently supporting one or more dependent piconets, the parent PNC shall select the dependent piconet PNC with the lowest DEVID to continue operation without interruption. The parent PNC shall notify the dependent PNC that it may continue operation by including the dependent DEVID in the PNC Shutdown IE, as described in 6.4.6, in the beacon. The other dependent PNCs, not seeing their DEVID in the PNC Shutdown IE, shall either cease operations, change channels, or join another piconet as a dependent piconet by the time of the last beacon sent by the parent PNC. If there is time, the dependent PNC should perform the shutdown procedure for its own piconet.

If the dependent PNC whose DEVID was listed in the parent PNC's PNC Shutdown IE is coordinating an IEEE 802.15.3 piconet, it shall remove the Parent Piconet IE from its Beacon frame, signifying that it is no longer a dependent piconet.

7.2.10.3 Parent PNC stopping a dependent piconet

If the parent PNC wishes to stop a child piconet, it shall terminate the stream allocated to the child piconet using the isochronous stream termination procedure, as described in 7.7.2.4. If the parent PNC wishes to stop a neighbor piconet, it shall either send a Disassociation Request command, as described in 6.5.2.4, to the neighbor PNC or terminate the stream allocated to the neighbor piconet using the isochronous stream termination procedure, as described in 7.7.2.4. In either case, the dependent PNC shall change channels, join another piconet as a dependent piconet, or immediately initiate its shutdown procedure, as described in 7.2.10.2. The parent PNC shall listen for the dependent PNC shutdown beacon sequence to determine when the dependent piconet CTA should be removed. The parent PNC may set a maximum time for the completion of the dependent shutdown sequence, after which the CTA will be removed regardless of the completion of the dependent shutdown procedure. If the dependent PNC is a neighbor that is not operating an IEEE 802.15.3 piconet, the parent PNC shall provide the same time as it would allow for its own shutdown sequence for the neighbor PNC to cease operations as a dependent piconet of the parent piconet before removing its private CTA.

7.2.10.4 Dependent PNC termination of a dependent piconet

After stopping piconet operations for its own piconet, as described in 7.2.10.2, a child PNC shall inform its parent PNC that it no longer requires channel time for child piconet operations by sending the parent PNC a Channel Time Request command terminating the CTA used for the child piconet.

After stopping piconet operations for its own piconet, as described in 7.2.10.2, a neighbor PNC shall inform its parent PNC that it no longer requires channel time for neighbor piconet operations by sending a Disassociation Request command to the parent PNC. Upon receiving a Disassociation Request command from a neighbor PNC, a parent PNC shall remove the CTA used by the neighbor piconet.

7.2.11 Non-PNC-capable DEVs

Simple DEVs may be implemented without providing support for the PNC role in a piconet. The implication of this is that these DEVs would be unable to form a piconet by themselves. Therefore, these DEVs should be of a type that are normally used only in conjunction with a DEV that provides PNC capability.

7.3 Starting and stopping pairnets

7.3.1 Scanning through channels

All DEVs shall use passive scanning to detect an active pairnet. That is, DEVs shall be in receive mode for a period of time in a channel no less than *mMinChannelScan* to look for Beacon frames from a PRC. If a particular PNID or PRC address to scan for is not specified with an MLME-SCAN.request, the DEV searches for any Beacon frame. If a particular PNID or PRC address to scan for is specified, the DEV shall ignore all received frames not matching the parameter or parameters contained in the request.

In addition, the searching DEV shall collect statistics on each channel scanned and save them in the ChannelRatingList, as described in 5.3.3.

DEVs search for pairnets by traversing through all available PHY channels. A DEV may search the channels in any order as long as all valid channels are contained in the search pattern. This provides a complete inventory of each channel.

While searching, if any frame type other than a Beacon frame is received, the searching DEV shall stay in the channel for a minimum of *mMinChannelScan* from the time of reception of first frame and look for a beacon from the PRC. The DEV shall scan all indicated channels to find pairnets before returning the scan information via the MLME-SCAN.confirm primitive. The DEV shall report only pairnets found due to the reception of a Beacon frame as a part of the MLME-SCAN.confirm primitive.

7.3.2 Starting a pairnet

An IEEE 802.15.3 pairnet begins when a PRC-capable DEV takes on the responsibility of being the PRC. Before connecting to any DEV, a pairnet shall disconnect any existing connections.

Before sending any Beacon frames, the PRC-capable DEV, being the PRC, shall initialize the Sequence Number fields for both data frames and command frames using the values given in 6.2.13 and shall initialize the Last Received Sequence Number fields for both data frames and command frames using the values given in 6.3.6.1. For the Beacon frame, the value of the Last Received Sequence Number field shall always be the initial value 0x3FF, and the value of the Last Received Frame Type field shall always be zero.

7.3.3 Stopping a pairnet

If the PRC is going to leave the pairnet, the PRC will shut down the pairnet operations. The DME may initiate the shutdown pairnet operations using an MLME-STOP.request primitive with the RequestType set to SHUTDOWN.

7.3.4 Non-PRC-capable devices

Simple PRDEVs may be implemented without providing support for the PRC role in a pairnet. The implication of this is that these PRDEVs would be unable to form a pairnet by themselves. Therefore, these PRDEVs should be of a type that are normally used only in conjunction with a PRDEV that provides PRC capability.

7.4 Association and disassociation with a piconet

7.4.1 General

Membership in a piconet depends on the security mode of the piconet. For a piconet that does not implement security, as described in 8.2.2, membership occurs immediately upon completion of the association process, as described in 7.4.2. For a piconet that implements security, as described in 8.2.3, membership occurs immediately upon receipt of the MLME-MEMBERSHIP-UPDATE.request primitive with the MembershipStatus parameter set to MEMBER. A DEV is removed from membership in a piconet via the disassociation process.

7.4.2 Association

Before a DEV has completed the association process, all frames sent to the PNC by the DEV shall be exchanged either in the CAP of the superframe or in an association MCTA.

An unassociated DEV initiates the association process by sending an Association Request command, as described in 6.5.2.2, to the PNC. When the PNC receives an Association Request command, it shall send an Association Response command, indicating that the DEV has been associated and the DEVID it has been assigned or that the request has been rejected with the reason for the rejection, as defined in 6.5.2.3. For association using MCTAs, the Association Response command, as described in 6.5.2.3, is sent in an MCTA with PNCID as source and UnassocID as destination.

The PNC shall acknowledge all correctly received Association Request commands, as described in 7.1, by sending an Imm-ACK. The ACK to an Association Request command does not mean that the DEV is associated. The PNC needs some time to ensure that the DEV should be allowed in the piconet, to ensure that there are enough resources available to support another DEV in the piconet, and to allocate a DEVID. The PNC may maintain a list of DEV addresses that are allowed to join the piconet. If the list is in use, when the PNC receives an Association Request command, the PNC shall consult the list to determine if the DEV address in the request is included in the list. If the DEV address is not in the list, the PNC shall send an Association Response command with the reason code set to “association denied,” as described in 6.5.2.3, indicating that the association failed. If the PNC determines that there are not enough resources available to support the new DEV, the PNC shall send an Association Response command with the reason code set to the appropriate value in 6.5.2.3. If the PNC determines that the DEV will be associated, the PNC shall send an Association Response command with the reason code set to “success.” The time difference between when the PNC sends the Imm-ACK to the Association Request command from a DEV and when it sends an Association Response command meant for the same DEV shall not exceed *mAssocRespConfirmTime*.

The Association Response command is not a directed frame. If an Imm-ACK was required for this command, when there were multiple DEVs trying to associate during the same time interval, all of them would try to ACK and collide. Therefore, the ACK Policy field for the Association Response command is set to no-ACK. Instead each DEV trying to associate shall compare its DEV address with the DEV Address field in the Association Response command and if there is a match, accept the DEVID for all future communications.

An unassociated DEV that receives the Association Response command with the DEV address matching its own shall send during the CAP or an association MCTA a second Association Request command with the SrcID field, as described in 6.2.5, set to its newly assigned DEVID. The PNC upon receiving this Association Request command shall respond with an Imm-ACK with the DestID set to the SrcID of the Association Request command. The PNC after acknowledging this second request shall then initialize the DEV Association IE with the requesting DEV’s DEVID, DEV address, DEV Capabilities field, and Association Status field set to “associated.” The requesting DEV upon receiving the Imm-ACK to its second Association Request command shall consider itself associated. All other DEVs that are members of the piconet receiving the beacon containing the DEV Association IE may use the DEV Association IE to update their internal list of associated DEVs in the piconet. The PNC shall ensure that the DEV Association IE announcement for a newly associated DEV complies with the rules for beacon announcements in 7.8.5.

The PNC starts the ATP timer once it has sent the Association Response command for the new DEV. The associating DEV needs to send the second Association Request command before the ATP timer expires. If the PNC receives the second association request command after the ATP timer expires, the PNC shall send the Disassociation Request command, as described in 6.5.2.4, to the DEV requesting association to indicate that it has failed the association process.

Figure 7-15 illustrates the message flow for a successful association process.

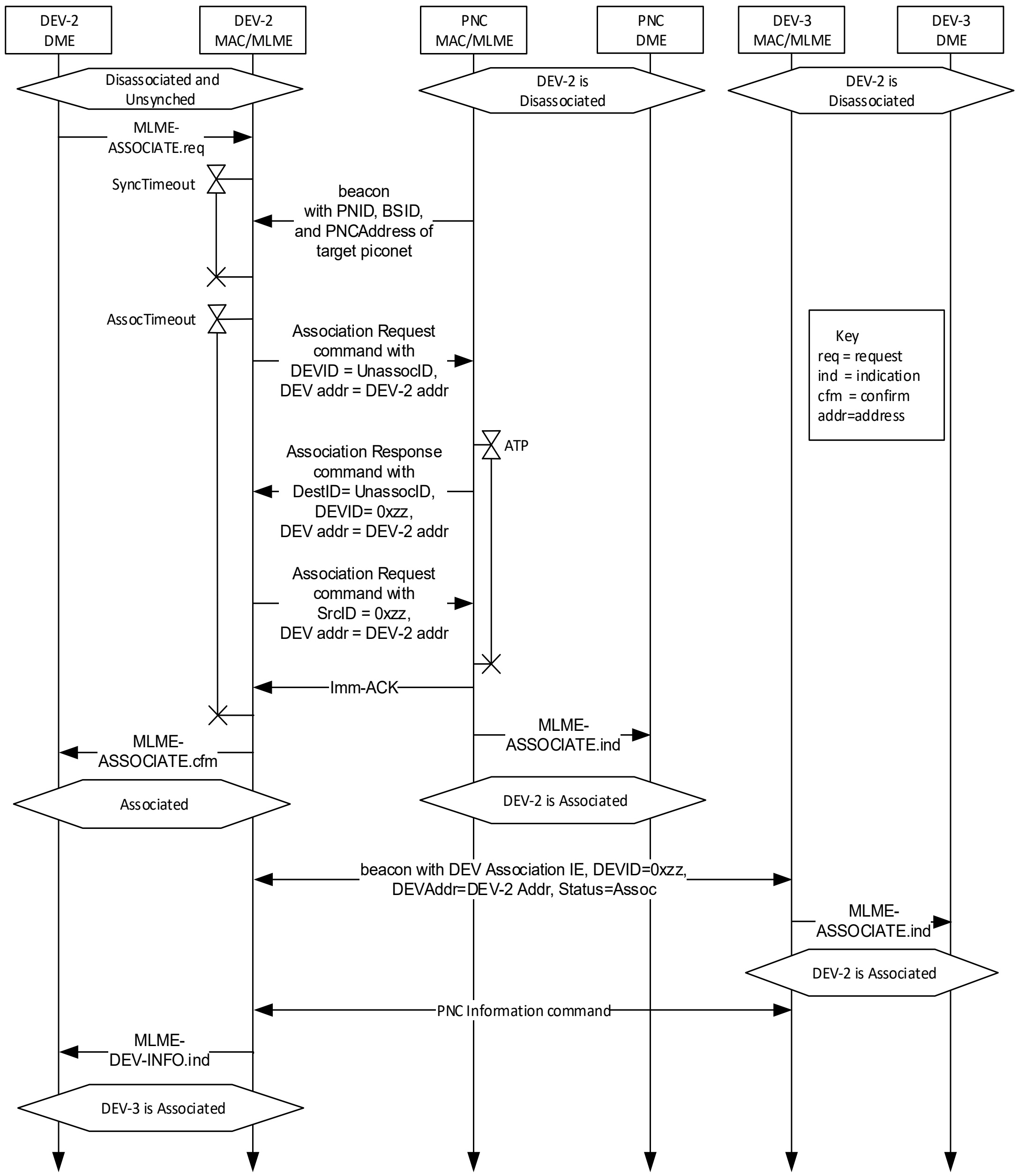

Figure 7-15—MSC of DEV-2 associating

When a PNC allows a DEV to associate, it shall assign a DEVID that is unique within the piconet. The first DEVID assigned should be randomly selected by the PNC. The PNC should assign DEVIDs so as to minimize the likelihood of collision with duplicate DEVID assignments within an interfering piconet and should assign the least recently used DEVIDs available.

NOTE—A simple method to accomplish these two goals is for the PNC to construct a list of available DEVIDs whenever it starts a new piconet. The first DEVID in the list is selected at random, followed by DEVIDs in monotonically increasing order that wrap around to one after DEVID 0xEC. This minimizes the chance that DEVIDs are duplicated if piconets with identical PNIDs interfere with each other. As DEVIDs become available for reassignment, they are added to the end of the list, which is thereby maintained in the least recently used order.

With the exception of the device that contains the PNC, there shall be a one-to-one correspondence between a DEV's device address and its assigned DEVID. The DEV that contains the PNC shall be assigned two DEVIDs: the PNCID shall be assigned to the PNC function within the DEV, and the other value of the DEVID shall be for use for all of the non-PNC traffic. When there is a coordination handover, as described in 7.2.4, the new PNC assumes the PNCID. The former PNC shall continue to use its non-PNCID DEVID for its non-PNC traffic. Hence the PNC is seen as two logical operational entities within the same DEV.

A DEVID released as a result of disassociation (see 7.4.5) shall not be reassigned until twice the disassociated DEV's ATP has elapsed.

After the association process is complete, the PNC broadcasts the PNC Information command, as described in 7.4.4.

7.4.3 Piconet services

Piconet services information is provided by the PNC, if supported, to an associating DEV upon request. In order to request the piconet services information, the associating DEV sets the Piconet Services Inquiry field in the Association Request command with the SrcID set to the newly assigned DEVID, as described in 6.5.2.2. If the DEV sets the Piconet Services Inquiry field, the PNC shall send a Piconet Services command, as described in 6.5.6.1, with DestID set to the newly assigned DEVID after that PNC has received the Association Request command with the SrcID set to the newly assigned DEVID from the DEV, as described in 7.4.2. This process is illustrated in Figure 7-16.

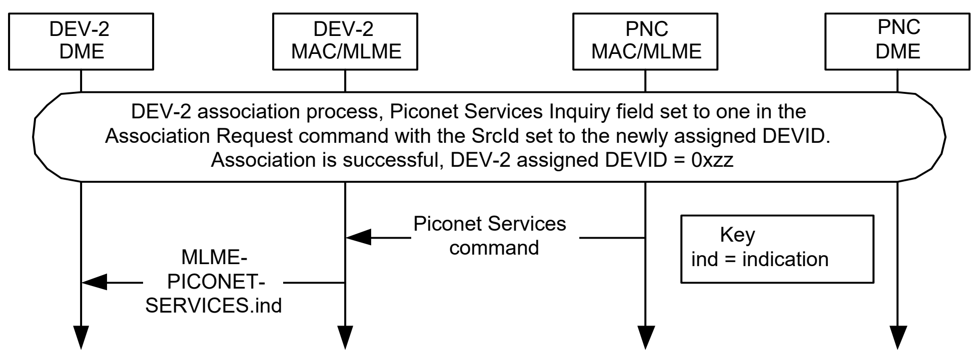

Figure 7-16—PNC sending the Piconet Services command to a newly associated DEV in response to a request in the association process

An associating DEV may inspect the Piconet Services IEs in the Piconet Services command, as described in 6.5.6.1, returned by the PNC to determine information about other DEVs.

DEVs that are members of the piconet may place their own Piconet Services IE in the PNC's record of piconet services by sending the Piconet Services IE to the PNC using the Announce command. The PNC then sends an Announce command with DestID set to the BcstID containing the Piconet Services IE that it has added to its internal record of piconet services. If the PNC supports this capability, it retains the Piconet Services IEs of DEVs that have been sent to the PNC via the Announce command. The PNC will only save Piconet Services IEs for which it has space. Thus it is possible that the PNC would not retain a DEV's Piconet Services IE. After a DEV disassociates from the piconet, the PNC shall delete the DEV's Piconet Services IE from its own record. The process of sending a Piconet Services IE to the PNC is illustrated in Figure 7-17.

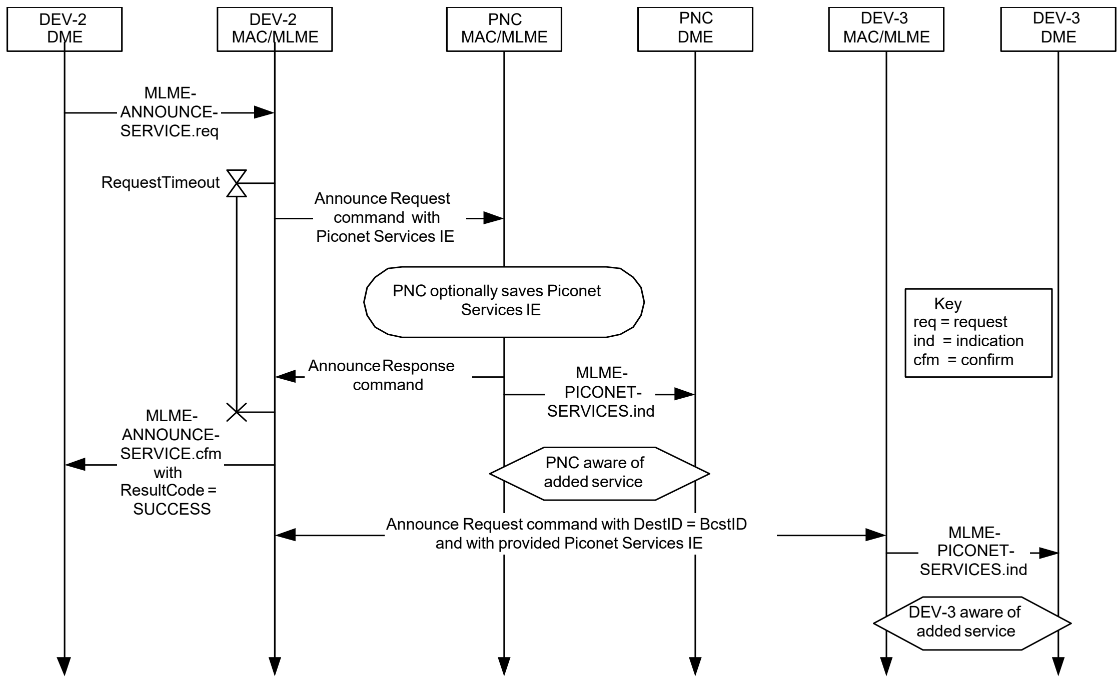

Figure 7-17—DEV sending the Piconet Services IE to the PNC

If a DEV sends a Probe Request command to the PNC requesting the Piconet Services IE, the PNC responds with Probe Response commands that contain all of the Piconet Services IEs that it has in its internal record. If a DEV has not provided a Piconet Services IE to the PNC, the PNC sends the Piconet Services IE in the Probe Response command with a zero length Content field. If the PNC did not have enough space to save the Piconet Services IE that a DEV provided, it shall send in the Probe Response command a Piconet Services IE with length 1, i.e., it contains only the DEVID. If the PNC does not support the Piconet Services IE or if its policy is not to broadcast the Piconet Services IE, it shall respond to the request with a Piconet Services IE with length 2, the DEVID field set to the PNCID, and the Services ID field set to the appropriate value, 6.4.23. The process of requesting the Piconet Services IEs from the PNC is illustrated Figure 7-18.

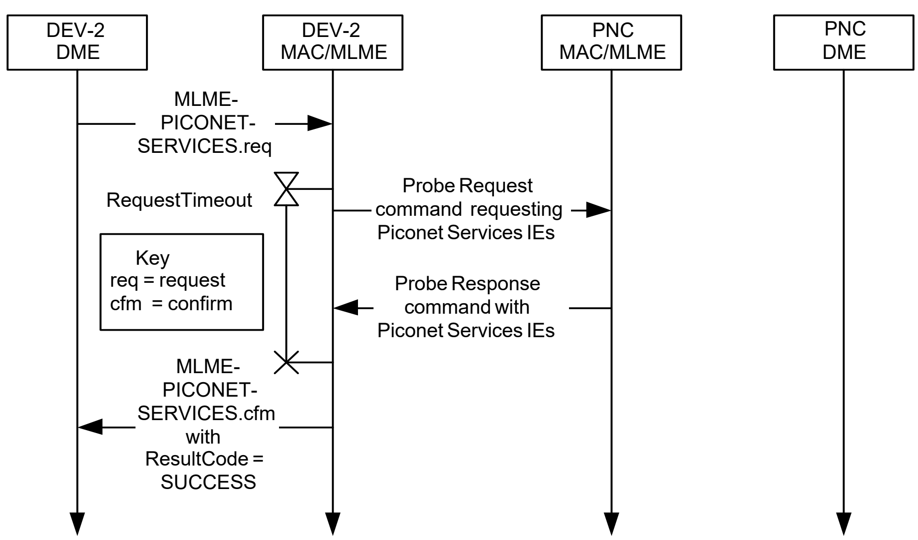

Figure 7-18—DEV requesting Piconet Services IEs from the PNC

If a DEV has a need for privacy, it is not required to provide information that would be available outside of the security operations of the piconet. The MAC PIB element *macDevServicesBroadcast*, as described in 5.4.3, indicates if the DEV will send the Piconet Services field. Likewise, the PNC is not required to furnish this information if it violates the security policy as set in the MAC PIB element *macPncServicesBroadcast*, as described in 5.4.2.

A DEV may use the Probe Request command, as described in 6.5.5.6, to request the Piconet Services IE from another DEV, the target DEV, in the piconet. If the target DEV supports sending the Piconet Services IE and its internal policy allows sending the information, the target DEV shall respond with a Probe Response command with the requested IE. If the target DEV does not support the piconet services IE or if its policy does not allow sending the piconet services IE, the target DEV shall respond with a Probe Response command with a zero-length Piconet Services IE, that is an IE with only the Element ID and Length fields.

It is outside of the scope of this standard to define the content or use of the Piconet Services field.

7.4.4 Broadcasting piconet information

The PNC shall broadcast the piconet information using the PNC Information command, as described in 6.5.5.3, after a DEV becomes a member of the piconet. This means that if security is required for the piconet, the PNC will wait until after secure membership has been established to broadcast the piconet information with the new DEV. In addition, the PNC shall send the piconet information for each of the DEVs that are a member of the piconet and any neighbor PNCs that are associated in the piconet at least once every *mBroadcastDevInfoDuration* via a PNC Information command. When the PNC broadcasts this command, the PNC shall include an entry for the DEV personality of the PNC as well as an entry for the PNCID. The DEV Info field for the PNCID shall contain the same information as the DEV Info field for the PNC's DEV personality with the exception of the DEVID field, which shall be set to the PNCID.

7.4.5 Disassociation

When a PNC wants to remove a DEV from the piconet, the PNC shall send a Disassociation Request command, as described in 6.5.2.4, to that DEV with an appropriate reason code. Similarly when a DEV wants to leave the piconet, the DEV shall send a Disassociation Request command to the PNC with an appropriate reason code.

All Disassociation Request commands, when received correctly, shall be acknowledged by the intended recipient.

All DEVs in the piconet shall send frames to the PNC often enough to assure that the ATP is not reached. If the PNC does not receive any frame originating from an associated DEV within this timeout duration, the PNC shall disassociate the DEV. The DEV may send a Probe Request command without requesting any information to cause the PNC to reset the ATP if the DEV does not have any other traffic that it needs to send to the PNC.

If the DEV is STP capable, as indicated by the STP field in its Capability IE, and is the originator of any allocated streams, it shall send an Announce command that includes the Stream Renew IE to reset the STP of all its streams. This command will also reset the ATP of the DEV.

If the DEV supports reporting the Piconet Channel Status, as defined in 7.2.6, it should include the Piconet Channel Status IE in the Announce command sent to the PNC that is used to reset the ATP.

If the beacons from the PNC are not received by the DEV for longer than the ATP, the DEV shall consider itself disassociated from the piconet and may try to associate again. The DEV notifies the DME that the ATP expired using the MLME-DISASSOCIATE.ind primitive with the ReasonCode set to DEV_ATP_EXPIRED. In addition, if an associated DEV receives a broadcast PNC Information command from the PNC that is missing its DEV Info field, i.e., none of the DEV Info fields (6.5.5.3) contains its DEV address, the DEV shall consider itself disassociated from the piconet.

The PNC shall send a Disassociation Request command to a DEV that sends a frame after its ATP has expired.

The PNC upon receiving a Disassociate Request command or an ATP expiration shall include in the beacon a DEV Association IE. The PNC shall ensure that the DEV Association IE announcement for a disassociated DEV complies with the rules for beacon announcements in 7.8.5. The PNC shall perform the stream termination procedures for each of the assigned CTAs with the disassociated DEV as the SrcID or DestID. The other DEVs that are members of the piconet shall use this information to update their internal DEV association table and to determine whether they will discontinue listening or transmitting in a CTA. Note that when a DEV is disassociated, it loses its DEVID, and so the PNC will reset the bits that refer to this DEVID in all of the relevant bitmaps, e.g., PS Status IE, PCTM IE, CWB IE. The PNC will also remove the disassociated DEV from any PS sets and multicast groups that it has joined and shall delete the Piconet Services IE, if any, for that DEV from its internal storage.

If the DEV is disassociated, the PNC shall terminate all of the streams allocated to the disassociated DEV with its DEVID as either the SrcID or DestID. The PNC follows the process described in 7.7.2.4 except that it does not send any of the commands that would have had the disassociated DEV as the destination.

Figure 7-19 illustrates the message flow for a disassociation initiated by a DEV.

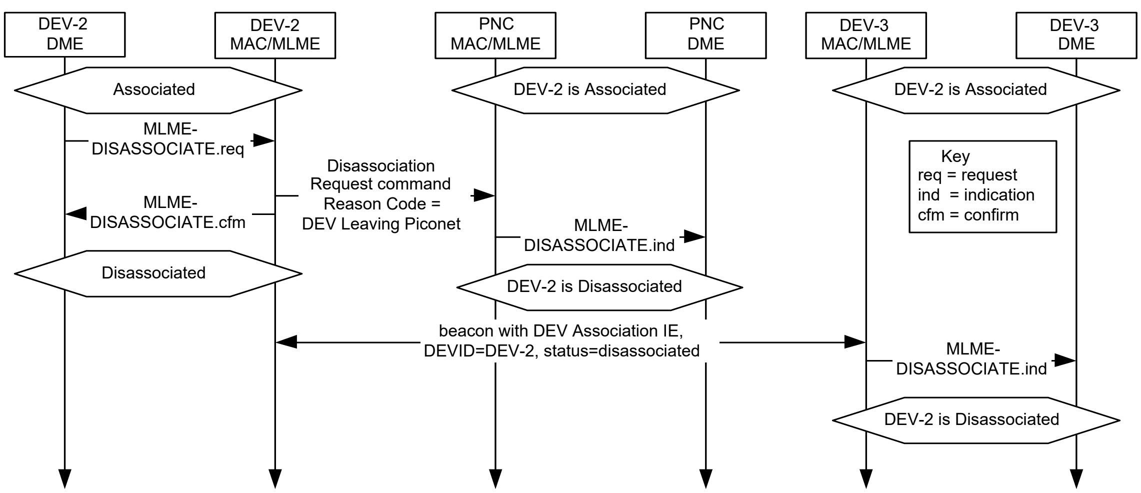

Figure 7-19—DEV-initiated disassociation MSC

Figure 7-20 illustrates the message flow for a disassociation initiated by the PNC.

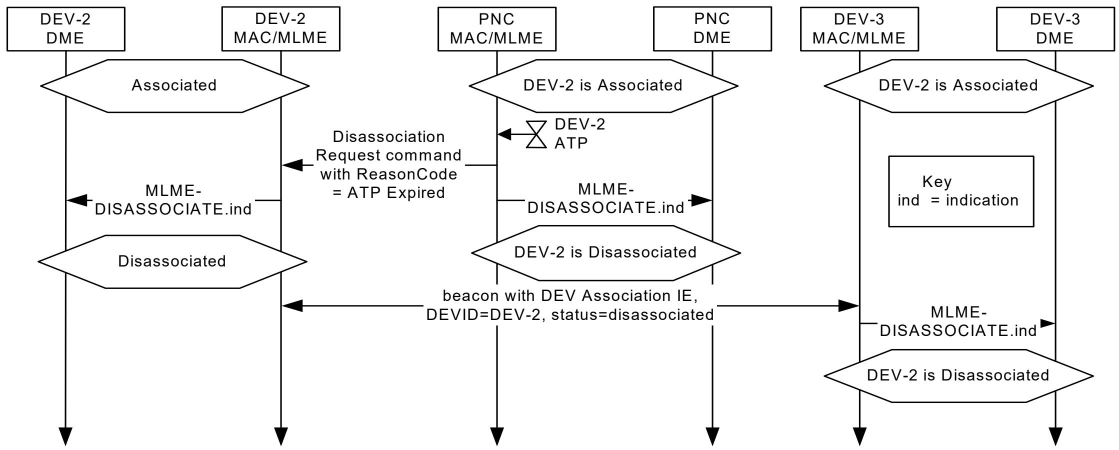

Figure 7-20—PNC-initiated disassociation MSC

7.5 Association and disassociation with a pairnet

7.5.1 Association

To start a pairnet, a PRDEV that is capable of acting as the PRC sends a Beacon frame with a randomly generated Next DEVID. After starting to send Beacon frames, the PRC shall not change the value of the Next DEVID for this new pairnet. If the PRC receives a Beacon frame sent by another PRC, the PRC may ignore this Beacon frame to continue sending Beacon frames as a PRC or stop sending Beacon frames to become a PRDEV. This selection is implementation dependent.

Before a PRDEV has completed the association process, all frames sent to the PRC from the PRDEV shall be exchanged in the Access Slots in PSP. An unassociated PRDEV initiates the association process by sending an Association Request command, as described in 6.5.2.2, to the PRC. Before sending an Association Request command frame, the PRDEV shall initialize the Sequence Number fields for both data frames and command frames and shall initialize the Last Received Sequence Number fields for both data frames and command frames. An unassociated PRDEV can send an Association Request command by selecting an access slot at random, after receiving every Beacon frame, and starting the Association timeout timer. The duration of an Access Slot consists of the length of an Association Request command and a SIFS. Carrier sense for sending an Association Request command is not required. Association Request commands shall be sent with No-ACK policy. When the PRC receives one of the Association Request commands, whose DEVID is the same value as the Next DEVID in the Beacon frame, it shall stop sending the Beacon frame, sending an Association Response command instead with the same timing as the Beacon frame or later, and start the Association timeout timer. If the PRC receives an Association Request command whose DEVID does not match the value of Next DEVID in the beacon frame, the PRC shall not respond to the command and shall continue sending beacons. If a PRDEV receives the Association Response command with the DEV address matching its own, the PRDEV becomes an associated PRDEV and sends the Stk-ACK to the Association Response command to the PRC. If the DEV address in the Association Response command does not match with the PRDEV's own DEV address, the PRDEV shall ignore the command. If an associated PRDEV or an unassociated non-PRC PRDEV that has initiated an association process by sending an Association Request command receives a Beacon frame with a different Next DEVID, it shall ignore this Beacon frame. The PRC may maintain a list of DEV addresses that are allowed to join the pairnet. If the list is in use, when the PRC receives the Association Request command, the PRC shall consult the list to determine if the DEV address in the request is included. If the DEV address is not in the list, the PRC shall send an Association Response command with the reason code set to "Association denied," as described in 6.5.2.3, indicating that the association failed.

The associated PRDEV shall start an Association timeout timer once it has sent a Stk-ACK and the PRC shall start an Association timeout timer once it has sent an Association Response command.

NOTE—No new primitives are defined. The reception of a Stk-ACK to the Association Response command may be sent to the PRC DME from the PRC MAC/MLME but the method is implementation dependent.

The MSC of a DEV associating with a PRC is shown in Figure 7-21.

The PRC determines that the association procedure has completed when it either receives: a) a Stk-ACK in response to the Association Response command, as shown in Figure 7-22, or b) a data frame from the DEV, as shown in Figure 7-23. The DEV that receives an Association Response command from the PRC determines that the association procedure has completed and transmits a Stk-ACK in response to the Association Response command. When a Stk-ACK or data frame is not received by the PRC after it sends an Association Response command, the PRC should change the state to Asynchronous Phase, and resend the Association Response command after a RIFS period, as shown Figure 7-24.

An additional setup procedure is required when using the HRCP SC PHY in MIMO mode as specified in 13.2.9.3.

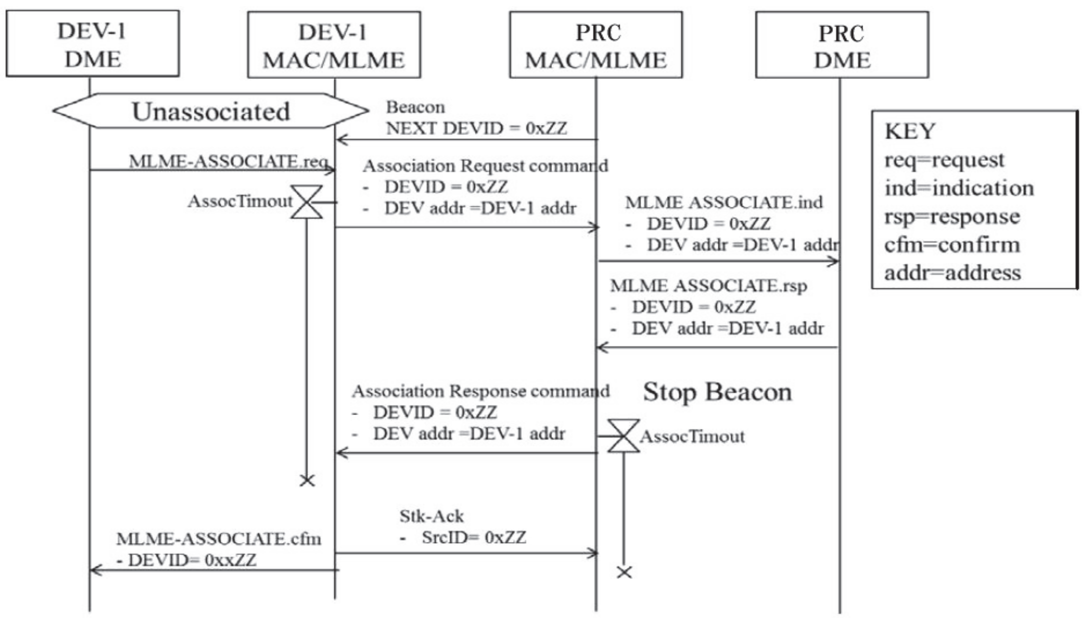

Figure 7-21—MSC of DEV-1 associating

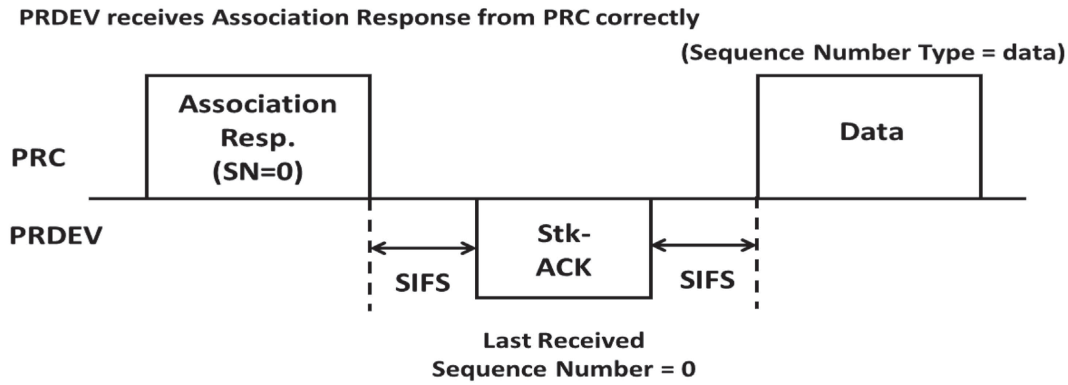

Figure 7-22—Association correctly completed (Case 1)

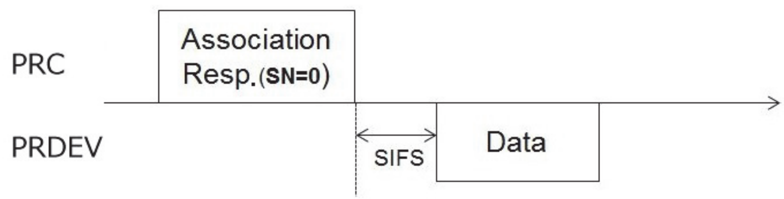

Figure 7-23—Association correctly completed (Case 2)

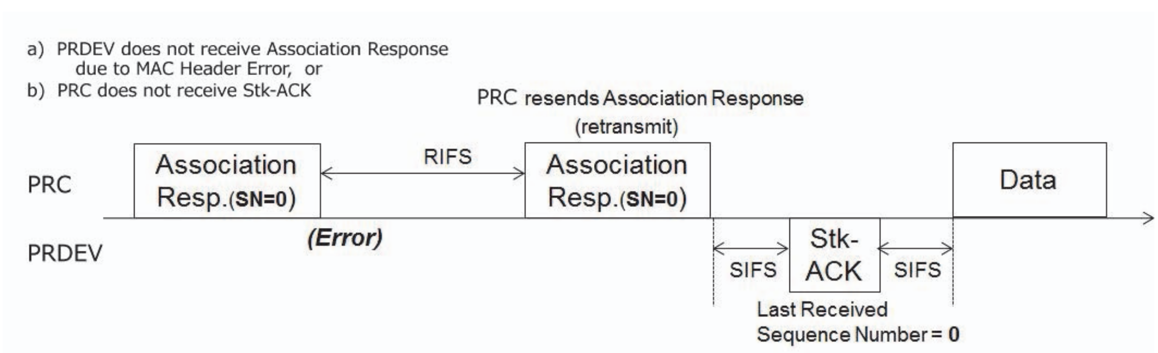

Figure 7-24—Recovery of Association Response command

7.5.2 Disassociation

When a PRC wants to remove a PRDEV from the pairnet, the PRC shall send a Disassociation Request command, as described in 6.5.2.4, to that PRDEV with an appropriate reason code. Similarly when a PRDEV wants to leave the pairnet, the PRDEV shall send a Disassociation Request command to the PRC with an appropriate reason code.

When a Disassociation Request command has been received correctly with a Stk-ACK policy, it shall be acknowledged by the other DEV. For a No-ACK policy, multiple Disassociation Request commands can be sent to maintain robustness.

The PRDEV in the pairnet shall send frames to the PRC often enough to assure that the association timeout period is not reached. If the PRC does not receive any frame originating from an associated PRDEV within this timeout duration, the PRC shall disassociate the PRDEV. The PRDEV may send a Probe Request command instead of Stk-ACK without requesting any information to cause the PRC to reset the ATP if the PRDEV does not have any other traffic that it needs to send to the PRC. The PRC shall send a Disassociation Request command to the PRDEV that sends a frame after its ATP has expired.

The PRC, upon receiving a Disassociation Request command or an ATP expiration, may send a new Beacon frame with a new Next DEVID, which will be assigned to the next PRDEV as DEVID.

7.5.3 Higher layer protocol setup during association procedure for a pairnet

Higher layer protocol setup, such as Internet Protocol layer setup or object exchange file transfer setup, may be performed during a pairnet's association procedure. This is made possible by using Higher Layer Protocol Information IE, as defined in 6.4.45, in the Beacon frame and association related commands. The content and format of the IE is out of scope of this standard and instead is determined by the entity identified in the IE.

A PRC or DEV may ignore any Higher Layer Protocol Information IE based on rules that are out of scope for this standard.

All PRCs may send Beacon frames with Higher Layer Protocol Information IE including Higher Layer Protocol Information.

All DEVs, which can understand the content of Higher Layer Protocol Information IEs on Beacon frames, may decide to send or not to send an Association Request command to that PRC based on the information in the IE. In such a case, when DEVs send association requests to the PRC, the Association Request command should have appropriate Higher Layer Protocol Information IE.

However, if a DEV cannot understand the Higher Layer Protocol Information IEs that it received, the DEV may send an Association Request command with its own Higher Layer Protocol Information IE.

A PRC may refuse association if it cannot understand the content of the Higher Layer Protocol Information IE in the Association Request command or if there is no appropriate IE in the Association Request command when the Beacon frames include Higher Layer Protocol Information IE.

The PRC should send an Association Response command with Higher Layer Protocol Information IE when such higher layer exists and requests such information.

7.6 Channel access

7.6.1 General

The channel time is divided into superframes, with each superframe beginning with a beacon. The superframe is composed of three major parts: the beacon, the optional CAP, and the CTAP, as shown in Figure 7-25. The CTAP is used for asynchronous and isochronous data streams as well as commands while the CAP may be used for commands and non-stream data, as regulated by the PNC. During the CAP, the DEVs access the channel in a distributed style using CSMA/CA and a backoff procedure. During the CTAP, the PNC controls the channel access by assigning CTAs to an individual DEV or to a group of DEVs with each CTA having a fixed start time and duration

.

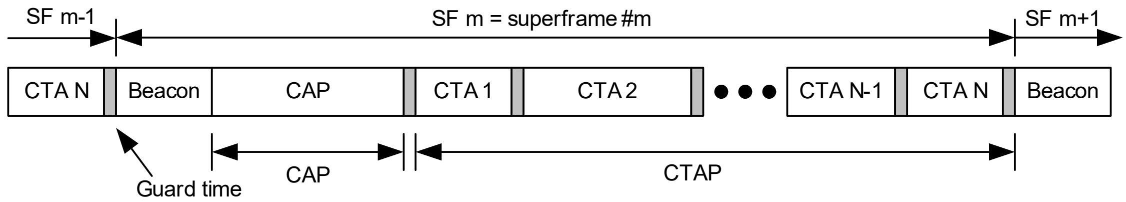

Figure 7-25—Superframe structure

Contention access methods are used in contention periods (CPs). The CPs defined for this standard are the CAP and contention access CTAs, which are defined to be association MCTAs, association CTAs, open MCTAs, and open CTAs.

7.6.2 Interframe space (IFS)

There are four IFSs that are defined: the minimum interframe space (MIFS), the short interframe space (SIFS), the backoff interframe space (BIFS), and the retransmission interframe space (RIFS). MIFS and BIFS are not used in pairnets. The actual values of the MIFS, SIFS, BIFS, and RIFS are PHY dependent and are defined as follows:

— In 11.2.7.2 for the 2.4 GHz PHY
— In 12.2.7 for the SC PHY
— In 12.3.6.5 for the HSI PHY
— In 12.4.2.2 for the AV PHY
— In 13.2.7.2 for the HRCP-SC PHY
— In 13.3.7.2 for the HRCP-OOK PHY
— In 14.2.7.2 for the THz-SC PHY
— In 14.3.7.2 for the THz-OOK PHY

The SIFS is the shortest interframe space when Rx-Tx turnaround time is required. All Imm-ACK frames, frames sent as a response frame for Imp-ACK, and Dly-ACK frames shall start transmission over the medium a SIFS after the end of the transmission of the previous frame that requested the ACK. The IFS between all received Imm-ACK frames and Dly-ACK frames and the next frame transmitted over the medium shall be no less than a SIFS. The MIFS is the shortest interframe space that can be taken when Rx-Tx turnaround time is not required. The IFS in a CTA between a frame and the next frame transmitted over the medium by the same DEV if the first frame had the ACK Policy field set to either no-ACK or Dly-ACK shall be no less than a MIFS.

During the CTAP, all DEVs shall use an IFS no less than a RIFS for retransmissions. During a contention period (CP), however, the retransmissions shall follow the CAP rules described in 7.6.3. The rules for acknowledgment and retransmissions are described in 7.12. The interframe space requirement for the beacon is ensured by the location of the CTAs, which is determined by the PNC, as described in 7.6.4.7.

Prior to association, the SIFS used by the PRC and DEV shall be equal to the default SIFS defined for the PHY. The PRC shall select the shortest SIFS duration supported by both the PRC and DEV, as indicated in the Supported SIFS field in the PRC Capability IE and the PRDEV Capability IE. After association, the SIFS used is indicated in the Pairnet Operation Parameters IE.

During the Synchronous Phase in PAP, all DEVs shall use SIFS and alternately exchange transmission rights. During the Asynchronous Phase in PAP, all DEVs shall use RIFS. The RIFS value of the PRC is always shorter than the one for the associated DEV. The RIFS values used within the pairnet shall be longer than a SIFS plus the time to acquire the Stk-ACK information in the MAC header. The rules for acknowledgment and retransmissions are described in 7.11.

7.6.3 Carrier sense multiple access with collision avoidance (CSMA/CA) in piconets

The basic medium access mechanism during the CAP is CSMA/CA. The CSMA/CA contention method is also used for open CTAs and association CTAs. The PNC controls the type of data or commands that may be sent in the CAP via the CAP Control field in the Piconet Mode field of the Piconet Synchronization Parameters field, as described in 6.3.1, in the beacon. A DEV shall only send frames of the type indicated by the Piconet Mode field in the beacon for the current superframe. The CAP Control field in the Piconet Mode field may be changed by the PNC from superframe to superframe.

Association CTAs are CTAs that have the SrcID set to the UnassocID and the DestID set to the PNCID. Association CTAs shall be used only to send Association Request commands to the PNC. Open CTAs are CTAs that have the SrcID set to the BcstID and the DestID set to a valid DEVID including GrpIDs, the McstID, and the BcstID. Open CTAs may be used to send either commands or data to the DEV or DEVs indicated by the DestID.

To minimize collisions, a transmitting DEV is required to first sense that the medium is idle for a random length of time. The MAC shall use the clear channel assessment (CCA) capabilities of the PHY to detect whether the channel is busy or idle. Only if the medium is idle after that time shall the DEV start its transmission. This process of waiting before transmission is termed “backoff.” The backoff procedure shall not be applied to the transmission of the beacon that is transmitted by the PNC at the beginning of the superframe.

During a CP, a DEV is allowed to transmit one frame at a time with backoff being applied to every frame attempted during the CP, except for the Imm-ACK frame. The PNC may send a command one SIFS after the Imm-ACK of a frame in a CP or one SIFS after a frame with ACK Policy field set to no-ACK in a CP. In this case, the PNC is not required to perform the backoff procedure before sending its frame.

In no case shall a DEV or the PNC extend its transmissions that started during a CP past the end of that CAP. If an Imm-ACK is expected for that frame, the remaining time in the CP needs to be long enough to accommodate the current frame, two SIFS times, and the Imm-ACK frame at the same PHY rate as the transmitted frame. If there is insufficient time remaining in the CP for the entire frame exchange sequence, then the DEV or the PNC shall not commence transmission of the frame.

The following backoff procedure shall be performed when sending frames (other than Imm-ACK) during a CP.

The backoff algorithm uses the following information:

— retry_count: An integer that takes on values in the range 0 to 3, inclusive.
— backoff_window(retry_count): A table that has values [7, 15, 31, 63].
— *pBackoffSlot*: A PHY-dependent parameter that is based on the amount of time it takes to sense the channel. For the 2.4 GHz PHY, this is defined in 11.2.7.2. For the SC PHY, this is defined in 12.2.7.2. For the HSI PHY, this is defined in 12.3.6.4. For the AV PHY, this is defined in 12.4.2.2.
— bw_random(retry_count): A random integer drawn from a uniform distribution over the interval [0,backoff_window(retry_count)]. The random number generated for a DEV should be statistically uncorrelated with random numbers generated by other DEVs. If the DEV does not possess a random number source, the random integer should be generated using its unique DEV address (and any other information that the implementer wishes to use) and a pseudo-random number generator (PRNG) such as MGF1, as defined in IEEE Std 1363™-2000 [B6]. Note that the current state of the PRNG should be maintained and subsequent backoffs should use subsequent integers in the pseudo-random sequence. It is important that designers recognize the need for statistical independence among the random number streams among DEVs.

The backoff time in a CP is measured at the air interface and indicates when a DEV may begin transmitting data. The DEV first waits a BIFS duration, as described in 7.6.2, from when the medium is determined to be idle before beginning the backoff algorithm. At the beginning of the CAP, the DEV may begin the backoff algorithm a SIFS after the end of the beacon transmission. If the PNC indicates that it is using an extended beacon, as described in 7.8.3, then the DEV shall wait until a SIFS after the last Announce command sent by the PNC as a part of the extended beacon before beginning the backoff procedure. At the beginning of a contention access CTA, the DEV may begin the backoff algorithm at the start time of the CTA.

The DEV shall then choose backoff_count = bw_random(retry_count) and shall maintain a counter for backoff_count that is decremented only when the medium is idle for the entire duration of *pBackoffSlot*. The retry_count shall be set to zero for the first transmission attempt of a frame. Whenever the channel is busy, the backoff counter shall be suspended. The channel shall be determined to be idle for the duration of a BIFS period before the backoff slot countdown is resumed. When the backoff counter reaches zero, the DEV may transmit a frame. When a backoff count of zero is drawn, the DEV can transmit immediately following the channel having been idle for a BIFS.

The backoff counter shall also be suspended outside of a CP duration. The backoff counter shall also be suspended if there is not enough time remaining in a CP for the DEV to send the frame. Note that the backoff counter is not maintained across superframes. A DEV shall choose a new backoff_count at the start of every CP. If the total time elapsed since the frame was queued for transmission has exceeded the transmission timeout specified for the frame, the backoff counter shall be reset, and the attempted transmission shall be canceled.

When a directed frame is transmitted and the expected ACK is not correctly received by the DEV, as described in 7.1, the retry_count shall be incremented but shall not be set to more than 3. The backoff_count shall then be set to bw_random(retry_count). If the maximum number of retries for that frame has not been exceeded, the backoff procedure is again resumed.

For DEVs supporting mmWave PHY, Blk-ACK is allowed in a CP. In a CP, the Data frame sent in response to a Blk-ACK request shall have a zero-length MAC Frame Body field. The directional use of CP is defined in 7.8.6.3.

7.6.4 Channel time allocation period channel access in piconets

7.6.4.1 General

Channel access in the CTAP is based on a TDMA method in which all CTAs have guaranteed start time and duration. The guaranteed start times enable both power saving and good QoS characteristics. All the CTAs for the current superframe are broadcast in the beacon.

7.6.4.2 Channel time allocations (CTAs)

The PNC divides the CTAP into channel time allocations (CTAs). A DEV that is given a directed CTA is guaranteed that no other DEVs will compete for the channel during the indicated time duration of the CTA. A DEV with a CTA may or may not make use of all the allocated time duration within the CTA. The selection of a stream, command, or asynchronous data for transmission during a CTA is determined locally by the DEV depending on the number of pending frames and the value of their User Priority fields. See B.3.2 for more information on priority management.

There are two types of CTAs: dynamic CTA and pseudo-static CTA. The type of a CTA requested is indicated in the Channel Time Request command, as specified in 6.5.7.2.

The PNC may move dynamic CTAs within the superframe on a superframe-by-superframe basis. This allows the PNC the flexibility to rearrange CTA assignments to optimize the utilization of the assignments.

The PNC moves a dynamic CTA by simply changing the CTA parameters in the beacon. Dynamic CTAs may be used for both asynchronous and isochronous streams.

If multiple CTAs per superframe were requested by the DEV in the Channel Time Request command, as described in 6.5.7.2, the PNC shall attempt to spread the CTAs out evenly within the superframe. If the PNC is unable to spread out the allocations, as described in F.1.4, it should deny the channel time request.

The PNC should attempt to allocate the CTAs of all SPS DEVs first in the superframe. Exceptions to placing these allocations first are as follows and in order of priority:

- QoS streams that need multiple CTAs within a superframe and require a location immediately following the beacon, if the CAP is used.
- If CAP is not used, a single CTA that follows the beacon without the *mFirstCtaGap* restriction and is one of the following:
  - An MCTA with the PNCID as the SrcID—directed, broadcast, or multicast
  - A single pseudo-static CTA

Pseudo-static CTAs shall be allocated only for isochronous streams and shall not be sub-rate allocations, as described in 6.5.7.2. If the PNC needs to change the duration or location of a pseudo-static CTA within the superframe, it shall change the corresponding CTA blocks in the beacon. The PNC shall not create any new CTAs for other stream indices that overlap with the old time interval of the pseudo-static CTAs for *mMaxLostBeacons* number of superframes. However, the PNC may overlap the old and new time intervals of the same pseudo-static CTA within a superframe as it does not create the possibility of frame collisions. If the PNC sees the transmission of a protocol data unit (PDU) during the new allocation by the source of the old allocation before the expiration of *mMaxLostBeacons* number of superframes, the PNC may reuse the old allocation for another pair of DEVs. When the source DEV of a pseudo-static CTA receives a beacon with the new CTA, it shall cease using the old CTA and begin using the new CTA. When the destination DEV of a pseudo-static CTA receives a beacon with the new CTA, it shall begin receiving during the new CTA and may also receive during the old CTA.

If the PNC needs to simultaneously change the positions of one or more of the CTAs in the superframe, including pseudo-static CTAs, the PNC may place the Piconet Parameter Change IE in the beacon with the Change Type field set to MOVE and the Superframe Duration field set to zero. After the superframe with the beacon number equal to the value in the Change Beacon Number field in the Piconet Parameter Change IE, a DEV shall not transmit until the DEV successfully receives a beacon with a beacon number greater than or equal to the value in the Change Beacon Number field in the Piconet Parameter Change IE.

While the PNC is changing the time interval of the pseudo-static CTA, it is possible for the destination DEV to miss traffic for up to *mMaxLostBeacons* superframes. If the destination wants to avoid this, it would need to listen for the entire superframe duration whenever it misses a beacon.

A private CTA is a CTA where the same DEV is both the source and the destination. A private CTA is not used for communication in the piconet. Instead, it is used to reserve channel time for some other use. For example, a private CTA would be used for a dependent piconet, as described in 7.2.8 and 7.2.9. Private CTAs shall be pseudo-static CTAs, so that its position and duration remain relatively constant for the other use. A DEV requests a private CTA by using a Channel Time Request command, as described in 6.5.7.2, with Target ID List field containing only its own DEVID and the Stream Index field set to the unassigned stream values, as defined in 6.2.8.

The PNC may allocate CTAs with the SrcID set to the BcstID. These CTAs are referred to as either open CTAs or open MCTAs, depending on the stream index, and use contention-based channel access instead of TDMA. The type of contention access that is used is indicated by the Stream Index field in the CTA block. A stream index set to the asynchronous stream index indicates that CSMA/CA is used while the MCTA stream index indicates that slotted aloha is used. A DEV may request that the PNC modify the frequency and duration of the open CTAs or open MCTAs by sending a Channel Time Request command, as described in 6.5.7.2, to the PNC with the DestID set to the BcstID and the stream index set to the MCTA index. The CTA Rate Factor field, CTA Rate Type field, Channel Time Request TU field, and Minimum Number of TUs field shall be set by the DEV to the desired duration and frequency requested for the open CTA allocation. The PNC is not required to allocate open CTAs or open MCTAs in the manner requested by a DEV; rather it may use this information to determine the frequency and duration of open CTAs or open MCTAs that it allocates. A DEV may modify its request for open CTAs or open MCTAs by sending another Channel Time Request command to the PNC with new parameters. A DEV is not allowed to terminate either an open CTA or an open MCTA. However, a DEV requests zero time for open CTAs or open MCTA by sending a Channel Time Request command to the PNC with the Minimum Number of TUs field set to zero.

The More Data field, as described in 6.2.2.8, in the Frame Control field is set to one to indicate that the source DEV could be sending more frames in the CTA. In order to save power at the destination DEV, a source DEV may indicate that it will not use the remaining time in the current CTA by setting the More Data field to zero. The source DEV may retransmit a frame with More Data set to zero for which an ACK was expected but was not received. If the destination DEV receives a frame with the More Data field set to zero with an ACK Policy other than no-ACK, it should continue to listen for an implementation-dependent time after sending an acknowledgment to make sure that the source DEV is not going to retransmit the frame because it did not receive the ACK. The source DEV may choose to send a zero-length frame with the More Data field set to zero when it has no more frames to send in a CTA.

The More Data field shall be ignored by the destination for all frames sent in the CAP, with the exception of any Announce commands used for the extended beacon.

7.6.4.3 Channel time allocation (CTA) and channel time usage

The DEVs that are members of the piconet shall use the Channel Time Request command, as described in 6.5.7.2, whenever they wish to make a change in their CTAs. Once a Channel Time Request command is received from a DEV, the PNC shall remember that as the outstanding request for that stream for every superframe until another Channel Time Request command for that stream is received from the DEV. The CTAs within the CTAP are based on the current pending requests from all the DEVs and the currently available channel time within the CTAP. The start time of each CTA is referenced to the start of the Beacon frame, as described in 7.8. The algorithm used to allocate the channel time and assign CTAs is outside of the scope of this standard.

The PNC shall not allocate any MCTAs or dynamic CTAs within *mFirstCtaGap* following the end of the beacon except with the PNC as the source.

When a source DEV has a frame of any type for a destination DEV, the source DEV may send it during any CTA for that source DEV and destination DEV pair or to use the CAP to communicate that frame. The source DEV may also send a frame to a destination DEV in any CTA assigned to that source even if the destination DEV is different from that indicated in the CTA block, provided the source DEV has determined that the destination DEV will be receiving in that CTA, as described in 6.4.12. The stream index in a transmitted frame shall be one of the following:

— The asynchronous stream index, as defined in 6.2.8
— The MCTA stream index, as defined in 6.2.8
— An established stream from the source DEV to the destination DEV

If the DestID of the CTA is the McstID or the BcstID, the source DEV may still send directed frames to any associated DEV. However, it is possible that the target DEV will not be receiving during the CTA if it is in a power save mode, as described in 7.17, or if it is not receiving multicast traffic, as described in 7.7.4.

In any superframe there may be one or more DEVs in the piconet that receives the beacon in error. This may not happen to the same DEV all the time but may happen to different DEVs at different times depending upon their location and type of interference to which they are subjected. If a DEV did not receive the beacon, it shall not transmit during the CAP or during any MCTA or dynamic CTA, except to ACK a directed frame sent to the DEV with the ACK Policy field set to one of Imm-ACK, Imp-ACK, or Dly-ACK Request. DEVs with pseudo-static CTAs are allowed to transmit during these CTAs as long as the number of consecutive lost beacons is less than or equal to *mMaxLostBeacons*. A DEV shall stop transmitting in its pseudo-static CTA when the number of consecutive lost beacons exceeds *mMaxLostBeacons*. If a DEV that is the destination of a pseudo-static CTA misses a beacon, it should listen for the entire duration of the superframe in case the pseudo-static CTA is in the process of being moved. Any DEV that misses a beacon may also listen for the entire duration of the superframe to receive frames for which it is the destination.

In no case shall a DEV extend its transmissions that started during a CTA beyond the end of that CTA. Hence, the source DEV shall check whether there is enough time remaining in the CTA for the transmission of current frame and SIFS. If an Imm-ACK or Dly-ACK is expected for that frame, the DEV shall check whether there is enough time remaining in the time slot to accommodate the current frame, 2 SIFS periods, and the Imm-ACK or Dly-ACK frame at the same PHY rate as the transmitted frame. If there is not enough time remaining for this entire frame exchange sequence, then the DEV shall abort the transmission and not use the remainder of the CTA.

The PNC may compute more than one superframe time allocation at a time and keep them repeating over time until the situation changes. If the PNC wants to increase the allocated channel time allocation on a regular basis, it shall allocate more time up to the maximum requested by sending a Channel Time Response command, as described in 6.5.7.3, to the source DEV and change the allocation(s) in the beacon. If the source DEV requires additional channel time it will need to use the stream modification procedure, as described in 7.7.2.3. The PNC may also reduce the channel time allocation for a stream by sending the Channel Time Response command to the source DEV and changing the allocations in the beacon.

In any individual superframe, the PNC may allocate more time for a dynamic CTA than the amount indicated in the Channel Time Response command.

7.6.4.4 Management CTAs

Management CTAs (MCTAs) are CTAs that have the stream index set to the MCTA stream index, as described in 6.2.8. A PNC may choose to use MCTAs instead of the CAP for sending command frames, unless otherwise restricted by the PHY, as described in 11.2.10. When MCTAs are used, the PNC shall ensure that sufficient MCTAs are allocated to allow for the transmission of commands to and from the PNC. There may not be any MCTAs in a superframe or there may be as few as a single MCTA in a superframe where the ownership of the MCTA changes from superframe to superframe. At the other extreme, there may be one or more uplink and downlink MCTAs per member DEV per superframe plus MCTAs for association. The PNC is responsible for determining the appropriate number of MCTAs in a superframe in the same way that the PNC is responsible for choosing the CAP size if a CAP is used. The PNC determines which DEVs will be allocated MCTAs and the frequency of the allocations. The PNC shall allocate at least one association MCTA every *mMctaAssocPeriod* if the CAP is not used.

An open MCTA is one where the SrcID is the BcstID, as described in 6.2.5, and the DestID is set to a valid DEVID including GrpIDs, the McstID, and the BcstID. Any DEV that is associated in the piconet may attempt to send a command frame to the DEV or DEVs that are indicated in the DestID in an open MCTA. An MCTA with the UnassocID as the SrcID is an association MCTA. Any DEV not currently associated in the piconet may attempt to send an Association Request command to the PNC in an association MCTA. Association Request commands shall not be sent in open MCTAs. Likewise, only Association Request commands shall be sent in association MCTAs. Open MCTAs with the DestID set to the PNCID enable the PNC to service a large number of DEVs with low MCTA requirements by using a minimum number of MCTAs. When there are few DEVs in a piconet, it might be more efficient to use MCTAs assigned to a DEV instead of using an open MCTA. Open MCTAs in which the DestID is not the PNCID are used by DEVs to send frames to other DEVs without having to request channel time.

It is the PNC's responsibility to determine the number and type of MCTAs to use for each superframe. A DEV may request the frequency of MCTA allocations by sending a Channel Time Request command, as described in 6.5.7.2, to the PNC with the stream index set to the MCTA stream index, as described in 6.2.8, and the CTA Rate Factor, as described in 6.5.7.2, set to the DEV's desired interval for uplink MCTAs, the Num Targets field set to one and the Target ID field set to the PNCID. All other parameters of the Channel Time Request Block field shall be set to zero and may be ignored by the PNC upon reception.

If commands are not allowed in the CAP, the PNC shall assign an MCTA with the new DEV's DEVID as the SrcID as soon as possible after a successful association, 7.4.2, preferably in the next superframe, in order to support fast connections.

The access mechanism for regular MCTAs, i.e., neither open nor association MCTAs, is TDMA, as described in 7.6.4.2.

7.6.4.5 Slotted aloha access

Slotted aloha is used for contention access in an open MCTA or an association MCTA. The access to an open or association MCTA shall be controlled by a contention window CW_a maintained by each DEV. The contention window shall be derived from the number a, where a is the number of retransmission attempts made by the DEV. For the first access attempt, a shall be set to zero. The size of the contention window, CW_a, is defined in Equation (7-1):

$$CW_a = \begin{cases} 256 & \text{for } 2^{a+1} \geq 256 \\ 2^{a+1} & \text{for } 2^{a+1} < 256 \end{cases} \tag{7-1}$$

The open or association MCTA used for the a^{th} retransmission attempt shall be chosen by a uniformly distributed random integer value, r_a, within the interval [1, CW_a]. The random number generated for a DEV should be statistically uncorrelated with random numbers generated by other DEVs. If the DEV does not possess a random number source, the random integer should be generated using its unique DEV address (and any other information that the implementer wishes to use) and a PRNG such as MGF1, as defined in IEEE Std 1363-2000 [B6]. Note that the subsequent retransmission attempts should use backoff counters drawn from subsequent entries in the pseudo-random list of integers. It is important that designers recognize the need for statistical independence among the random number streams among DEVs.

The DEV shall start counting r_a beginning with the open or association MCTA(s) in the current superframe and continue across superframes. The lack of an Imm-ACK indicates the failure of the previous access attempt.

The first open or association MCTA after the DEV begins the access process is specified by number $r = 1$. The open or association MCTA with number equal to r_a is the MCTA that the DEV shall access. The DEV shall not access the MCTA before its counter has reached the open or association MCTA with the number $r = r_a$. After receiving an ACK, a shall be reset to zero.

7.6.4.6 Allocation of MCTAs

The PNC shall indicate in every beacon, as described in 6.3.1, the rate at which it will be allocating either open MCTAs or directed uplink MCTAs in the MCTA Allocation Rate field in the Piconet Synchronization Parameters field. If the PNC is not using either open MCTAs or directed uplink MCTAs it shall indicate this with the appropriate value of the MCTA Allocation Rate field, as described in 6.3.1.1. Likewise, if the PNC will not be guaranteeing the rate at which MCTAs will be allocated, it shall also indicate this in the MCTA Allocation Rate field.

The intent of the MCTA Allocation Rate field is to enable the DEVs in the piconet to approximately determine the length of time required to send a command to the PNC. This information might be used to set the timeout parameters for the MLME primitives, as described in 5.3.

7.6.4.7 Guard time

In a TDMA system, guard times are required to keep transmissions in adjacent CTAs from colliding. In addition, a SIFS time is required to ensure sufficient turnaround time between transmissions. A CTA is defined by the start time and the duration, as specified in the CTA IE. Guard time is the time between the end of one CTA and the start of the next CTA. Including SIFS as part of CTAs and allocating guard time between CTAs ensures that transmissions are spaced by at least a SIFS. Figure 7-26 is an illustration of the allocation of the guard time such that the transmissions are separated by at least a SIFS if the owners of adjacent CTAs drift towards the other CTA. The PNC shall allocate sufficient guard time between CTAs to ensure that transmissions in adjacent CTAs do not overlap.

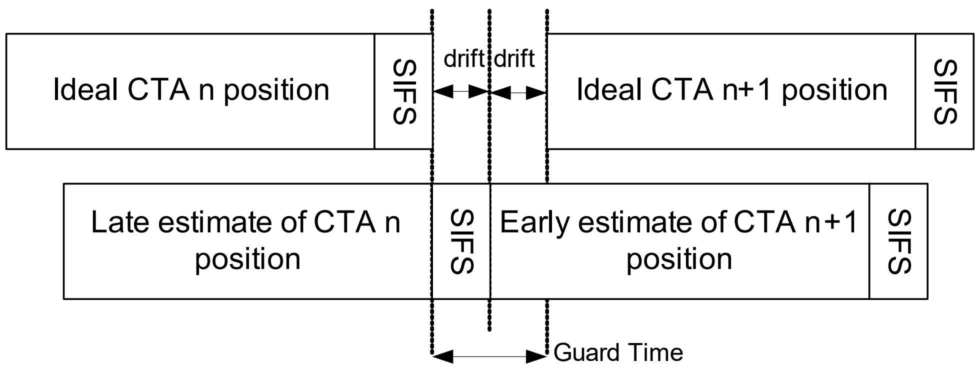

Figure 7-26—Guard time

The required guard time depends on the maximum drift between a DEV's local time and the ideal time. This drift is a function of the time elapsed since a synchronizing reference event. In an IEEE 802.15.3 piconet, the synchronizing event is the start of the preamble of a beacon. The maximum drift, MaxDrift, is calculated as shown in Equation (7-2):

$$MaxDrift = \text{Clock accuracy} \times \text{interval} \tag{7-2}$$

Propagation delay will also affect timing uncertainty, but in a piconet, the 10 m range limits propagation delay to around 33 ns, or even 66 ns for DEVs 20 m apart at opposite ends of a piconet. This is much lower than the resolution of the CTA timing, and it is ignored when calculating the guard time.

The PNC may calculate a single worst-case guard time for all CTAs in the superframe, or it may calculate and assign guard time based on the type of CTA (dynamic or pseudo-static) and the position of the CTA in the superframe.

Pseudo-static CTAs require longer guard times than dynamic CTAs because pseudo-static CTAs allow transmission even when up to *mMaxLostBeacons* are missed by the transmitting DEV. Guard times are calculated based on the worst-case drift in a superframe and the maximum allowed number of lost beacons for each of the adjacent CTAs. Guard time may be calculated by the PNC as shown in Equation (7-3):

$$GuardTime = (MaxLostBeacons_{CTA\_n} + MaxLostBeacons_{CTA\_n+1} + 2) \times MaxDrift \tag{7-3}$$

MaxLostBeacons for each CTA depends on whether the CTA is pseudo-static or dynamic. MaxLostBeacons is zero for a dynamic CTA and is equal to *mMaxLostBeacons* if the CTA is pseudo-static. The PNC calculates the MaxDrift using the superframe duration and the clock accuracy, *pClockAccuracy*. The PNC then calculates the start time and duration of each CTA such that there is sufficient guard time between the end of one CTA and the start of the next CTA.

A DEV transmitting in a CTA starts transmission of the preamble for the first frame at the point that it calculates is the start of the CTA based on its local clock. In the case of no-ACK or delayed-ACK, the transmitting DEV shall ensure that there is enough time remaining in the CTA to transmit the frame and allow for a SIFS before the end of the CTA as calculated by that DEV, as shown in Figure 7-27.

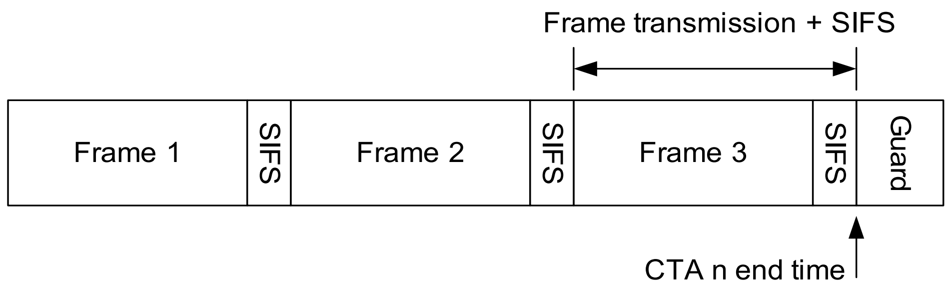

Figure 7-27—SIFS and Guard time at the end of a CTA—no-ACK

If Imm-ACK is used, the transmitting DEV shall also ensure there is enough time for the ACK and another SIFS as shown in Figure 7-28.

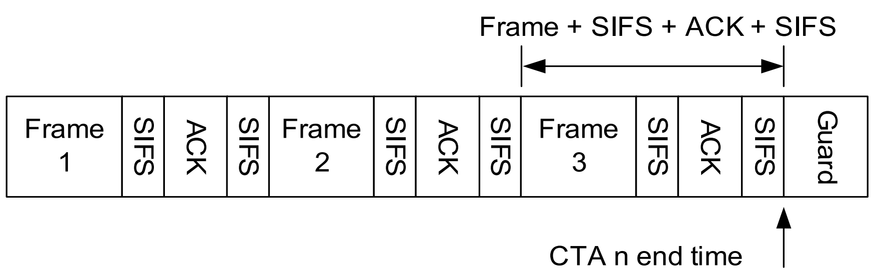

Figure 7-28—SIFS, ACK, SIFS, and Guard time at the end of a CTA—Imm-ACK

As with any CTA, the PNC shall include sufficient guard time between the last CTA in the superframe and the beacon as shown in Figure 7-29.

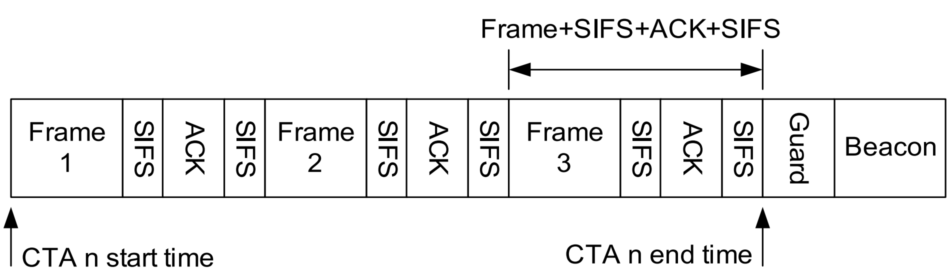

Figure 7-29—Guard time at the end of the superframe

The PNC shall begin transmission of the beacon preamble at the point in time that it calculates is the start of the superframe based on its local clock. All DEVs will resynchronize their clock based on the beacon arrival.

Because the clock in one DEV may be fast and another may be slow relative to the ideal time, a DEV that is expecting to receive either the beacon or a frame during the CAP or in a CTA shall begin receiving before the time that it calculates to be the start of the beacon, CAP, or CTA and shall continue receiving after the time that it calculates to be within one SIFS of the end of the CTA. The amount of time that the DEV listens before the start of the CTA and after the end of the CTA is up to the implementer. It may be calculated based on the type of CTA, the superframe duration, and *pClockAccuracy*. The DEV shall be able to receive a frame that is transmitted within the bounds of allowable transmission for the CTA, accounting for the worst-case drift.

7.6.4.8 Calculating channel time requests

Each DEV sends channel time requests to the PNC to indicate the amount of channel time required for transmission.

The requesting DEV shall include the frame transmission time, if known *a priori*, and the ACK transmission time, if used, and one MIFS or SIFS as appropriate per frame or ACK when calculating channel time requests. Figure 7-30 shows an example of channel time being requested for a CTA where Imm-ACKs are used.

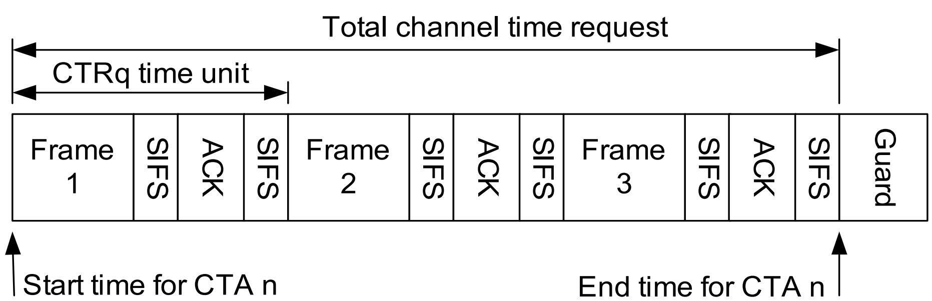

Figure 7-30—Channel time request for frames with immediate ACKs

When No-ACK is used, the channel time request is calculated differently because there is a MIFS in between each frame in the CTA instead of a SIFS. However, there is a SIFS at the end of the CTA to allow time for the DEVs to switch from transmit to receive and from receive to transmit. Figure 7-31 shows an example of a channel time request when no-ACK is used.

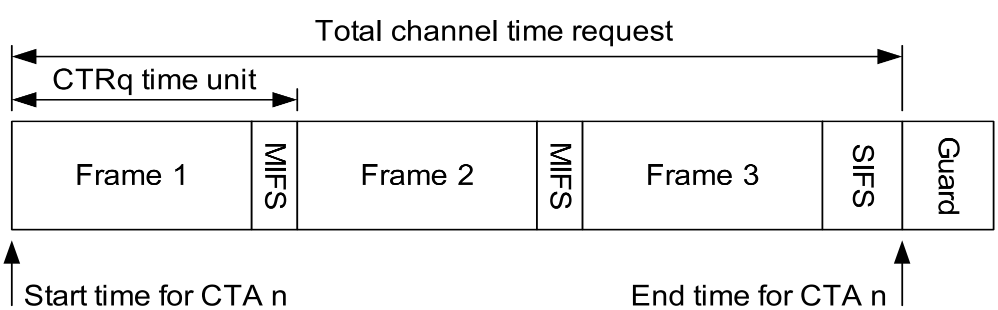

Figure 7-31—Channel time request with no ACKs

A Channel Time Request TU in the CTA may cover more than one frame as shown in Figure 7-32.

If the frame size is not known a priori, it is up to the requesting DEV to decide the amount of channel time to request for the CTA. The calculation method in Figure 7-32 is preferred for better CTA efficiency.

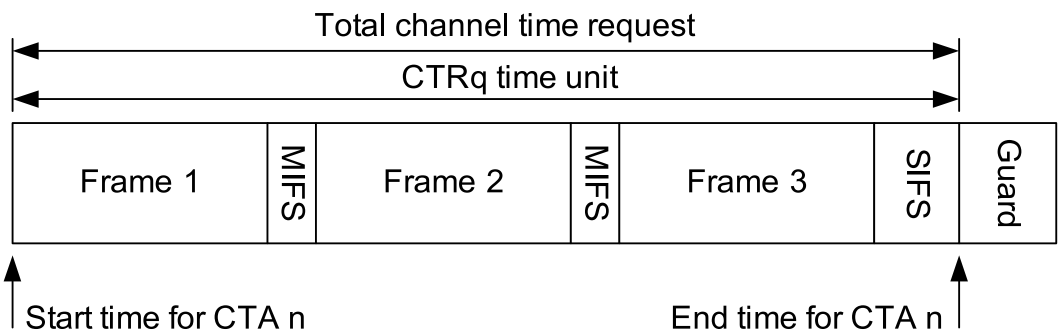

Figure 7-32—Channel Time Request TU covering multiple frames

7.6.4.9 Relinquishing CTA time to another DEV

The PNC gives transmit control to the DEV that is the SrcID of a CTA for the duration of the CTA. The DEV that has transmit control in a CTA may, subject to the restrictions in this subclause, relinquish the remaining time in a CTA to another DEV. The DEV that relinquishes the channel time is referred to as the *originating DEV* while the DEV that is given the transmit control of the time in the CTA is referred to as the *target DEV.* The DEV that is the SrcID of the CTA begins the CTA with transmit control for the CTA.

The originating DEV relinquishes the remaining time in the CTA to a target DEV by setting the CTA Relinquish field in the header of a frame which has the DestID set to the DEVID of the target DEV. Transmit control of the CTA can only be exchanged between the source DEV of the CTA and the destination DEV(s). If there are multiple destination DEVs for the CTA, transmit control may be given to any one of them.

The originating DEV should not initiate a CTA relinquish procedure unless it has determined that the receiving DEV has set the CTA Relinquish Supported field to one in the DEV Capabilities, as defined in 6.4.12.

A DEV that receives transmit control of a CTA keeps control until the CTA end time or until it relinquishes the transmit control back to source DEV of the CTA.

The target DEV, after listening for an *mCtaRelinquishTimeout* following reception of a frame with the CTA Relinquish field set, may send a data frame to any potential DestID of the CTA. The target DEV listens to verify that the originating DEV is not retrying the frame that had the Relinquish CTA field set.

If the DEV relinquishing time in a CTA does not correctly receive a header, as described in 7.1, from the DEV to which it relinquished the CTA within *mCtaRelinquishTimeout* time, it shall maintain transmission control of the channel.

The CTA Relinquish procedure shall not be used in the CAP or in any contention CTA.

Any DEV that is responding to frames in a CTA may set the More Data field in the header of an Imm-ACK frame or data frame to indicate to the current CTA transmit control owner that they would like to get a transmission opportunity. The current CTA transmit control owner may choose to relinquish the CTA, send a frame with an Implied ACK request, or to ignore the request.

If the DEV that has transmit control of the CTA wishes to relinquish the CTA and has no data to send, it may set the CTA Relinquish field in an Imm-ACK frame. The same rules apply as when the CTA Relinquish field is used in a data frame.

The target of a CTA relinquish shall relinquish the CTA only to DEV that is SrcID of the CTA. The current CTA transmit control owner shall always initiate the relinquish frame exchange.

Figure 7-33 illustrates the process of relinquishing transmit control to another DEV in the piconet.

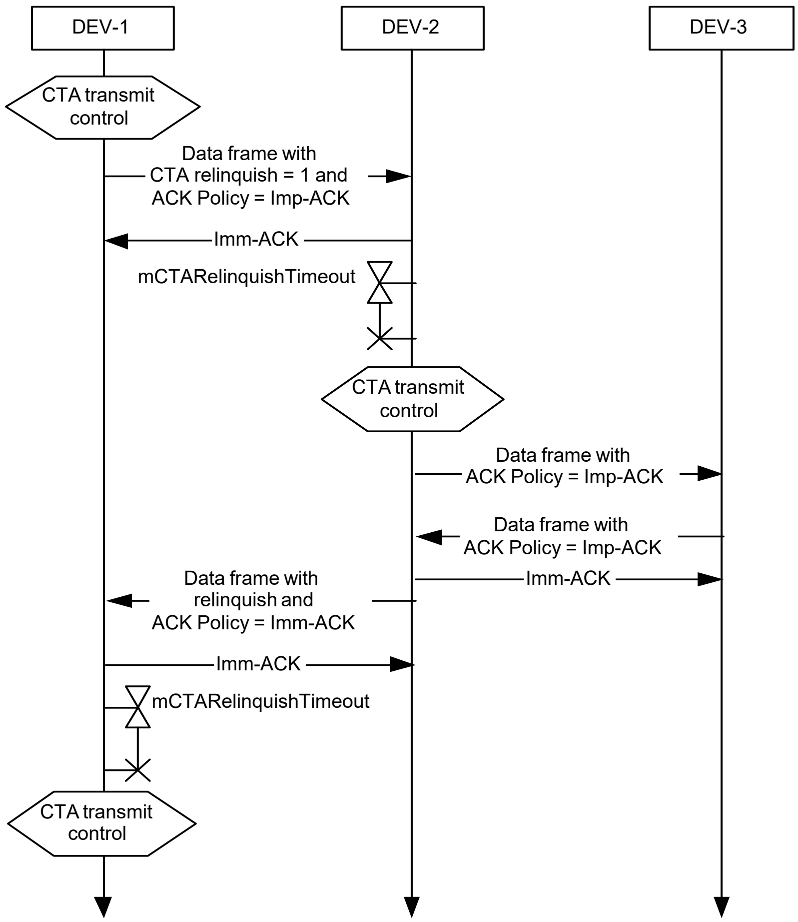

Figure 7-33—MSC for relinquishing CTA time when the target DEV has data to send

7.6.5 PAP after association

Stk-ACK is used for acknowledgment of data and command frames and is indicated in the MAC header and may be piggybacked with the data payload. The PAP has two phases, Synchronous Phase and Asynchronous Phase. The DEVs that comprise the pairnet determine individually which phase they are in internally. The Synchronous Phase is either after completion of link setup or when the ping-pong transmission between the PRC and the DEV continues, i.e., when frame exchange continues using SIFS. Otherwise, it will be the Asynchronous Phase. After receiving a frame that has either a MAC header error or has not received a frame after SIFS from the end of its transmission, the Asynchronous Phase starts. When the DEV receives a frame with the correct MAC header within the RIFS, the DEV determines that it entered the Synchronous Phase. The recovery procedure during the Asynchronous Phase is described in 7.12.7.2.

During the Synchronous Phase, if the DEV of either side receives the frame and the MAC header has no errors, following the SIFS, the DEV transmits the frame to the other DEV. This continues in an alternate fashion. When the PRC or the associated DEV has no data frame to transmit, only the MAC header is transmitted. After completion of link setup, the DEV that has scheduled data transmission may access the medium by SIFS.

During the Asynchronous Phase, each of the DEVs within the pairnet accesses the medium with RIFS and shall transmit the frame with only the MAC header. The Stk-ACK information shall always be set in the MAC header of the transmitted frames. The PRC and the associated DEV use different RIFS values.

7.7 Channel time management for piconets

7.7.1 General

Channel time management in an IEEE 802.15.3 piconet involves the following:

- The creation, modification, and termination of isochronous data streams between two or more DEVs in the piconet.
- The reservation and termination of asynchronous channel time for the exchange of asynchronous data between two or more DEVs in the piconet.

A DEV may support one or more streams depending upon the application it is designed to support. A PNC needs to support as many isochronous streams as it desires to source and sink.

7.7.2 Isochronous stream management

7.7.2.1 General

Creating, modifying, and terminating isochronous streams between two or more DEVs in a piconet is accomplished via negotiation between the originating DEV and the PNC using the Channel Time Request and Channel Time Response commands, as described in 6.5.7.2 and 6.5.7.3. Once a stream index and its CTA are established, the CTA may be modified or terminated.

Only a DEV that is either a member of the piconet or associated as a neighbor PNC shall send a Channel Time Request command to the PNC.

There is no absolute guarantee of the length of delay between the time of the request and the reception of a beacon containing the requested CTA.

The StreamGrpID parameter of the MLME-CREATE-STREAM.request and MLME-MODIFY-STREAM.request primitives allows a DME to define associations between streams for the purpose of sharing channel time. When a StreamGrpID other than zero is assigned, a CTA should be shared fairly among all streams within the group associated with the StreamGrpID. The DME is responsible for assigning unique StreamGrpIDs for stream groups.

The PNC may split a super-rate allocation into more CTAs than required to satisfy the CTA Rate Factor and may also split a sub-rate allocation into more than one CTA in the same superframe.

A DEV may request the creation (and subsequent modification and termination) of allocations, isochronous and asynchronous, in which the DestID is the PNCID. Likewise, the PNC may create allocations, both isochronous and asynchronous, in which the SrcID is the PNCID. However, while these allocations are transferred in the PNC handover process, the purpose for the allocations might be lost when the DEV that is the PNC changes. The PNC shall only allocate multiple CTAs per superframe if the time allocated for each CTA is no more than one Channel Time Request TU different from any of the other CTAs allocated for that stream.

7.7.2.2 Isochronous stream creation

In the case where the originating DEV is going to request a new isochronous stream with a target DEV, the originating DEV shall send a Channel Time Request command, as described in 6.5.7.2, to the PNC with the following parameter values:

— Target ID List field is set to the DEVID with which the originating DEV is requesting a new stream.
— Stream Index field is set to the unassigned stream value, as described in 6.2.8.
— Stream Request ID field is set to a unique value between 1 and 255 for the duration of the negotiation.
— Priority field is set to a value between 0b011 and 0b110, as defined in Figure 6-18.
— All the other Channel Time Request command parameters are set to appropriate values, as defined in 6.5.7.2.

The PNC upon receiving the Channel Time Request command from the originating DEV shall respond with a Channel Time Response command, as described in 6.5.7.3, to the originating DEV with the following Channel Time Response command field values if the requested channel time is available:

— The Stream Index field is set to an unused value other than the asynchronous stream index. This indicates that the isochronous stream has been allocated channel time. The PNC should assign the least recently used stream index available.
— The Available Number of TUs field is set to a value greater than or equal to the Minimum Number of TUs and less than or equal to the Desired Number of TUs requested.
— The Reason Code field is set to “success.”

The PNC may update the beacon with the newly assigned isochronous stream CTAs before it receives an ACK to the Channel Time Response command from the originating DEV.

The PNC shall announce the creation of all streams with the CTA Status IE, as described in 6.4.11, using the beacon information announcement mechanism, as described in 7.8.5. The PNC shall issue the initial CTA for the stream in the superframe indicated in the CTA Status IE.

The CTA Status IE shall have the stream index of the new allocation, the Channel Time Request Control field from the corresponding Channel Time Request command, and the CTA Sub-Rate field set appropriately, 6.4.11. In addition, the PNC shall allocate the first CTA of the stream in the superframe with the beacon number, as described in 6.3.1.1, indicated by the Start Beacon Number field of the CTA Status IE.

If, however, either the requested channel time is not available or the PNC is not able to support the requested priority, as described in B.3.2, the PNC shall respond to the requesting DEV with the following parameter values:

— The Stream Index field shall be set to the unassigned stream value, as described in 6.2.8.
— The Available Number of TUs field shall be set to the number of TUs that the PNC had available for allocation to this request.
— The Reason Code field shall be set to “priority unsupported,” “channel time unavailable,” or “unable to allocate as pseudo-static CTA” value.

If the request is for a private pseudo-static CTA, and the PNC will not support the creation of a child piconet, it shall respond with the reason code set to “request denied.”

If the stream index is set to beam-forming stream value then the request is for a CTA for beam forming. If the PNC grants this request, it shall assign the stream index value to be the beam-forming stream index.

The requesting DEV upon receiving this Channel Time Response command and indicated parameters may accept its denied request as final, it may resend its original request, or it may modify its original request with new parameters.

If the target DEVID is not a member of the piconet, the PNC shall respond to the requesting DEV with the following parameter values:

— The Stream Index field shall be set to the unassigned stream value.
— The Available Number of TUs field set to zero.
— The Reason Code field shall be set to either “target DEV not a member” or “target DEV unassociated” depending on the status of the target DEVID.

DEVs perform multicast negotiations at a higher layer. A DEV may set up a multicast stream either by using a GrpID assigned by the PNC, as described in 7.7.4, or by using the McstID. A DEV enables reception of a multicast stream that has the DestID set to the McstID with the MLME-MULTICAST-RX-SETUP.request. This primitive tells the MAC to receive frames from a particular source DEV with the DestID set to the McstID and with the stream index specified in the primitive. A DEV enables the reception of multicast traffic addressed to the GrpID by joining a multicast group, as described in 7.7.4.

If the target DEV is in DSPS and is part of the same SPS Set as the channel time request, and the PNC grants the allocation, the PNC shall respond with a Channel Time Response command with a Reason Code of SUCCESS. Neither the originator nor the target DEV need to change PM modes for the stream to be allocated.

However, if the target DEV is either in DSPS mode and is not part of the same SPS Set as the channel time request or in APS mode, and the PNC grants the channel time request, the PNC shall set the Reason Code in the Channel Time Response command to “Success, DEV in PS mode.” The PNC shall place the PCTM IE in the beacon with a bit set for the target DEV, as described in 6.4.9.

When the Target DEV in DSPS or APS mode receives a beacon with its bit set in the PCTM IE, it shall send a PM Mode Change command to the PNC. If the DEV is going to remain in a power save mode it shall set the PM Mode field in the PM Mode Change command to the appropriate value, either “SPS” or “APS.” The PNC shall then terminate the stream, as described in 7.7.2.4.

If the power save DEV is going to listen to the new allocation, it shall set the PM Mode field in the PM Mode Change command to ACTIVE. The PNC shall then begin allocating the channel time in the beacon for the stream. The PNC shall no longer set the bits for the DEV in the PS Status IEs.

If the PNC does not receive the PM Mode Change command from the power save DEV within a timeout determined by the PNC, the PNC shall terminate the channel time request, as described in 7.7.2.4, and unset the PS DEV’s bit in the PCTM IE.

An STP capable DEV shall renew its allocated streams at least once every ATP by sending an Announce command with the Stream Renew IE to the PNC containing the stream index of every stream that the originator wants to keep. Streams that have been modified within the current ATP may be omitted from the Stream Renew IE sent during the same ATP.

If the Target DEV is in DSPS mode, after the PNC sets the DSPS DEV’s bit in the PCTM IE, the PNC shall provide in the DSPS DEV’s next wake superframe an MCTA with the DSPS DEV as the source and the PNC as the destination that is long enough to handle a PM Mode Change command, a Channel Time Request command with four isochronous Channel Time Request Block fields, and the associated Imm-ACKs and SIFSs. This allows the DSPS DEV to request a change to one of the current channel time allocations, to request new channel time, or to request that a channel time allocation be terminated.

Figure 7-34 illustrates the sequence of messages involved in successfully establishing a DEV-2 to DEV-3 stream in a piconet.

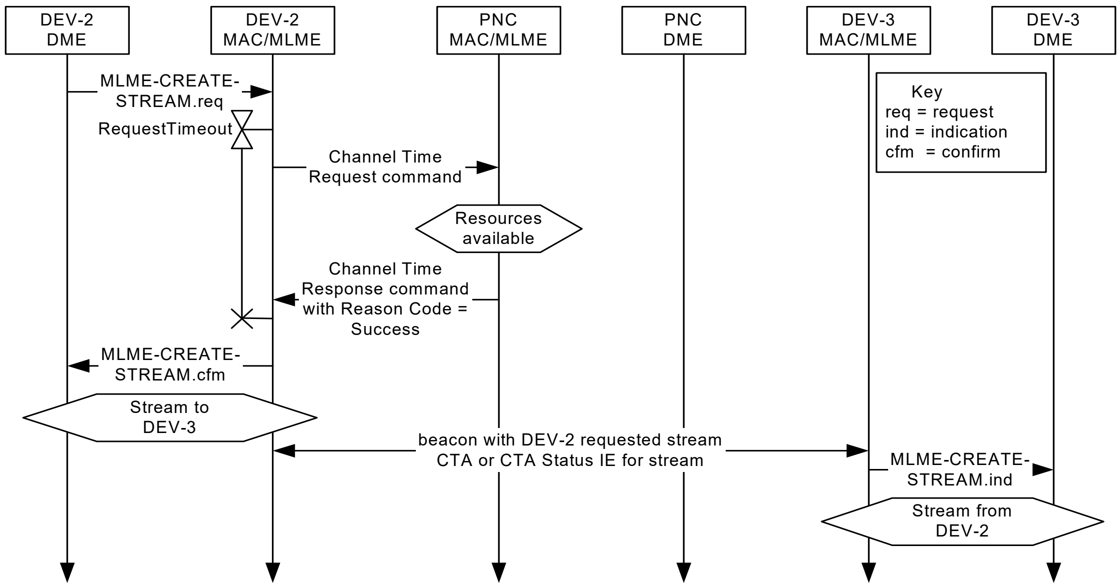

Figure 7-34—MSC for creating a DEV-2 to DEV-3 stream

Figure 7-35 illustrates the sequence of messages involved in an unsuccessful attempt to establish a DEV-2 to DEV-3 stream in a piconet.

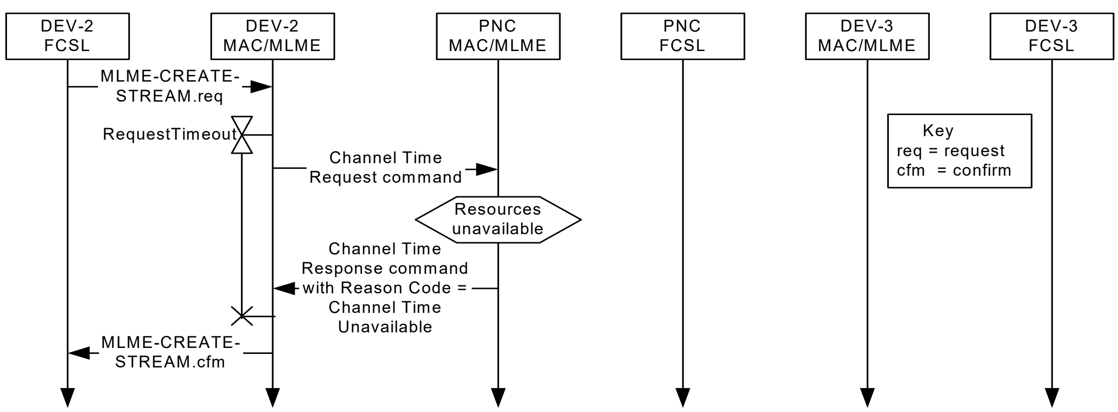

Figure 7-35—MSC for a denied DEV-2 to DEV-3 stream

7.7.2.3 Isochronous stream modification

Only the originating DEV or the PNC may modify an established isochronous stream. The originating DEV that is requesting a modification of the channel time allocated to one of its streams shall send the PNC a Channel Time Request command with the following parameter values:

— Target ID List field is set to the DEVID with which the originating DEV has an established stream.
— Stream Request ID field set to zero.
— Stream Index field set to the index of the stream to be modified.
— The CTA Type field shall be set to the same value as in the original request for that stream index.

— All the other Channel Time Request command parameters are set to appropriate values, as defined in 6.5.7.2.

The PNC upon receiving the Channel Time Request command shall check to see if the requested resources are available. If the requested channel time is not available, the PNC shall:

— Send a Channel Time Response command with:
  — The Available Number of TUs equal to the previously assigned Available Number of TUs.
  — The Stream Request ID field set to zero.
  — The Reason Code field set to “channel time unavailable.”
— Make no modification to the existing beacon CTA blocks for the DEV requesting the modification.

If the requested channel time is available, the PNC shall:

— Reserve the requested channel time.
— Send a Channel Time Response command where:
  — The Stream Request ID field is set to zero.
  — The Reason Code field is set to “success.”

  and
  — The “new” Available Number of TUs field is greater than the “previous” Available Number of TUs field and is less than or equal to the Desired Number of TUs (for a requested an increase in channel time),

  or
  — The “new” Available Number of TUs field less than or equal to the Desired Number of TUs field (for a requested decrease in channel time).
— Build a new beacon with the modified CTA.

NOTE—If a request is rejected, the Available Number of TUs is calculated using the Channel Time Request TU associated with the last accepted request. Only if the request is accepted will the Channel Time Request TU size for an exiting stream be modified.

The PNC shall announce the modification of all streams with the CTA Status IE, as described in 6.4.11, using the beacon information announcement mechanism, as described in 7.8.5. The PNC shall issue the first modified CTA for the stream in the superframe indicated in that IE. If the target DEV is in DSPS mode, the PNC shall also allocate an uplink MCTA in the same superframe as when the CTA is first allocated with the DSPS DEV as the source and the PNC as the destination that is long enough to handle a PM Mode Change command, a Channel Time Request command with four isochronous Channel Time Request Block fields, and the associated Imm-ACKs and SIFSs.

A dependent PNC, the originator DEV, may handover control of the dependent piconet’s CTA to another DEV, the target DEV, in the parent piconet. The target DEV shall be either a member of the piconet or a DEV that has associated as a neighbor PNC, as described in 7.2.9. To handover control of the dependent piconet’s CTA, the originator DEV shall send a Channel Time Request command to the parent PNC with the following parameters:

— The Num Targets field set to one.
— The Target ID List field containing the DEVID of the target DEV that is to receive control of the CTA.
— The Stream Request ID field set to zero.
— The Stream Index field set to the stream index of a CTA that has already been allocated to the dependent PNC as a private, pseudo-static CTA.
— All other fields set to the same values as in the last successful Channel Time Request for this Stream Index.

If the target DEV indicated in the Target ID List is either a member of the parent piconet or is an associated neighbor PNC and the Channel Time Request command has the correct entries as indicated previously, the parent PNC shall grant the request to change the source and destination for the stream and shall send a Channel Time Response command to the originator with the Reason Code set to “Success.” The PNC shall continue to place the CTA block for the allocation in the beacon but shall change the SrcID and DestID to be equal to the target DEV’s DEVID. Once the PNC has changed the SrcID and DestID in the CTA block, the target DEV will have gained control of the CTA and will be allowed to request modification or termination of the allocation.

If the target DEV is not a member of the piconet and it is not an associated neighbor PNC, the parent PNC shall reject the request and shall send a Channel Time Response command to the originator with the Reason Code set to either “target DEV unassociated” or “target DEV not a member” depending on the status of the target DEV.

If the Channel Time Request command has improper entries, e.g., the Stream Index does not exist or the Stream Index is not associated with a private, pseudo-static CTA, then PNC shall reject the request and shall send a Channel Time Response command to the originator DEV with the Reason Code set to “request denied.”

If the DEV is STP capable, a request to modify a stream will reset the STP of that stream.

Figure 7-36 illustrates the message sequence involved in requesting a modification to an existing stream.

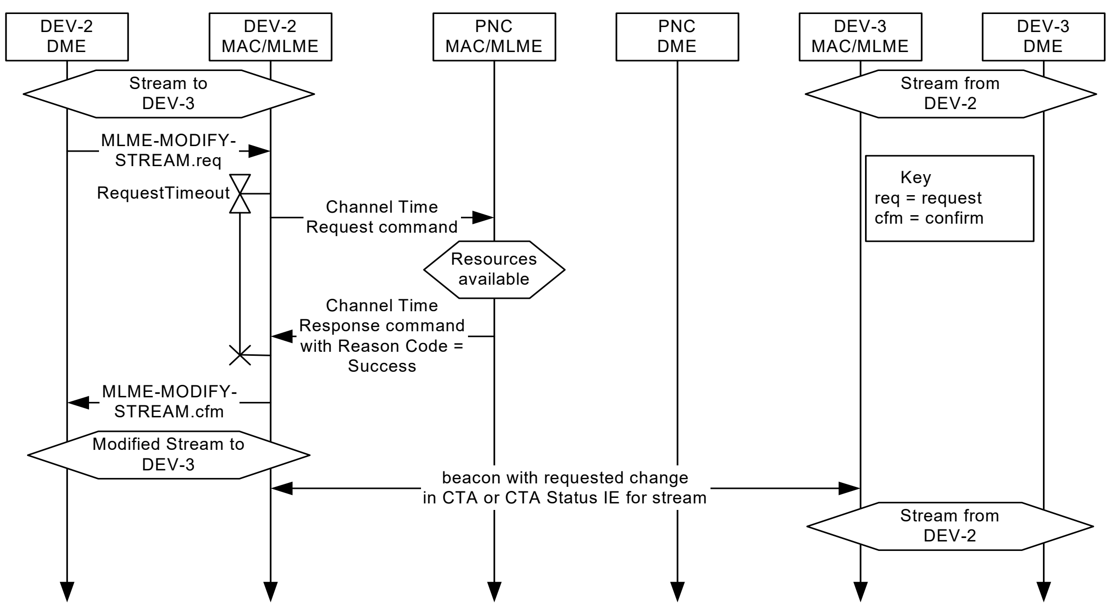

Figure 7-36—MSC for modifying a stream

Figure 7-37 illustrates the message sequence involved when a requested stream modification for an existing stream is denied.

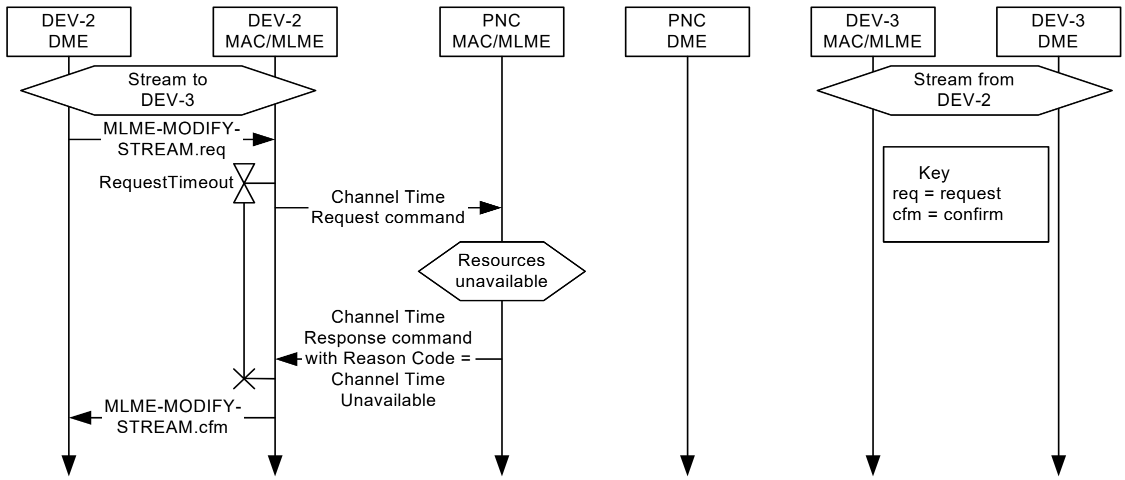

Figure 7-37—MSC for a denied stream modification

The MSC for the handover of the control of a private, pseudo-static CTA is illustrated in Figure 7-38.

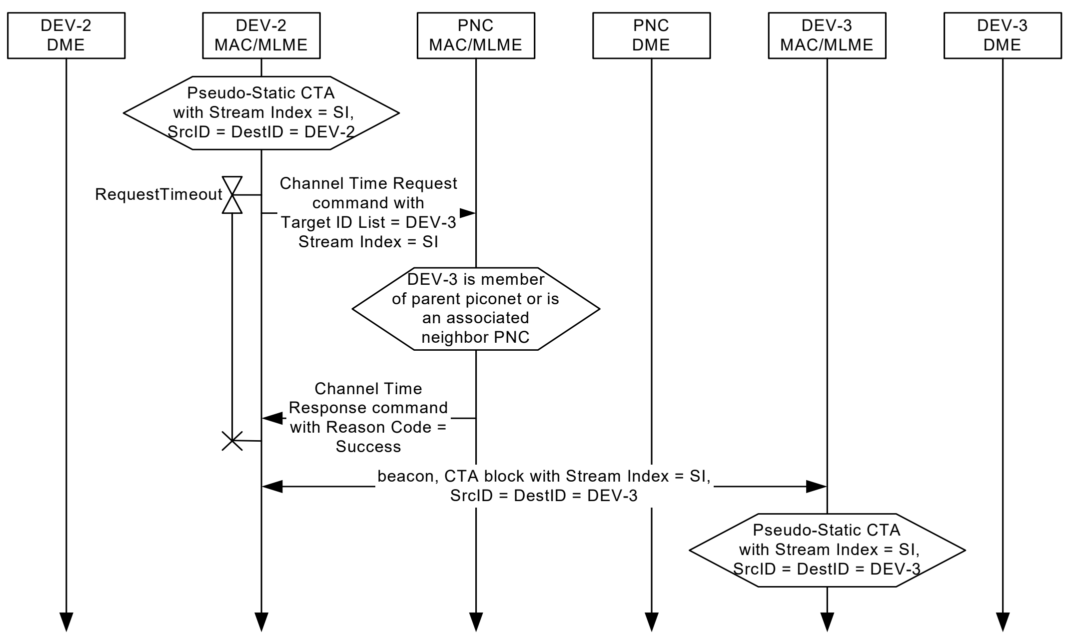

Figure 7-38—MSC for handing over control of a private, pseudo-static CTA

7.7.2.4 Isochronous stream termination

Only the PNC, the originating DEV, or the target DEV may terminate an established stream. In the case of multicast or broadcast streams, only the originating DEV or the PNC may terminate the stream. In the case where either the originating DEV or the target DEV desires to terminate a specific stream, it shall send the PNC the Channel Time Request command with the following parameter values:

— Target DEVID List field is set to the DEVID of the DEV to which the originating DEV has an established stream.
— Stream Index field shall be set to the value of the stream to be terminated.

— All other fields shall be set to zero.

The PNC, upon receiving a Channel Time Request command from a DEV requesting stream termination, shall respond with an Imm-ACK. In the case where the originating DEV is requesting a stream termination, the PNC shall then notify the target DEV of the termination via CTA Status IE in the beacon. The CTA Status IE shall have the stream index set to the value of the terminated stream and the Terminate field set to indicate that the stream was terminated, as described in 6.4.11. The PNC shall ensure that the CTA Status IE announcements comply with the rules for beacon announcements in 7.8.5. Figure 7-39 illustrates the MSC for termination of a stream by a source DEV.

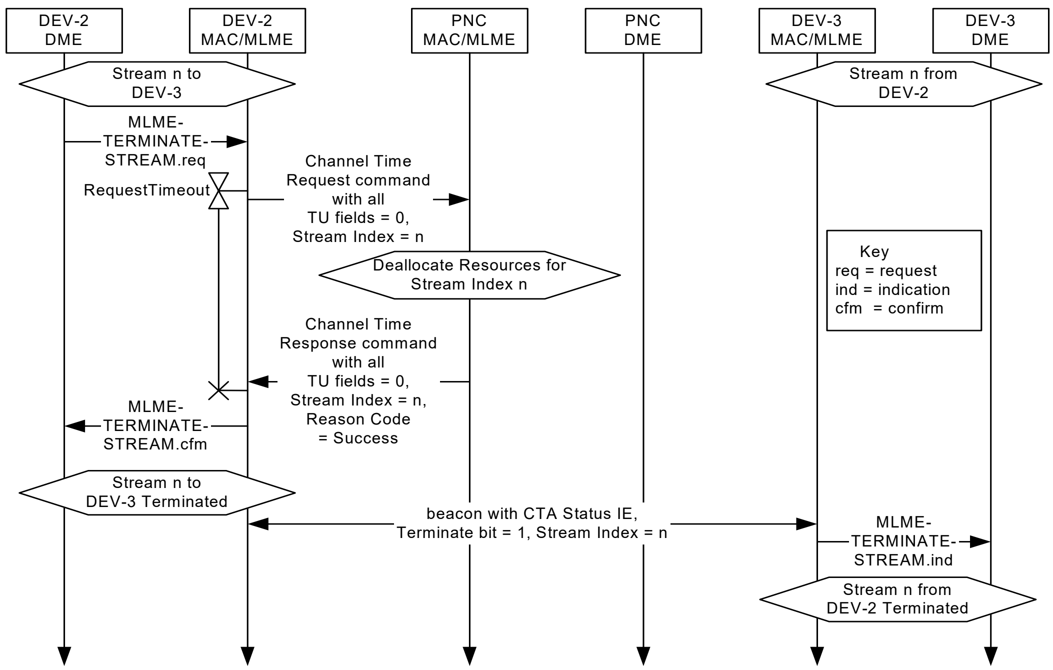

Figure 7-39—MSC of source DEV-2 requesting termination of its stream

In the case where the target DEV is requesting a stream termination, the PNC shall then notify the originating DEV of the termination via a Channel Time Response command. Figure 7-40 illustrates the MSC for termination of a stream by a target DEV.

In the case where the PNC decides to terminate an originating DEV's stream, the PNC shall notify the source DEV via a Channel Time Response command and the target DEV via a CTA Status IE in the beacon. The CTA Status IE shall have the stream index set to the value of the terminated stream and the Terminate field set to indicate that the stream was terminated, as described in 6.4.11. The PNC shall ensure that the CTA Status IE announcements comply with the rules for beacon announcements in 7.8.5. Figure 7-41 illustrates the termination of a source DEV's stream by the PNC.

If the originator of a stream is STP capable, the PNC may terminate any stream that has not been renewed within the ATP of the originator of the stream. If the PNC terminates the stream, it shall use the stream termination procedure described in this subclause for the PNC terminating a stream with the Reason Code in the Channel Time Response command set to "STP expired."

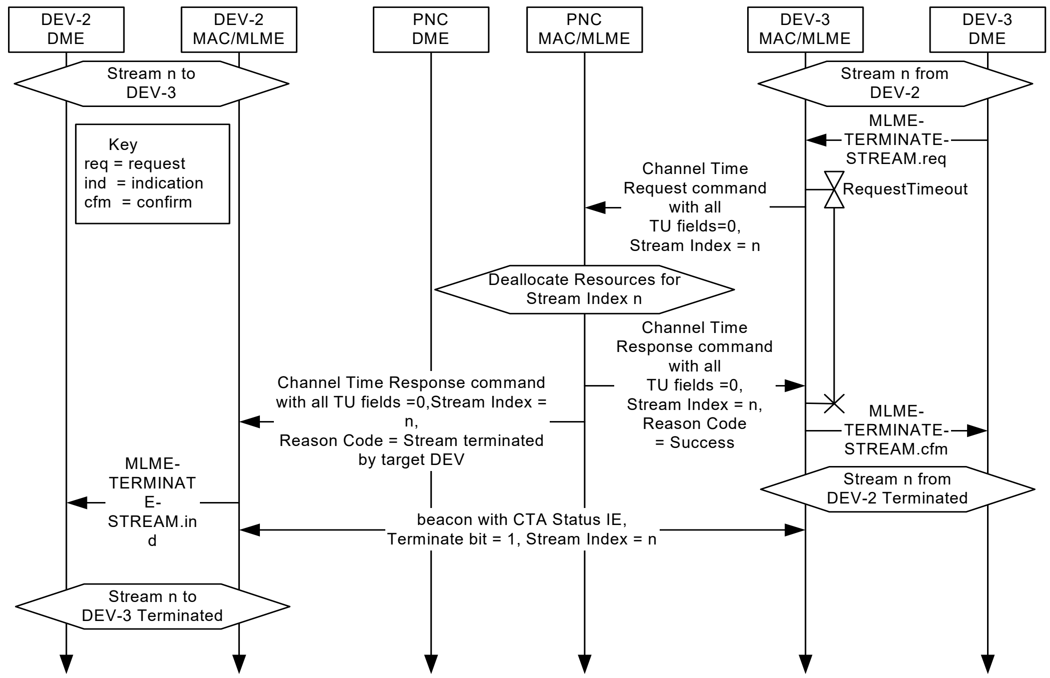

Figure 7-40—MSC of target DEV-3 requesting termination of source DEV-2's stream

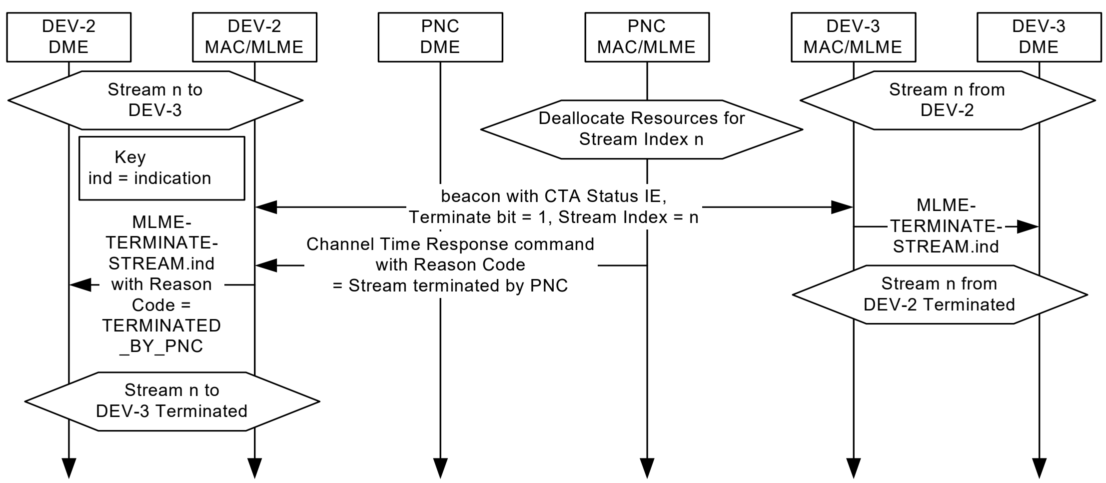

Figure 7-41—MSC of PNC terminating a stream

7.7.3 Asynchronous channel time reservation and termination

7.7.3.1 Asynchronous channel time reservation

There are two methods for requesting asynchronous channel time, as follows:

- Request a single CTA for multiple target DEVs, i.e., group asynchronous channel time requests.
- Request individual CTAs for each of the target DEVs, i.e., an individual asynchronous channel time request.

The DEV requesting asynchronous channel time shall only use one of the two methods at a time. The DEV switches between the two methods by sending a Channel Time Request command that utilizes the new method. If the DEV changes methods, the PNC shall drop previously received asynchronous channel time requests from that DEV. A DEV shall not send a Channel Time Request that requests both types of asynchronous allocations and the PNC shall reject any request received from a DEV that requests both types of asynchronous allocations.

When a DEV is requesting the creation or modification of channel time for asynchronous data transmissions with a target DEV or DEVs, the originating DEV shall send a Channel Time Request command, 6.5.7.2, to the PNC with the following parameter values:

- The target ID list shall contain either:
  - A list of all of the target DEVs. Only one Channel Time Request Block field is used for all destinations with the same TU for all of the target DEVs.

  or
  - Only one DEV in the destination list. In this case the originating DEV may send multiple Channel Time Request Block fields in the command and the TU may be different in each of the Channel Time Request Block fields.
- Stream Index field shall be set to zero.
- Priority field shall be set to a value of either 0b000 or 0b001, as defined in B.3.2.
- The DSPS Set Index, PM Channel Time Request Type, CTA Type, and CTA Rate Type fields shall be set to zero and may be ignored upon reception.
- All the other Channel Time Request command parameters are set to appropriate values, as defined in 6.5.7.2.

The PNC upon receiving the Channel Time Request command from the originating DEV shall respond with an Imm-ACK to the requesting DEV. If the requested channel time is available, the PNC places the CTA block(s) in a beacon with the source and target DEVID fields appropriately set.

In the case of a group asynchronous allocation, the PNC shall place multiple CTA blocks in the beacon, one for each of the destinations. Each CTA block shall have the asynchronous stream index and the same SrcID, start time, and duration but different DestIDs. The PNC may also split a group asynchronous allocation into several CTAs in a single superframe, with any such CTA again announced by multiple CTA blocks that overlap in time but have different DestIDs. Such splits shall only be done on the TU boundaries.

For an individual asynchronous allocation, if any of the DestIDs is currently in a power save mode, the PNC shall allocate the CTAs for the power save DEV(s) in the DEV’s next wake beacon, either system or DSPS. It is the responsibility of the source DEV to be able to handle different CTAs for destination DEVs in power save mode as opposed to DEVs in ACTIVE mode.

For group asynchronous allocations, if some of the DEVs are in a power save mode, then the PNC shall allocate as much as possible of the requested channel time in their wake beacons. The rest of the channel time may be allocated in non-wake beacons.

There is no guarantee of the delay between the time of the request and the reception of a beacon containing the requested CTA. If a frame’s timeout interval expires while waiting for its requested CTA in the beacon, an MAC-ASYNC-DATA.confirm shall be sent with the ReasonCode set to TX_TIMEOUT.

When the PNC allocates an asynchronous CTA it decrements its count of unallocated TUs by the number of TUs allocated in the CTA. When the count of unallocated TUs reaches zero, the PNC no longer allocates the CTA and drops the channel time request.

If the request is rejected, the PNC shall send a Channel Time Response command indicating the rejection. Note that if an asynchronous request is queued by the PNC but that channel time is not immediately available due to resource constraints, that does not constitute a rejection of the request.

A new asynchronous channel time request to a target DEV replaces the previous request for that target DEV and unallocated TUs from the previous request shall be replaced by the current request. A new group asynchronous channel time request replaces all previous asynchronous requests and unallocated TUs from all previous asynchronous requests shall be replaced by the current request. The originating DEV may change the request method, TU size, destinations, and desired TUs between subsequent asynchronous channel time requests.

The PNC may time out the request for an asynchronous CTA and purge them after *mAsyncRequestLifetime*. The requesting DEV should resend a new request after *mAsyncRequestLifetime* if it desires more channel time.

Figure 7-42 illustrates the sequence of messages involved in reserving channel time for the exchange of asynchronous data between DEV-2 and DEV-3 in a piconet.

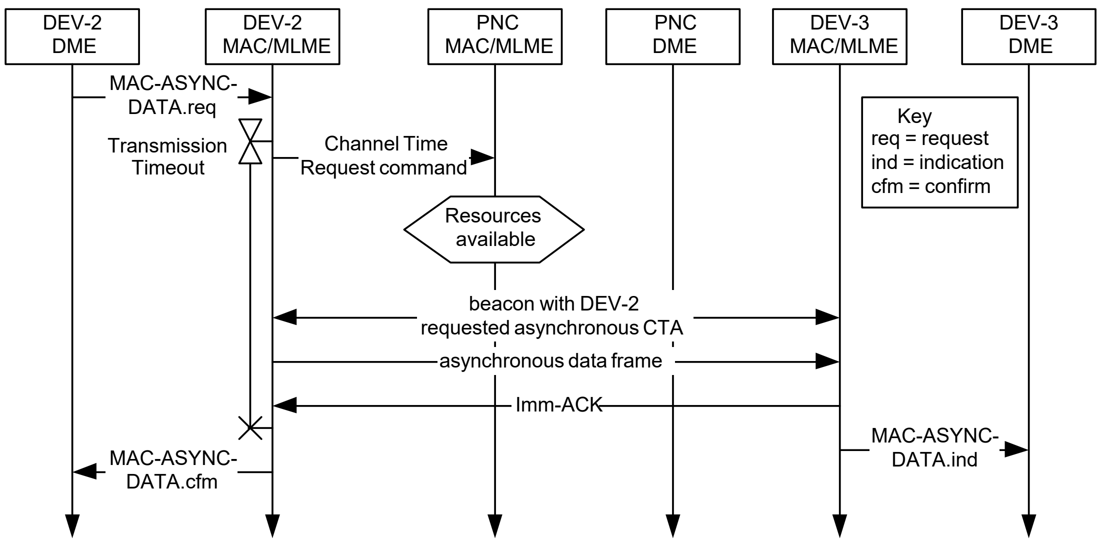

Figure 7-42—MSC for reserving asynchronous data channel time

If the target DEV is in APS mode and the PNC grants the channel time request, the PNC shall set the Reason Code in the Channel Time Response command to “Success, DEV in PS mode.” The PNC shall place the PCTM IE in the beacon with a bit set for the target DEV, as described in 6.4.9.

When the Target DEV in APS mode receives a beacon with its bit set in the PCTM IE, it shall send a PM Mode Change command to the PNC. If the DEV is going to remain in APS mode it shall set the PM Mode field in the PM Mode Change command to “APS.” The PNC shall then terminate the asynchronous channel time, as described in 7.7.3.2.

If the power save DEV is going to listen to the new allocation, it shall set the PM Mode field in the PM Mode Change command to ACTIVE. The PNC shall then begin allocating the channel time in the beacon for the asynchronous allocation. The PNC shall no longer set the bits for the DEV in the PS Status IEs.

If the PNC does not receive the PM Mode Change command from the APS DEV within a timeout determined by the PNC, the PNC shall terminate the channel time request, 7.7.3.2, and unset the DEV’s bit in the PCTM IE.

7.7.3.2 Asynchronous channel time termination

Only the PNC or the originating DEV shall be allowed to terminate an asynchronous CTA. In the case where the originating DEV is going to terminate a specific asynchronous CTA, it shall send to the PNC the Channel Time Request command with the following parameter values:

- TargetID List field shall be set to the DEVIDs of the DEVs with whom the originating DEV is going to terminate the asynchronous connection.
- Stream Index field shall be set to the asynchronous stream index, as described in 6.2.8.
- All other fields shall be set to zero.

The PNC, upon receiving the Channel Time Request command from a DEV requesting termination of the asynchronous channel time, shall respond with an Imm-ACK and shall cease allocating the channel time.

In the case where the PNC terminates the asynchronous channel time, the PNC shall notify the source DEV via a directed Channel Time Response command, as described in 6.5.6.2, with the reason code set to the appropriate value.

7.7.4 Multicast group configuration

Group MAC addresses are defined in IEEE Std 802. Because IEEE Std 802.15.3 uses DEVIDs for addressing, the PNC may assign a DEVID to be used for a group address. The PNC also keeps track of all of the DEVs that request the use of a particular group address by maintaining a list of their DEVIDs and the associated group address. A group of DEVs that have been registered with the PNC using a particular group address is called a *multicast group*.

A DEV requests a DEVID for a group address, called a GrpID, from the PNC using the Multicast Configuration Request command, 6.5.11.1, with the Group Address field set to the desired group address and the Action field set to “Join.” If a GrpID is not currently assigned as a DEVID for that Group Address and the PNC has the resources available, the PNC should assign a GrpID for the Group Address and respond to the originating DEV with the Multicast Configuration Response command, 6.5.11.2. If the request was successful, the PNC adds the originating DEV to the multicast group associated with the GrpID.

If the PNC has already assigned a GrpID for the address in the Group Address field and the PNC has the resources available, it shall add the originating DEV’s DEVID to the multicast group.

If the originating DEV’s request is granted, the PNC shall send the Multicast Configuration Response command to the originating DEV with the GrpID field set to the value assigned to that group address and the Reason Code set to “Success.”

If the address in the Group Address field does not correspond to a valid group address, the PNC shall not assign a GrpID and shall send the Multicast Configuration Response command to the originating DEV with the GrpID set to zero and the Reason Code field set to “Failure, not a valid group address.”

If the PNC is unable to fulfill the originating DEV’s request for a GrpID, the PNC shall send the Multicast Configuration Response command to the originating DEV with the GrpID set to zero and the Reason Code field set to the appropriate value.

When a DEV no longer needs to use the group address, it shall send the Multicast Configuration Request command to the PNC with the Group Address field set to the address and the Action field set to “Leave.” When the PNC receives this command, it shall remove the DEV from the multicast group and respond with the Multicast Configuration Response command with the GrpID field set to zero, the Group Address field set to the same value as in the request command, and the Reason Code field set to “Success.” The PNC shall always respond to a properly formatted Multicast Configuration Request command with the Action field set to “Leave” by sending a Multicast Configuration Response command with the Reason Code set to “Success.” If the address in the Group Address field corresponds to a multicast group that has the originating DEV as a member, the PNC shall remove the DEV from the multicast group.

If the PNC is unable to support an existing multicast group, it shall terminate all CTAs with the GrpID as either the SrcID or DestID using the stream termination procedure in 7.7.2.4. The PNC shall also place a DEV Association IE in the beacon, following the repetition rules described in 7.8.5, with the DEVID set to the GrpID and the Association Status field set to disassociated. When a DEV receives such an IE, it will know that the multicast group no longer exists.

If a multicast group no longer has any members, either due to disassociation or requests from the DEVs to leave the group, the PNC shall free the GrpID. A GrpID shall be allocated and reused according to the rules for assigning DEVIDs described in 7.4.2. A GrpID shall not be reported in the PNC Information command.

During PNC handover, the old PNC shall send one or more Announce commands, as defined in 6.5.6.2, to the new PNC with the Group ID IEs, as defined in 6.4.25, that correspond to the GrpIDs that are currently in use.

The MSC for a DEV successfully joining a multicast group is illustrated in Figure 7-43.

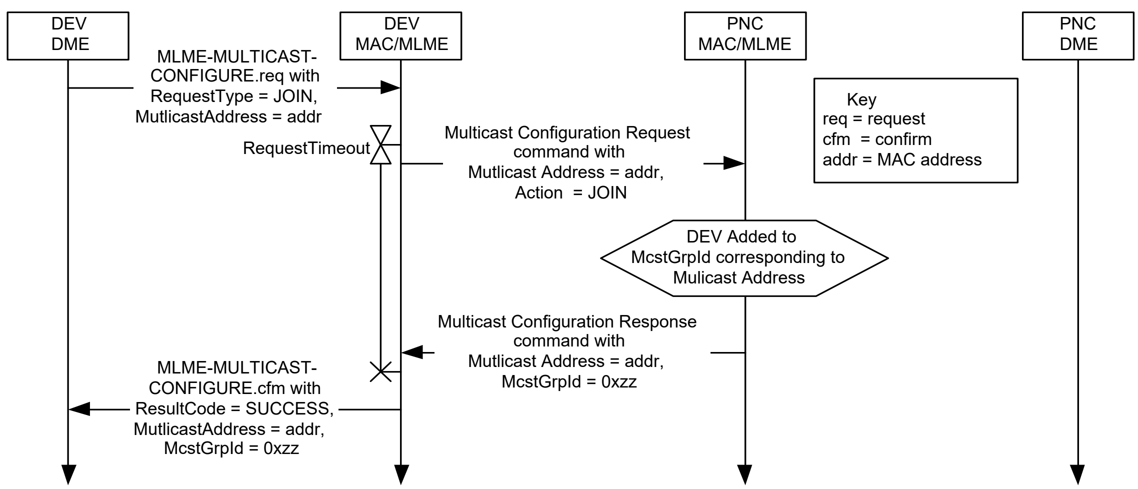

Figure 7-43—MSC for a DEV successfully joining a multicast group

The MSC for leaving a multicast group is illustrated in Figure 7-44

.

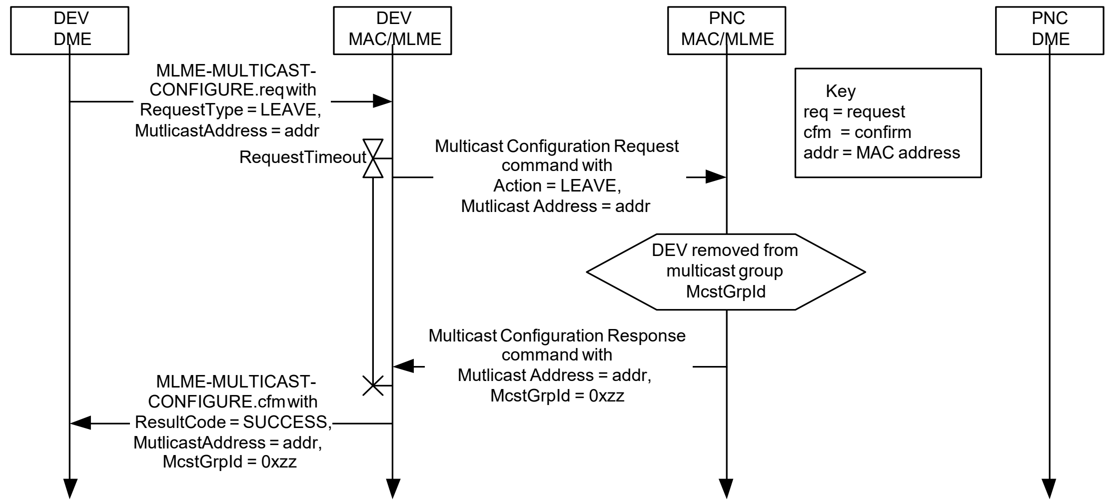

Figure 7-44—MSC for a DEV leaving a multicast group

7.8 Synchronization for piconet

7.8.1 General

All DEVs within a single piconet shall be synchronized to the PNC's clock. In addition, child or neighbor PNCs shall synchronize their piconet's time usage to the parent PNC's beacon and their CTA. The beacon sent at the beginning of every superframe contains the information necessary to time-synchronize the DEVs in the piconet. See 6.3.1 for the definition of the timing parameters sent in the beacon.

Each DEV in the piconet, including the PNC, shall reset its superframe clock to zero at the beginning of the beacon preamble, as shown in Figure 7-45. All times in the superframe shall be measured relative to the beginning of the beacon preamble. If a DEV does not hear a beacon, it should reset its superframe clock to zero at the instant where it expected to hear the beginning of the beacon preamble.

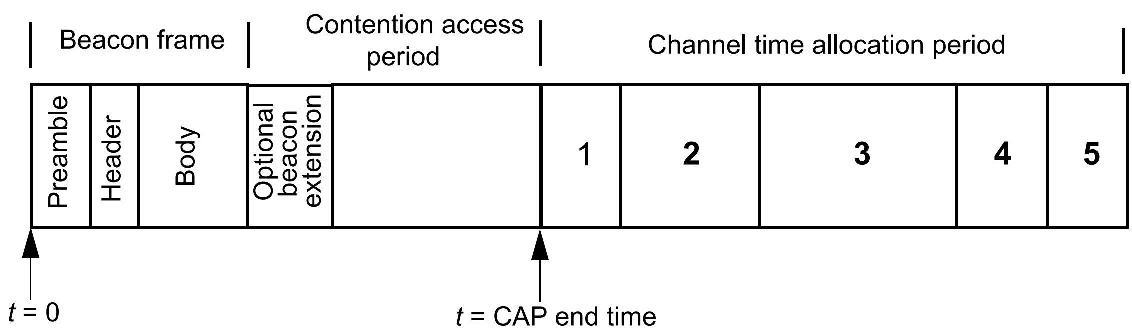

Figure 7-45—Piconet timing relative to the beacon

7.8.2 Time accuracy

A compliant implementation shall maintain the accuracy of the timer to be at least as accurate as *pClockAccuracy*.

7.8.3 Beacon generation

The PNC shall send a beacon at the beginning of each superframe using the Beacon frame described in 6.3.1. The PNC may transmit quasi-omni beacons, as described in 7.8.6.2, between the PNC and a DEV and to enable DEVs located in different directional antenna coverage areas to join the same piconet if the PHY does not require support of omni-directional modes.

If the PNC determines that the Beacon frame is too large or if it is going to split the information in the Beacon frame, it may send one or more Announce commands with the SrcID set to the PNCID and the DestID set to the BcstID following the beacon. This is called an *extended beacon*. Unless it is specified otherwise, the term *beacon* applies to both the Beacon frame and the Announce commands that make up the extended beacon. The IFS between the Beacon frame, the first Announce command, and any additional Announce commands shall be less than a SIFS and greater than a MIFS. If the PNC sends some of the beacon information in the broadcast Announce commands, it shall set the More Data field to indicate more data in the Frame Control field of the Beacon frame and in all but the last Announce command frame used to communicate the IEs. The CAP or the CTAP, if the CAP is not present, begins after the last Announce command that is part of the extended beacon. The PNC shall send CTA IEs, BSID IE, and the Parent Piconet IE, if present, only in the Beacon frame and not in any of the broadcast Announce commands. The Announce commands are sent to the BcstID, and so the ACK Policy field shall be set to no-ACK in these frames.

If all of the CTA IEs contained in the beacon are identical to the CTA IEs transmitted in the previous beacon, the PNC may set the CTA IEs Unchanged field appropriately in the Piconet Mode field of the Piconet Synchronization Parameters field, 6.3.1.1.

If all of the IEs (other than CTA IEs) contained in the beacon are identical to the IEs (other than CTA IEs) transmitted in the previous beacon, the PNC may set the Other IEs Unchanged field appropriately in the Piconet Mode field of the Piconet Synchronization Parameters field, 6.3.1.1.

The PNC shall transmit the beacon such that the time between beacons is the superframe duration with an error of no more than *pClockAccuracy* times the superframe duration. The PNC changes the superframe position or duration using the procedures indicated in 7.14.2 and 7.14.3, respectively.

7.8.4 Beacon reception

All of the DEVs that are associated shall use the beacon start time, CAP end time, and the CTA IEs contained in the beacon to start their transmissions. The superframe duration and the CAP end time in the beacon, as described in 6.3.1, are used to accurately mark the beginning and the end of the CTAP. A lost beacon is defined as one for which the FCS is not valid or when a DEV has not received a beacon at the expected time.

If a DEV receives a Beacon frame with the CTA IEs Unchanged field set in the Piconet Mode field of the Piconet Synchronization Parameters field, and if the DEV has correctly received the previously transmitted beacon, the DEV may assume that the CTA IEs contained in the Beacon frame are identical to the CTA IEs received in the previous beacon. If the CTA IEs Unchanged field is not set, or if the previous Beacon frame was not correctly received, then the DEV shall consider that the CTA IEs in the current beacon contain new information.

If a DEV receives a Beacon frame with the Other IEs Unchanged field set in the Piconet Mode field of the Piconet Synchronization Parameters field, and if the DEV has correctly received the previously transmitted beacon, the DEV may assume that the IEs (other than CTA IEs) contained in the Beacon frame and any Announce commands that make up the beacon are identical to the IEs (other than CTA IEs) received in the previous beacon. If the Other IEs Unchanged field is not set, or if the previous beacon (Beacon frame and any Announce commands) was not correctly received, then the DEV shall consider that the IEs (other than CTA IEs) in the current beacon contain new information.

7.8.5 Beacon information announcement

The PNC sends several IEs in its beacons to inform the DEVs in the piconet about constant or temporary conditions. Some IEs are sent in every beacon, while others are only sent if certain operations are in use, such as power save or a dependent piconet. Some of these IEs are listed in Table 7-2.

Table 7-2—IEs included in beacons as needed

IE	Format	Usage
AS IE for piconet	6.4.8	7.19
Pending channel time map (PCTM) IE for piconet	6.4.9	7.17.3, 7.17.4
PS Status IE for piconet	6.4.20	7.17
CWB IE for piconet	6.4.21	7.17.3

Other IEs are only sent as an announcement of a changed condition in the piconet. These IEs could be for the benefit of all DEVs or for a particular DEV. IEs that are not sent in every beacon are called *announcements* and are listed in Table 7-3.

Table 7-3—Repeated beacon announcements

IE	Subclause	Announced in	Intended for	Subclause
DEV Association IE for piconet	6.4.5	*mMinBeaconInfoRepeat*	All DEVs	7.4.2, 7.4.5
PNC Shutdown IE for piconet	6.4.6	*mMinBeaconInfoRepeat*	All DEVs	7.2.10.2
Piconet Parameter Change IE	6.4.7	*mMinBeaconInfoRepeat*	All DEVs	7.14, 7.15.2, 7.15.3
PNC Handover IE for piconet	6.4.10	*mMinBeaconInfoRepeat*	All DEVs	7.2.4
CTA Status IE for piconet	6.4.11	*mMinBeaconInfoRepeat*	DestID	7.7.2.2, 7.7.2.3
Next PNC IE for piconet	6.4.27	*mMinBeaconInfoRepeat*	All DEVs	7.2.6

If the intended recipient of an IE is all DEVs, the following rules apply:

— The IE shall be sent in at least *mMinBeaconInfoRepeat* consecutive beacons.
— If any DEV is in PSPS, the IE announcement shall be made in a system wake beacon and in at least *mMinBeaconInfoRepeat*–1 consecutive beacons following the system wake beacon.

If the intended recipient of an IE is one individual DEV, the following rules apply:

— The IE shall be sent in at least *mMinBeaconInfoRepeat* consecutive beacons.
— If the DEV is in PSPS mode, the IE announcement shall be made in a system wake beacon and in at least *mMinBeaconInfoRepeat*–1 consecutive beacons following the system wake beacon.

— If the DEV is in DSPS mode, the IE announcement shall be made in one of the DEV's DSPS set wake beacons and in at least *mMinBeaconInfoRepeat*-1 consecutive beacons following the DEV's DSPS set wake beacon.

A CTA Status IE is considered to be intended for all DEVs if the DestID contained in that IE is the BcstID or McstID. Otherwise, the CTA Status IE is intended for the DEV defined by DestID.

7.8.6 Superframe support for directional PHYs

7.8.6.1 Overview

If a PHY does not support omni-directional modes of operation or if it allows DEVs that do not support omni-directional modes of operation, then the superframe, beacon, and CAP need additional features to enable theses DEVs to operate. In particular, the superframe structure for DEVs using these PHY modes will be different, as illustrated in Figure 7-46.

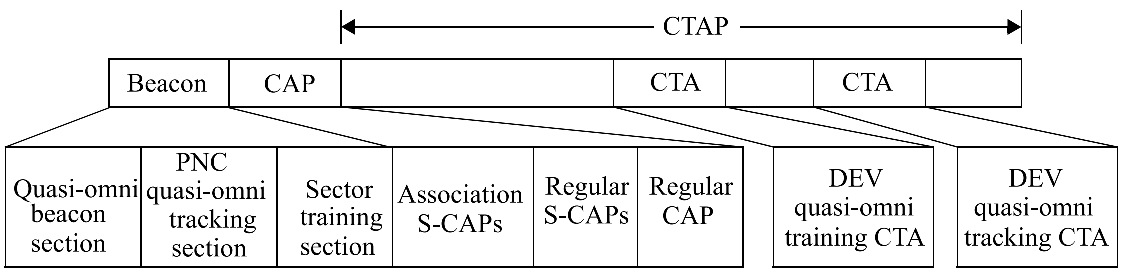

Figure 7-46—Superframe structure for quasi-omni capable PHYs

The optional procedures that enable DEVs to associate and operate with PHYs that do not have omni-directional modes are described in the following subclauses. A summary of the types of CTAs used in the standard is given in F.1.1.

7.8.6.2 Quasi-omni beacon

The quasi-omni beacon shall be structured as illustrated in Figure 7-47.

Quasi-omni beacon		
Quasi-omni beacon section ($I^{(1,t)}$ frames)	PNC quasi-omni tracking section ($I^{(1,t)}$ QTs)	Sector training section ($J^{(1,t)}$ STs)

Figure 7-47—Beacon structure for quasi-omni capable PHYs

The quasi-omni beacon section shall be formatted as illustrated in Figure 7-48.

Beacon frame TX direction # 0, $Q_0^{(1,t)}$	BSIFS	…	Beacon frame TX direction # $i^{(1,t)}$, $Q_{i^{(1,t)}}^{(1,t)}$	BSIFS	…	Beacon frame TX direction # $I^{(1,t)}-1$, $Q_{I^{(1,t)}-1}^{(1,t)}$	BSIFS

Figure 7-48—Quasi-omni beacon section structure

The PNC quasi-omni tracking section shall be formatted as illustrated in Figure 7-49.

QT # 0, $Q_0^{(1,t)}$	BSIFS	…	QT # $i^{(1,t)}$, $Q_{i^{(1,t)}}^{(1,t)}$	BSIFS	…	QT # $I^{(1,t)}–1$, $Q_{I^{(1,t)}-1}^{(1,t)}$	BSIFS

Figure 7-49—PNC quasi-omni tracking section structure

The sector training section shall be formatted as illustrated in Figure 7-50.

ST # 0, $S_0^{(1,t)}$	BSIFS	…	ST # $j^{(1,t)}$, $S_{j^{(1,t)}}^{(1,t)}$	BSIFS	…	ST # $J^{(1,t)}–1$, $S_{J^{(1,t)}-1}^{(1,t)}$	BSIFS

Figure 7-50—Sector training section structure

Quasi-omni coverage during beaconing shall be supported by repeating the Beacon frame $I^{(1,t)}$ times and sending each repetition with a different TX antenna, quasi-omni direction, or array pattern. Each repetition is followed by a BSIFS, as defined in 15.1. The $I^{(1,t)}$ antenna directions or array patterns shall be identified by indices zero through $I^{(1,t)} – 1$. The number of Beacon frame repetitions $I^{(1,t)}$ and the index of the current beacon transmit direction are parameters in the Synchronization IE, as described in 6.4.29. In the following, the term *direction* shall be used to refer to an antenna direction or an array pattern.

When the PNC Quasi-omni Tracking Present field in the Synchronization IE is set to one, then a PNC quasi-omni tracking section shall follow the quasi-omni beacon section to allow devices in the piconet to track the PNC quasi-omni directions. The PNC quasi-omni tracking section consists of $I^{(1,t)}$ quasi-omni training (QT) sequences transmitted in the $I^{(1,t)}$ PNC quasi-omni directions. The $I^{(1,t)}$ QT sequences may be transmitted at once in a superframe or distributed over multiple superframes. The QT sequence shall be identical to the long preamble.

When the Sector Training field in the Synchronization IE is set to one, then a sector training section shall follow the PNC quasi-omni tracking section to enable proactive beam forming, as described in 15.7. The sector training section consists of $J^{(1,t)}$ sector training (ST) sequences transmitted in the $J^{(1,t)}$ PNC sector directions. The $J^{(1,t)}$ ST sequences may be transmitted once in a superframe or distributed over multiple superframes. The ST sequences shall be identical to the long preamble.

7.8.6.3 Directional CAP

The CAP period for directional communication may be divided into three sections—an association section, a regular sub-CAP (S-CAP) section, and a regular CAP section as shown in Figure 7-51. If PNC allows new association requests, an association section exists at the beginning of the CAP period. The association section may be further divided into a set of $I^{(1,r1)}$ equal size association S-CAPs corresponding to some of the $I^{(1,r)}$ PNC different quasi-omni receive directions within one superframe or distributed over multiple superframes ($r1 \leq r$). The regular S-CAP section is divided into the $I^{(1,r1)}$ equal size S-CAPs corresponding to some of the $I^{(1,r)}$ PNC different quasi-omni receive directions within one superframe or distributed over multiple superframes. Each S-CAP is received by the PNC using a different antenna receive direction.

CAP								
Association S-CAP				Regular S-CAP				Regular CAP
Association S-CAP #0	Association S-CAP #1	…	Association S-CAP #$I^{(1,r1)}$–1	S-CAP #0	S-CAP #1	…	S-CAP #$I^{(1,r1)}$–1	

Figure 7-51—Quasi-omni CAP structure

The association CAP shall be used solely for devices to send Association Request commands to the PNC and for the Imm-ACK from the PNC to the Association Request command. The regular CAP and regular S-CAP may be used for all other command and data exchanges. It is up to the implementer to determine the method used to achieve omni-directional communications for the SC and HSI PHY modes in the regular CAP.

If the AAS field in the piconet synchronization parameters field is set to zero, indicating that the PNC is SAS, the number of S-CAPs shall be one-to-one with the number of beacons ($I^{(1,t)} = I^{(1,r)}$). That is, the PNC transmit direction used for the i^{th} Beacon frame transmission shall be used for the i^{th} S-CAP.

If the AAS field in the piconet synchronization parameters field is set to one, indicating that the PNC is AAS, the number of S-CAPs shall be equal to $I^{(1,r)}$, which is specified in the Synchronization IE, as described in 6.4.29. The special case where $I^{(1,r)} = 1$ indicates that the PNC is omni capable on reception.

Before two peer DEVs communicate in the regular CAP and/or regular S-CAPs, the two DEVs may perform beam forming, as described in 15.5, if they do not know the antenna directions to point to each other.

To support directional peer communication in the regular CAP and/or regular S-CAPs, the source DEV may send the PNC an Announce command with Directional Peer IE with the Request/Release field set to one, as described in 6.4.43. If the PNC allows the directional peer communication, the PNC shall include the Directional Peer IE in the beacon to announce that the two devices will use regular CAP and/or regular S-CAPs for directional peer communication.

After the source and destination DEVs receive the Directional Peer IE in the beacon from the PNC, the two DEVs may switch their antenna directions and communicate with each other in the regular CAP and/or regular S-CAPs. After the two devices complete directional peer communication, the source device may send the PNC an Announce command with Directional Peer IE with the Request/Release field set to zero, and the PNC shall remove Directional Peer IE from its beacon.

Directional peer-to-peer communication between two non-PNC DEVs is not bound by the S-CAP boundaries.

7.8.6.4 Directional association

A DEV that is not omni capable on reception and supports multiple receive directions shall implement directional association. Let $I^{(1,t)}$ and $I^{(1,r)}$ be the number of PNC quasi-omni transmit and receive directions, respectively, and let $I^{(2,t)}$ and $I^{(2,r)}$ be the number of DEV quasi-omni transmit and receive directions, respectively, of a DEV that wants to find a PNC and associate with that PNC.

The best and second best pair of PNC transmit and DEV receive or DEV transmit and PNC receive directions are referred to as the *best antenna direction pair* and *second best antenna direction pair,* respectively.

While searching for a PNC, a DEV shall listen to quasi-omni beacons at all $I^{(2,r)}$ quasi-omni receive directions to find the best and second best antenna direction pairs based on LQI measurement. The DEV shall use the best DEV quasi-omni receive direction to receive further transmissions from the PNC when quasi-omni transmission is used. The DEV shall include the information of the best PNC quasi-omni transmit direction in its Association Request commands, as defined in 6.5.2.2, to inform the PNC the best PNC quasi-omni transmit direction for further quasi-omni transmissions to the DEV.

The DEV shall track the best and second best antenna direction pairs during the quasi-omni beacon section and PNC quasi-omni tracking section of beacons based on LQI measurement. If the beacon quality in the second best antenna direction pair is better than the best antenna direction pair, the PNC and the DEV shall switch to the second best antenna direction pair, which then becomes the new best antenna direction pair.

The association procedure of a DEV depends on the antenna types at both the PNC and the DEV. The DEV shall transmit one or multiple Association Request commands, as described in 6.5.2.2. If both the DEV itself and the PNC utilize SAS antennas, this command shall be transmitted on DEV antenna quasi-omni direction $I^{(2,t)} = I^{(2,r)}$, during the S-CAP with index, $I^{(1,t)}$.

If either the DEV or the PNC is AAS, the DEV does not know which quasi-omni transmit direction to use, nor which S-CAP is the best. In this case, the DEV shall transmit the Association Request command at different antenna direction pairs during association S-CAPs in one or multiple superframes until it receives an Association Response command successfully or association timeout. The Association Request command shall be sent individually instead of back to back, with random backoff applied to each one. The backoff window used by the DEV to send the Association Request command shall be the same for the cycle of $I^{(2,t)}$ quasi-omni transmit directions. If a complete cycle of quasi-omni transmit direction fails, then the backoff window shall be increased, as defined in 7.6.3. The Association Response command shall include the information of the antenna direction pair from which it receives the Association Request command. The Association Response command may be sent in a CTA.

Once an Association Response command or an Imm-ACK to the Association Request command is received successfully, as defined in 7.4.2, the DEV shall cease the transmission of Association Request command in association S-CAPs and ignore other copies of the same Association Response command from PNC. The DEV shall use the regular S-CAP of the PNC quasi-omni receive direction carried in the Association Response command for all further CAP transactions with the PNC before the completion of the best antenna direction pair searching. In addition, the DEV shall use the DEV transmit quasi-omni direction carried in the Association Response command before the completion of the best antenna direction pair searching.

7.8.6.5 DEV quasi-omni transmit direction training and tracking

Since the antenna direction pair found at the association stage may not be the best pair due to possible collisions in CAP, the PNC shall reserve a CTA to search for the best and the second best antenna direction pairs when channel has free time. The CTA structure for DEV quasi-omni transmit direction training is illustrated in Figure 7-52.

Quasi-omni training DEV → PNC $I^{(2,t)}$ cycles	SIFS	PNC → DEV Announce command with Feedback IE $Q^{(1,t)}_{i^{(1,t)}} \to Q^{(2,r)}_{i^{(2,r)}}$	SIFS	DEV → PNC Imm-ACK $Q^{(2,t)}_{i^{(2,t)}} \to Q^{(1,r)}_{i^{(1,r)}}$	SIFS

Figure 7-52—CTA structure for DEV quasi-omni transmit direction training

The structure of the quasi-omni training from DEV to PNC is illustrated in Figure 7-53.

Quasi-omni training cycle #0, $I^{(2,r)}$ cycles	…	Quasi-omni training cycle #$i^{(2,t)}$, $I^{(2,t)}$ cycles	…	Quasi-omni training cycle #$I^{(2,t)}-1$, $I^{(2,t)}$ cycles

Figure 7-53—Quasi-omni training DEV –› PNC structure

The structure of the quasi-omni training cycle is illustrated in Figure 7-54.

QT #0 $Q^{(2,t)}_{i^{(2,t)}} \to Q^{(1,r)}_0$	QT #1 $Q^{(2,t)}_{i^{(2,t)}} \to Q^{(1,r)}_1$	…	QT #$I^{(2,t)}-1$ $Q^{(2,t)}_{i^{(2,t)}} \to Q^{(1,r)}_{i^{(1,r)}-1}$	SIFS

Figure 7-54—Quasi-omni training cycle structure

The quasi-omni training consists of $I^{(2,t)}$ cycles. During each cycle, the DEV shall send $I^{(1,r)}$ repetitions of a QT sequence, as defined in 7.8.6.2, in the same direction. Each cycle except the last one shall end with a BSIFS. The $I^{(2,t)}$ cycles shall be sent in $I^{(2,t)}$ different directions, namely $[Q^{(2,t)}_0, Q^{(2,t)}_1, \ldots, Q^{(2,t)}_{I^{(2,t)}-1}]$.

During a cycle, the PNC shall attempt to receive each of the $I^{(1,r)}$ quasi-omni training sequences using a different direction. The $I^{(1,r)}$ different directions, $[Q^{(1,r)}_0, Q^{(1,r)}_1, \ldots, Q^{(1,r)}_{I^{(1,r)}-1}]$, during a cycle shall correspond to the PNC's quasi-omni receive directions.

At the completion of the full $I^{(2,t)}$ cycles, the PNC will have had an opportunity to receive a QT sequence using each combination of DEV2 transmit quasi-omni direction (0 to $I^{(2,t)} - 1$) and PNC receive quasi-omni direction (0 to $I^{(1,r)} - 1$). Based on this information, the PNC selects the best quasi-omni pair, i.e., DEV2's optimal transmit quasi-omni direction, $I^{(2,t)}_{i^{(2,t)}}$, and the PNC optimal transmit and receive sector, $I^{(1,r)}_{i^{(1,r)}}$.

Following the quasi-omni training, the PNC shall transmit its quasi-omni feedback in a Feedback IE by sending an Announce command with Imm-ACK requested. The Announce command shall be sent in the optimal transmit quasi-omni direction, $Q^{(1,t)}_{i^{(2,t)}}$, and DEV shall listen on its optimal receive direction, $Q^{(1,r)}_{i^{(2,r)}}$. The Feedback IE informs DEV of its optimal transmit quasi-omni direction, $Q^{(2,t)}_{i^{(2,t)}}$, second best quasi-omni direction, and the corresponding LQIs.

The PNC shall track the best and second best antenna direction pairs based on LQI measurement by allocating a tracking CTA periodically when channel has free time. The CTA structure for DEV quasi-omni transmit direction tracking is illustrated in Figure 7-55. If the signal quality in the second best pair is better than the best pair, the PNC and the DEV shall switch to the second best pair, which becomes the new best antenna direction pair thereafter.

DEV –› PNC Quasi-omni training	SIFS	PNC –› DEV Announce command with Feedback IE	SIFS	DEV –› PNC Imm-ACK	SIFS

Figure 7-55—CTA structure for DEV quasi-omni transmit direction tracking

The structure for quasi-omni tracking DEV → PNC is illustrated in Figure 7-56.

QT #i, $Q_{i\,(2,t)}^{(2,t)}$	BSIFS	QT #j, $Q_{j\,(2,t)}^{(2,t)}$

Figure 7-56—Quasi-omni direction training cycle structure

7.9 Synchronization for pairnet

7.9.1 General

All PRDEVs within a pairnet shall be synchronized to the PRC's clock during PSP. The beacon sent at the beginning of every superframe contains the information necessary to time-synchronize the DEVs in the pairnet. See 6.3.1 for the definition of the timing parameters sent in the beacon.

Each DEV in the pairnet, including the PRC, shall reset its superframe clock to zero at the beginning of the beacon preamble, as shown in Figure 7-57. All times in the superframe shall be measured relative to the beginning of the beacon preamble. If a DEV does not hear a beacon, it should reset its superframe clock to zero at the instant where it expected to hear the beginning of the beacon preamble.

After association, PAP is used instead of CAP, and the end time is the same as the end of the superframe. Therefore, no synchronization is necessary.

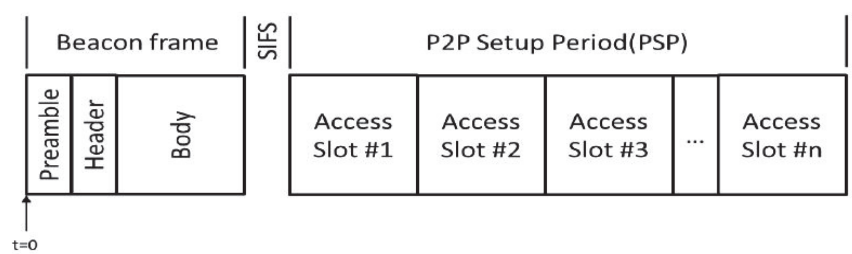

Figure 7-57—Pairnet timing relative to the beacon

7.9.2 Time accuracy

A compliant implementation shall maintain the accuracy of the timer to be at least as accurate as *pClockAccuracy*.

7.9.3 Beacon generation

The PRC shall send a beacon at the beginning of each superframe using the Beacon frame described in 6.3.1 before the reception of the Association Request command. An extended beacon shall not be supported.

The PRC shall transmit the beacon such that the time between beacons is the superframe duration with an error of no more than *pClockAccuracy* times the superframe duration.

7.9.4 Beacon Information announcement

The PRC sends several IEs in its beacons to inform the DEVs in the pairnet about constant or temporary conditions. Some of these IEs are listed in Table 7-4.

Table 7-4— IEs included in beacons as needed

IE	Format	Usage
MIMO Information IE	6.4.44	13.2.9.1
Higher Layer Information IE	6.4.45	7.5.3

7.10 Fragmentation and defragmentation for piconets

Fragmentation may be performed at the transmitting DEV on each MSDU. In addition, certain commands, i.e., MAC command data units (MCDUs), may be fragmented, as indicated in 6.5. All fragments shall be of equal size, except the last fragment, which may be shorter. Once the MSDU/MCDU is fragmented and a transmission attempted, it shall not be refragmented. The smallest size of a fragment, excluding the last fragment, shall be at least *pMinFragmentSize*. A DEV indicates its preferred fragment size for reception in the Preferred Fragment Size field in the DEV Capabilities field, as described in 6.4.12, that it sends to the PNC when the DEV associates with the piconet.

Each fragment shall be sent with the Last Fragment Number field set to the highest fragment number of the current MSDU/MCDU, which is one less than the total number of fragments of the current MSDU/MCDU.

The first fragment shall be sent with the Fragment Number field set to zero. Each subsequent fragment shall be sent with the Fragment Number field incremented. However, the Fragment Number field shall not be incremented when a fragment is retransmitted.

All fragments of the same MSDU/MCDU shall have the same MSDU/MCDU number.

Defragmentation of an MSDU/MCDU is the reassembly of the received fragments into the complete MSDU/MCDU. The MSDU/MCDU shall be completely reassembled in the correct order before delivering it to the frame convergence sublayer (FCSL). Any MSDU/MCDU with missing fragments shall be discarded.

The receiving DEV shall not deliver an MSDU/MCDU to the FCSL until all of the fragments have been obtained. The receiving DEV may discard the fragments of an MSDU/MCDU if it is not completely received within a timeout determined by the receiving DEV. The destination DEV may also discard the oldest incomplete MSDU/MCDU if otherwise a buffer overflow would occur. If the no-ACK policy is used, the destination DEV shall discard an MSDU/MCDU immediately if a fragment is missing.

A DEV shall support concurrent reception of fragments of at least three MSDU/MCDUs including isochronous streams, asynchronous data, and commands.

For frames with Imm-ACK mechanism, a DEV shall not send another fragment or frame with the same stream index to the same DEV until the sending DEV has received an Imm-ACK frame response to that frame or it has timed out on sending the frame.

If Dly-ACK is used, unacknowledged fragments from multiple MSDUs belonging to the same stream may be retransmitted in the same burst. In this case it is the responsibility of the destination DEV to deliver the MSDUs in the correct order to the FCSL.

7.11 Aggregation

7.11.1 Piconet standard aggregation

Figure 7-58 illustrates the aggregation process. The originating DEV, upon receiving an MSDU, maps it into a subframe payload. If the length of the MSDU exceeds the predetermined value indicated in the Preferred Fragment Size field in Capability IE, as defined in 6.4.12, the MSDU shall be fragmented and mapped into multiple subframe payloads. Each MSDU is assigned an unique MSDU number for identification. If fragmentation is adopted, each fragment is assigned a fragment number for identification within the MSDU. All the fragments of the same MSDU shall have the same MSDU number.

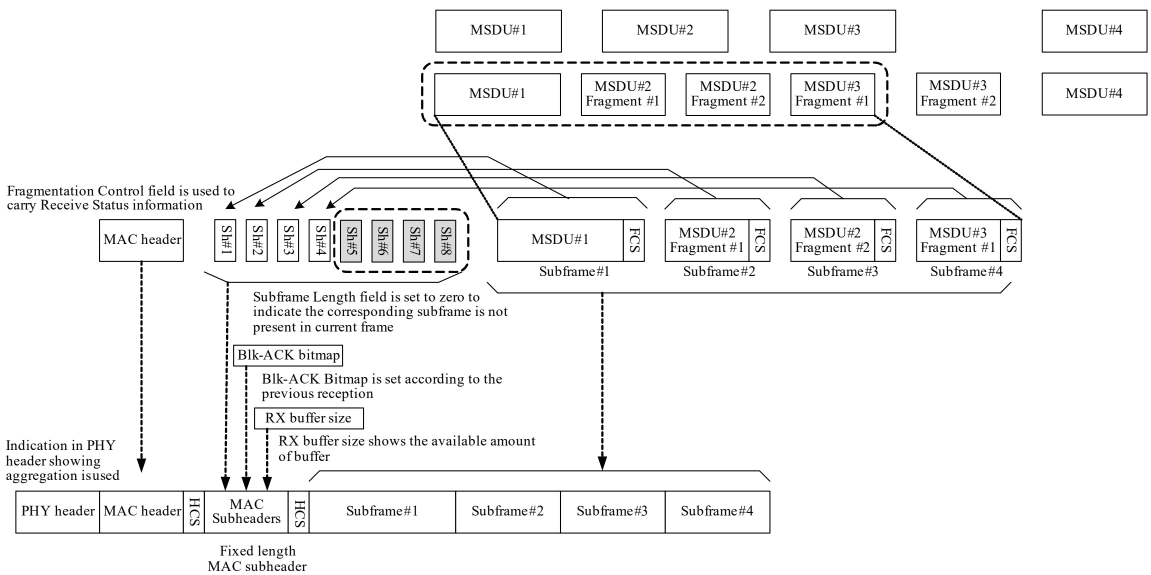

Figure 7-58—Aggregation at originating DEV

A subheader is created and configured, as defined in 6.2.11.2, for each subframe to contain the necessary information that helps the target DEV to retrieve the original data. The ACK Policy field in MAC header shall be set to Blk-ACK, as described in 6.2.2.5.

All the subheaders are combined together to form the MAC subheader. As specified in 6.2.11.2, up to 8 subframes aggregated into a single frame. All unused subframes have zero length in the Subframe Length field.

Figure 7-59 illustrates the deaggregation process. After receiving the aggregated frame, the target DEV divides it into subframes according to the information in MAC subheader, and validates each subframe by FCS. To recreate the original MSDU, the target DEV uses the MSDU Number field and the Fragment Number field in the subheader, as defined in 6.2.11.2.

The standard aggregation supports unidirectional and bidirectional data transmission by attaching ACK information with data. For unidirectional transmission, the target DEV upon receiving an aggregated frame replies an empty Data frame with the Blk-ACK Bitmap field properly configured to support retransmission, as described in 7.12.6.2. For bidirectional data transmission, the target DEV sends ACK information with a Data frame that may contain a Data Payload field to the originating DEV.

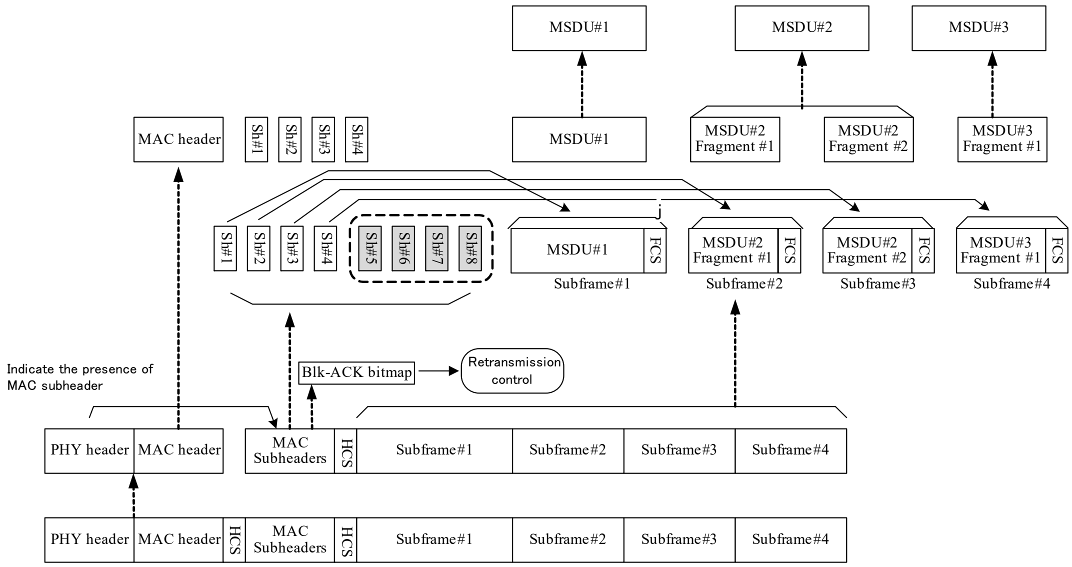

Figure 7-59—Deaggregation at target DEV

To avoid buffer overflow, before starting real data transmission, the originating and target DEV may exchange an empty data frame with the RX Buffer Size field properly configured, as defined in 6.2.11, to inform each other the available receiving buffer size. This information is used by the DEVs to adjust the subframe number and subframe length when sending real data.

The Fragmentation Control field in the MAC header of an aggregated frame shall always be used to carry the Receive Status field, as defined in 12.1.9.3.

7.11.2 Piconet low-latency aggregation

Before starting low-latency aggregation, the originating and target DEVs shall exchange the desired CTA relinquish duration value by sending CTA Relinquish Duration IE, as described in 6.4.33, in an Announce command, as defined in 6.5.6.2. Low-latency aggregation shall not be allowed in CP because the usage of relinquish operation is not allowed in CP.

The DEV chooses the smaller value among its own CTA relinquish duration and that of the target DEV to start the bidirectional CTA relinquish duration (BCRD) timer.

In low-latency aggregation mode, the originating DEV receives MSDUs from the FCSL and aggregates them into the Frame Payload field. Each MSDU is formatted, as described in 6.2.11.3. The originating DEV shall also aggregate zero-length MSDUs if there is no MSDU available from the FCSL until an MSDU is available, or until the BCRD timeout has expired. If the originating DEV has not received any MSDU from the FCSL until the beginning of the next BCRD period, it may transmit a zero-length frame with the ACK Policy field in MAC header set to No-ACK, the Aggregation field set to zero, and the Low-latency Mode field set to one in the PHY header (for the purpose of maintaining the link alive). Each MSDU transmitted is assigned an MSDU number on its first transmission that is used to identify it uniquely in the MSDU sequence delivered to the destination FCSL. The sequence number of zero-length MSDU is assigned the most recent sequence number transmitted by the source for which it has received an ACK. The MSDU sequence number in a zero-length MSDU shall be ignored by the destination. No retransmission is required for zero-length MSDU.

When the target DEV receives an aggregated MAC frame, it verifies the Low-latency Aggregation field, as defined in 12.2.4.3.2.

The target DEV reads the Frame Payload field, parsing each aggregated MSDU based on the received MSDU subheader. For each MSDU subheader, the destination validates its HCS. If the HCS is not valid, the destination shall continue searching for a match of a valid MSDU HCS, shifting the next expected MSDU on octet alignment until one is found. If the FCS is correct, the destination passes the MSDU in order to the FCSL, withholding any trailing MSDUs until all intermediate MSDU fragments are retransmitted and received correctly.

The target DEV sets the corresponding bit in the next transmitted Blk-ACK Bitmap field according to its order offset taking the value of MSDU Response Number field of the transmitted MAC subheader as reference. The MSDU Response Number field is updated upon reception of a valid MAC subheader, from the MSDU Request Number field. In case the MSDU order offset is not sequential to the previous MSDU order offset, the corresponding range of bits shall be cleared in the Blk-ACK Bitmap field.

The target DEV shall inform the originating DEV the available receiving buffer size using RX buffer size field, as defined in 6.2.11. The originating DEV adjusts the maximum number of MSDUs to send in the next frame to avoid buffer overflow.

If the MSDU is incorrectly received, the corresponding bit is set to zero in the next transmitted Blk-ACK Bitmap field according to its order offset in the current aggregated frame.

The source and destination interchange roles during the CTA, every BCRD or less, utilizing the CTA relinquish mechanism described in 7.6.4.9, to enable minimum latency retransmission of incorrectly received MSDUs.

7.11.3 Pairnet aggregation

Pairnets using the THz PHY shall not use aggregation.

Pairnet data transmission shall use the pairnet aggregation frame format.

Figure 7-60 illustrates the aggregation process. The originating PRC or DEV, upon receiving an MSDU, maps it into a payload. If the length of the MSDU exceeds the Preferred Payload Size negotiated during the association process, the MSDU shall be fragmented and mapped into multiple payloads. Each payload is assigned a unique sequence number for identification. A payload that is not a last fragment of an original MSDU shall have a payload with the length of the Preferred Payload Size in the Operation Parameters field.

Each fragment shall have a unique sequence number assigned in ascending order.

Command frames shall have a separate and unique successive sequence number. The initial number of both sequence numbers is zero.

A subheader is created and configured, as defined in 6.3.6, for each subframe to contain the necessary information that helps the target DEV to retrieve the original data. The ACK Policy field in MAC header shall be set to Stk-ACK as described in 6.2.2.6.

Padding octets shall be appended to each subframe except for the last subframe in the aggregated frame to make the subframe a multiple of *n* bits in length, where *n* is the largest unit of subframe padding supported by both the transmitter and the receiver of the aggregated frame. A 32-bit unit of padding is mandatory and another optionally supported padding unit is indicated by the Supported Unit of the Subframe Padding field in the PRC Capability and PRDEV Capability. The content of these padding octets is not specified.

As specified in 6.2.13, up to 256 subframes are aggregated into a single frame. Figure 7-61 illustrates the deaggregation process. After receiving the aggregated frame, the target DEV divides it into subframes according to the information in MAC subheader and validates each subframe by subheader HCS and payload FCS. To recreate the original MSDU, the target DEV uses the Sequence Number field and Last fragment field in the subheader, as defined in 6.3.6.1.

For non-secure and secure Aggregated Multi-protocol Data frames, in the case where a single MSDU is fragmented into multiple subframes, the Data ID and the Data Header fields shall be copied to each relevant subframe fields.

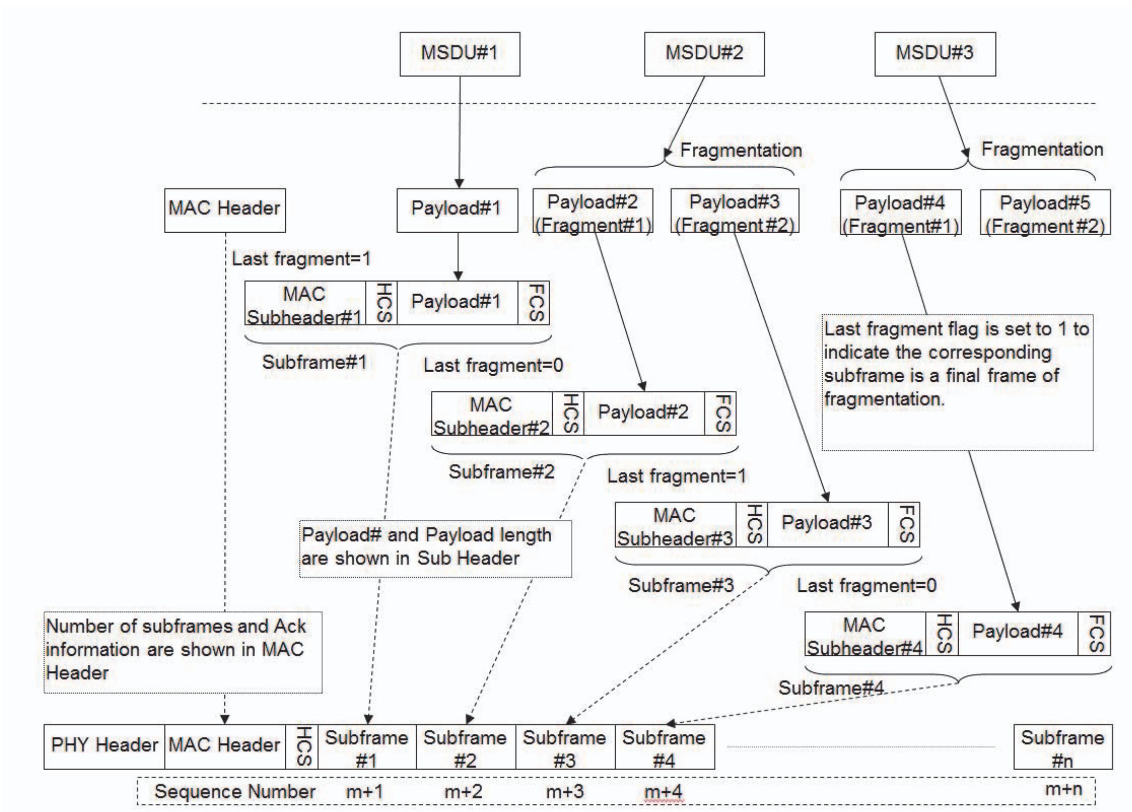

Figure 7-60—Aggregation at originating DEV

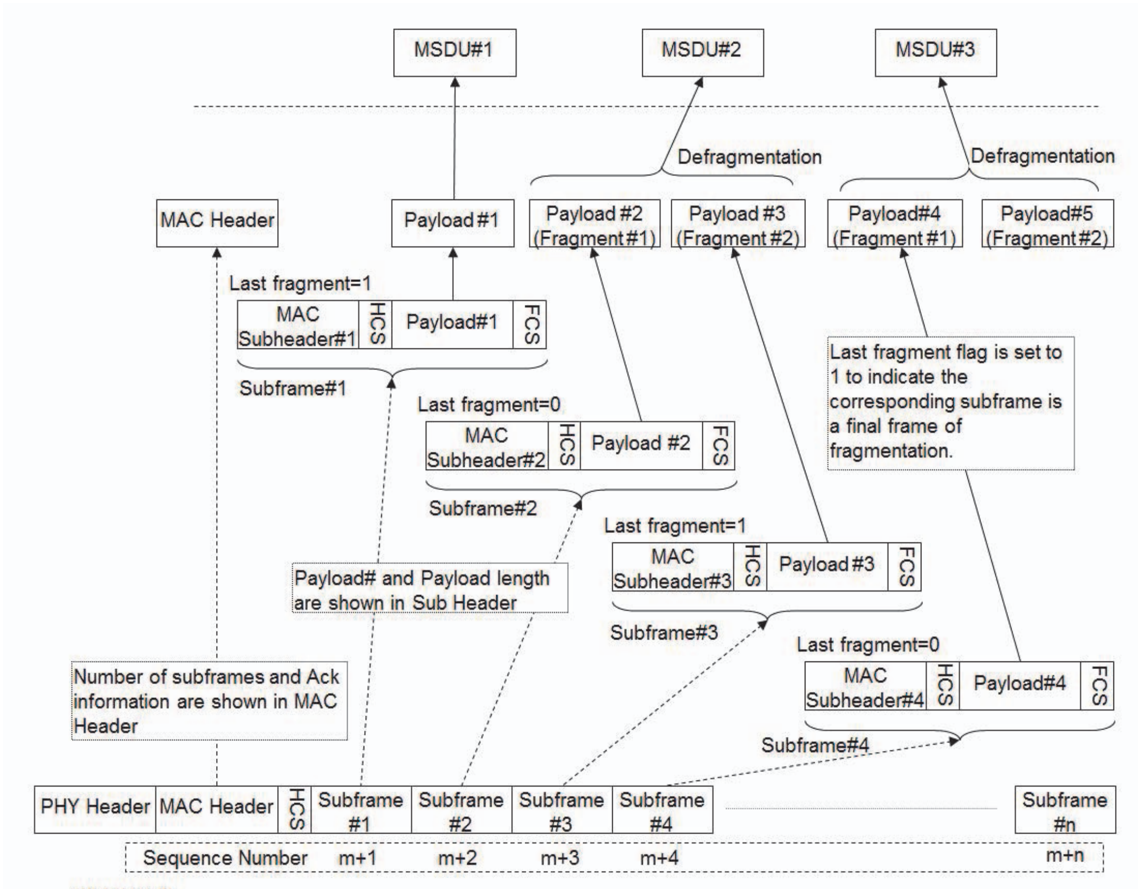

Figure 7-61—Deaggregation at target DEV

7.12 Acknowledgment and retransmission

7.12.1 Overview

The acknowledgment types defined for this standard are as follows:

- No acknowledgment (no-ACK)
- Immediate acknowledgment (Imm-ACK)
- Delayed acknowledgment (Dly-ACK)
- Implied acknowledgment (Imp-ACK)
- Block acknowledgment (Blk-ACK)
- Stack acknowledgment (Stk-ACK)

7.12.2 No acknowledgment (no-ACK)

A transmitted frame with the ACK Policy field set to indicate no-ACK shall not be acknowledged by the intended recipient(s). The transmitting DEV assumes that the frame is successful for all its local management entities and proceeds to the next frame scheduled for transmission. The ACK Policy field in broadcast and multicast addressed frames shall be set to no-ACK upon transmission.

7.12.3 Immediate acknowledgment (Imm-ACK) for piconets

A directed frame that expects an Imm-ACK shall have the ACK Policy field in that directed frame set to indicate the same, as defined in 6.2.2.5. If the intended recipient of a directed frame correctly receives the frame, it shall send the Imm-ACK frame, as described in 7.6.2.

7.12.4 Delayed acknowledgment (Dly-ACK) for piconets

Dly-ACK shall be used only for directed stream data frames, i.e., isochronous connections, where the Dly-ACK mechanism has been set up with negotiation between the source and destination DEVs. The Dly-ACK mechanism is initiated by the source DEV sending a single data frame with the ACK Policy field set to Dly-ACK Request.

If the MAC header was correctly received, as described in 7.1, and the destination DEV accepts the use of Dly-ACK, it shall respond with a Dly-ACK frame, acknowledging the received data frame, if it was correctly received, and setting the Max Burst field to a value representing the maximum number of *pMaxFrameBodySize* MPDUs the source DEV may send in one burst. Because the receiver buffer requirement is equal to Max Burst field times *pMaxFrameBodySize*, the source may send as many smaller frames as will fit in the receive buffer window, up to a maximum of Max Frames, as provided in the Dly-ACK frame, as described in 6.3.2.2. The destination DEV may change the value of the Max Burst field and the Max Frames field in each Dly-ACK frame.

If the destination DEV wants to decline the use of the Dly-ACK mechanism, it shall reply with an Imm-ACK frame.

If the value of the Max Burst field is set to zero, the source DEV shall stop transmitting in the current CTA and reopen the Dly-ACK mechanism by sending a single frame with the ACK Policy field set to Dly-ACK Request in the next CTA for this stream. If the value is not zero, the source DEV may continue transmission in the current CTA, if the time is available.

A DEV shall send a Dly-ACK frame in response to a correctly received MAC header, as described in 7.1, that has the ACK Policy set to Dly-ACK request and the DestID set to the DEVID of this DEV or, if applicable, the PNCID. Because the Dly-ACK frame is sent even if the FCS failed on the data frame with the ACK Policy set to Dly-ACK request, the DEV shall check the contents of the Dly-ACK frame to determine which, if any, MSDUs were correctly received.

A DEV should not initiate a Dly-ACK procedure unless it has determined that the receiving DEV supports Dly-ACK by checking the DEV Capabilities field in the Capability IE, as defined in 6.4.12.

The source DEV may change the ACK policy in a stream from Dly-ACK to Imm-ACK or no ACK by sending a frame with the ACK Policy field set to one of those values. This has the effect of canceling the Dly-ACK policy and the source shall use the Dly-ACK negotiation procedure before restarting the Dly-ACK mechanism. The receiver shall no longer maintain the ACK status of any previous frames sent with the ACK Policy field set to Dly-ACK.

The source DEV may send any MPDUs including retransmissions and new MPDUs, as long as the total size of transmitted data is less than size indicated by the Max Burst value set by the destination DEV.

If the Dly-ACK frame is not received when requested, the last data frame of the burst is repeated until the Dly-ACK frame is received. The source DEV may send an empty data frame that was not in the original burst as an alternative to resending the last data frame, as long as the total number of frames, including the empty one, does not exceed Max Frames. The source DEV shall not start or resume burst transmissions until a Dly-ACK frame is received.

The destination MAC shall deliver MSDUs for each isochronous stream in ascending MSDU number order to its FCSL. If necessary to accomplish this, a destination MAC may discard correctly received, as described in 7.1, (and potentially acknowledged) frames.

Figure 7-62 illustrates the Dly-ACK negotiation. In the figure, the term *Alt* indicates that two alternate outcomes are illustrated in the MSC.

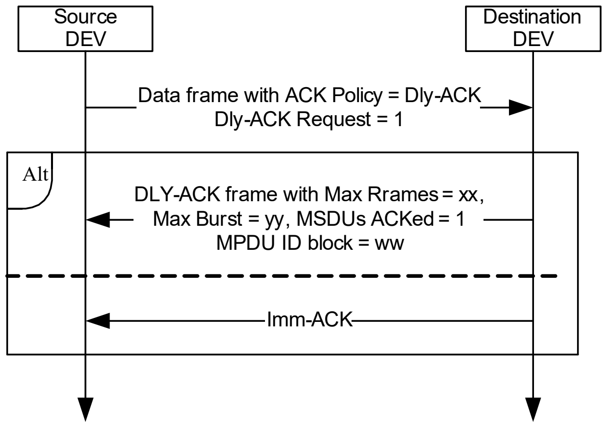

Figure 7-62—MSC for Dly-ACK negotiation

7.12.5 Implied acknowledgment (Imp-ACK) for piconets

Imp-ACK is one method that allows a CTA to be used for bidirectional data transfer. With Imp-ACK, the ACK is implied when the target DEV sends any frame, called the *response frame*, in response to a frame that has an ACK policy of Imp-ACK.

A DEV should not initiate an Imp-ACK procedure unless it has determined that the receiving DEV supports Imp-ACK by checking the DEV Capabilities field in the Capability IE, as defined in 6.4.12.

The originating DEV sets the ACK Policy to Imp-ACK in a frame to begin the Imp-ACK procedure. Only the DEV that is the source of a CTA shall initiate the Imp-ACK procedure. When the target DEV successfully receives a frame with the ACK policy set to Imp-ACK, the DEV shall respond with either an Imm-ACK frame or with a command or data frame. The target DEV shall only send a frame to the originating DEV, unless transmit control of the CTA has been relinquished, as described in 7.6.4.9.

If the target DEV sends a frame to the originating DEV in response to a frame with ACK policy of Imp-ACK, the target DEV shall set the ACK policy of the response frame to one of the following; no-ACK, Imm-ACK, or Imp-ACK.

If there is not sufficient time in the CTA for the response frame and any required acknowledgments before the end of the CTA, the target DEV shall respond with an Imm-ACK. If the target DEV does not know the end time of the current CTA, it shall only send an Imm-ACK frame as the response frame. The target DEV shall only send one frame in response to a frame with the ACK policy set to Imp-ACK. The response frame may be of any length as long as the frame and any required ACKs do not exceed the end of the current CTA.

If the target DEV successfully receives a MAC header for a frame with the ACK policy set to Imp-ACK but does not successfully receive the Frame Payload field, i.e., the FCS check fails, it may still send a data or command frame in response. In this case, the target DEV shall set the Imp-ACK NAK field, 6.2.2.9, to indicate that it did not successfully receive the frame from the originating DEV in the response frame.

As in any CTA, a DEV should only transmit to a DEV that it knows is listening to the CTA. Imp-ACK shall not be used for broadcast or multicast frames, i.e., frames with the DestID set to BcstID, McstID, or a GrpID. Imp-ACK shall not be used in the CAP or a contention CTA.

The use of Imp-ACK to enable bidirectional communication in a CTA does not imply that the stream is bidirectional as well. All streams in this standard are unidirectional. Only the originating DEV is allowed to send frames using the stream index assigned by the PNC. Frames transmitted in the opposite direction in the CTA shall use either the asynchronous stream index or a stream index that has been assigned to the DEV that is sending the frame.

If the destination DEV in a CTA has data to send to the source DEV of that CTA, it should set the More Data field in Imm-ACK frames that it sends to the source. This will indicate to the source that the destination is requesting that the source either use Imp-ACK or relinquish the CTA to the destination, as described in 7.6.4.9.

Figure 7-63 shows an example of the use of Imp-ACK where successive frames have an ACK policy of Imp-ACK.

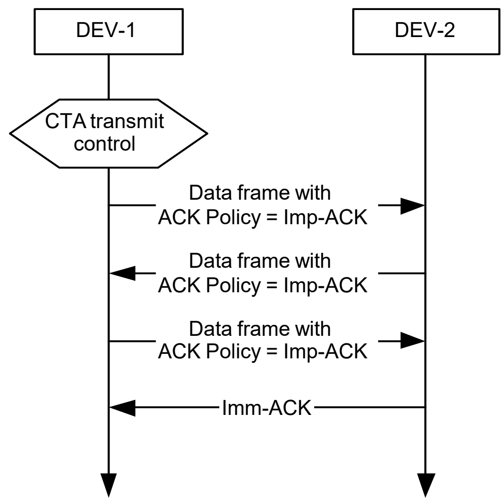

Figure 7-63—Example of the use of Imp-ACK where the responding frame is sent with an ACK policy of Imp-ACK

Figure 7-64 shows an example of the use of Imp-ACK where one of the frames fails the FCS verification and the destination responds with the Imp-ACK NAK field set.

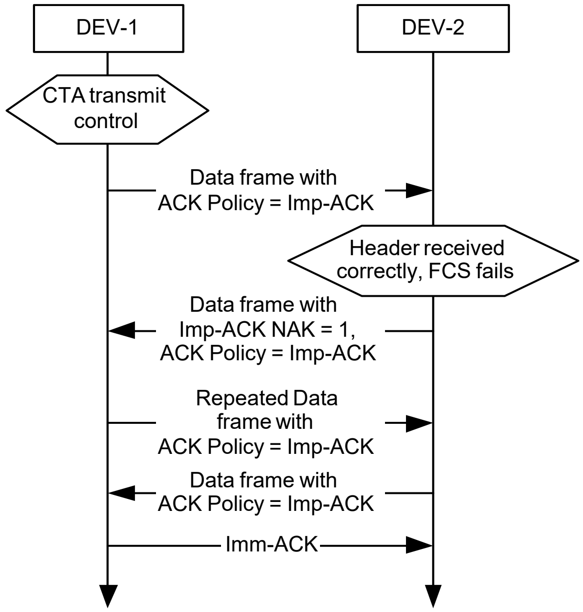

Figure 7-64—Example of the use of Imp-ACK where one of the frames fails FCS verification

7.12.6 Block acknowledgment (Blk-ACK) for piconets

7.12.6.1 General

Blk-ACK shall only be used with an aggregated frame. The destination, upon receiving an aggregated frame, checks each subframe. Based on the status of the subframe, either correctly or incorrectly received, the corresponding ACK bit or MSB/LSB ACK bit of the Blk-ACK Bitmap field of the MAC subheader in the Data frame will be set as described in 6.2.11.2. The originating DEV, after reading the Blk-ACK Bitmap field in MAC subheader, handles subframe retransmission.

For example, if the destination correctly receives subframes 0–4, 6, and 8–10, then the destination would set bits 0–4, 6, and 8–10 to one and the rest of the bits in the Blk-ACK Bitmap fields to zero.

The control of retransmission is different according to the different aggregation mode, standard or low latency.

7.12.6.2 Blk-ACK for standard aggregation

For standard aggregation, the rules in this subclause apply.

If the MAC subheader is incorrectly received by the target DEV, all the subframes in the frame are considered invalid.

If the MAC subheader is correctly received, but some of the subframes are incorrectly received, then the target DEV responds with a Data frame with the Blk-ACK Bitmap field in the MAC subheader set to indicate the subframes that were incorrectly received, as described in 6.2.11. If the data transmission is unidirectional from originating DEV to target DEV, the Data frame sent back by the target DEV shall be an empty Data frame.

In any frame sent by the originating DEV, any retransmitted subframes should be put in the original order. The MSDU Number field in MAC header shall be set to the value of the first subframe, no matter it is a retransmission or a newly aggregated subframe.

If the originating DEV wants to drop certain retransmissions, it advances the MSDU number in MAC header to inform the target DEV that any subframe with the MSDU number older than this will not be retransmitted anymore.

In case that the Blk-ACK is lost, the originating DEV may retransmit the same frame after RIFS. The subframes contained will be the same as those of the previous transmitted frame. The Retry field in the MAC header shall be set to one to indicate to the target DEV that the same frame is retransmitted. For any frame that contains different subframes from previous transmission, the Retry field in MAC header shall be set to zero.

If the subframe contains only MSB information, the LSB ACK in the Blk-ACK Bitmap field in the MAC header shall be set to zero and shall be ignored upon reception. Likewise, if the a subframe contains only LSB information, the LSB ACK in the Blk-ACK Bitmap field shall be set to zero and shall be ignored upon reception.

7.12.6.3 Blk-ACK for low-latency aggregation

For low-latency aggregation, the rules in this subclause apply.

When the source receives the information in the Blk-ACK Bitmap field, it inserts the incorrectly received MSDUs for retransmission in the next frame to be sent.

The originating DEV of a Blk-ACK shall maintain the Blk-ACK Bitmap field according the rules in this subclause.

The originator maintains a list of recently transmitted MSDU numbers for retransmissions.

The list of recently transmitted MSDUs shall be updated per transmission frame, and with each Blk-ACK received, to reflect any outstanding MSDU not acknowledged.

The originating DEV may abort retransmissions of a specific MSDU by advancing the value of the MSDU Request Number field, as defined in 6.2.11.3, beyond the specific MSDU sequence number. When the MSDU Request Number is advanced beyond a missing MSDU sequence number, the receiver updates its MSDU Response Number accordingly and assumes the skipped MSDU sequence number is discarded.

For each newly transmitted MSDU, its MSDU number is appended to the end of the list. The number of MSDUs in the list may grow up to the limit of the *mMaxSubframeNum*, as defined in 7.20.

If the MAC subheader is correctly received, the Blk-ACK Bitmap field is parsed. For each bit that is set, the entry of the MSDU number is removed from the list, updating the relative order of all remaining MSDUs (that were not acknowledged) and subsequently their offset order in the following retransmission attempt on the next transmission frame.

If the MAC subheader is not correctly received, the Blk-ACK Bitmap field is considered as corrupt. In this case, the source DEV may retransmit all the MSDUs from the previous frame.

If the subframe contains only MSB information, the LSB ACK in the Blk-ACK Bitmap field in the MAC header shall be set to zero and shall be ignored upon reception. Likewise, if the a subframe contains only LSB information, the LSB ACK in the Blk-ACK Bitmap field shall be set to zero and shall be ignored upon reception.

7.12.7 Stack acknowledgment (Stk-ACK) for pairnets

7.12.7.1 Ping-pong transmission and Stk-ACK (Synchronous Phase)

Stk-ACK is used for acknowledgment of data and command frames and channel access control as described in 7.6.2. It is also used for re-transmission control for the pairnet aggregation frame that is defined in 7.11. The destination, upon receiving an aggregated frame, checks each subframe from the beginning. Based on the status of the subframe, either correctly or incorrectly received, the last received sequence number is set in the TX and ACK Information field of the MAC header, and shall be sent in the next transmission. A sequence number is given to each subframe by each DEV, a cyclic and continuous value beginning from zero. The originating DEV, after reading the TX and ACK Information field in MAC header, handles subframe retransmissions.

Figure 7-65 illustrates the ping-pong transmission behavior of a pairnet aggregated frame. During the Synchronous Phase, both DEVs in the pairnet perform ping-pong transmissions with a SIFS between frames. If DEV receives subframes from N+1 to N+4 without any data error, DEV shall set N+4 in the TX and ACK Information field of the Stk-ACK of the next transmission. Stk-ACK may be sent as a piggyback on the next frame transmission. If the DEV does not have any data to send in its transmission phase, the DEV shall send a Stk-ACK with the last received sequence number without data to maintain the ping-pong Synchronous Phase. If DEV detects any errors in either the subheaders or the subframes, as illustrated in Time #3 of Figure 7-65, then DEV discards the subframe that was in error and all following subframes, and would set the sequence number of the last error-free subframe in the TX and ACK Information field of the MAC header of the next transmission frame. The ascending subframe order shall be maintained in the retransmission. The same retransmission behavior shall be applied if the destination DEV causes buffer overflow, as illustrated in Time #4 of Figure 7-65. However, the destination shall set the Buffer Full field to one and inform the source DEV of the buffer overflow. The source should read the Buffer Full field and use the appropriate transmission controls.

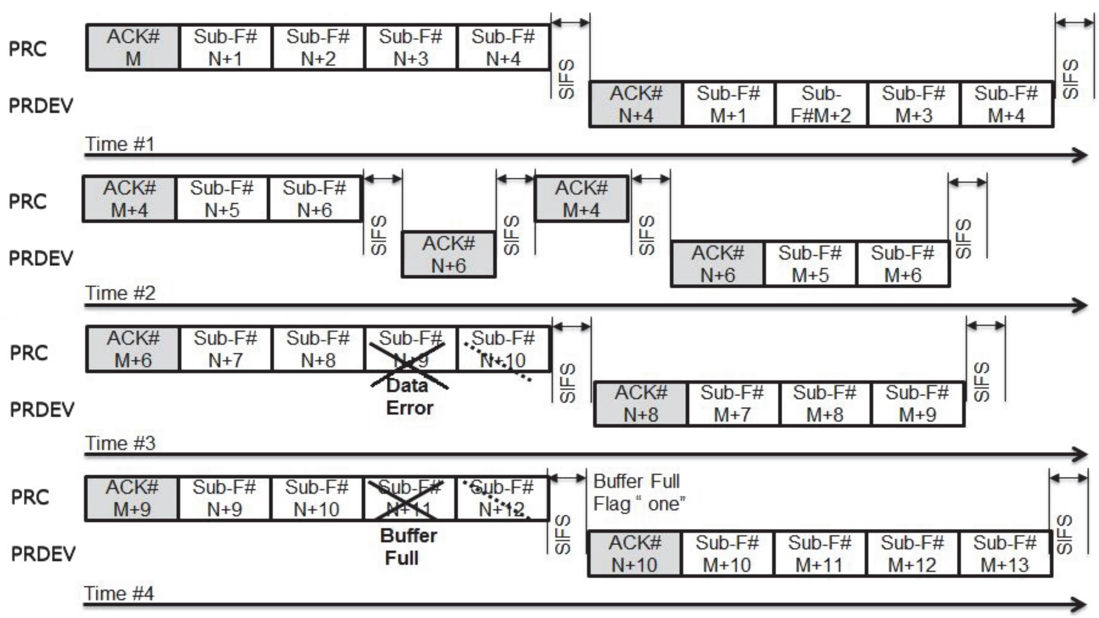

Figure 7-65—Ping-pong channel access with Stk-ACK (synchronized state)

7.12.7.2 Recovery Process (Asynchronous Phase)

If the destination detects a MAC header error, namely the destination lost the ACK frame, DEVs in the pairnet shall enter the Asynchronous Phase. When the PHY reports that there is an error at the destination, then the DEV may also enter the Asynchronous Phase. When the DEV enters Asynchronous Phase, the DEV shall start Recovery Process. Figure 7-66 illustrates the recovery process. As described in 7.6.5, each of the DEVs within the pairnet accesses the medium with RIFS and shall transmit the frame with no data payload. The Stk-ACK information shall be always set in the MAC header of the transmitted frames. The PRC and the associated DEV use different RIFS values. RIFS shall be longer than the time domain frame length with no data payload. When the DEV receives a frame with the correct MAC header within RIFS, the DEV determines that it entered the Synchronous Phase.

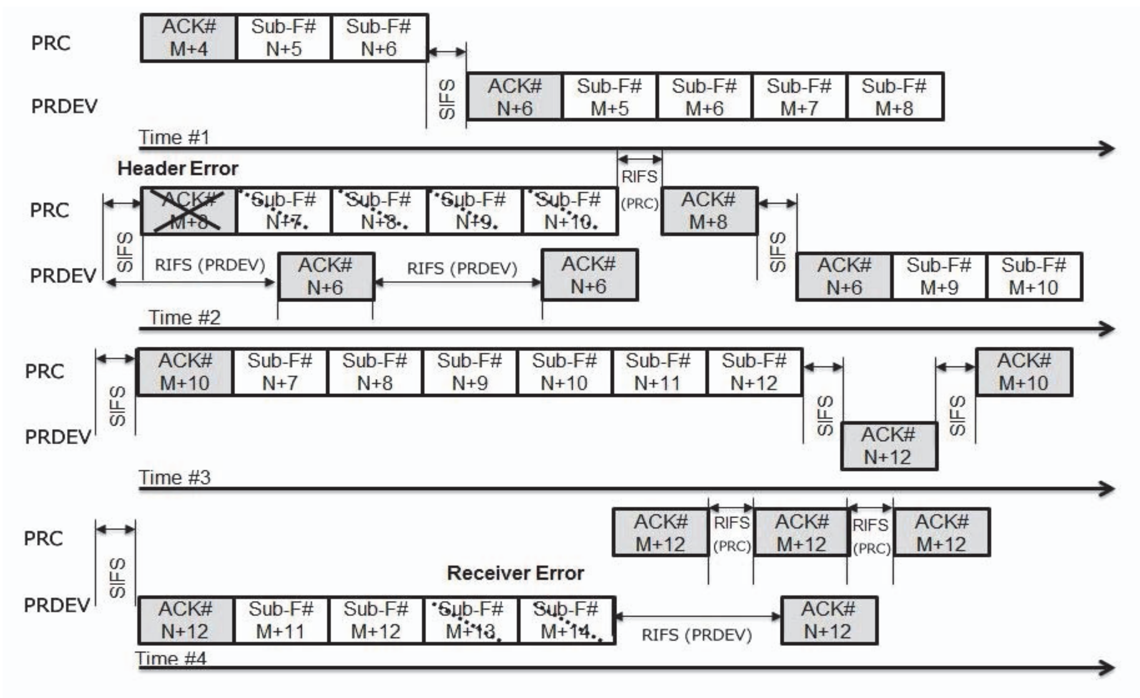

Figure 7-66—Synchronization recovery process

7.12.8 Retransmissions

During the CAP, retransmissions shall follow the backoff rules, as specified in 7.6.3.

During CTAs within the CTAP when an Imm-ACK or Dly-ACK is expected but is not received, the source DEV may start the retransmission of the frame (or the transmission of a new frame if the failed frame's retransmission limit has been met) a RIFS after the end of the last frame transmitted as long as there is enough channel time remaining in the CTA for the entire frame exchange.

A DEV determines the number of times a frame is retried before the DEV discards that frame. If the DEV gives up on a fragment of an MSDU/MCDU, the DEV shall discard all MPDUs of that MSDU/MCDU.

7.12.9 Duplicate detection

Because the DEV sending the data frame may not correctly receive an ACK, duplicate frames may be sent even though the intended recipient has already received and acknowledged the frame. Hence all DEVs shall detect such multiple receptions and indicate the data frames to the higher layers only once. The SrcID, Stream Index, Fragmentation Control field, Retry field, and PNID are used to detect multiple receptions of the same frame.

7.13 Piconet peer discovery

7.13.1 General

Each DEV that is a member of the piconet may use the PNC Information Request command, as described in 6.5.5.2, to obtain information about other DEVs in the piconet. In addition the DEV may use the Probe Request command, as described in 6.5.5.6, to obtain other information required for peer-to-peer communication. The remote scan procedure is used by the PNC to determine channel conditions. All DEVs in the piconet are able to use the Channel Status Request, as described in 6.5.8.2, and Channel Status Response, as described in 6.5.8.3, commands to gather information about the quality of their link with another DEV.

7.13.2 PNC information request

A DEV may request information about either a single DEV in the piconet or about all of the DEVs in the piconet by sending a PNC Information Request command, as described in 6.5.5.3, to the PNC. If the DEV is requesting information about only a single DEV in the piconet, it shall set the DEVID in the command to the ID of that DEV. If the DEV is requesting information about all of the DEVs in the piconet, it shall set the DEVID in the command to the BcstID, as described in 6.2.5. If the originating DEV requests information about a single DEV that is not a member of the piconet, then the PNC shall send a PNC Information command with length zero. Otherwise, the PNC shall send the PNC Information command with the information requested by the originating DEV.

Figure 7-67 illustrates the sequence of messages involved in acquiring PNC information regarding a specific DEV or all of the DEVs that are members of the piconet.

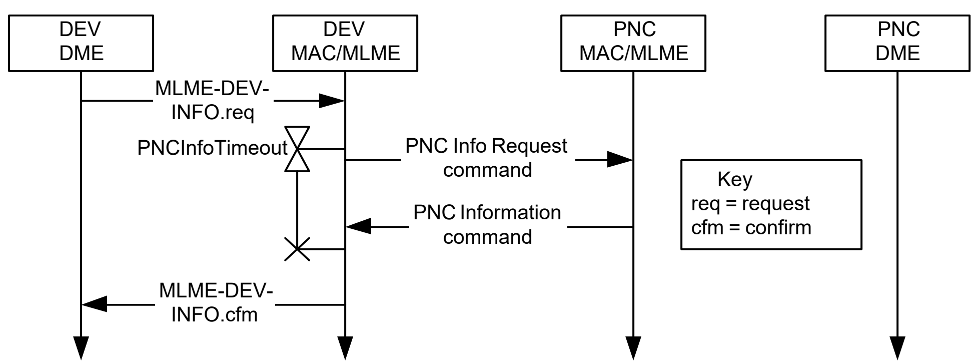

Figure 7-67—MSC for acquiring information regarding a specific DEV or all of the DEVs from the PNC using the PNC Information Request and PNC Information commands

7.13.3 Probe request and response

The Probe Request command provides the ability to request IEs from a target DEV. If the target DEV receives the Probe Request command, it shall respond to the originator with Probe Response command(s) that shall have the IEs requested by the originator.

A DEV may request information about an isochronous stream by sending a Probe Request command requesting the CTA Status IE, as described in 6.4.11, with the Request Index field set to the stream index of the stream for which CTA information is requested. If the Request Index field is set to zero, the DEV is requesting information about all isochronous streams directed to the requesting DEV and to the BcstID and McstID. The PNC shall respond to a Probe Request command containing a request for the CTA Status IE by sending Probe Response command(s) containing the appropriate CTA Status IE(s).

If a DEV requests the CTA Status IE for a stream index that is not currently allocated by the PNC, the PNC shall respond with a Probe Response command containing a CTA Status IE with the following information:

— The SrcID and DestID fields both set to the UnassocID.
— The Stream Index field set to the requested stream index.
— All other fields set to zero.

Any DEV may send the Probe Request command with the Information Requested field set to zero and ACK Policy field set to Imm-ACK to any other DEV in the piconet to determine if the destination DEV is still present in the piconet and is within range of the sending DEV.

A DEV that is going to send a Probe Request command to a DEV operating in a power save mode should consider the operation of those modes, as described in 7.17, to determine the appropriate time to send the Probe Request command and the time to expect a response.

The types of IEs that are allowed to be requested or to be sent in the response depend on the status of the originator as either a DEV or PNC. The rules for requesting a specific IE are listed in Table 6-26 while the rules for responding to the request for a specific IE are listed in Table 6-27.

7.13.4 Announce

The Announce command provides the ability to send unrequested IEs to a target DEV. This command shall have one or more IEs that the originator is sending to the target.

A DEV that is going to send an Announce command to a DEV operating in a power save mode should consider the operation of those modes, as described in 7.17, to determine the appropriate time to send the Announce command.

The types of IEs that are allowed to be sent depend on the status of the originator as either a DEV or PNC. The rules for sending a specific IE are listed in Table 6-28.

7.13.5 Channel status request

The Channel Status Request command, as described in 6.5.8.2, may be used by any DEV in the piconet to get information from a target DEV about the link quality between the two DEVs. The originating DEV sends the Channel Status Request command to the target DEV to start the process. When the target DEV receives the request, it shall send a Channel Status Response command to the originating DEV with all of the information specified in 6.5.8.3.

The information conveyed in the channel status process is based on the results of an attempted data transfer between the two DEVs. Thus, the Channel Status Request command should only be used for DEVs that are actively participating in a data transfer as the information would not have much meaning otherwise. The PNC also uses this command to get the channel status information from the DEVs in the piconet, as described in 7.15.2, 6.5.8.2, and 6.5.8.3.

7.13.6 Remote scan

Remote scan is a procedure by which a PNC may request that a target DEV in the piconet initiate a channel scan on the PNC's behalf of the specified channels and to report the results of the scanned channels back to the PNC. The PNC may then use the results of the remote scan to initiate a change in maximum transmit power for the piconet, initiate a channel change to a channel with better channel characteristics, or other action. The algorithm for determining which procedure to execute is outside of the scope of this standard.

The PNC may optionally allocate channel time in the CTAP so that there is quiet time for the remote DEV to scan the channel for other piconets.

One of the reasons that the PNC requests a remote scan is because it determines that the current channel is unsatisfactory for continued operation of the piconet. This allows the PNC to get information about other wireless networks that may be out of range of the PNC but in range of some of the DEVs in the piconet. The PNC is able to get this information while it continues allocating channel time and generating beacons for its piconet.

The PNC initiates a remote scan by sending the Remote Scan Request command with a list of channels to a DEV in the piconet. The target DEV should accept the request to perform a channel scan on behalf of the PNC. If the DEV does not accept the request, it shall respond to the PNC by sending a Remote Scan Response command with a ReasonCode set to “request denied.” The PNC upon receiving this response may send a Remote Scan Request command to another piconet DEV, initiate its own channel scan, or take other action.

In the case where the target DEV does accept the request, the target DEV shall initiate a series of OpenScan channel scans, as described in 7.2.2, based upon the channel list passed to it in the Remote Scan Request command. The target DEV shall scan each of the channels requested by the PNC. When the target DEV has finished scanning the channels, it shall respond via the Remote Scan Response command to the PNC with the Remote Piconet Description Set parameters and the Channel Rating List field. The PNC upon receiving this information may then determine that there is a new channel with better characteristics than its current channel. The PNC may use this information to initiate the change channel procedure. The PNC may decide instead that a more appropriate solution to the current channel impairment is to increase/decrease transmission power level of the piconet. The PNC may also decide to do nothing or to take other unspecified action.

A DEV that is receiving beacons from more than one PNC may send an unsolicited Remote Scan Response command to its PNC with a Piconet Description Set representing the interfering PNCs. A DEV shall not report overlapping piconets if it determines that the beacons were received from a child or IEEE 802.15.3 neighbor piconet that is associated with the DEV’s current piconet.

7.13.7 PNC channel scanning

PNC channel scanning is a procedure by which the PNC is able to determine the channel characteristics of not only its current operating channel but also the channel characteristics of other channels. The PNC may use the results of its channel scans to determine whether the current channel in which it is operating has acceptable characteristics or that there is another channel(s) with better channel characteristics than its current channel.

The PNC may allocate CTAs such that there is unallocated channel time in the CTAP. This provides quiet time for the PNC to scan channels for other IEEE 802.15.3 piconets, non-IEEE 802.15.3 wireless networks, or interference.

If the PNC initiates a scan of one or more alternate channels, the PNC shall not transmit a beacon for one or more beacon intervals. The PNC shall not suspend beacon transmissions for more than twice *mMinChannelScan*. The PNC, upon returning to its current channel and resuming the transmission of its beacons, shall increment the Time Token field by the number of beacons not sent during the time the PNC was scanning one or more alternate channels.

After scanning both its current channel and other channels, the PNC may initiate one of these options:

a) Do nothing since the PNC has determined that none of the alternate channels were better than its current channel.
b) Initiate the dynamic channel change procedure described in 7.15.2.
c) Increase or decrease the max tx power level of the piconet, as described in 7.15.3.
d) Initiate some other unspecified action.

The algorithm for determining when to initiate any of these actions is outside the scope of this standard.

7.13.8 Channel probing

Channel probing is an optional function to allow the source DEV to choose the best MCS for the current channel. A DEV may use the Channel Status Request, as described in 6.5.8.2, and Channel Status Response, as described in 6.5.8.3, commands to gather information of the link quality with another DEV. In addition, when the DEVs are capable of beam forming, the quality of the link may be obtained by beam-forming procedure, as described in 15.5. During data streaming, the destination DEV reports the channel status information, as defined in 12.1.9.3, to the source DEV. After learning the channel status, the source DEV may switch to an MCS that is best adapted to the channel condition. For the SC/HSI PHY, the channel status information may be written in the fragmentation field of the MAC header of a Imm-ACK or Dly-ACK, as defined in 6.3.5. For AV PHY, it may be written in the fragmentation field of the MAC header of an Imm-ACK or Dly-ACK frame.

7.14 Changing piconet parameters

7.14.1 General

A PNC shall not change either the pseudo-static CTAs or the pseudo-static CTA blocks during a piconet parameter change. If the parent needs to move a pseudo-static CTA because the superframe duration is being reduced, it shall do so prior to using the superframe duration change process, as described in 7.14.3. A child or IEEE 802.15.3 neighbor PNC shall use the value of the Change Beacon Number field in the Piconet Parameter Change IE from the parent’s beacon to calculate the Change Beacon Number field in the Piconet Parameter Change IE in its own beacon. The exceptions to this are as follows:

— When the parent is changing its PNID or BSID.
— When a child or neighbor PNC decides not to change channels with the parent PNC and is shutting down, as described in 7.15.2.

When the Change Beacon Number field is used with changing parameters, the PNC shall ensure that the Piconet Parameter Change IE announcement complies with the rules for beacon announcements in 7.8.5. The PNC shall not place more than one Piconet Parameter Change IE, as described in 6.4.7, in any beacon.

The PNC shall not be required to wait until all of the DEVs in power save modes are in ACTIVE mode before changing a piconet parameter. A power save DEV is not required to switch to ACTIVE mode when there is a piconet parameter change in progress. However the power save DEV shall update its internal values for the new piconet parameter at the time indicated in the Piconet Parameter Change IE.

7.14.2 Moving beacon

The PNC may move the relative position of its beacon. Moving a beacon means that the superframe duration is unchanged while the position of the beacon is moved.

If the PNC wishes to move its beacon position, it shall insert the Piconet Parameter Change IE, as described in 6.4.7, into NbrOfChangeBeacons beacons with the change type set to MOVE and the superframe timing set to the delay of the first beacon after this sequence compared to previous beacon transmission time. See Figure 7-68.

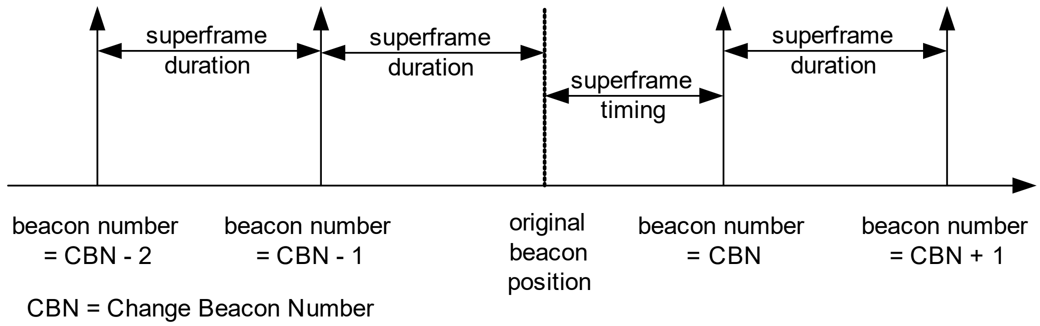

Figure 7-68—Moving the beacon position

7.14.3 Changing superframe duration

The PNC may change the duration of its superframe. If the PNC wishes to change its superframe duration, it shall insert the Piconet Parameter Change IE, as described in 6.4.7, into NbrOfChangeBeacons beacons with the Change Type set to SIZE and the Superframe Timing set to the size of the superframe that follows the first beacon after this sequence. See Figure 7-69.

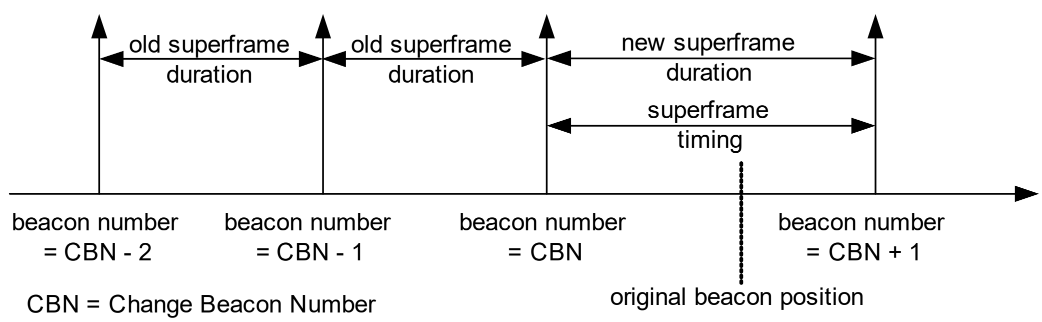

Figure 7-69—Changing superframe duration

7.14.4 Setting the PNID or BSID

The BSID is used to provide a way to identify the piconet. The PNC is able to change the BSID via the Piconet Parameter Change IE in the beacon, as described in 6.4.7, using the process described in this subclause. The BSID is preserved in the PNC handover process, and it may be persistent when the PNC restarts a piconet that ended without handing over control to a PNC-capable DEV.

The PNC shall choose a PNID when it starts a piconet; the PNID should be selected randomly. An existing piconet's PNID shall be changed only if the PNC detects another piconet with the same PNID on any channel. The same PNID may be persistent when the PNC restarts a piconet that ended without handing over control to a PNC-capable DEV.

If the PNC detects that another piconet is using the same BSID in its operational area, it may change the BSID, but it is not required to change it. If the PNC detects that another piconet is using the same PNID within range of the PNC on any channel or if it is informed of this via the Overlapping PNID IE, as described in 6.4.22, the PNC shall choose another PNID and change it via the Piconet Parameter Change IE in the beacon. The PNC shall not simultaneously change both the PNID and the BSID.

If a DEV detects a piconet within its range on any channel with the same PNID, it shall send an Announce command, as described in 6.5.6.2, to the PNC including an Overlapping PNID IE, as described in 6.4.22, that contains the current PNID and channel index. Once this command has been sent successfully, the DEV shall not send this information again until after the current PNID has been changed by the PNC.

Before changing its PNID, a parent PNC shall scan for the PNIDs of other piconets, including all of its dependent piconets. The PNC shall not change its PNID to the same value as that of any other piconet that it detects.

If the PNC decides to change the PNID or BSID, the PNC shall send a beacon with the Piconet Parameter Change IE indicating the new PNID or BSID. The DEVs that received the beacon with Piconet Parameter Change IE shall change the PNID or BSID to the new value at the time of the beacon with a beacon number equal to the Change Beacon Number field in the previous Piconet Parameter change IEs.

The MSC in Figure 7-70 describes the BSID change process.

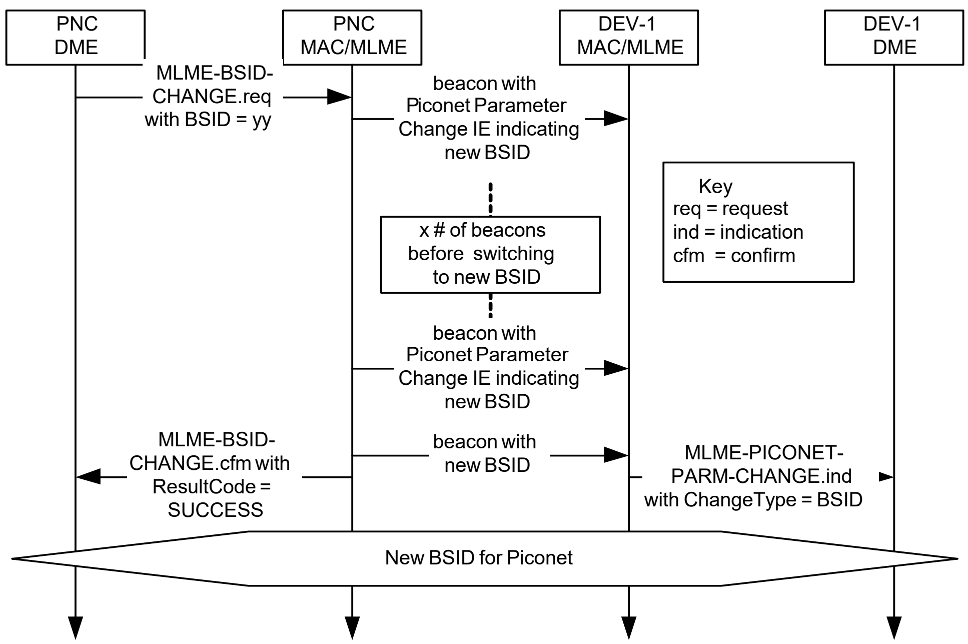

Figure 7-70—MSC for changing the BSID for a piconet

7.14.5 Maintaining synchronization in dependent piconets

A dependent PNC receiving a parent beacon with a Piconet Parameter Change IE that has the Change Type field set to MOVE or SIZE shall immediately insert the appropriate Piconet Parameter Change IE into its beacons. If the Change Type field is set to SIZE, a dependent PNC shall change the length of the private CTA used to reserve time for the operations of the parent piconet in the first beacon it broadcasts in the parent superframe of the new superframe size.

7.15 Piconet interference mitigation

7.15.1 General

The PNC should periodically listen in the current channel to detect interference, the presence of other IEEE 802.15.3 piconets, or the presence of other wireless networks. The PNC should also periodically listen to other overlapping channels for the presence of the same types of DEVs.

If the PNC detects the presence of another IEEE 802.15.3 piconet, four of the methods that are available to mitigate the interference between the two piconets are as follows:

— The PNC may join the other piconet to form a child piconet, as described in 7.2.8.
— The PNC may join the other piconet to form a neighbor network, as described in 7.2.9.
— The PNC may change channels to one that is unoccupied, as described in 7.15.2.
— The PNC may reduce the maximum transmit power in the piconet, as described in 7.15.3.

If the PNC determines that there is either an interferer or a non-IEEE 802.15.3 wireless network operating in the PNC's current channel or overlapping with the current channel, two of the methods that are available to mitigate the interference are as follows:

— The PNC may change channels to one that is unoccupied, as described in 7.15.2.
— The PNC may reduce the maximum transmit power in the piconet, as described in 7.15.3.

7.15.2 Dynamic channel selection

The PNC initiates dynamic channel selection if it determines that the current conditions of its channel are poor. The PNC may use one or more of the following methods to make this determination:

— The PNC may perform a PNC channel scanning procedure, defined in 7.13.7.
— The PNC may use the remote scan procedure, defined in 7.13.6.
— The PNC may collect the channel status from its member DEVs by sending a Channel Status Request command, defined in 6.5.8.2, to request that the DEVs provide their channel status via a Channel Status Response command, as described in 6.5.8.3.

The algorithm required to use the channel status information when deciding whether to change channels is outside the scope of this standard.

The PNC shall initiate a dynamic channel change procedure only after it has performed a PNC channel scan, as described in 7.13.7, and has determined that there is one or more channels with better characteristics than exist in its current operating channel. Note that in addition to the PNC channel scan, the PNC is able to use other methods, described previously, to determine which channel to use as the new channel. If the PNC decides to initiate a dynamic channel change, the PNC shall broadcast the Piconet Parameter Change IE, as described in 6.4.7, with the change type set to CHANNEL in its current channel via its beacon for *NbrOfChangeBeacons* consecutive beacons. The Piconet Parameter Change IE shall contain the channel index of the new channel to which the PNC will be moving the piconet, and the Change Beacon Number field that contains the beacon number of the first beacon that will be sent on the new channel. The channel change shall take effect starting with the first beacon with a beacon number equal to the Change Beacon Number field in the previous Piconet Parameter Change IEs. The DEVs that received a beacon containing the Piconet Parameter Change IE shall change from their current channel to the new channel before the first expected beacon on the new channel.

No DEV shall transmit on the new channel until a beacon has been correctly received, as described in 7.1, on the new channel.

Dependent piconets shall either change to the new channel with the parent PNC or cease operations. If a dependent PNC ceases operation due to a channel change, it may attempt to restart its piconet in the original channel, as described in 7.2.3.

The PNC is not required to wait until all of the DEVs in power save modes are in ACTIVE mode before changing channels. A power save DEV is not required to switch to ACTIVE mode when there is a channel change in progress. However, the power save DEV should change channels at the time indicated in the Piconet Parameter Change IE.

In the last beacon transmitted prior to a channel change, the PNC should provide time at the end of the superframe for the DEVs in the piconet to change channels. The time required for a DEV to change channels is *pChannelSwitchTime*.

NOTE—One method to allow time for the channel change is for the PNC not to place CTAs just prior to the first beacon that will be sent on the new channel.

The PNC shall ensure that the Piconet Parameter Change IE announcement complies with the rules for beacon announcements in 7.8.5.

Figure 7-71 illustrates the message sequence involved in transitioning the current piconet to a new channel.

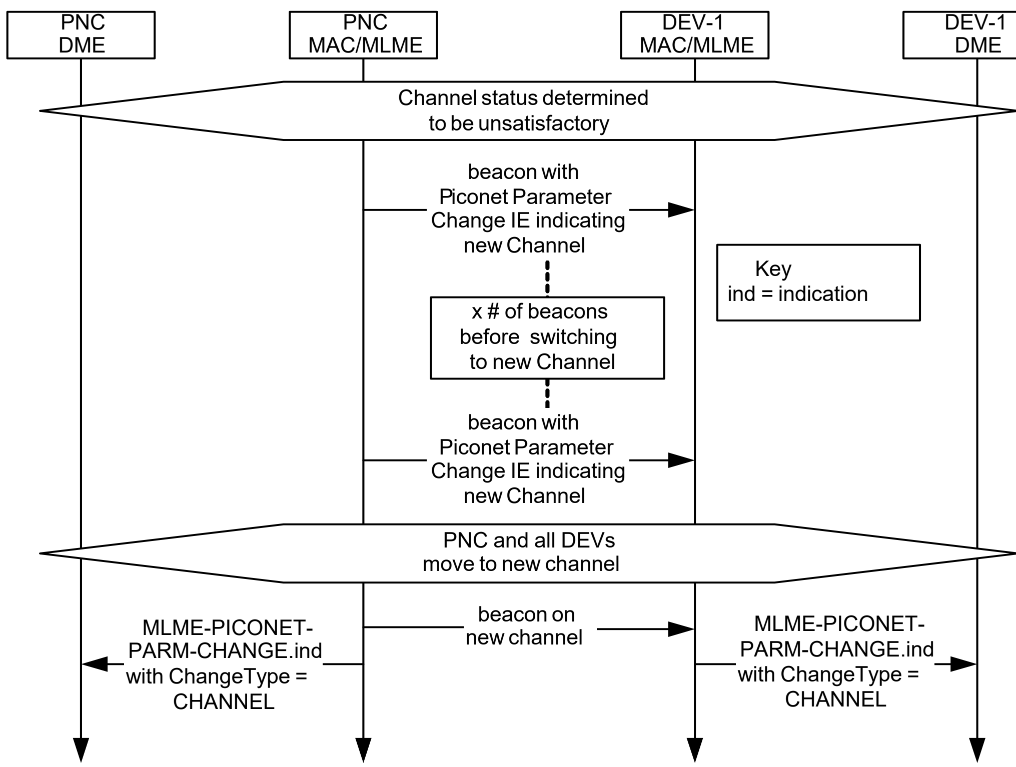

Figure 7-71—MSC for changing the piconet's channel

When a parent PNC changes channels, the dependent PNC may change channels as well. If the dependent PNC is going to change channels with the parent PNC, it puts the appropriate Piconet Parameter Change IE into its beacons when it receives a beacon from the parent indicating a pending channel change. The dependent PNC shall then switch channels to the channel indicated in the Piconet Parameter Change IE at the appropriate time.

A dependent PNC may change to a different channel even if the parent PNC does not change channels. In this case, the child or IEEE 802.15.3 PNC shall remove the Parent Piconet IE from its beacon after the channel change since it is no longer a child or neighbor of that piconet. The dependent PNC should also disassociate from the parent piconet when it changes channels without the parent PNC.

7.15.3 Transmit power control (TPC)

7.15.3.1 General

Two independent forms of TPC are available for IEEE 802.15.3 systems: a maximum power for the CAP, the beacon, and directed MCTAs, and an adjustable power in a CTA. The goal of TPC in the CAP is to prevent one DEV from having better access to the medium in the CAP due to a higher transmit power level. Adjustable transmitter power in the CTA is intended to support reduced power usage as well as reducing the overall interference levels generated by the piconet. The transmit power used in a CTA is not limited by the transmit power used for the CAP, beacon, or directed MCTAs.

7.15.3.2 Maximum transmitter power for the CAP, beacon, and directed MCTAs

The PNC may choose a maximum transmit power level for the CAP, beacon, and directed MCTAs, excluding association MCTAs. The PNC shall convey this information to the DEVs via the Beacon frames using the Max TX Power Level field in the beacon. The PNC shall not set the Max TX Power Level below the *pMinTpcLevel* for the PHY, which is defined in 11.5.9 for the 2.4 GHz PHY. All DEVs within the piconet shall set their nominal transmit power level for frames in the CAP or directed MCTAs, excluding association MCTAs, to be no more than the value indicated in the Max TX Power Level field in the beacon, as described in 6.3.1. DEVs shall comply with the maximum transmit power within 10×*mMaxLostBeacons* superframes following the beacon in which the DEV detects the change. Likewise, the PNC shall set its nominal transmit power for the beacon to be no more than the value that is indicated in the beacon.

7.15.3.3 Adjustable transmitter power in a CTA

Each DEV participating in a CTA may request that the other DEV with which it is communicating in the CTA either increase or decrease its transmitter power level. The originating DEV shall use the Transmit Power Change command, as described in 6.5.8.6, to request a change in the power level setting of the target DEV for all CTAs assigned between the two DEVs. The target DEV shall increase or decrease its transmit power level as indicated in the Transmit Power Change command if the power level setting is supported by the target DEV. If the power level change is not supported by the target DEV, it shall use the closest implemented TX power level that is greater than the requested level provided that is within the allowable range. The target DEV shall apply the change in the power level for all CTAs assigned between the two DEVs.

A DEV may also change its transmit power based on its own estimation of the channel.

7.16 Multi-rate support

A compliant PHY may support more than one data rate. In each PHY there will be one mandatory base rate specified for the purposes described in this subclause. In addition to the base rate, the PHY may support rates that are both faster and slower than the base rate. A DEV shall send a frame with a particular data rate to a destination DEV only when the destination DEV is known to support that rate. The allowed data rates and the mandatory base rate are PHY dependent and defined as follows:

- In 11.3 for the 2.4 GHz PHY
- In 12.2.3.1 for the SC PHY mode
- In 12.3.3.1 for the HSI PHY mode

— In 12.4 for the AV PHY mode
— In 13.2 for the HRCP-SC PHY mode
— In 13.3 for the HRCP-OOK PHY mode
— In 14.2 for the THz-SC PHY mode
— In 14.3 for the THz-OOK PHY mode

In order to determine the rates that are supported by a target DEV in the piconet or pairnet, the DEV shall use one of the following four methods:

a) Check the capabilities of the target DEV broadcast by the PNC when the DEV becomes a member of the piconet.
b) Send a Probe Request command, as described in 6.5.5.6, to the target DEV to request its Capability IE, as described in 6.4.12.
c) Request the information from the PNC using the PNC Information Request command, as described in 6.5.5.2.
d) Check the capabilities of the target DEV exchanged during the association process of the pairnet, as described in 5.3.6.

Similarly, each DEV may periodically use the Channel Status Request command, as described in 6.5.8.2, to obtain the channel status information from other DEVs and decide the PHY rate to be used in transmissions to those other DEVs. Additionally, the channel quality may be evaluated by the presence or absence of ACKs to transmitted frames. This information may be used to determine if the rate of transmission or the power level needs to change.

If the Dly-ACK mechanism is used, all frames in a burst shall be sent with the same rate. The Dly-ACK frame shall be sent with the same rate as the rate of the last frame in the burst that requested the Dly-ACK.

The allowed PHY rates for each of the different types of frames are listed in Table 7-5.

Table 7-5—Allowed PHY data rates for frames

Frame type	Allowed PHY data rates
All broadcast and multicast addressed frames (including the beacon and commands)	Base rate
Imm-ACK	Same rate as the frame that is being ACKed
Dly-ACK	Same rate as the last frame of the burst being ACKed
Association request	Base rate
Association response	Base rate
Disassociation request	Base rate
Directed command frame	Any rate supported by both the source and destination.
Directed data frame	Any rate supported by both the source and destination.

HRP frames in the AV PHY with the ACK policy set to Imm-ACK shall be ACKed with the directional ACK, as described in 6.2.12.6. The data rate for the payload in a directional ACK with payload is selected by the DEV sending the directional ACK. Blk-ACK policy shall not be used with HRP AV PHY frames.

7.17 Power management for piconet

7.17.1 General

There are four power management (PM) modes defined for piconets: ACTIVE, APS, PSPS, and DSPS. The latter three modes are collectively referred to as power save (PS) modes. A DEV that is in ACTIVE, APS, PSPS, or DSPS mode is said to be an ACTIVE DEV, an APS DEV, a PSPS DEV, or a DSPS DEV, respectively. In any given PM mode, a DEV may be in one of two power states, either AWAKE or SLEEP states. AWAKE state is defined as the state of the DEV where it is either transmitting or receiving. SLEEP state is defined as the state in which the DEV is neither transmitting nor receiving. A DEV, regardless of its PM mode, is allowed to enter the SLEEP state during a CTA for which it is neither the source nor the destination. A DEV is also allowed to enter the AWAKE state during any time when it is in a power save mode.

The wake beacon for a DEV is defined as the PNC-defined system wake beacon for DEVs in PSPS mode, as described in 7.17.2, and the wake beacon of the DSPS set for a DEV in DSPS mode, as described in 7.17.3. A DEV in DSPS mode may be in multiple DSPS sets and therefore may have multiple wake beacons because each of those DSPS sets may have its own wake beacon. The wake beacon for a DEV in APS mode occurs at times determined by the DEV and is unknown to the PNC and other DEVs in the piconet. Unlike the DSPS and PSPS wake beacons, the wake beacon of the DEV in APS mode is not periodic and is only guaranteed to happen once per ATP period for that DEV.

A DEV shall always establish membership with the piconet in ACTIVE mode.

Table 7-6 lists the rules for the four modes of operation defined in this standard. Each entry indicates the state required, either AWAKE or SLEEP, for the DEV.

Table 7-6—Power management rules for superframe elements

Superframe portion	ACTIVE	APS in wake superframe	PSPS or DSPS DEV in wake superframe
Beacon	AWAKE	AWAKE	AWAKE
CAP	AWAKE	May SLEEP	AWAKE
CTA with BcstID as DestID (including MCTAs)	AWAKE	May SLEEP	AWAKE
CTA with McstID as DestID (including MCTAs)	May SLEEP	May SLEEP	May SLEEP
CTA with DEV as SrcID or DestID (including MCTAs)	AWAKE	May SLEEP	AWAKE
All other CTAs and unallocated time (between CTAs)	May SLEEP	May SLEEP	May SLEEP

The PNC shall support one PS set for APS and one PS set for PSPS. In addition the PNC shall support at least one DSPS set, i.e., a PS set with PS Set Index between 2 and 253.

7.17.2 Piconet synchronized power save (PSPS) mode

A DEV in PSPS mode shall listen to all system wake beacons, as announced by PNC, and is required to be in the AWAKE state during system wake superframes, as indicated in Table 7-6. The wake beacon for PSPS

DEVs is determined by the PNC. PSPS mode is identified by a PS Set Index equal to one. If a DEV in PSPS mode does not correctly receive the system wake beacon, it shall be in the AWAKE state during the expected beacon transmission times to receive the following beacons until a beacon is correctly received, as described in 7.1.

The PNC shall announce the system wake beacon in the Next Wake Beacon field in the PS Status IE with PS Set Index field equal to one in the beacon. If none of the DEVs are in PSPS mode, the PNC may omit the PS Status IE for PS Set Index one from the beacon. In that case every beacon is a system wake beacon for the purpose of beacon information announcements, as described in 7.8.5.

It is the responsibility of the DEV using PSPS mode to synchronize with the system wake beacon before entering the sleep state. Because the PSPS DEV at some point will need to send commands to the PNC, e.g., the PM Mode Change command, the PNC needs to take this into consideration when allocating MCTAs if the CAP is not available for sending commands.

Any DEV that is going to use the PSPS mode shall send the SPS Configuration Request command, as described in 6.5.9.4, to the PNC with the Operation Type field set to "join," SPS Set Index set to one, and the Wake Beacon Interval set to its desired system wake beacon interval. The valid range for a requested Wake Beacon Interval is defined in 6.5.9.4. Upon reception of this command, the PNC shall ACK the command and respond with an SPS Configuration Response command with the next wake beacon. The PNC uses the information in the Wake Beacon Interval field from all participating PSPS DEVs to determine the system wake beacon interval. The actual system wake beacon interval may not correspond to any of the PSPS DEVs requested wake beacon interval.

A DEV may send the SPS Configuration Request command more than once if its system wake beacon interval requirement changes. If the DEV no longer wishes to use PSPS mode, it shall send the SPS Configuration Request command with the Operation Type field set to 'leave' and the PS Set Index set to one to leave the PSPS set.

A DEV shall send a PM Mode Change command to the PNC with the PM Mode field set to SPS and receive the ACK from the PNC before entering the PSPS mode. When the PNC receives this command, it shall terminate all super-rate streams for which the DEV is the destination, as described in 7.7.2.4, and set the DEVID Bitmap field in the PS Status IE appropriately, as described in 6.4.20.

The PS Status IE in the beacon for PS Set Index value of one with the bit for the DEV's DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched into PSPS mode.

When the DEV is going to switch to ACTIVE mode, it shall send a PM Mode Change command to the PNC with the PM Mode field set to ACTIVE. Once this command is sent, the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC. The PNC is not required to align sub-rate allocations for a PSPS DEV with the system wake beacon.

7.17.3 Device synchronized power save (DSPS) mode

7.17.3.1 General

DSPS mode allows a DEV that is sensitive to power utilization to synchronize its AWAKE state with other DEVs. The DSPS mode is based on grouping DEVs that have similar power save requirements into DSPS sets. These DSPS sets are managed by the PNC, but the parameters of the sets are determined by the DEVs.

7.17.3.2 Creation, use, and management of DSPS sets

In order to use DSPS mode, a DEV is required to first join a DSPS set. Each DSPS set has two associated parameters: the Wake Beacon Interval and the Next Wake Beacon. The DSPS set is identified by an index value called the DSPS Set Index that is between 2 and 253, inclusive. The Wake Beacon Interval is the number of superframes between two successive wake beacons of that DSPS set. This value is set by the DEV when it creates the DSPS set. The Next Wake Beacon parameter is the beacon number, as described in 6.3.1.1, corresponding to the immediate next wake beacon of that DSPS set. This parameter is set by PNC when it creates the DSPS set. Both of these parameters shall be maintained by the PNC.

Any DEV that is a member of the piconet may request the information about the existing DSPS sets by sending a PS Set Information Request command, as described in 6.5.9.2, to the PNC. The PNC shall respond by sending a PS Set Information Response command, as described in 6.5.9.3, that provides the parameters of all of the PS sets currently in use within the piconet.

The DEV may select a DSPS set to join. If there are not any DSPS sets currently in existence that match the DEV's requirements, the DEV may request the formation of a new DSPS set by setting the SPS Set Index field in the SPS Configuration Request command, as described in 6.5.9.4, to the "Unallocated DSPS set" value and the Operation Type field set to "join." The DEV shall set the Wake Beacon Interval field to its requested value. The valid range for the Wake Beacons Interval field is defined in 6.5.9.4. This value shall not be changed while the DSPS set has any members. The PNC shall respond to the request by sending the SPS Configuration Response command, as described in 6.5.9.5, to the DEV indicating success or the reason that the request failed. If the DSPS set is created, the PNC assigns a DSPS Set Index to it. The PNC shall assign a unique number between 2 and 253 for the DSPS set index. The PNC includes in the SPS Configuration Response command a value for the Next Wake Beacon field set to the beacon number, as described in 6.3.1.1, of the first wake beacon for the new DSPS set. Once a DSPS set is created, the PNC shall keep the Next Wake Beacon for that set updated at all times. The maximum number of DSPS sets supported by the PNC is implementation dependent up to a maximum of 252.

The PNC may require that all PS sets have a unique Wake Beacon Interval. For example, the PNC may reject a request to create a PS set with a Wake Beacon Interval of 4 if there is a PS set that already has this value. If the DEV requires this Wake Beacon Interval, it may join the existing PS set.

A DEV may join an existing DSPS set by sending the SPS Configuration Request command to the PNC with the SPS Set Index field set to the index of an existing DSPS set and the Operation Type field set to "join." All other parameters shall be ignored. The PNC shall confirm or reject the request by sending the SPS Configuration Response command to the DEV. Since a DEV may support multiple applications with different requirements, a DEV may register in more than one DSPS set at a given time.

A DEV that no longer needs to be in a DSPS set shall send the SPS Configuration Request command to the PNC with Operation Type field set to "leave." The PNC shall not send the SPS Configuration Response command to the requesting DEV if the Operation Type field was set to "leave."

When the last member of a DSPS set has left, the DSPS set shall be terminated by the PNC. The PNC may also delete a DSPS set. The PNC deletes a DSPS set by placing the PCTM IE, 6.4.9, in the beacon and setting the bit for each of the DEVs that are members of that DSPS set and are in a PS mode. The PNC also removes from the beacon, if present, the PS Status IE of the DSPS set that the PNC is deleting. When the PNC determines that a DEV in that DSPS set is in AWAKE state, it shall send an SPS Configuration Response command to the DEV with SPS Set Index field set to the value of the DSPS set index that is being deleted and the Reason Code field set to "DSPS set deleted by PNC." If a DEV is in DSPS mode and it receives a beacon that does not have the PS Status IE for its DSPS set, the DEV should send a PS Set Configuration Request command to the PNC to determine if the DSPS set has been deleted.

7.17.3.3 Changing DSPS mode and operation

DSPS DEVs alternate between DSPS mode and ACTIVE mode depending on the amount and type of data traffic without leaving any of the DSPS sets that the DEV has joined. The PM Mode Change command, as described in 6.5.9.6, is used by a DEV to inform the PNC that it is changing its power management mode. A DEV shall have joined one or more DSPS sets before it will be allowed to switch into DSPS mode.

If the DEV is going to change its power management mode from ACTIVE to DSPS, the DEV shall send the PM Mode Change command, as described in 6.5.9.6, to the PNC with the PM Mode field set to SPS. The PNC shall then set the bit in the DEVID Bitmap field for the DEV in each PS Status IE that corresponds to an SPS set of which the DEV is a member. If the DEV is the source or destination of any streams, not including broadcast or multicast, which are not using a DSPS wake beacon interval, the PNC shall terminate those streams, as described in 7.7.2.4, when the DEV changes to DSPS mode. The PNC does not automatically terminate any streams when the DEV changes from DSPS to ACTIVE mode.

If the DEV has joined PS set one (PSPS) in addition to other DSPS sets before issuing the PM Mode Change command, the DEV shall transition into a combined PSPS and DSPS mode. In either case, the DEV shall not consider itself in either DSPS or PSPS mode until it has received an Imm-ACK to the PM Mode Change command.

The presence of a PS Status IE in the beacon with the bit for the DEV's DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched into PSPS mode.

If the DEV is going to change its power save mode from DSPS to ACTIVE, the DEV shall send the PM Mode Change command, as described in 6.5.9.6, to the PNC with the PM Mode field set to ACTIVE. Once this command is sent the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap field in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC. When the PNC correctly receives the PM Mode Change command with the PM Mode field set to ACTIVE, it shall no longer set the bit(s) for the DEV in the PS Status IE(s). If the PM Mode field indicates the current mode of the DEV, then no change is requested.

The PNC shall create one PS Status IE in the beacon for each DSPS set that has at least one member currently in DSPS mode. When a DSPS set is not in use, the PNC shall discontinue inserting its PS Status IE in the beacon. The PNC needs to ensure that the number and size of the PS Status IEs do not cause the beacon or extended beacon to exceed its maximum allowed size, as described in 6.2.

Other DEVs may use the information in the PS Status IE to learn when to transmit to a DSPS DEV. In addition, the PS Status IE informs DSPS capable DEVs which DSPS set is required in order to synchronize data transfers to the DEVs in DSPS mode.

It is possible that DEVs in DSPS mode will not receive broadcast messages. If a DEV requires the opportunity to receive all of the broadcast streams, it should not use DSPS mode. If a source DEV needs that the DEVs in DSPS mode have an opportunity to receive the broadcast frame, then the source DEV should send the frame in all of the superframes required to reach the DSPS DEVs in their wake beacons.

The PNC may grant an ACTIVE DEV's channel time request to create or modify a stream with PM Channel Time Request Type field set to ACTIVE and a DSPS mode DEV as the TargetID using the processes defined in 7.7.2.2 and 7.7.2.3.

If the CTA Rate Type field of a new allocation is set to sub-rate with the DSPS DEV as the destination but not aligned with one of the DEV's DSPS sets, the DSPS DEV may take any one of the following actions:

— Stay in DSPS mode while listening to the additional beacons required for the new allocation.

— Switch to ACTIVE mode using the PM Mode Change command.
— Terminate the stream, as described in 7.7.2.4.

7.17.3.4 CTA timing in DSPS mode

A DSPS Channel Time Request is a Channel Time Request command, as described in 6.5.7.2, with the PM Channel Time Request Type field set to DSPS. This requests that the PNC allocate channel time during the wake superframes of the specified DSPS set. An additional condition placed on the timing is that the value of the CTA Rate Factor field shall not be less than the number of superframes between wake beacons, i.e., the Wake Beacon Interval. Since the CTA Rate Factor field, like the Wake Beacon Interval, is a power of 2, the rate of DSPS CTAs also is a power of 2 sub-rate of the wake beacon rate, as illustrated in Figure 7-72. For example, in case 3 of Figure 7-72 the Wake Beacon Interval is $2 = 2^1$ while the CTA Rate Factor is $8 = 2^3$. Thus, the CTAs occur every fourth wake beacon, $4 = 8/2$. An example of values that are not allowed would be a Wake Beacon Interval of 4 and a CTA Rate Factor of 2. The reason that this is not allowed is that the CTAs would occur more often than the DEV was waking to listen to the beacon.

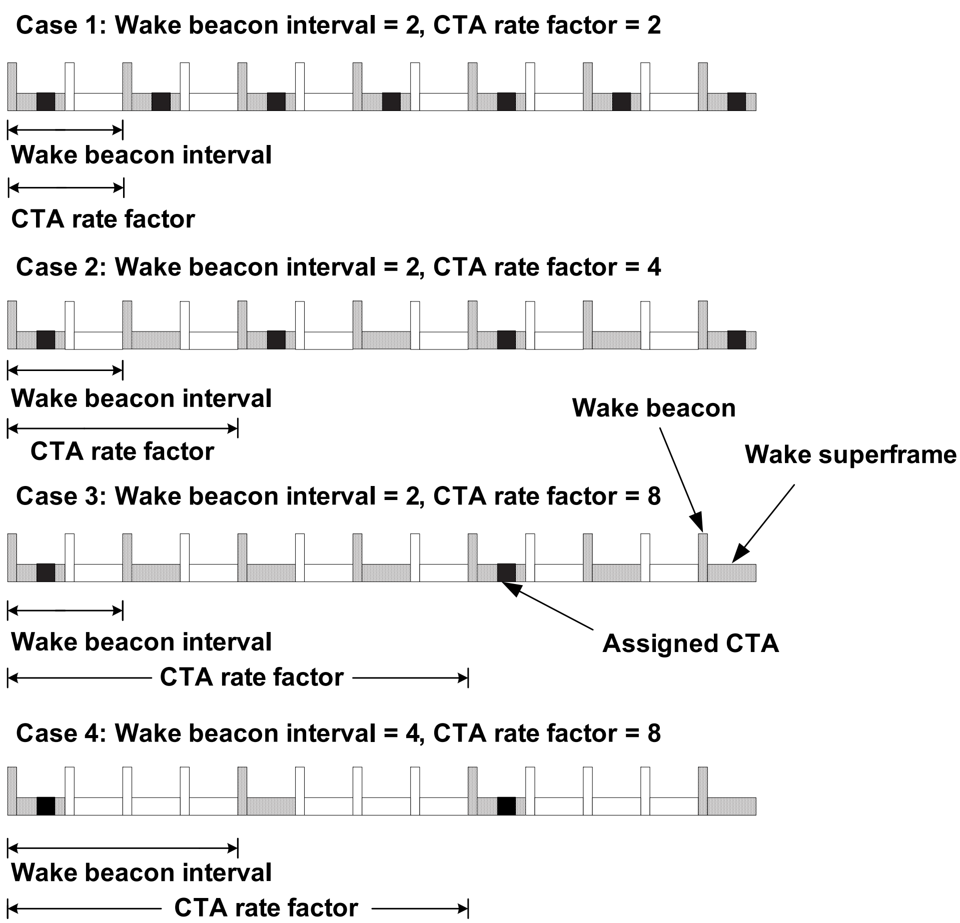

Figure 7-72—Illustration of DSPS set and sub-rate CTAs

If the PNC grants a DSPS channel time request, the PNC shall allocate CTAs only in the wake beacons of that DSPS set unless there is insufficient channel time for the allocation. If the PNC determines that it is unable to provide the requested CTA in a wake beacon, the PNC shall not allocate the CTA and shall take one of the following three actions:

a) Continue attempting to allocate the CTA at the appropriate times.
b) Terminate the stream, as described in 7.7.2.4.

c) Set the appropriate bits in the CWB IE, as described in 6.4.9, for the SrcID and DestID to indicate that those DEVs should wake up for the next beacon to see if there will be a CTA in the next beacon. The PNC may set the CWB bits in up to three consecutive beacons until it is able to allocate the CTA. Although this behavior is permitted, an undesirable reduction in battery life will result from unnecessary wake periods for DSPS DEVs.

A DSPS DEV shall listen to every wake beacon regardless of the frequency with which CTAs are allocated. If the DSPS DEV is the DestID of any CTA in the wake beacon, then the DSPS DEV shall listen during the assigned CTA in that wake superframe.

Figure 7-73 shows four equivalent PNC CTA arrangements for a Wake Beacon Interval of 2 and the CTA Rate Factor of 8. The PNC chooses the wake superframes to position the assigned CTAs at an overall rate of 1 CTA per 8 superframes.

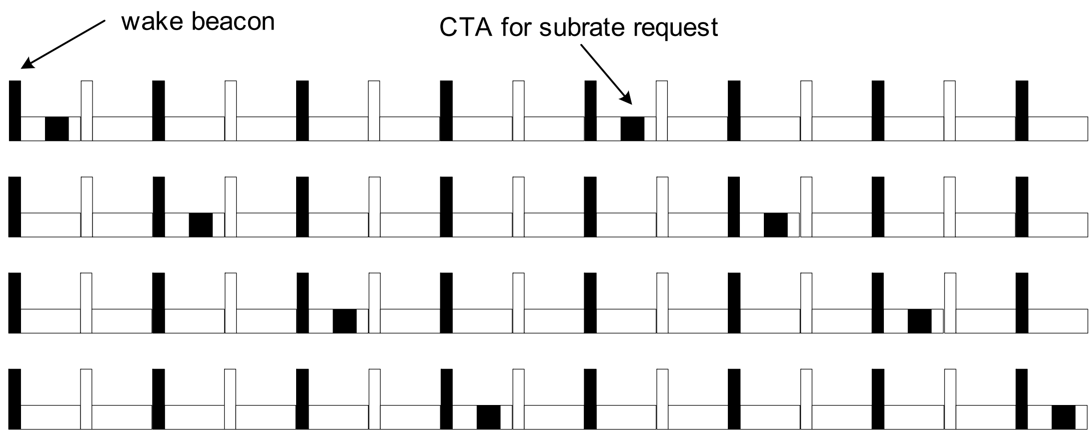

Figure 7-73—Equivalent CTAs for wake beacon interval = 2 and CTA rate factor = 8

The example in Figure 7-74 shows that Wake Beacon Interval and CTA Rate Factor together provide the ability of the DEV to trade off power savings and superframe loading, i.e., the number of CTAs allocated to any one superframe. In this example, the PNC minimizes the superframe loading for the eight DSPS DEVs, if they are members of different DSPS sets, because there is one wake beacon every other superframe, rather than the minimum, one wake beacon every eight superframes if power saving were maximized.

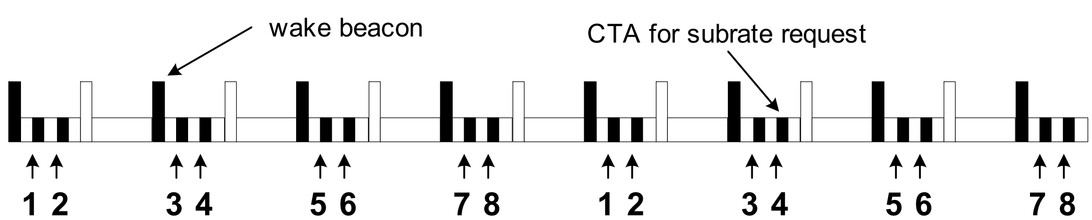

Figure 7-74—Minimum superframe loading of CTAs for 8 DEVs requesting wake beacon interval = 2 and CTA rate factor = 8

7.17.4 Asynchronous power save (APS) mode

APS mode allows a DEV to conserve power by remaining in SLEEP state for extended periods of time. The only responsibility of a DEV in APS mode is to communicate with the PNC before the end of its ATP in order to preserve its membership in the piconet.

In the APS mode the DEV is not required to listen to any beacons or other traffic until it changes to either ACTIVE or a different power save mode using the PM Mode Change command, as described in 6.5.9.6. APS mode shall not be used in combination with any other power save mode. A DEV shall not use the SPS Configuration Request command to set parameters for the APS Set Index.

All DEVs in APS mode need to send at least one acknowledged frame to the PNC during their ATP in order to avoid being disassociated from the piconet, as described in 7.4.5. Because the APS DEV will need to send a frame to the PNC at least once during its ATP, the PNC needs to take this into consideration when allocating MCTAs if the CAP is not available for sending commands.

A DEV shall send a PM Mode Change command to the PNC with the PM Mode field set to APS and receive the ACK before entering APS mode. When the PNC receives this command, it shall set the DEVID Bitmap field in the PS Status IE appropriately, as described in 6.4.21. The PNC shall terminate all streams and asynchronous data allocations that have the APS DEV as either the source or destination ID.

The PS Status IE in the beacon with the bit in the DEVID Bitmap field for the DEV's DEVID set shall serve as indication to other DEVs in the piconet that its peer has switched to APS mode. The PS Set Index of 0 shall only be used for APS DEVs. Although a PS Set Index is assigned to the DEVs in APS mode, the DEVs in this mode all act independently, unlike the DEVs that are members of other PS sets.

The DEV may leave APS mode by sending a PM Mode Change command to the PNC with the PM Mode field set to ACTIVE. Once this command is sent the DEV shall regard itself as in the ACTIVE mode whether the command was acknowledged by the PNC or not. If the PNC does not set the DEVID Bitmap in the PS Status IE appropriately, the DEV should resend the PM Mode Change command to the PNC.

The PNC may grant an ACTIVE mode DEV's channel time request, with PM Channel Time Request Type field set to ACTIVE and with an APS mode DEV as the DestID using the processes defined in 7.7.2.2 for isochronous allocations and in 7.7.3.1 for asynchronous allocations.

7.17.5 MSCs for power save modes

Figure 7-75 illustrates the message flow for a DEV inquiring about the PS sets that are currently defined within the PNC.

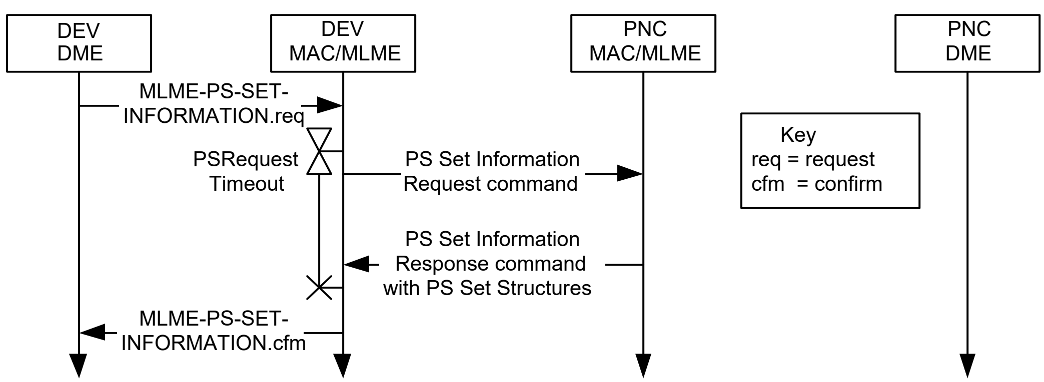

Figure 7-75—MSC for PS set information exchange

Figure 7-76 illustrates the message flow for a DEV requesting the creation of a new DSPS set.

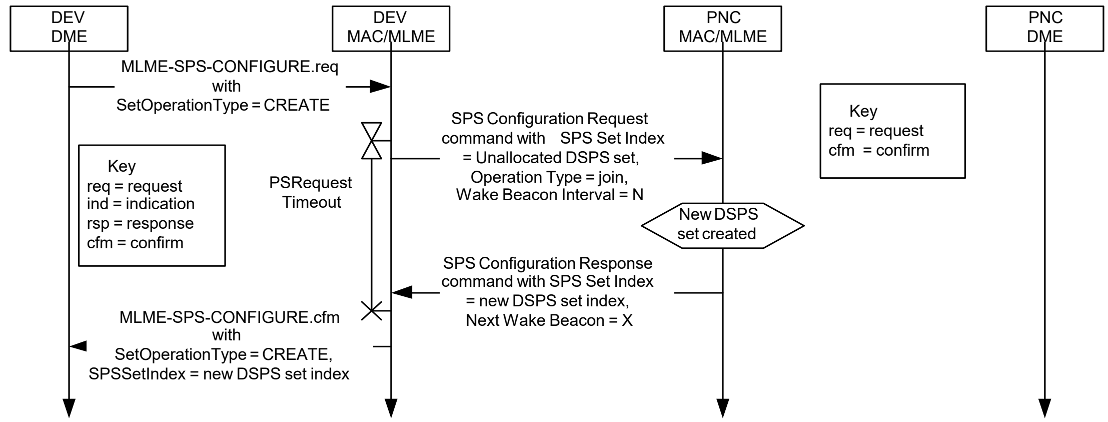

Figure 7-76—MSC for DSPS set creation

Figure 7-77 illustrates the message flow for a DEV requesting to add itself to an existing SPS set.

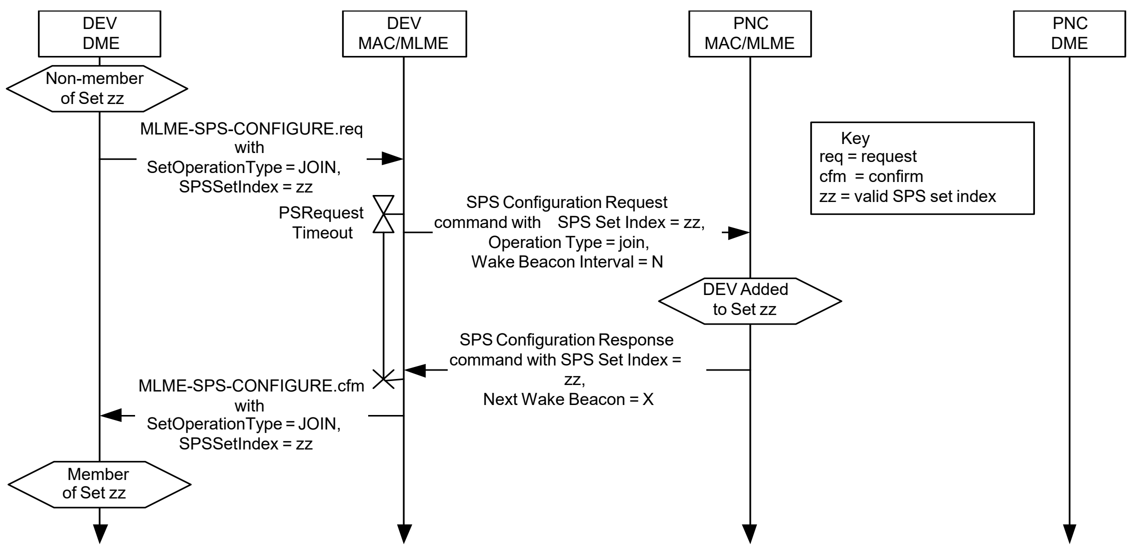

Figure 7-77—MSC showing a DEV joining an existing SPS set

Figure 7-78 illustrates the message flow for a DEV requesting to leave an SPS set.

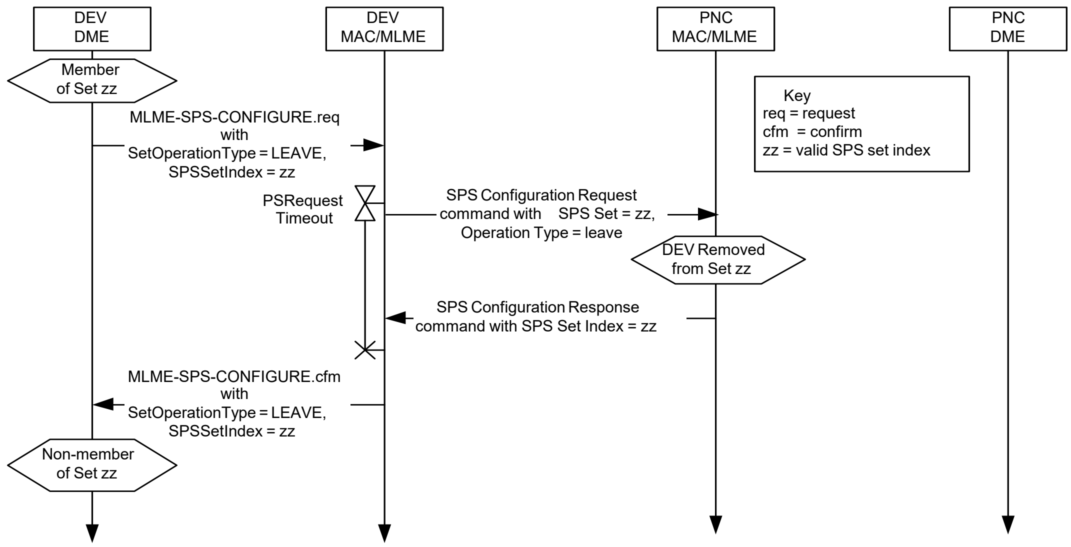

Figure 7-78—MSC showing a DEV leaving an SPS set

Figure 7-79 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from ACTIVE to an SPS mode when that DEV is a member of one or more SPS sets.

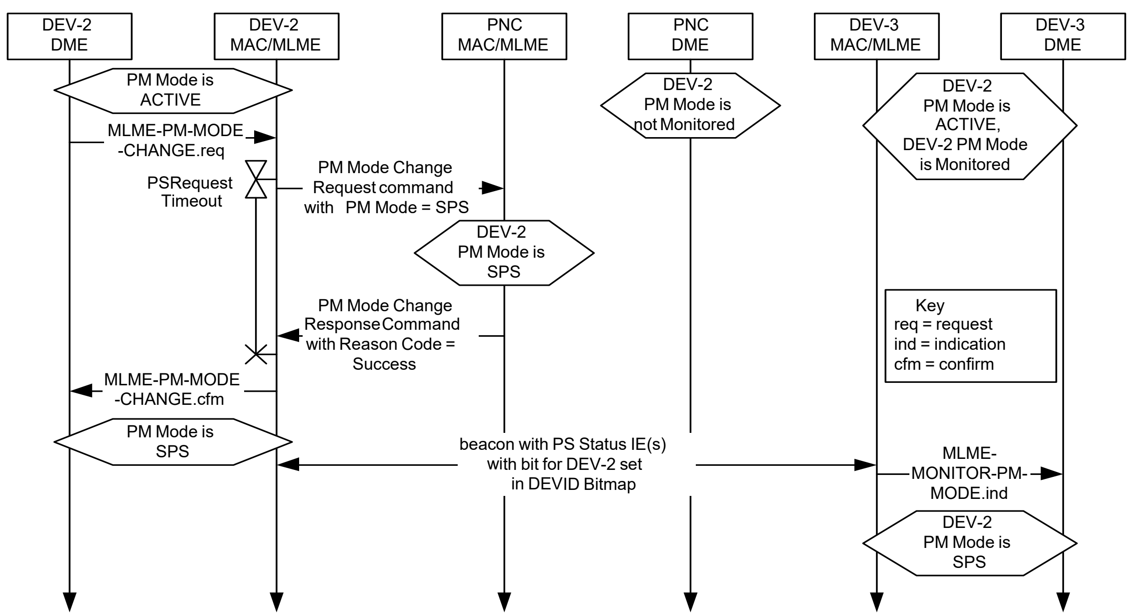

Figure 7-79—MSC showing DME initiated PM mode change from ACTIVE to an SPS mode

Figure 7-80 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from ACTIVE to APS mode.

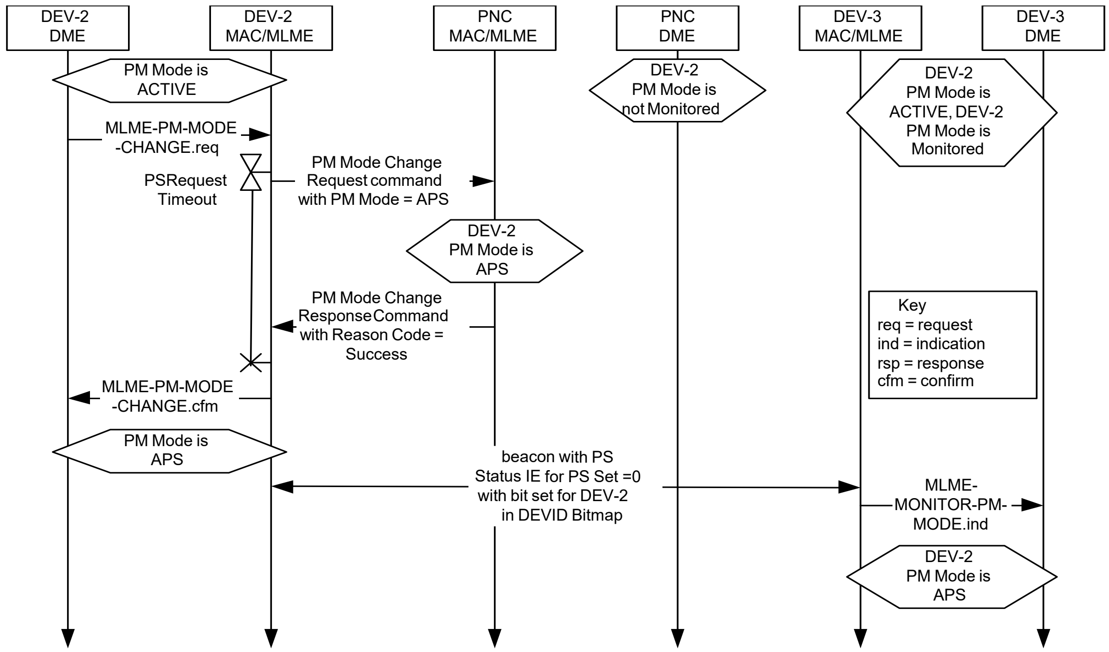

Figure 7-80—MSC showing DME initiated PM mode change from ACTIVE to APS

Figure 7-81 illustrates the message flow for a DEV DME requesting to change the current PM mode of operation from one of the power save modes to ACTIVE mode.

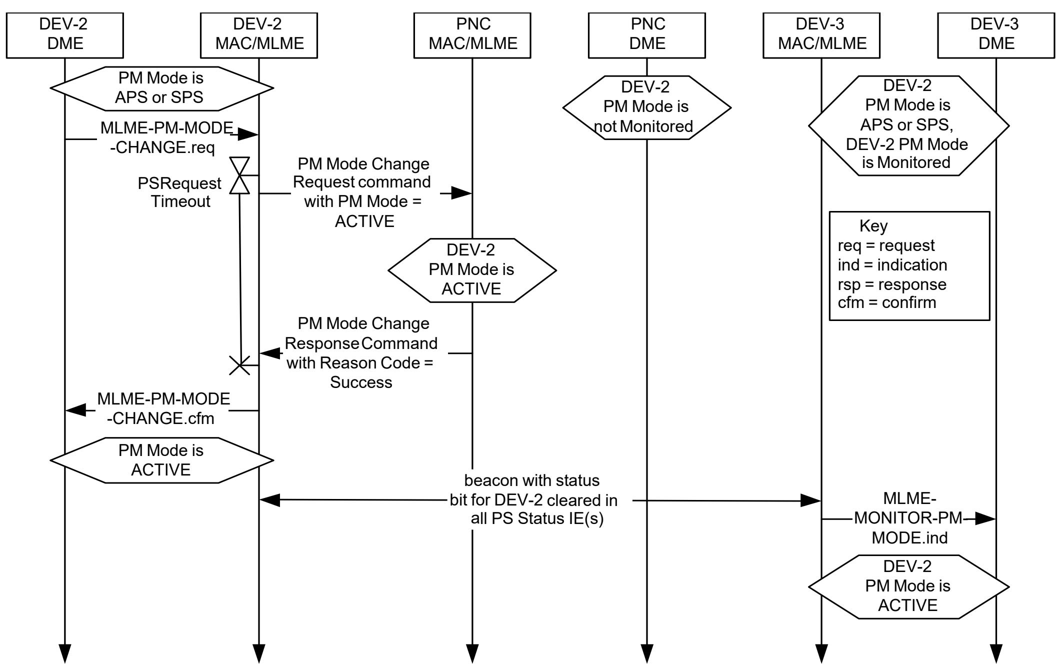

Figure 7-81—MSC showing DME initiated PM mode change from any PS mode to ACTIVE

Figure 7-82 illustrates the message flow for a DEV MLME changing the PM mode of operation from DSPS to ACTIVE due to an ACTIVE mode channel time allocation.

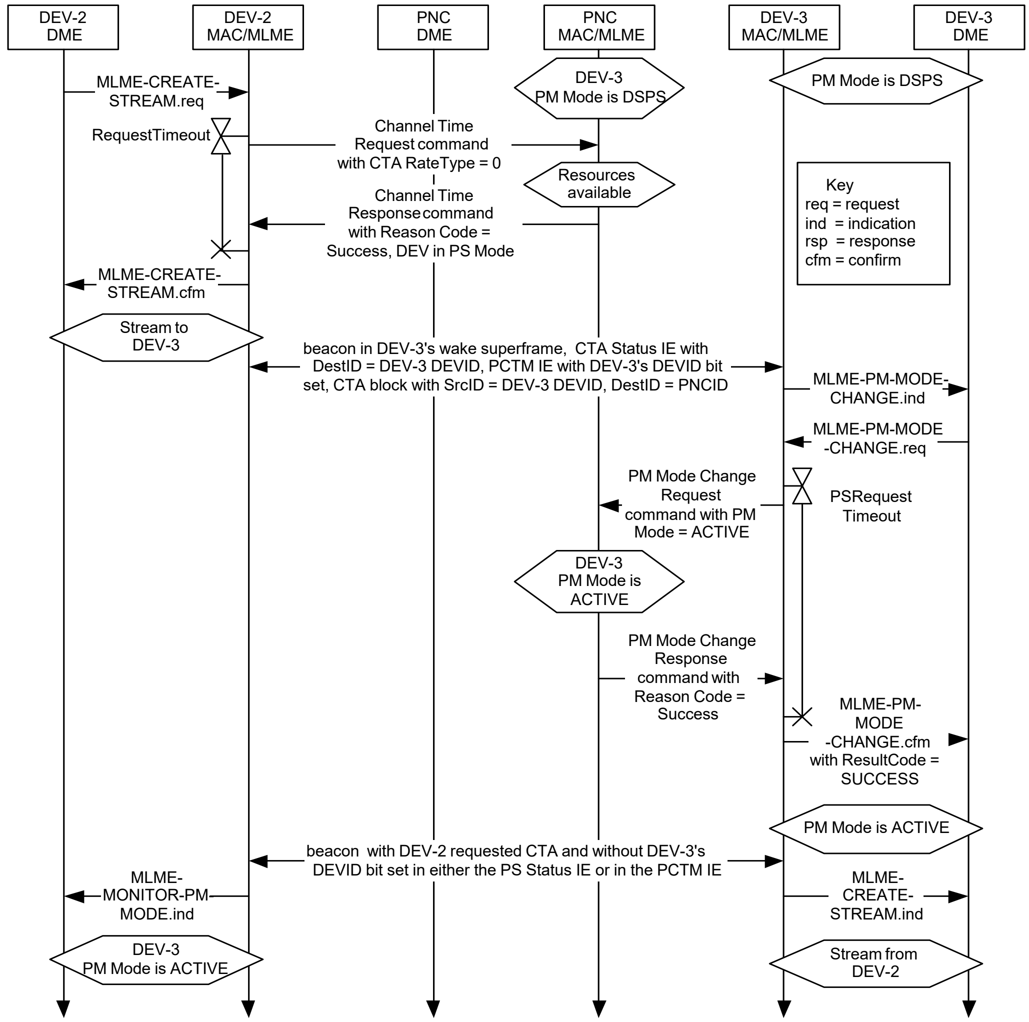

Figure 7-82—MSC showing MLME initiated PM mode change from DSPS to ACTIVE

7.18 Power management for pairnet

LLPS mode is a power saving mode which may be used only for pairnet in Associated State. LLPS mode allows conserving power by going into sleep state when both DEVs do not have data to send. Three LLPS control parameters, LLPS Start, LLPS Interval, and LLPS Extend shall be determined during the association process as described in 6.4.11.

Figure 7-83 shows the time domain LLPS mode behavior. Each DEV in ping-pong phase shall send an ACK if the DEV does not have data to send during its transmission turn, as described in 7.12.7.1. If the duration of alternating consecutive ACKs exceeds value of LLPS Start, then either DEV may send a Sleep Request to go to sleep state during an LLPS Interval. Sleep request is indicated by the DEV Sleep field in the ACK. The DEV that received Sleep Request without error may go into sleep state during the LLPS Interval. The DEV that did not correctly receive an ACK after a sleep request shall go into Recovery Process as explained in 7.12.7.2. LLPS Interval shall be set shorter than ATP to avoid disassociation.

After completing LLPS Interval with Sleep State, both DEVs shall wake up and return to ping-pong transmission via the Recovery Process. A DEV may restart LLPS Interval if the duration of alternating consecutive ACKs exceeds LLPS extend. However, if a DEV sends a data frame even once during ping-pong transmission, LLPS Start shall be applied.

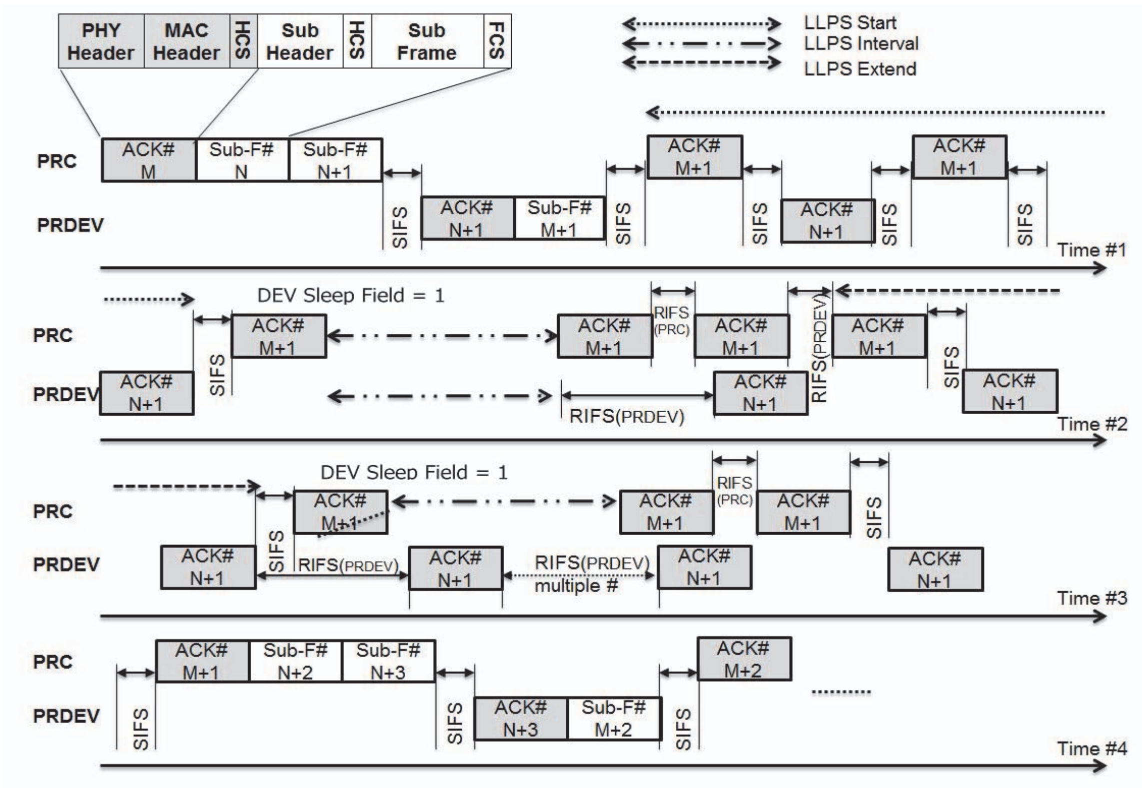

Figure 7-83—LLPS Time Domain Behavior

7.19 AS IE operation

The AS IE is used to implement out-of-scope features that require additional functionality by both the PNC and one or more of its piconet member DEVs. The additional functionality is defined as an enhancement that does not violate the standard and allows DEVs that do not have the functionality to operate normally. The Application Specific Data field in this IE provides the messages that are only interpreted by the targeted DEV.

Multiple AS IEs may be placed in the beacon by the PNC. The designer should minimize the size of each AS IE used to support the custom application.

To place a new AS IE in the beacon, the DEV sends the AS IE Request command to the PNC with the Request Type field set to "add" and the AS IE Index field set to zero, as shown in Figure 7-84. The requesting DEV selects a unique value of the Request ID field in the AS IE Request command, 6.5.10.3, to be able to determine which AS IE Response command is associated with this request. The AS IE Index field shall be ignored by the PNC in an AS IE Request command when the value of the Request Type field is set to "add."

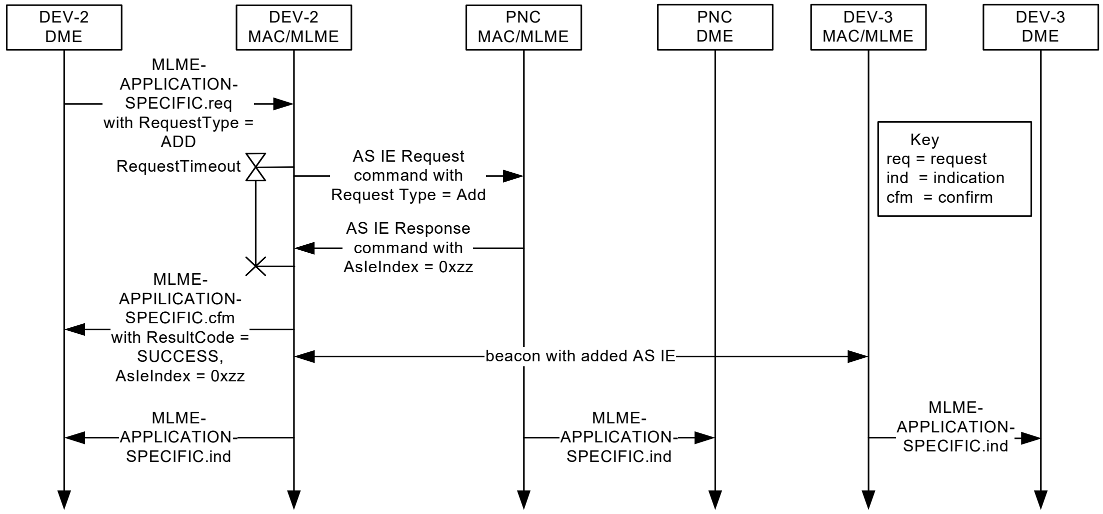

Figure 7-84—MSC for a DEV adding an AS IE to the beacon

If the PNC grants the request to place the AS IE in the beacon, it shall respond with an AS IE Response command to the originating DEV with the Request ID set to the same value as in AS IE Request command, the AS IE Index set to an unused value, and the Reason Code set to "success."

If the PNC refuses the request to place the AS IE in the beacon, it shall send an AS IE Response command to the originating DEV with the Request ID set to the same value as in the AS IE Request command and the Reason Code set to the appropriate value. The AS IE Index shall be set to zero by the PNC and shall be ignored by the originating DEV upon reception when the Reason Code is other than "success."

A DEV may modify an existing AS IE by sending the AS IE Request command to the PNC with the Request Type field set to "modify," the AS IE Index field set to the value assigned by the PNC for that AS IE, and the Request ID field set to a value unique for that DEV. This process is illustrated in Figure 7-85.

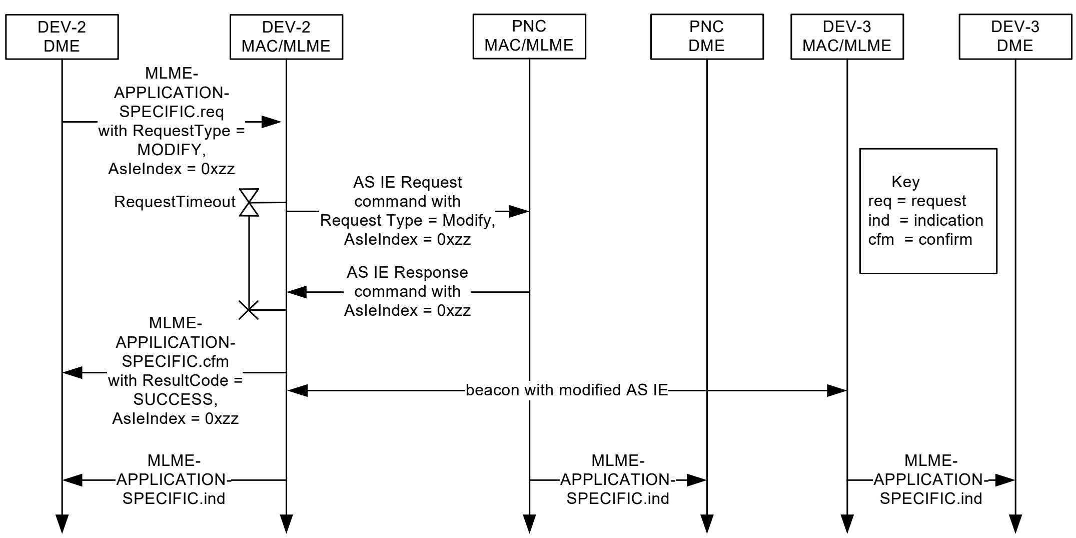

Figure 7-85—MSC for a DEV modifying an AS IE in the beacon

If the PNC allows the request to modify an existing AS IE, it shall respond with an AS IE Response command to the originating DEV with the Request ID and AS IE Index set to the same values as in AS IE Request command and the Reason Code set to "success."

If the PNC refuses the request to modify the AS IE in the beacon, it shall send an AS IE Response command to the originating DEV with the Request ID and the AS IE Index set to the same values as in the AS IE Request command and the Reason Code set appropriately. If the PNC refuses a request to modify an AS IE, it does not delete the AS IE, but it shall continue placing the existing AS IE in the beacon.

A DEV may also request that the PNC remove the AS IE from the beacon by sending the AS IE Request command to the PNC with the Request Type field set to "remove" and the AS IE Index field set to value for the AS IE that is to be removed. If the PNC successfully receives this command with the AS IE Index value that corresponds to an AS IE that is being currently managed by the PNC, it shall no longer place that AS IE in the beacon and shall send an AS IE Response command to the DEV with the Reason Code set to "success" and the Request ID and AS IE Index set to the same values as in AS IE Request command. This procedure is illustrated in Figure 7-86.

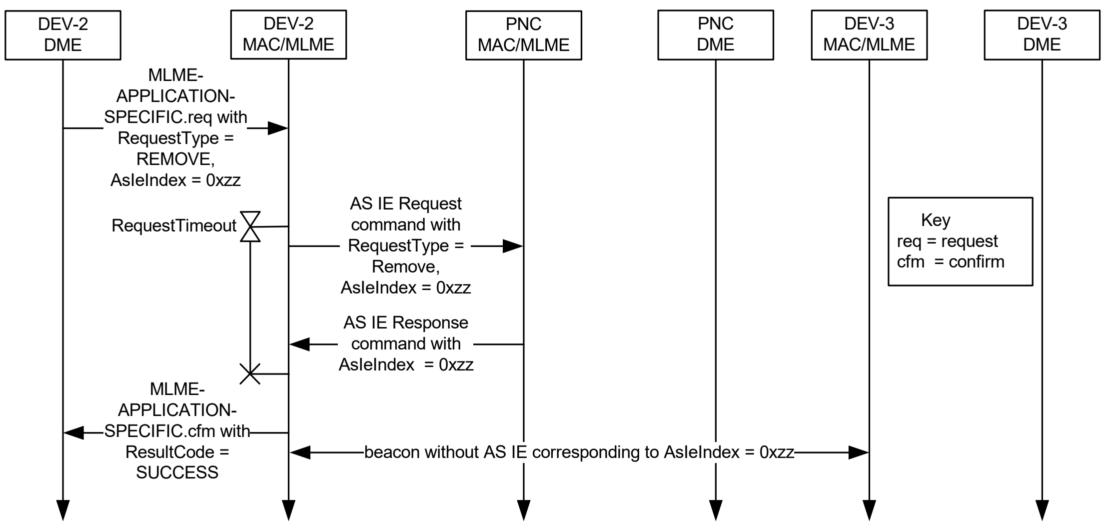

Figure 7-86—MSC for a DEV removing an AS IE from a beacon

If the value of the AS IE Index in an AS IE Request command with Request Type set to "modify" or "remove" does not correspond to an AS IE that is currently being managed by the PNC, the PNC shall refuse the request with the Reason Code set to "Unassigned AS IE Index."

Only the DEV that requested an AS IE be added to the beacon shall be allowed to modify or remove that AS IE.

Once the PNC has allowed a request to add or modify an AS IE, it shall place the AS IE in every beacon until the originating DEV either has been disassociated or has requested that the PNC remove the AS IE from the beacon. The PNC shall remove all AS IEs that were sent by a DEV when that DEV disassociates from the piconet. AS IEs are not transferred during handover, and so DEVs that need AS IEs in the beacon will need to send AS IE Request commands to the new PNC once it has taken over the piconet.

7.20 MAC sublayer parameters

The parameters that define some of the MAC characteristics are given in Table 7-7.

Table 7-7—MAC sublayer parameters

Parameter	Value
mMinChannelScan	*mMaxSuperframeDuration*
mBroadcastDevInfoDuration	64×*mMaxSuperframeDuration*
mAssocRespConfirmTime	4×*mMaxSuperframeDuration*
mMinSuperframeDuration	1000 µs
mMaxSuperframeDuration	65535 µs
mMaxLostBeacons	4
mMctaAssocPeriod	150 ms
mFirstCtaGap	100 µs
mMinBeaconInfoRepeat	4
mAsyncRequestLifetime	1 s
mMaxKeyChangeDuration	65535 ms
mMaxTimeTokenChange	65535
mMaxNumValidDEVs	243
mCtaRelinquishTimeout	3 × SIFS + 2 × (duration of Imm-ACK frame)
mMaxSubframeNum	For standard aggregation mode = 8, for low-latency mode = 256
mMaxSubframeSize	For standard aggregation mode = 10478575, for low-latency mode = 256

Additional characteristics that are PHY dependent are indicated in Table 7-8 for the 2.4 GHz PHY.

Table 7-8—MAC sublayer parameters—2.4 GHz PHY dependent

Parameter	Subclause
SIFS	11.2.7.2
MIFS	11.2.7.2
BIFS	11.2.7.2
RIFS	11.2.7.2
pBackoffSlot	11.2.7.2
pChannelSwitchTime	11.2.7.6
pMinTpcLevel	11.5.9
p*MaxTransferUnitSize*	11.2.8.3

Table 7-8—MAC sublayer parameters—2.4 GHz PHY dependent (*continued*)

Parameter	Subclause
pMaxFrameBodySize	11.2.8.2
pClockAccuracy	11.5.5
pLengthHcs	11.2.9
pMinFragmentSize	11.2.8.4

Additional characteristics that are PHY dependent are listed in Table 7-9 for the SC PHY.

Table 7-9—MAC sublayer parameters—SC PHY dependent

Parameter	Subclause
SIFS	12.2.7.2
MIFS	12.2.7.2
BIFS	12.2.7.2
RIFS	12.2.7.2
pBackoffSlot	12.2.7.2
pMaxTransferUnitSize	12.2.8.3
pMaxFrameBodySize	12.2.8.2
pClockAccuracy	12.2.5.3
pLengthHcs	12.2.4.3.3
pMinFragmentSize	12.2.8.4

Additional characteristics that are PHY dependent are listed in Table 7-10 for the HSI PHY.

Table 7-10—MAC sublayer parameters—HSI PHY dependent

Parameter	Subclause
SIFS	12.3.6.5
MIFS	12.3.6.8
BIFS	12.3.6.6
RIFS	12.3.6.6
pBackoffSlot	12.3.6.6
pMaxTransferUnitSize	12.3.7.5
pMaxFrameBodySize	12.3.7.4
pClockAccuracy	12.3.5.2

Table 7-10—MAC sublayer parameters—HSI PHY dependent (*continued*)

Parameter	Subclause
pLengthHcs	12.3.4.5.1
pMinFragmentSize	12.3.7.6

Additional characteristics that are PHY dependent are listed in Table 7-11 for the AV PHY.

Table 7-11—MAC sublayer parameters—AV PHY dependent

Parameter	Subclause
SIFS	12.4.2.2.3
MIFS	12.4.2.2.5
BIFS	12.4.2.2
RIFS	12.4.2.2
pBackoffSlot	12.4.2.2
pMaxTransferUnitSize	12.4.2.3.3
pMaxFrameBodySize	12.4.2.3.2
pClockAccuracy	12.4.5.3
pLengthHcs	12.4.2.4
pMinFragmentSize	12.4.2.3.4

The parameters that define some of the pairnet MAC characteristics are given in Table 7-12.

Table 7-12—MAC sublayer parameters for HRCP-SC PHY, HRCP-OOK PHY, THz-SC PHY, and THz-OOK PHY dependent

Parameter	Values
mMinSuperframeDuration	*pMaxBeaconLengthTime +SIFS+ pDAccessSlot*
mMaxSuperframeDuration	*pMaxBeaconLengthTime+SIFS+max(pNAccessSlot)*pDAccessSlot*
mMaxSubframeSize	8192 octets
mMaxNumValidDEVs	2
mMaxTimeTokenChange	65535

Additional characteristics that are PHY dependent are indicated in Table 7-13 for the HRCP SC PHY.

Table 7-13—MAC sublayer parameters—HRCP SC PHY dependent

Parameter	Values or subclause
pNAccessSlot	1–6
pDAccessSlot	8.5 μs ≤ (pMaxAssocReqCommandTime + SIFS) ≤ 10.5μs
pMaxFrameBodySize	Defined in 13.2.8.2
pMaxBeaconLengthTime	8 μs
pMaxAssocReqCommandTime	8 μs

Additional characteristics that are PHY dependent are listed in Table 7-14 for the HRCP OOK PHY.

Table 7-14—MAC sublayer parameters—HRCP-SC PHY, THz-SC PHY, and THz-OOK PHY dependent

Parameter	Values or subclause
pNAccessSlot	mod((Beacon Interval – pMaxBeaconLengthTime-SIFS)/pDAccessSlot)
pDAccessSlot	pMaxAssocReqCommandTime+SIFS
pMaxFrameBodySize	Defined in 13.3.8.2
pMaxBeaconLengthTime	Unspecified (depends on optional IEs)
pMaxAssocReqCommandTime	Unspecified (depends on optional IEs)

7.21 Unequal error protection (UEP)

The following three types of UEP may be used for flexible error protection of a frame consisting of MSB and LSB subframes:

— UEP Type 1 protects an aggregated frame that consists of MSB subframes, LSB subframes, or both MSB and LSB subframes by using the different FECs.

— UEP Type 2 protects an aggregated frame that consists of MSB subframes, LSB subframes, or both MSB and LSB subframes by using different MCSs. The UEP Specific IE, as defined in 6.4.31, is used in UEP Type 2 only for the originating DEV to inform the target DEV of the separating position of the MSBs and LSBs. If necessary, a DEV using UEP Type 2 may send an Announce command with the UEP Specific IE to dynamically change the separating position.

— UEP Type 3 protects MSB and LSB in a subframe unequally by either applying different FEC coding rates to MSBs and LSBs or mapping MSBs and LSBs to a skewed constellation. The capability of supporting UEP by FEC coding and/or by the skewed constellation is indicated in the Capability IE, as defined in 6.4.12.

All UEP types, UEP Type 1, UEP Type 2, and UEP Type 3, may be used in SC PHY, as described in 12.2.3.9. UEP Type 3 may be used in HSI PHY, as described in 12.3.3.1, and AV PHY, as described in 12.4.

It is required that originating and target DEVs understand the UEP type and MCS information of each other before sending frames that use UEP. This is accomplished by the DEVs reporting their UEP capability in the DEV Capability IE in the association process. If the source and destination do not support the same UEP type, then only EEP frames shall be exchanged between the DEVs.

7.22 Sync frame transmission and virtually dependent piconet

Sync frame transmission is a function that mitigates co-channel interference due to a hidden PNC node. It also provides a method to obtain synchronization among independent piconets. Sync frame transmission is mandatory for PNC-capable DEVs when operating as a mmWave PNC, as defined in 12.1.10, and is optional for non-PNC-capable mmWave DEVs. A mmWave DEV indicates if it is capable of sending Sync frames by setting the Sync Frame Capable field in the Capabilities IE, 6.4.12.

The PNC controls the transmission of Sync frames by DEVs in the piconet by sending a Sync Frame Frequency IE, as defined in 6.4.42, in a Announce command to a DEV that is capable of sending Sync frames. The PNC sets the frequency with which the frames will be sent by setting the Sync Frame TX Frequency field to the appropriate value.

When a DEV that is capable of sending Sync frames receives the Sync Frame Frequency IE from the PNC, it sends a Sync frame in the first transmission opportunity in the next superframe. The DEV continues sending the Sync frame every Sync Frame TX Frequency superframe. DEVs may use the response frame in an Imp-ACK exchange to send the Sync frame. DEVs may use either data or Imp-ACK transmit opportunities for sync frame transmission.

A DEV that receives a Sync frame from another piconet may utilize the information in the frame to obtain network synchronization and mitigate interference. A PNC may use Sync frames that it receives from DEVs in another piconet to synchronize to PNC timing and forming a virtually dependent piconet. The PNC may then adjust the CTAs in its piconet to avoid interference with the other piconet.

8. Security

8.1 Security mechanisms

8.1.1 General

Security mechanisms provided by this standard allow security services to be implemented to control the admission of DEVs into a security relationship between the PNC or PRC and a DEV or between two ordinary DEVs and protect the information and integrity of communications between DEVs in a security relationship. This standard also provides a symmetric cryptography mechanism to assist in providing security services. Additional security services need to be provided by the higher layers to ensure proper management and establishment of the symmetric keys used in this standard.

The background assumptions used in designing this security solution are outlined in C.1.

8.1.2 Security membership and key establishment

The method by which a DEV becomes a member of a security relationship and obtains the appropriate key is outside of the scope of this standard. The Security Message command has been included as a special command to assist in the implementation of vendor-defined protocols for establishing security relationships and any related data. It can be achieved with higher layer protocols that are not specified in this standard. The MAC/MLME is informed of changes to the membership of a security relationship and the key for that relationship with the MLME-MEMBERSHIP-UPDATE primitive, as described in 5.3.8.2.

Because the association protocol is unprotected, to maintain security, an authentication algorithm needs to include the DEVID among the items that are validated.

8.1.3 Key transport

All keys that are transmitted from one DEV to another shall be encrypted as specified in the key request, as described in 8.4.4, and distribute key protocols, as described in 8.4.3. For example, key transport is used to provide a copy of the piconet group data key or pairnet group data key to a DEV.

8.1.4 Data encryption

Data encryption uses a symmetric cipher to protect data from being read by parties without the cryptographic key. Data may be encrypted either by using a key shared by all associated DEVs or by using a key shared between only two DEVs.

8.1.5 Data integrity

Data integrity uses an integrity code to protect data from being modified by parties without the cryptographic key. It further provides assurance that data came from a party with the cryptographic key. Integrity may be provided using a key shared by all piconet DEVs or using a key shared by all pairnet DEVs, or using a key shared between only two DEVs. All secure data frames that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indication, and no other action is taken on the frame by the MLME.

8.1.6 Beacon integrity protection

The beacon may be integrity-protected. This integrity protection provides evidence to all the DEVs in the piconet or pairnet that the PNC or PRC of the secure piconet or pairnet transmitted the beacon. Under normal operations, the integrity check on the beacon provides evidence that the piconet or pairnet is operating properly and that no security changes have occurred. When not scanning for other piconets, if the integrity check on the beacon fails, the DEV is alerted to the fact that the DEV does not have its security state synchronized with the PNC or PRC.

8.1.7 Command integrity protection

The integrity of commands may be protected just like any other data. Integrity protected commands sent between the PNC and a DEV or PRC and a PRDEV shall be protected using the PNC-DEV management key or PRC-DEV management key. All secure commands that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indication, and no other action is taken on the frame by the MLME.

8.1.8 Freshness protection

To prevent replay of old messages, a strictly increasing time token is included in the beacon. A DEV shall reject as invalid a received beacon with a time token less than or equal to the current time token. For pairnets, a DEV shall further check the SFC and the SECID included in the Beacon frame and shall reject as invalid the Beacon frame if the SFC value in the Beacon frame is not strictly greater than the last SFC value received from that DEV corresponding to the key identified by the SECID. The last SFC value received shall be only updated after the received integrity code corresponding to the SFC value of the received frame or subframe is successfully verified. In addition, for piconets, the time token is included in the nonce, as described in 9.2.4, for each secure frame, as described in 6.2, so the integrity check will fail if a frame is replayed in a different superframe. For pairnets, a DEV shall check the SECID included in each secure frame and the SFC value of each secure frame or subframe, and shall reject as invalid the received frame or subframe if the SFC value corresponding to the frame or subframe is not strictly greater than the last SFC value received from that DEV corresponding to the key identified by the SECID to detect whether the frame or subframe is replayed or not. The last SFC value received shall be only updated after the received integrity code corresponding to the SFC value of the received frame or subframe is successfully verified. A DEV maintains two values for freshness. The CurrentTimeToken is the time token value found in the beacon for the current superframe. For piconets, the CurrentTimeToken is used to protect all messages sent and check all messages received during that superframe. For pairnets, the CurrentTimeToken, together with the SFC value, is used to check beacon freshness, and only the SFC value is used to check freshness of other frames. The LastValidTimeToken is used by the DEV to ensure that the security of the beacons have not been compromised.

8.2 Security modes

8.2.1 General

The security mode indicates whether a DEV is currently implementing frame protection in the piconet or pairnet. The security mode in use is determined by the *macSecurityOptionImplemented* entry in the MAC PIB.

8.2.2 Security mode 0

A DEV operating in security mode 0 shall not perform any cryptographic operations on MAC frames. While in this mode, if the MAC receives a frame with the SEC field set to one, the MAC shall discard the frame, and the MLME shall return an MLME-SECURITY-ERROR.indication to the higher layer with the ReasonCode set to UNAVAILABLE-KEY.

8.2.3 Security mode 1

Security mode 1 provides a mechanism for a device to perform cryptographic security on frames transmitted in the piconet or pairnet. DEVs operating in security mode 1 use symmetric-key cryptography to protect frames using encryption and integrity.

While in mode 1, the cryptographic operations used for secure frames exchanged with the PNC or PRC and with other members of the piconet or pairnet security group shall be performed as specified by the symmetric key security operations. While in this mode, if the MAC receives a frame with the SEC field in the Frame Control field set to a value different than expected, as defined in Table 6-47 for piconet and Table 6-48 for pairnet, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

8.3 Security support

8.3.1 General

The security policies determine the actions taken to preserve the security of the piconet or pairnet. Subclauses 8.3.2 through 8.3.10 specify the methods that are provided in this standard to support specific security policies.

8.3.2 PNC handover in piconets

When a PNC chooses to hand over the PNC role to another DEV in the piconet, the security relationships with the old PNC no longer apply to the new PNC. When the old PNC hands over the piconet information using a PNC Information command, as described in 6.5.5.3, the list of associated DEVs is passed to the new PNC.

PNC handover does not affect the group membership, so it does not require a rekey of the group key. However, in a piconet with payload protection, the command functions of the PNC that relate to specific DEVs are not supported until the new PNC has established secure membership with each DEV in the piconet. When the PNC role has been handed over, the new PNC should create CTAs for each of the associated DEVs to establish secure membership with the new PNC.

The old PNC may send security information about the new PNC to the other DEVs in the piconet and send security information about all of the DEVs that are secure members in the piconet to the new PNC when it hands over the role of the PNC. This is accomplished by sending a directed Security Information command, as described in 6.5.5.5, to the new PNC with the security information of the piconet in it and by sending a broadcast Security Information command or a directed Security Information command to each member of the piconet with the security information of the new PNC.

8.3.3 Changes in the group data key

When the PNC or PRC changes the piconet group data key or pairnet group data key, the PNC or PRC shall transmit the new key to all of the members of the piconet or pairnet that are in ACTIVE mode using the Distribute Key Request command, as described in 6.5.3.4.

For piconet, once the Distribute Key Request command has been issued for all of the members of the piconet that are in ACTIVE mode, the PNC may change the SECID in the beacon. When a DEV receives a valid Distribute Key Request command, as described in 6.5.3.4, from the PNC, the DEV shall use the new key for all outgoing secure frames that require the use of the piconet group data key once it sees the corresponding SECID in the beacon. The DEV may continue to accept frames protected by the old piconet group data key for up to *mMaxKeyChangeDuration* since the DEV last received a valid beacon protected by the old piconet-wide group data key.

For pairnet, once the Distribute Key Request command has been issued for the member of the pairnet that are in ACTIVE mode, the PRC shall change the SECID in the outgoing secure frames. When a DEV receives a valid Distribute Key Request command, as described in 6.5.2.3 from the PRC, the DEV shall use the new key for all outgoing secure frames that require the use of the pairnet group data key once it sees the corresponding SECID in the received frame.

If a DEV receives a beacon with a time token greater than the last known time token, but with a SECID that does not match the SECID of the known key, the DEV shall send a Key Request command to the PNC or PRC to obtain the new key.

8.3.4 Joining a secure network

If a DEV wishes to join a secure piconet or secure pairnet, it should associate with the PNC or PRC in order to be assigned a local DEVID. Once the piconet DEV is associated, the PNC shall allocate an MCTA if commands are not allowed in the CAP. The DEV or PNC or PRC may choose to send Probe Request and/or Announce commands to each other to either request or transmit IEs, including Vendor Defined IEs. The DEV and PNC or PRC may also exchange additional data frames or Security Message commands. After the DEV has associated and exchanged the desired information with the PNC or PRC, the DEV shall establish secure membership. The process by which secure membership is established is outside of the scope of this standard.

8.3.5 Membership update

When the DME determines that there has been a change of membership status with a particular DEV or when a management key is changed, the DME shall issue an MLME-MEMBERSHIP-UPDATE.request to its MLME. This membership status change or key change may be the result of a successful establishment of a security relationship, key update process, termination of a security relationship, or some other event. The process by which this change occurs is outside the scope of this standard.

When the MLME receives the MLME-MEMBERSHIP-UPDATE.request, it shall first examine the TrgtId to determine the membership relationship to modify. If the TrgtId is the PNCID or PRCID, the management key corresponds to the management key for the relationship with the PNC or PRC, and the MembershipStatus indicates whether the DEV is a secure member of the piconet or piconet. Otherwise, the management key is for a peer-to-peer relationship with the DEV indicated by the TrgtId, and the MembershipStatus indicates whether the DEV shares a secure relationship with that peer DEV. If the OrigId is the PNCID or PRCID, then the management key corresponds to the management key for a secure relationship with a DEV in the piconet or pairnet and will be used for PNC-related frames or PRC-related frames.

If the TrgtId is the PNCID or PRCID and the MembershipStatus is set to MEMBER, the DEV is a secure member of the piconet or pairnet. If the TrgtId is the PNCID and the MembershipStatus is set to NON-MEMBER, the DEV is no longer a secure member of the piconet or pairnet.

The MembershipStatus field indicates to the MLME whether the DEV is currently maintaining secure relationship information with the target DEV. If the MembershipStatus is set to NON-MEMBER, the MLME shall securely delete the management key, the data key, and the related SECID, key type, and key originator values corresponding to that TrgtId. When a DEV is not a member of a security relationship with a peer DEV, the DEV shall select keys for secure frame processing as if the DEV does not have an individual relationship with that peer DEV, as described in 8.3.10. When a DEV is not a member with the PNC or PRC, the DEV is not a secure member of the piconet or pairnet and shall select keys for secure processing as if the

DEV does not have a piconet group data key or PNC-DEV management key, or pairnet group data key or PRC-DEV management key as described in 8.3.10.

If the MembershipStatus is set to MEMBER, the MLME shall examine the KeyInfoLength field to determine if a new key is being added or a key is being deleted. If the KeyInfoLength field is set to zero, the MLME shall securely delete the key and SECID corresponding to the management key for that relationship. When the key is deleted, the DEV is unable to transmit or successfully receive frames to any DEV that require protection with the management key, as described in 8.3.10, but the DEV may continue to use the data key corresponding to that relationship and the piconet group data key or pairnet group data key. This may occur, for instance, during PNC handover, in which the management key with the PNC is no longer valid (since the PNC has changed), but the piconet group data key is still valid.

If the MembershipStatus is set to MEMBER and the KeyInfoLength field is not 0, the MLME shall set the SECID, key originator field, and key for the management key of this relationship to the values in the SECID, KeyOriginator, and KeyInfo fields, respectively, from the MLME-MEMBERSHIP-UPDATE.request.

8.3.6 Secure frame generation

When a DEV wishes to send a secure frame, it shall use the keying material required for the type of frame and by the relationship between the sending DEV and the receiving DEV. For each security relationship, there are two keys used to protect secure frames: a management key and a data key. Table 8-1 provides a listing of which of the keys shall be used to protect secure frames and which frames shall be sent without security for piconet DEVs. Table 8-2 provides the listing for PRDEVs. A DEV shall not send a secure frame if the only key selection in Table 8-1 or Table 8-2 is “none.” A DEV shall not send an unprotected frame or a frame with an incorrect SECID when security is required for that frame. If the DEV is unable to find the corresponding key that is to be used, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not transmit the requested frame.

A PNC in a piconet or PRC in a pairnet using security shall send secure beacons protected with the piconet group data key or pairnet group data key stored in the MAC/MLME. For each superframe, the PNC or PRC shall increment the time token and transmit a secure beacon with the SEC field in the Frame Control field set to one.

Key selection for secure frames is described in 8.3.10.

If the piconet DEV is able to obtain the appropriate keying material, the DEV shall use the CurrentTimeToken and secure frame counter for the corresponding SECID to construct the CCM nonce, Figure 9-1, used to protect the secure frame. If the PRDEV is able to obtain the appropriate keying material, the DEV shall use the secure frame counter for the corresponding SECID to construct the GCM nonce, Figure 10-1, used to protect the secure frame. The SECID included in the frame shall be the value corresponding to the keying material being used.

For piconets, the integrity code shall be computed as specified in 9.3.2. The result of the integrity code computation shall be encrypted as specified in 9.2.2 and placed in the Integrity Code field in the secure frame. The encryption operation shall be applied only to the integrity code, the key that is transmitted in a Distribute Key command or Request Key Response command and the payload of data frames. The result of the encryption operation shall be inserted into the frame in the place of the data that was encrypted. The DEV shall then compute the FCS over the modified frame.

For pairnets, authenticated encryption shall be applied as specified in 10.4.3, to the GCM input specified in 10.3.2. The encryption shall be applied only to the key that is transmitted in a Distribute Key command or Request Key Response command and the payload of data frames. The result of the encryption operation shall be inserted into the frame in the place of the data that was encrypted. The resulting integrity code shall be placed in the Integrity Code field in the secure frame. The DEV shall then compute the FCS over the modified frame.

A piconet DEV shall send only frames that have increasing SFCs in a superframe, except for frames that are retransmitted with the same SFC without any intervening frames having been sent. A PRDEV shall send only frames or subframes that have increasing SFC values for a single key corresponding to the SECID indicated in the transmitted frames.

8.3.7 Updating CurrentTimeToken

If the DEV is able to determine that it missed a beacon or that the beacon was corrupted and if CurrentTimeToken is less than LastValidTimeToken + *mMaxTimeTokenChange* – 1, the DEV should increment the CurrentTimeToken to maintain synchronization with other DEVs in the piconet or pairnet.

8.3.8 Secure frame reception

Before any security operations have been performed on a received frame, the DEV shall check the FCS. For pairnets, if the FCS check for a subframe in the received aggregated frame fails, then the subframe with the FCS check failure and the other subsequent subframes in the aggregated frame shall be ignored by the DEV. Table 8-1 provides a listing of the keys that shall be used to protect secure frames and the frames that shall be sent without security for piconets. Table 8-2 provides the listing for pairnets. A DEV may ignore any secure frame if the only key selection in Table 8-1 or Table 8-2 is "none." A DEV shall ignore any non-secure frame or a secure frame with an incorrect SECID when security is required.

An associated device that has not yet received the piconet group data key or pairnet group data key shall accept all secure beacons and ignore the integrity code, SECID, and secure frame counter. When the DEV has received the piconet group data key or pairnet group data key, it shall set the LastValidTimeToken and CurrentTimeToken to be the time token in that beacon.

When a DEV receives a secure Beacon frame, as defined in 6.3.1.3 for piconet and 6.3.1.4 for pairnet, the DEV shall determine if the received time token is greater than the CurrentTimeToken and less than the LastValidTimeToken + *mMaxTimeTokenChange*. If not, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-TIME-TOKEN and shall not perform any additional operations on the received beacon. The DEV shall also determine if the SECID matches the SECID of the piconet group data key or pairnet group data key stored in the MAC/MLME, or the SECID of a valid old piconet group data key or old pairnet group data key, as described in 8.3.6. If the SECID matches, a PRDEV shall further check the SFC included in the Beacon frame and the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-SFC and shall not perform any additional operations on the received Beacon frame if the SFC value in the Beacon frame is not strictly greater than the last SFC value received from that DEV corresponding to the key identified by the SECID. The last SFC value received shall be only updated after the received integrity code corresponding to the SFC value of the received frame or subframe is successfully verified. If the SECID does not match, the DEV may request a new piconet group data key or pairnet group data key, as described in 8.3.3. If these checks succeed, the DEV shall check the integrity code on the beacon using the piconet group data key or pairnet group data key. If this succeeds, the DEV shall accept the beacon and set the LastValidTimeToken and CurrentTimeToken to be the time token in the beacon.

When a DEV receives a secure non-Beacon frame, it shall use the appropriate keying material depending on the type of frame, SECID, and SrcID found in the frame. If the SECID in the frame does not correspond to known keying material in the receiving DEV, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to UNAVAILABLE-KEY and shall not perform any additional operations on the received frame. For piconets, a DEV shall reject all frames that do not have an SFC that is strictly greater than the last SFC received from that DEV in that superframe. For pairnets, a DEV shall reject all frames or subframes that do not have a corresponding SFC value that is strictly greater than the last SFC value received from that DEV corresponding to the key identified by the SECID in the received frames, and the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to BAD-SFC and shall not perform any additional operations on the rejected frames or subframes. The last SFC value received shall be only updated after the received integrity code corresponding to the SFC value of the received frame or subframe is successfully verified.

If there are no previous security errors in the processing of the frame, the DEV shall apply the operations defined by the symmetric key security operations to the frame, as defined in 9.3.2 for piconets and 10.3.2 for pairnets. If any of the security operations fail, the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to FAILED-SECURITY-CHECK and shall not perform any additional operations on the received frame. For pairnets, if the integrity code check for a subframe in the received aggregated frame fails, then the MLME shall return an MLME-SECURITY-ERROR.indication to the DME with the ReasonCode set to FAILED-SECURITY-CHECK and shall not perform any additional operations on the subframe with the integrity code check failure or the other subsequent subframes in the aggregated frame. If the security operations have been successfully performed and the frame has been modified appropriately, the DEV may then continue to process the frame.

While operating in mode 1, if the MAC receives a command frame with the SEC field in the Frame Control field set to a value different than expected, as defined in Table 6-47 for piconet and Table 6-48 for pairnet, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

8.3.9 Selecting the SECID for a new key

For each management and data key used in the piconet or pairnet, the key originator in the relationship shall select the 2-octet SECID, as described in 6.2.10.2, that identifies the key.

When a PNC-capable DEV starts a secure piconet, as described in 7.2.3, or a PRC-capable DEV starts a secure pairnet, it shall select a SECID and a symmetric key to be used for beacon protection. Because there are no other DEVs in the piconet or pairnet when the PNC-capable DEV or PRC-capable DEV starts a piconet or pairnet, this key is not distributed to any other DEVs. Once another DEV joins the piconet or the pairnet, the PNC or PRC will update the key and SECID, as indicated in 8.3.4.

8.3.10 Key selection

The key used to protect a particular frame depends on the purpose of the frame and the membership states of the DEV. If the DEV is a member of a secure piconet (i.e., the DEV is the PNC or the DEV is a secure member with the PNC) or if the DEV is a member of a secure pairnet (i.e., the DEV is the PRC or the DEV is a secure member with the PRC), the DEV will have entries for the piconet group data key or the pairnet group data key and for the PNC-DEV management key or the PRC-DEV management key. If the DEV has a secure relationship with a peer-DEV (i.e., the DEV is a secure member with a peer DEV), the DEV will have entries for a peer-to-peer data key and a peer-to-peer management key that it shares with that DEV. For any given frame, the DEV shall either send the frame without security or with the single key that is required for that frame, as indicated in Table 8-1 for piconets and Table 8-2 for pairnets. All secure commands between the PNC and other DEVs shall be protected with the PNC management key or PRC management key. All secure data frames with the PNC as either the DestID or SrcID, all secure broadcast frames, and all secure beacons shall be protected with the piconet group data key or the pairnet group data key. If two DEVs in a secure piconet do not have a peer-to-peer security relationship, they shall use the piconet group data key for commands that are required to be sent securely and they shall use the piconet group data key for secure data frames transmitted between them. Table 8-1 and Table 8-2 summarize the keys that shall be used for each type of frame.

Table 8-1—Key selection for secure frames

Frame type or command	None	PNC-DEV mgmt key	Piconet group data key	Peer-to-peer mgmt key	Peer-to-peer data key	Comment
Beacon frame			X			All secure Beacon frames shall be protected by the piconet group data key.
Imm-ACK frame	X					Immediate acknowledgment frames shall not be secured with any key.
Dly-ACK frame	X					Delayed acknowledgment frames shall not be secured with any key.
Data frame			X		X	Only secure data frames shall be exchanged between DEVs that have a secure relationship. Secure data frames between DEVs that share a peer-to-peer relationship shall use the peer-to-peer data key; otherwise, they shall use the piconet group data key.
Multi-protocol Data frame			X		X	Only secure Multi-protocol Data frames shall be exchanged between DEVs that have a secure relationship. Secure Multi-protocol Data frames between DEVs that share a peer-to-peer relationship shall use the peer-to-peer data key; otherwise, they shall use the piconet group data key.
Announce Response command	X	X	X	X		If the Announce response command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
PM Mode Change Response command		X				
AS IE Request command		X				
AS IE Response command		X				
Multicast Configuration Request command		X				

Table 8-1—Key selection for secure frames (*continued*)

Frame type or command	None	PNC-DEV mgmt key	Piconet group data key	Peer-to-peer mgmt key	Peer-to-peer data key	Comment
Multicast Configuration Response command		X				
Association Request command	X					Association Request commands shall not be secured with any key.
Association Response command	X					Association Response commands shall not be secured with any key.
Disassociation Request command	X	X				Disassociation Request commands shall not be secured with any key before the DEV establishes secure membership in the piconet and shall be protected by the PNC-DEV management key otherwise.
Request Key command		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Request Key Response command		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Distribute Key Request command		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
Distribute Key Response command		X		X		The management key for the relationship (peer-to-peer or PNC-DEV) shall be used for this command.
PNC Handover command		X				
PNC Handover Response command		X				
PNC Handover Information command		X				
PNC Information Request command		X				
PNC Information command		X	X			If the PNC Information command is sent as a directed frame from the PNC to a DEV, the PNC-DEV management key shall be used. If the PNC Information command is sent as a broadcast frame, the piconet group data key shall be used.

Table 8-1—Key selection for secure frames (*continued*)

Frame type or command	None	PNC-DEV mgmt key	Piconet group data key	Peer-to-peer mgmt key	Peer-to-peer data key	Comment
Probe Request command	X	X	X	X		If the Probe Request command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Probe Response command	X	X	X	X		If the Probe Response command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Piconet Services command	X					
Announce	X	X	X	X		If the Announce command is sent to or from the PNC before the DEV becomes a secure member of the piconet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
Channel Time Request command	X	X				If the communicating parties are the PNC and a neighbor PNC, the Channel Time Request command shall not be protected with any key. Otherwise, the PNC-DEV management key shall be used.
Channel Time Response command	X	X				If the communicating parties are the PNC and a neighbor PNC, the Channel Time Response command shall not be protected with any key. Otherwise, the PNC-DEV management key shall be used.

Table 8-1—Key selection for secure frames (*continued*)

Frame type or command	None	PNC-DEV mgmt key	Piconet group data key	Peer-to-peer mgmt key	Peer-to-peer data key	Comment
Channel Status Request command		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key for the relationship (peer-to-peer or PNC-DEV) shall be used.
Channel Status Response command		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key for the relationship (peer-to-peer or PNC-DEV) shall be used.
Remote Scan Request command		X				
Remote Scan Response command		X				
Transmit Power Change command		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.
PM Mode Change command		X				
SPS Configuration Request command		X				
SPS Configuration Response command		X				
PS Set Information Request command		X				
PS Set Information Response command		X				
Security Message	X					
Vendor Defined		X	X	X		If the DEVs do not share an individual relationship, the piconet group data key shall be used. Otherwise, the management key (peer-to-peer or PNC-DEV) for the relationship shall be used.

Table 8-2—Key selection for secure pairnet frames

Frame type or command	None	PRC-DEV mgmt key	Pairnet group data key	Comment
Beacon frame			X	All secure Beacon frames shall be protected by the pairnet group data key.
Stk-ACK frame	X			Stk-ACK frames shall not be secured with any key.
Data frame			X	Only secure data frames shall be exchanged between DEVs that have a secure relationship. The pairnet group data key shall be used for secure data frames between DEVs in pairnet.
Association Request	X			Association Request commands shall not be secured with any key.
Association Response	X			Association Response commands shall not be secured with any key.
Disassociation Request	X	X		Disassociation Request commands shall not be secured with any key before the DEV establishes secure membership in the pairnet and shall be protected by the PRC-DEV management key otherwise.
Request Key		X		The management key for the relationship shall be used for this command.
Request Key Response		X		The management key for the relationship shall be used for this command.
Distribute Key Request		X		The management key for the relationship shall be used for this command.
Distribute Key Response		X		The management key for the relationship shall be used for this command.
Security Information Request		X		
Security Information		X		
Probe Request	X	X	X	If the Probe Request command is sent to or from the PRC before the DEV becomes a secure member of the pairnet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.
Probe Response	X	X	X	If the Probe Request command is sent to or from the PRC before the DEV becomes a secure member of the pairnet, the command shall not be secured by any key. If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.
Transmit Power Change		X	X	If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.

Table 8-2—Key selection for secure pairnet frames (*continued*)

Frame type or command	None	PRC-DEV mgmt key	Pairnet group data key	Comment
Array Training		X	X	If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.
Array Training Feedback		X	X	If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.
Security Message	X			
Vendor Defined		X	X	If the DEVs do not share an individual relationship, the pairnet group data key shall be used. Otherwise, the PRC-DEV management key for the relationship shall be used.

8.4 Protocol details

8.4.1 General

The following protocol details include all cryptographic components and headers for the frames. The headers should be interpreted as being headers in the MAC frames. In addition, each element should be interpreted as specified in Clause 6. Note that all frame transmissions described are sent with the ACK Policy field set to Imm-ACK for piconets and Stk-ACK for pairnets unless specified otherwise. The ACK frames do not affect the security of the protocols and are omitted from all diagrams.

8.4.2 Security information request and distribution

A DEV establishing membership in a security relationship may request or send security information to another DEV. This most often is done directly before or during the PNC handover process, but may be done at any time.

Figure 8-1 illustrates the message flows for Security Information Request and Security Information commands between two peer DEVs.

Figure 8-2 illustrates the message flows for an Security Information Request from the new PNC to the old PNC. This operation is not applicable for pairnets.

Figure 8-3 illustrates the message flows for an Security Information Request from the new PNC to the old PNC. This operation is not applicable for pairnets.

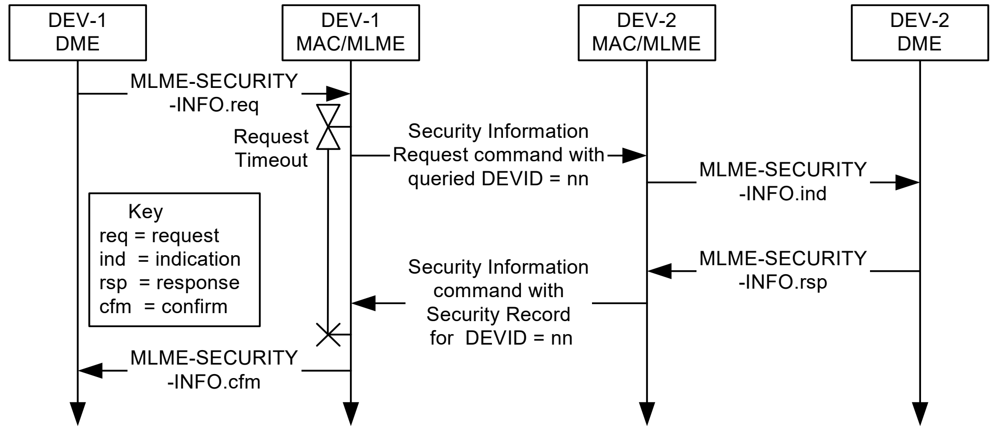

Figure 8-1—MSC for DEV-DEV Security Information request

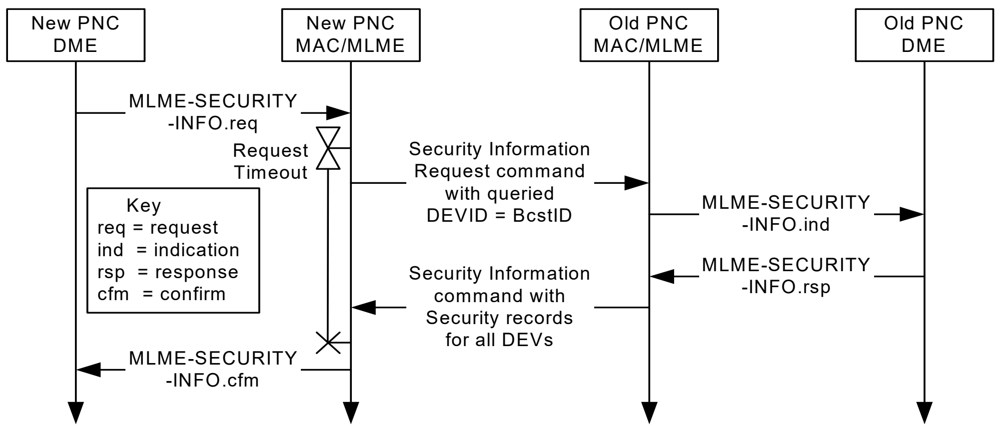

Figure 8-2—MSC for new PNC-old PNC security information transfer

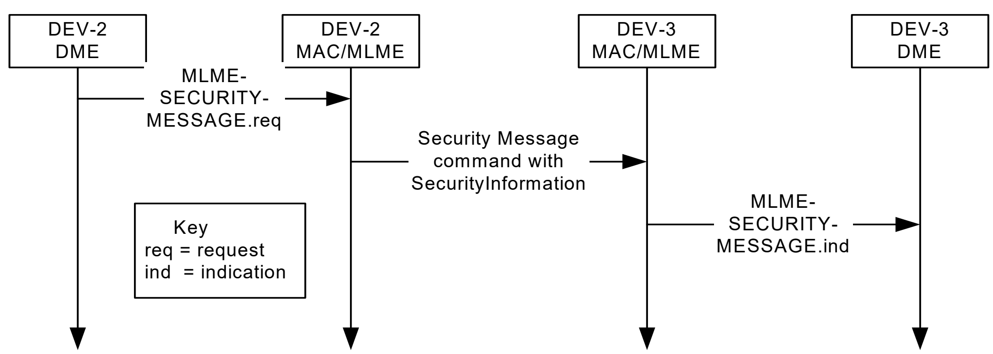

Figure 8-3—MSC for sending security information with the Security Message command

8.4.3 Key distribution protocol

In a secure piconet, in a secure pairnet, or in a secure peer-to-peer relationship, the key originator may wish to update the current data protection key by initiating the key distribution protocol described here. For a change in the piconet group data key or pairnet group data key, the PNC or PRC sends the new piconet group data key or new pairnet group data key to each member of the piconet or pairnet using the Distribute Key Request command. DEVs do not respond to a Distribute Key Request command sent by the PNC or PRC, other than with an Imm-ACK or Stk-ACK if the frame FCS is valid. Note that the Imm-ACK or Stk-ACK does not indicate that the Distribute Key Request command has passed cryptographic verification, only that the FCS was valid. For a change in a peer data key, the key originator in the relationship initiates the key distribution protocol. The key originator sends the Distribute Key Request command to the DEV with which it is updating the key. A DEV that successfully receives a Distribute Key Request command that also passes the data authentication, 11.4.2, shall respond to the key originator with the Distribute Key Response command. The key originator should initiate this protocol with each DEV with their respective shared key whenever the key is updated.

8.4.4 Key request protocol

In a secure piconet or pairnet, if a DEV receives a frame or beacon with an unknown SECID, it may initiate the request key protocol in order to obtain the unknown key from the key originator of the relationship. The DEV initiates the protocol by sending the Request Key command to the key originator. When the key originator receives a Request Key command that has a valid Integrity Code, it checks to see whether it has a secure relationship with the requesting DEV. If there is a secure relationship, the key originator sends the Request Key Response command to the requesting DEV using the management key for that secure relationship.

9. Security specifications for piconets

9.1 Modes for security

When symmetric key security operations are selected, DEVs perform secure operations in mode 1. This mode is defined in 8.2. Symmetric key security operations are not defined for mode 0.

9.2 Symmetric cryptography building blocks

9.2.1 Notational conventions

When transmitting and interpreting security material in this standard, the first byte transmitted shall be the first byte of the security material and represented on the left of the other bytes. The bit ordering within the byte for security operations shall be MSB first and LSB last. The octet ordering shall be most significant octet first and least significant octet last. This ordering shall be irrespective of the transmission order of the bits. See Figure 4-3 for the mapping of bit transmission order to most significant or least significant.

A bit string is a sequence of symbols over the set {0,1}. The notation $x \| y$ is the right-concatenation of two bit strings x and y of length m and n whose result is the string z of length $m + n$ that is constructed with x on its leftmost (most significant) m symbols and with y on its rightmost (least significant) n symbols.

9.2.2 CCM combined encryption and data authentication

The combined symmetric encryption and data authentication mechanisms used in the symmetric key security operations consist of the generation of an integrity code followed by the encryption of plaintext data and the integrity code. The output consists of the encrypted data and the encrypted integrity code.

The symmetric authentication operation consists of the generation of an integrity code using a block cipher in cipher block chaining (CBC) mode computed on a nonce followed by (optional) padded authentication data followed by (optional) padded plaintext data. The verification operation consists of the computation of this integrity code and comparison to the received integrity code.

The symmetric encryption operation consists of the generation of a key stream using a block cipher in counter mode with a given key and nonce and performing an XOR of the key stream with the plaintext and integrity code. The decryption operation consists of the generation of the key stream and the XOR of the key stream with the ciphertext to obtain the plaintext and integrity code.

All of the preceding operations shall be performed as specified in 9.4. The parameters for these operations shall be as specified in 9.2.3.

9.2.3 CCM parameters

The CCM operations shall be parameterized by the following selections: the AES encryption algorithm as specified in 9.2.5, the length in octets of the length field L shall be 2 octets, the length of the authentication field M shall be 8 octets, the nonce shall be formatted as specified in 9.2.4.

9.2.4 Nonce value

In order to preserve the security of the symmetric algorithms, the nonce used for CCM encryption and authentication shall be unique. As a result, the DEV shall not reuse any Secure Frame Counter field value within a single superframe that is intended for a particular DEVID (as this would cause a repeated nonce). This uniqueness is guaranteed by the use of the SrcID, which guarantees that different DEVs sharing the same key will use a different nonce, by the time token, which is different for every superframe with a given key and by the DestID and secure frame counter, which guarantee uniqueness within a superframe as long as a DEV does not send more than 65 536 frames to a particular DestID within that superframe. If a frame is retransmitted and a single bit in the header or Frame Payload field has changed, a new nonce shall be used. To ensure this, each time a frame is retransmitted, the value of the Secure Frame Counter field shall be incremented.

The nonce that is input to the CCM algorithm shall be formatted as illustrated in Figure 9-1.

Octets: 1	**1**	**6**	**2**	**3**
SrcID	DestID	Time Token	Secure Frame Counter	Fragmentation Control Field

Figure 9-1—CCM nonce format

Figure 9-2 specifies the format of the nonce for aggregated frame.

Octets: 1	**1**	**6**	**2**	**1**	**2**
SrcID	DestID	Time Token	Secure Frame Counter	Secure Subframe Counter	Padding

Figure 9-2—CCM nonce format for aggregated frame

The SrcID field, DestID field, Secure Frame Counter field, and Fragmentation Control field shall be set to the values that are included in the frame that is being protected.

The Time Token field shall be set to the value of CurrentTimeToken.

The Secure Subframe Counter field shall be included to generate an unique nonce for each subframe that is aggregated. The counter shall be incremented for each subframe in a frame.

The Padding field shall be set to 0xCFCF.

9.2.5 AES encryption

The AES encryption algorithm used for symmetric key security operations shall be performed as specified in FIPS Pub 197. This encryption algorithm is parameterized by the use of 128-bit keys and 128-bit block size.

9.3 Symmetric cryptography implementation

9.3.1 Symmetric cryptography data formats

Table 9-1 specifies the length and meaning of the symmetric cryptography-related specific data elements from 9.2.2. The operations performed to obtain the variable data values are specified in 9.3.2.

Table 9-1—Symmetric cryptography frame object formats

Notation	Length	Value	Description
Encrypted key	16	Variable	The encrypted key consists of the result of the encryption of a 16-octet key (not including the integrity code) using CCM encryption, as specified in 9.2.2.
Integrity code	8	Variable	The integrity code consists of the encrypted integrity code that is the result of a CCM computation, as specified in 9.2.2, that is computed along with the encrypted seed.
Encrypted data	Variable	Variable	The encrypted data consists of the result of the encryption of the specified data (not including the integrity code) using CCM encryption, as specified in 9.2.2.

9.3.2 Symmetric cryptographic operations

Figure 9-3 specifies the length information and data input to the CCM operation for secure beacons. The Auth Data Length *l(a)* shall be set to the length of all of the Piconet Sync Parameters field plus the sum of the lengths of the IEs that are included in the beacon. The Enc Data Length, *l(m),* shall be set to zero. The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and integrity code.

Octets: 10	2	2	13	L_1	…	L_n
Frame Header	SECID	SFC	Piconet Sync Parameters	IE-1	…	IE-*n*

2	2
Auth Data Length	Enc Data Length

Figure 9-3—CCM input for secure beacons

Figure 9-4 specifies the length information and data input to the CCM operation for secure commands. For all commands except for the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data and the Enc Data Length, *l(m),* shall be set to zero. For the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data minus 16 (the length of the key) and the Enc Data Length, *l(m),* shall be set to 16 (the length of the key). The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS, and integrity code.

Octets: 10	2	2	2	2	L_1	L_2
Frame Header	SECID	SFC	Command Type	Length	Auth Data	Enc Data

2	2
Auth Data Length	Enc Data Length

Figure 9-4—CCM input for secure commands

Figure 9-5 specifies the length information and data input to the CCM operation for Secure Data frames. The Auth Data Length, *l(a)*, shall be set to 14 and the Enc Data Length, *l(m)*, shall be set to the length of the Data Payload field. The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS, and integrity code.

Octets: 10	2	2	L_1
Frame header	SECID	SFC	Data Payload

2	2
Auth Data Length	Enc Data Length

Figure 9-5—CCM input for Secure Data frames

Figure 9-6 specifies the length information and data input to the CCM operation for Secure Multi-Protocol Data frames. The Auth Data Length, *l(a)*, shall be set to 14 and the Enc Data Length, *l(m)*, shall be set to the sum of the lengths of the Data ID field, the Data Header field, and the Data Payload field. The data input to CCM shall be taken in the order it is received in the frame, omitting the HCS, FCS, and integrity code.

Octets: 10	2	2	2	L_1	L_2
Frame header	SECID	SFC	Data ID	Data Header	Data Payload

2	2
Auth Data Length	Enc Data Length

Figure 9-6—CCM input for Secure Multi-Protocol Data frames

9.4 CCM mode

9.4.1 General

CCM is a generic authenticate-and-encrypt block cipher mode. CCM is defined for use with block ciphers with a 128-bit block size, such as AES.

For CCM mode there are two parameter choices to be made (see Table 9-2). The first choice is M, the size of the authentication field. The choice of the value for M involves a trade-off between message expansion and the probability that an attacker will be able to undetectably modify a message. Valid values for M are 4, 6, 8, 10, 12, 14, and 16 octets. The second choice is L, the size of the length field. This value requires a trade-off between the maximum message size and the size of the nonce. Different applications require different trade-offs, so L is a parameter. Valid values for L are 2 to 8 octets (the value $L = 1$ is reserved).

Table 9-2—Parameters of CCM mode

Name	Description	Field size	Encoding of field
M	Number of octets in authentication field	3 bits	$(M–2)/2$
L	Number of octets in length field	3 bits	$L–1$

9.4.2 Inputs

To send a message, the sender shall provide the following information (see Table 9-3):

— An encryption key K suitable for the block cipher.
— A nonce N of $15 – L$ octets. Within the scope of any encryption key K, the nonce value shall be unique. That is, the set of nonce values used with any given key shall not contain any duplicate

values. Using the same nonce for two different messages encrypted with the same key destroys the security properties of this mode.

— The message *m*, consisting of a string of *l(m)* octets where $0 \le l(m) < 2^{8L}$. The length restriction ensures that *l(m)* will be able to be encoded in a field of *L* octets.
— Additional authenticated data *a*, consisting of a string of *l(a)* octets where $0 \le l(a) < 2^{64}$. This additional data is authenticated but not encrypted, and is not included in the output of this mode. It may be used to authenticate plaintext headers or contextual information that affects the interpretation of the message. If there is no additional data to authenticate, the string shall be zero length.

Table 9-3—Inputs for CCM

Name	Description	Field size	Encoding of field
K	Block cipher key	Depends on block cipher	String of octets
N	Nonce	15–*L* octets	Not specified
m	Message to be encrypted and sent	*l(m)* octets	String of octets
a	Additional authenticated data	*l(a)* octets	String of octets

9.4.3 Data authentication

The first step is to compute the authentication field *T*. This is done using cipher block chaining-message authentication code (CBC-MAC). First a sequence of blocks $B_0, B_1, \ldots, B_n$ is defined and then CBC-MAC is applied to these blocks.

The blocks shall be ordered as shown in Figure 9-7.

Octets: 16	16	16 × (*n*–2)	16
B_0	B_1	B_2 to B_{n-1}	B_n

Figure 9-7—Authentication block ordering

The first block B_0 shall be constructed as shown in Equation (9-1):

$$B_0 = \text{Flags} \mathbin{||} \text{Nonce} \mathbin{||} l(m) \qquad (9\text{-}1)$$

The value *l(m)* is 16 bits and is encoded in most-significant-octet first order.

The Flags field is 8 bits and shall be constructed as shown in Equation (9-2):

$$\text{Flags} = \text{Reserved} \mathbin{||} \text{Adata} \mathbin{||} \text{M} \mathbin{||} \text{L} \qquad (9\text{-}2)$$

The *L* field is 3 bits and encodes the size of the length field used to store *l(m)*. The parameter *L* may take on the values from 2 to 8 (the value *L*=1 is reserved). This value is encoded in the field using the values from 1 to 7 by choosing the field value as *L*–1 (the zero value is reserved).

The *M* field is 3 bits encodes the value of *M* as (*M*–2)/2. As *M* may take on the even values from 4 to 16, the field may take on the values from 1 to 7.

The Adata field is 1 bit and is set to zero if *l(a)*=0, and set to one if *l(a)*>0.

The Reserved field is 1 bit and shall be set to zero.

If *l(a)*>0 (as indicated by the Adata field), then one or more blocks of authentication data are added. These blocks contain *l(a)* and *a* encoded in a reversible manner. The string that encodes *l(a)* shall be constructed as follows:

— If $0 < l(a) < 2^{16}–2^8$ then the length field shall be encoded as two octets that contain the value *l(a)* in most-significant-octet first order.
— If $2^{16}–2^8 \le l(a) < 2^{32}$ then the length field shall be encoded as six octets consisting of the octets 0xff, 0xfe, and four octets encoding *l(a)* in most-significant-octet-first order.
— If $2^{32} \le l(a) < 2^{64}$ then the length field shall be encoded as ten octets consisting of the octets 0xff, 0xff, and eight octets encoding *l(a)* in most-significant-octet-first order.

This is summarized in Table 9-4. Note that all fields are interpreted in most-significant-octet first order.

Table 9-4—Length encoding for additional authentication data

First two octets	Followed by	Comment
0x0000		Reserved
0x0001 … 0xFEFF		For $0 < l(a) < 2^{16} – 2^8$
0xFF00 … 0xFFFD		Reserved
0xFFFE	four octets *l(a)*	For $2^{16} – 2^8 \le l(a) < 2^{32}$
0xFFFF	eight octets *l(a)*	For $2^{32} \le l(a) < 2^{64}$

The blocks encoding *a* shall be formed by the string that encodes *l(a)* followed by *a* itself, and splitting the result into 16-octet blocks, padding the last block with zeros if necessary. These blocks shall be appended as the octets following the first block B_0. These blocks, if created, shall be formatted as shown in Figure 9-8, where the length of a is $16 \times (k – 1) – L_1 + L_2$.

Octets: L_2 = 2, 6 or 10	$16 – L_1$	$16 \times (k – 2)$	L_2 = 1 to 16	$16 – L_2$
l(a)	First octets of *a*	Next octets of *a*	Final octets of *a*	0
B_1		B_2 to $B_{k–1}$	B_k	

Figure 9-8—Authentication block ordering for additional authentication blocks

After the (optional) additional authentication blocks have been added, the next step is to form the message blocks. The message blocks are formed by splitting the message *m* into 16-octet blocks, padding the last block with zeros if necessary. If the message *m* consists of the empty string, then no blocks shall be added in this step.

These blocks, if created, shall be formatted as shown in Figure 9-9, where the length of a is $16 \times (k-1) - L_1 + L_2$.

Octets: 16	**$16 \times (n - k - 2)$**	**L_3**	**$16 - L_3$**
First octets of *m*	Next octets of *m*	Final octets of *m*	0
B_{k+1}	B_{k+2} to B_{n-1}	B_n	

Figure 9-9—Authentication block ordering for message blocks

The result is a sequence of blocks B_0, B_1, …, B_n. The CBC-MAC shall be computed by Equation (9-3), Equation (9-4), and Equation (9-5):

$$X_1 = \mathrm{E}(K, B_0) \tag{9-3}$$

$$X_{i+1} = \mathrm{E}(K, X_i \oplus B_i) \text{ for } i = 1, \ldots, n \tag{9-4}$$

$$T = \text{first-}M\text{-octets } (X_{n+1}) \tag{9-5}$$

where E() is the block cipher encryption function and T is the integrity code value. Note that the last block B_n is XORed with X_n and encrypted with the block cipher to give T.

9.4.4 Encryption

The message data shall be encrypted with CTR mode. The key stream blocks are defined by Equation (9-6):

$$S_i = \mathrm{E}(K, A_i) \tag{9-6}$$

for $i = 0, 1, 2, \ldots$.

The values A_i are constructed as shown in Equation (9-7):

$$A_i = \text{Flags} \,||\, \text{Nonce} \,||\, \text{counter } i \tag{9-7}$$

The counter *i* field is 16 bits and is encoded as most-significant-octet first order.

The Flags field is 8 bits and shall be constructed as shown in Equation (9-8):

$$\text{Flags} = \text{Reserved} \,||\, \text{Null} \,||\, L \tag{9-8}$$

The L field is 3 bits and is encoded using the same encoding as in B_0.

The Null field is 3 bits and shall be set to zero. This ensures that all the A blocks are distinct from B_0, which has the non-zero encoding of M in this position.

The Reserved field is 2 bits and shall be set to zero.

Bit 6 corresponds to the Adata field in the B_0 block, but this field is not used here.

The message is encrypted by XORing the octets of message *m* with the first *l(m)* octets of S_1, S_2, S_3, …, S_{n-k} ordered as shown in Figure 9-10. Note that S_0 is not used to encrypt the message.

Octets: 16	**16**	**16 × (*n* – *k* – 2)**	L_3
S_1	S_2	S_3 to S_{n-k-1}	First L_3 octets of S_{n-k}

Figure 9-10—Block ordering for encryption

The authentication value *U* shall be computed by encrypting *T* with the key stream block S_0 and truncating it to the desired length, as shown in Equation (9-9):

$$U = T \oplus \text{first-}M\text{-octets}(S_0) \quad (9\text{-}9)$$

9.4.5 Output

The final result *c* consists of the encrypted message, followed by the encrypted authentication value *U*.

9.4.6 Decryption

To decrypt a message, the following information is required:

— The encryption key *K*
— The nonce *N*
— The additional authenticated data *a*
— The encrypted and authenticated message *c*

Decryption starts by recomputing the key stream to recover the message *m* and the integrity code value *T*. The message and additional authentication data is then used to recompute the CBC-MAC value and check *T*.

If the *T* value is not correct, the receiver shall not reveal any information except for the fact that *T* is incorrect. In particular, the receiver shall not reveal the decrypted message, the value *T*, or any other information.

9.4.7 Restrictions

All implementations shall limit the total amount of data that is encrypted with a single key. The sender shall ensure that the total number of block cipher encryption operations in the CBC-MAC and encryption together shall not exceed 2^{61}. (This allows close to 2^{64} octets to be encrypted and authenticated using CCM.) Receivers that do not expect to decrypt the same message twice may also implement this limit.

The recipient shall verify the CBC-MAC before releasing any information such as the plaintext. If the CBC-MAC verification fails, the receiver shall destroy all information, except for the fact that the CBC-MAC verification failed.

9.4.8 List of symbols

Table 9-5 provides a list of the symbols used for the above specification of CCM.

Table 9-5—List of symbols

Name	Description	Size	Comment
a	Additional authenticated data	$l(a)$ octets	Use empty string if not desired.
A_i	Counter block to generate key stream	16 octets	Contains block counter, nonce, and flags.
B_i	Input block for CBC-MAC	16 octets	Together encode N, L, M, m, and a uniquely.
c	Ciphertext	$l(m) + M$ octets	Includes the encrypted integrity code.
K	Block cipher key	N/A	At least 128 bits, preferably 256 bits.
L	Number of octets in length field	3 bits	Values 1 … 8, encoded in 3 bits as $L - 1$.
m	Message to be encrypted and sent	$l(m)$ octets	Subject to $0 \leq l(m) < 2^{8L}$.
M	Number of octets in authentication field	3 bits	Values 4, 6, 8, …, 16. Encoded value is $(M - 2)/2$.
N	Nonce	15–L octets	Nonce should never be repeated for same key.
S_i	Block of the encryption key stream	16 octets	Use S_0, S_1, S_2, … to encrypt m and T.
T	Unencrypted authentication tag	M octets	
U	Encrypted authentication tag	M octets	Appended as the higher order octets to the message after encryption.
X_i	Intermediate value of CBC-MAC	16 octets	

10. Security specifications for pairnets

10.1 Modes for security

When symmetric key security operations are selected, DEVs perform secure operations in mode 1. This mode is defined in 8.2. Symmetric key security operations are not defined for mode 0.

10.2 Symmetric cryptography building blocks

10.2.1 Notational conventions

When transmitting and interpreting security material in this standard, the first byte transmitted shall be the first byte of the security material and represented on the left of the other bytes. The bit ordering within the byte for security operations shall be MSB first and LSB last. This ordering shall be irrespective of the transmission order of the bits. See Figure 4-3 for the mapping of bit transmission order to most significant or least significant.

10.2.2 Galois/Counter Mode (GCM) combined encryption and data authentication

The security operation for pairnets is based on the GCM mode of the AES encryption algorithm. GCM provides confidentiality, authentication, and integrity for secure frames defined in this standard. The SFC field provides message freshness as a defense against replay attacks. The SFC field and the Time Token field in the secure Beacon frames provide message freshness for the secure Beacon frames. GCM is constructed from a symmetric key block cipher with a block size of 128 bits, such as the AES algorithm. GCM is defined in NIST SP 800-38D.

GCM combines a counter mode, called *GCTR*, for confidentiality and a universal hashing function, called *GHASH*, defined over a binary Galois field for authentication and integrity. GCM is comprised of two functions: an authenticated encryption function that encrypts confidential data and computes an integrity code on both the encrypted data and any additional, selected, unencrypted portion of data, and an authenticated decryption function that decrypts the encrypted data and verifies the integrity code. Each of these functions is relatively efficient and parallelizable. Consequently, high-throughput implementations are possible.

The security operations using GCM shall be performed as specified in 10.4. The parameters for these operations shall be as specified in 10.2.3.

10.2.3 GCM parameters

The GCM operations shall be parameterized by the following selections: the AES encryption algorithm as specified in 10.2.5, the length of the integrity code shall be 16 octets, the length of the GCM nonce shall be 12 octets and it shall be formatted as specified in 10.2.4.

10.2.4 Nonce value

In order to preserve the security of the symmetric algorithms, the nonce used for GCM encryption and authentication shall be unique for a given key. As a result, the DEV shall not reuse any SFC field value with a given key (as this would cause a repeated nonce).

This uniqueness is guaranteed by the use of the DEV address of the source DEV and the SFC. The DEV address is globally unique and guarantees that two different DEVs sharing the same key will use a different nonce. The DEV address of the source DEV and the secure frame counter guarantee uniqueness of the nonce for a given key as long as a DEV does not send more than 2^{48} frames or subframes to the other DEV in the pairnet.

If a frame or a subframe is retransmitted and a single bit in the header or frame body has been changed, a new nonce shall be used. To implement this, each time a frame or subframe is retransmitted, the value of the Secure Frame Counter shall be incremented.

The nonce that is input to the GCM algorithm shall be formatted as illustrated in Figure 10-1.

Octets:6	6
DEV address (source DEV)	Secure Frame Counter (SFC)

Figure 10-1—GCM nonce format

The DEV address field shall be set to the DEV address of the source DEV.

The Secure Frame Counter field is set to the value of the SFC corresponding to the transmitted frame or subframe field that is being protected. The SFC field is defined in 6.2.10.3. If the transmitted frame is an aggregated frame, only the Secure Frame Counter of the first subframe is explicitly included in the aggregated frame, and the Secure Frame Counter value for other subframes shall be incremented for each subframe in the aggregated frame, starting from the value explicitly indicated in the SFC field of the transmitted frame.

NOTE—The value of the Secure Frame Counter field is independent from the value of the sequence number in the MAC header and they do not need to match.

10.2.5 AES encryption

The AES encryption algorithm used for symmetric key security operations shall be performed as specified in FIPS Pub 197. This encryption algorithm is parameterized by the use of 128-bit keys and 128-bit block size. Only AES-128 GCM shall be used for pairnets.

10.3 Symmetric cryptography implementation

10.3.1 Symmetric cryptography data formats

Table 10-1 specifies the length and meaning of the symmetric cryptography-related specific data elements from Clause 6. The operations performed to obtain the variable data values are specified in 10.3.2.

Table 10-1—Symmetric cryptography frame object formats for GCM

Notation	Length	Value	Description
Encrypted key (see NOTE)	16	Variable	The encrypted key consists of the result of the encryption of a 16-octet key (not including the integrity code) using GCM encryption, as specified in 10.2.2.
Integrity code	16	Variable	The integrity code consists of the encrypted integrity code that is the result of a GCM computation, as specified in 10.2.2, that is computed along with the encrypted seed.
Encrypted data	Variable	Variable	The encrypted data consists of the result of the encryption of the specified data (not including the integrity code) using GCM encryption, as specified in 10.2.2.
NOTE—Encrypting a key with GCM requires a unique nonce. The key is transmitted in secure command frames protected using a management key. And the Nonce defined in 10.2.4, which is guaranteed to be unique, is used for secure command frames. A group data key and a management key may use separate Secure Frame Counter.			

10.3.2 Symmetric cryptographic operations

Figure 10-2 specifies the length information and data input to the GCM operation for secure Beacon frames. The Auth Data Length in octets, *l(a),* shall be set to the length of the Frame Header, SECID, SFC, Time Token, all of the Pairnet Synchronization Parameters field plus the sum of the lengths of the IEs that are included in the Beacon frame. The Enc Data Length in octets, *l(p)*, shall be set to zero. The data input to GCM shall be taken in the order it is received in the frame, omitting the HCS, FCS and Integrity Code.

Octets: 10	2	6	6	15	L_1	…	L_n
Frame Header	SECID	SFC	Time Token	Pairnet Synchronization Parameters	IE-1	…	IE-*n*

2	2
Auth Data Length	Enc Data Length

Figure 10-2—GCM input for secure Beacon frames

Figure 10-3 specifies the length information and data input to the GCM operation for secure commands. For all commands except for the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data including Frame Header, SECID, SFC, MAC Subheader, Command Type and Length plus the length of the Payload field in the command frame. The Enc Data Length, *l(p)*, shall be set to zero. For the Request Key Response command and Distribute Key Request command, the Auth Data Length, *l(a)*, shall be set to the length of all of the protected data minus 16 (the length of the key) and the Enc Data Length, *l(p)*, shall be set to 16 (the length of the key). The data input to GCM shall be taken in the order it is received in the frame, omitting the HCS for the Frame Header, FCS and Integrity Code.

Octets: 10	2	6	4	2	2	L_1	L_2
Frame Header	SECID	SFC	MAC Subheader	Command Type	Length	Auth Data	Enc Data

2	2
Auth Data Length	Enc Data Length

Figure 10-3—GCM input for secure commands

Figure 10-4 specifies the length information and data input to the GCM operation for secure data frames. The GCM operation is applied to each subframe in the data frame separately. For the first subframe, the Auth Data Length 1, $l_1(a)$, which is the Auth Data Length for the first subframe, shall be set to 22, which is the length of the Frame Header, SECID, SFC, and the MAC Subheader of the first subframe, and the Enc Data Length 1, $l_1\ (p)$, which is the Enc Data Length for the first subframe, shall be set to the length of the Payload field in the first subframe.

For the *n*-th subframe, the Auth Data Length *n*, $l_n(a)$, which is the Auth Data Length for the *n*-th subframe, shall be set to 4, which is the length of the MAC Subheader of the *n*-th subframe, and the Enc Data Length *n*, $l_n\ (p)$, which is the Enc Data Length for the *n*-th subframe, shall be set to the length of the Payload field in the *n*-th subframe.

The data input to GCM for each subframe shall be taken in the order it is received in the frame, omitting the FCS, Integrity Code, and Padding in the subframe. The HCS for the Frame Header shall be also omitted.

Octets: 10	**2**	**6**	**4**	$\mathbf{L_1}$
Frame Header	SECID	SFC	MAC Subheader 1	Payload 1

2	**2**
Auth Data Length 1	Enc Data Length 1

……………

Octets: 4	$\mathbf{L}_{\boldsymbol{n}}$
MAC Subheader *n*	Payload *n*

2	**2**
Auth Data Length *n*	Enc Data Length *n*

Figure 10-4—GCM input for secure data frames

Figure 10.4 specifies the length information and data input to the GCM operation for Secure Multi-Protocol Data frames. The GCM operation is applied to each subframe in the data frame separately. For the first subframe, the Auth Data Length 1, $l_1(a)$, which is the Auth Data Length for the first subframe, shall be set to 22, which is the length of the Frame Header, SECID, SFC, and the MAC Subheader of the first subframe, and the Enc Data Length 1, $l_1\ (p)$, which is the Enc Data Length for the first subframe, shall be set to the sum of the lengths of the Data ID field, Data Header field, and Data Payload field in the first subframe.

For the *n*-th subframe, the Auth Data Length *n*, $l_n(a)$, which is the Auth Data Length for the *n*-th subframe, shall be set to 4, which is the length of the MAC Subheader of the *n*-th subframe, and the Enc Data Length *n*, $l_n\ (p)$, which is the Enc Data Length for the *n*-th subframe, shall be set to the sum of the lengths of the Data ID field, Data Header field, and Data Payload field in the *n*-th subframe.

The data input to GCM for each subframe shall be taken in the order it is received in the frame, omitting the FCS, Integrity Code, and Padding in the subframe. The HCS for the Frame Header shall be also omitted.

Octets: 10	**2**	**6**	**4**	**L_1**			**2**	**2**
Frame Header	SECID	SFC	MAC Subheader 1	Data ID	Data Header	Data Payload 1	Auth Data Length 1	Enc Data Length 1

…………..

Octets: 4	**L_n**			**2**	**2**
MAC Subheader *n*	Data ID	Data Header	Data Payload *n*	Auth Data Length *n*	Enc Data Length *n*

Figure 10-5—GCM input for Secure Multi-Protocol Data frames

10.4 GCM mode

10.4.1 General

GCM is a generic authenticate-and-encrypt block cipher mode. GCM is defined for use with block ciphers with a block size of 128 bits, such as AES. The GCM parameters for pairnets are specified in 10.2.3.

10.4.2 Inputs for authenticated encryption

To send a message, the sender shall provide the following information (see Table 10-2):

— An encryption key K of 16 octets suitable for the block cipher.

— A nonce N of 12 octets. Within the scope of any encryption key K, the nonce value shall be unique. That is, the set of nonce values used with any given key shall not contain any duplicate values. Using the same nonce for two different messages encrypted with the same key destroys the security properties of this mode. The nonce value specified in 10.2.4 shall be used.

— The data to be encrypted, p, consisting of a string of $l(p)$ octets where $0 \leq l(p) \leq 2^{36} - 32$. If there is no data to be encrypted in the message, the string shall be zero length. The inputs for encryption are defined in 10.3.2.

— Additional authenticated data (AAD), a, consisting of a string of $l(a)$ octets where $0 \leq l(a) < 2^{61}$. This additional data is authenticated but not encrypted, and is not included in the output of this mode. It may be used to authenticate plaintext headers or contextual information that affects the interpretation of the message. If there is no additional data to authenticate, the string shall be zero length. The inputs for additional authenticated data are defined in 10.3.2.

The bit lengths of N, p, and a shall be multiples of 8, so that these values are octet strings.

NOTE—The maximum subframe size in pairnets meets the length requirement of m and a.

Table 10-2—Inputs for GCM

Name	Description	Field size	Encoding of field
K	Block cipher key	16 octets	String of octets
N	Nonce	12 octets	Not specified
p	Data to be encrypted	$l(p)$ octets	String of octets
a	AAD	$l(a)$ octets	String of octets

10.4.3 Authenticated encryption

The inputs for encryption that are defined in 10.3.2 shall be divided into 16-octet message blocks as shown in Figure 10-6. The blocks are ordered in the same order they are received in the frame, from P_1 to P^*_n.

The message block consists of n blocks, where n = CEIL[$l(p)$ /16].

The bit length u of the last block P^*_n may be less than 128 bits and the relationship between $l(p)$, n, and u is shown in Equation (10-1):

$$l(p)*8 = (n - 1)*128 + u, \text{ where } 1 \leq u \leq 128 \tag{10-1}$$

Octets: 16	…	16	1 –16
P_1	….	P_{n-1}	P^*_n

Figure 10-6—Block ordering for encryption

There is no need to pad the input to meet the 16 octet boundary.

The corresponding ciphertext blocks to each message block are denoted as $C_1, C_2, \ldots, C_{n-1}, C^*_n$, where the number of bits in the last block C^*_n is u.

Similarly, the inputs for AAD that are defined in 10.3.2 shall be divided into 16-octet message blocks as shown in Figure 10-7. The blocks are ordered in the same order they are received in the frame, from A_1 to A^*_m.

The message block consists of m blocks, where m = CEIL[$l(a)$ /16].

The bit length v of the last block A^*_m may be less than 128 bits and the relationship between $l(a)$, m, and v is shown in Equation (10-2):

$$l(a)*8 = (m - 1)*128 + v, \text{ where } 1 \leq v \leq 128 \tag{10-2}$$

Octets: 16	…	16	1 – 16
A_1	….	$A_{m\text{-}1}$	A^*_m

Figure 10-7—Block ordering for AAD

The procedure defined in NIST SP 800-38D shall be used for authenticated encryption. The procedure described below is provided as an informative overview of the authenticated encryption procedure.

The two main functions used in GCM are block cipher encryption and multiplication over the field $\mathrm{GF}(2^{128})$. The following notation and parameters are used for specifying the GCM operation:

— The block cipher encryption of the value X with the key K is denoted as $E(K, X)$.

— The multiplication of two elements $X, Y \in GF(2^{128})$ is denoted as $X \cdot Y$. GCM shall use the polynomial shown in Equation (10-3):

$$f = 1 + \alpha + \alpha^2 + \alpha^7 + \alpha^{128} \tag{10-3}$$

— The addition of X and Y is denoted as $X \oplus Y$. Addition in this field is equivalent to the bitwise exclusive-or operation.

— The convention for interpreting strings as polynomials is “little endian”. That is, if α is the variable of the polynomial, then the block $x_0x_1...x_{127}$ corresponds to the polynomial shown in Equation (10-4):

$$x_0 + x_1\,\alpha + x_2\,\alpha^2 + ... + x_{127}\,\alpha^{127} \tag{10-4}$$

The function len() returns a 64-bit string containing the nonnegative integer describing the number of bits in its argument, with the LSB on the right.

— The expression 0^l denotes a string of l zero bits.

— $A||B$ denotes the concatenation of two bit strings A and B.

— The function $\mathrm{MSB}_t(S)$ returns the bit string containing only the most significant (leftmost) t bits of S

— The symbol {} denotes the bit string with zero length.

— GHASH is a keyed hash function using the hash subkey, denoted H, which shall be generated by applying the block cipher to the “zero” block.

— IV is the initialization vector and the 12 octet nonce N defined in 10.2.4 shall be used as IV. IV is used for generating the initial counter value, denoted Y_0.

— The function incr() treats the rightmost 32 bits of its argument as a nonnegative integer with the least significant bit on the right, and increments this value modulo 2^{32}. That is, the value of incr($F||I$) is $F||(I + 1 \bmod 2^{32})$. incr() is used for generating successive counter values, denoted Y_i.

The authenticated encryption shall be processed as shown in Equation (10-5) through Equation (10-10):

$$H = E(K, 0^{128}) \tag{10-5}$$

$$Y_0 = IV \,\|\, 0^{31}1 \tag{10-6}$$

$$Y_i = \mathrm{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \tag{10-7}$$

$$C_i = P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n \tag{10-8}$$

$$C_n^* = P_n^* \oplus \mathrm{MSB}_u(E(K, Y_n)) \tag{10-9}$$

$$T = \mathrm{MSB}_t[\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0)] \tag{10-10}$$

T is the t-bit length Integrity Code. The value of t shall be fixed to 128 in this standard to use the 16 octet length Integrity Code.

The function GHASH is defined by $\mathrm{GHASH}(H, A, C) = X_{m+n+1}$, where the A is the sequence of blocks $A_1, A_2, .., A^*_m$ and the C is the sequence of blocks $C_1, C_2, .., C^*_n$.

The variables X_i for $i = 0, ..., m + n + 1$ are defined as shown in Equation (10-11) through Equation (10-16):

$$X_i = 0 \text{ for } i = 0 \tag{10-11}$$

$$X_i = (X_{i-1} \oplus A_i) \cdot H \text{ for } i = 1, \ldots, m-1 \tag{10-12}$$

$$X_i = (X_{m-1} \oplus (A_m^* \| 0^{128-v})) \cdot H \text{ for } i = m \tag{10-13}$$

$$X_i = (X_{i-1} \oplus C_{i-m}) \cdot H \text{ for } i = m+1, \ldots, m+n-1 \tag{10-14}$$

$$X_i = (X_{m+n-1} \oplus (C_n^* \| 0^{128-u})) \cdot H \text{ for } i = m+n \tag{10-15}$$

$$X_i = [X_{m+n} \oplus (\mathrm{len}(A) \| \mathrm{len}(C))] \cdot H \text{ for } i = m+n+1 \tag{10-16}$$

10.4.4 Outputs from authenticated encryption

There are two outputs from the authenticated encryption processing:

- A ciphertext c, which is the sequence of blocks $C_1, C_2, .., C^*_n$, whose length is exactly that of the plaintext p.
- An Integrity Code T, whose length is 16 octets.

10.4.5 Inputs for authenticated decryption

For authenticated decryption process, the following information is required:

- The encryption key K of 16 octets.
- The nonce N which is used as the IV whose length is 12 octets.
- The additional authenticated data a.
- The encrypted and authenticated message c.
- The Integrity Code T, whose length is 16 octets.

The received secure frame is parsed to construct these inputs except the encryption key K.

10.4.6 Authenticated decryption

The authenticated decryption operation is similar to the authenticated encryption operation, but with the order of the hash step and encrypt step reversed. The procedure defined in NIST SP 800-38D shall be used for authenticated decryption. The procedure described as follows is provided as an informative overview of the authenticated decryption procedure.

The authenticated decryption shall be processed as shown in Equation (10-17) through Equation (10-22):

$$H = E(K, 0^{128}) \tag{10-17}$$

$$Y_0 = IV \| 0^{31}1 \tag{10-18}$$

$$T' = \mathrm{MSB}_t[\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0)] \tag{10-19}$$

$$Y_i = \mathrm{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n \tag{10-20}$$

$$P_i = C_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n \tag{10-21}$$

$$P_n^* = C_n^* \oplus \mathrm{MSB}_u[E(K, Y_n)] \tag{10-22}$$

The tag *T'* that is computed by the decryption operation is compared to the Integrity Code *T* in the received secure frame associated with the ciphertext *C*. If the two values match, then the ciphertext is returned. Otherwise, the receiver shall not reveal any information except for the fact that the Integrity Code *T* is incorrect. In particular, the receiver shall not reveal the decrypted message, the value *T*, or any other information.

10.4.7 Restrictions

The sender shall make sure that the total number of invocations of the authenticated encryption function using a given key shall not exceed 2^{48}. Receivers that do not expect to decrypt the same message twice may also implement this limit.

The recipient shall verify the Integrity Code before releasing any information such as the plaintext. If the Integrity Code verification fails, the receiver shall destroy all information, except for the fact that the Integrity Code verification failed.

The recipient shall use the Time Token and SFC in the received Beacon frame to detect replay attacks on the Beacon frame and check beacon freshness. To detect replay attacks on other frames, the recipient shall use the SFC in the received frame. The recipient shall discard the received frame if the replay attack is detected.

10.4.8 List of symbols

Table 10-3 provides a list of the symbols used for the above specification of GCM.

Table 10-3—List of symbols

Name	Description	Size	Comment
a	Additional authenticated data (AAD)	$l(a)$ octets	Use empty string if not desired.
A_i	Input block for AAD	16 octets	The last block may be less than 16 octets.
p	Data to be encrypted	$l(p)$ octets	Use empty string if not desired.
P_i	Input block for encryption	16 octets	The last block may be less than 16 octets.
K	Block cipher key	16 octets	—
N	Nonce	12 octets	Nonce should never be repeated for the same key.
c	Ciphertext	$l(p)$ octets	The length is exactly that of the plaintext p.
C_i	Ciphertext block	16 octets	The last block may be less than 16 octets.
T	Integrity Code	16 octets	—

11. PHY specification for the 2.4 GHz band

11.1 Overview of the 2.4 GHz PHY

This clause specifies the PHY for a single carrier system that supports up to five modulation formats with coding at 11 Mbaud to achieve scalable data rates. The formats, coding, and data rates are given in Table 11-1.

Table 11-1—Modulation, coding, and data rates for 2.4 GHz PHY

Modulation type	Coding	Data rate
QPSK	8-state TCM	11 Mb/s
DQPSK	none	22 Mb/s
16-QAM	8-state TCM	33 Mb/s
32-QAM	8-state TCM	44 Mb/s
64-QAM	8-state TCM	55 Mb/s

This standard is based on the established regulations for Europe, Japan, Canada and the United States. The regulatory documents listed below are for information only and are subject to change or revision at any time. The regulatory domains are enumerated in a vector called *phyRegDomainsSupported*, specified in 11.7, and are indicated by the parameter *phyCurrentRegDomain*.

Europe (except France and Spain):

Approval standards: European Telecommunications Standards Institute (ETSI)
Documents: ETS 300-328 [B1], ETS 300-826
Approval authority: National type approval authorities

Japan:

Approval standards: Association of Radio Industries and Businesses (ARIB)
Document: ARIB STD-T66
Approval authority: Ministry of Post and Telecommunications (MPT)

United States:

Approval standards: Federal Communications Commission, USA
Documents: 47 CFR, Part 15, Sections 15.205, 15.209, 15.249

Canada:

Approval standards: Industry Canada, Canada
Document: GL36

11.2 General requirements

11.2.1 Operating frequency range

This PHY operates in the 2.4 GHz to 2.4835 GHz frequency range as allocated by the regulatory agencies in Europe, Japan, Canada, and the United States as well as any other areas where the regulatory bodies have allocated this band.

11.2.2 Radio frequency (RF) power measurements

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

11.2.3 Channel assignments

A total of 5 channels in two sets are assigned for operation. The first set is the high-density mode, which allocates 4 channels while the second is an IEEE 802.11b co-existence mode, which allocates 3 channels. Since the two outer channels of the sets overlap, there are a total of 5 channels allowed for operation. The assigned channels are shown in Table 11-2. A compliant IEEE 802.15.3 implementation shall support all 5 channels.

Table 11-2—2.4 GHz PHY channel plan

CHNL_ID	Center frequency	High-density	IEEE 802.11b coexistence
1	2.412 GHz	X	X
2	2.428 GHz	X	
3	2.437 GHz		X
4	2.445 GHz	X	
5	2.462 GHz	X	X

The *phyCurrentChannel* is the CHNL_ID of the current channel. For the purpose of the Remote Scan Request and Remote Scan Response commands, as described in 6.5.8.4 and 6.5.8.5, respectively, the Channel Index field is the CHNL_ID in Table 11-2.

11.2.4 Scanning channels

A DEV may, in the course of a scan, change to an IEEE 802.11b channel for the purpose of detecting the presence of IEEE 802.11b networks.

When a DEV is scanning to start a piconet, it should scan all five channels to decrease the probability of choosing an occupied channel.

If a DEV is capable of identifying an IEEE 802.11b network and it does identify an IEEE 802.11b network while scanning, it should use the IEEE 802.11b coexistence channel set. It should also rate the channels where IEEE 802.11b networks were identified as the worst channels. If multiple IEEE 802.11b networks are detected, the DEV should order them based on an estimate of the amount of traffic and the power level in the channel.

11.2.5 Unwanted emissions

Conformant implementations shall comply with the in-band and out-of-band emissions for all operational modes as set by the applicable regulatory bodies.

11.2.6 Operating temperature range

A conformant implementation shall meet all of the specifications in this standard for ambient temperatures from 0 °C to 40 °C.

11.2.7 PHY layer timing

11.2.7.1 General

The values for the PHY layer timing parameters are defined Table 11-3.

Table 11-3—PHY layer timing parameters

PHY parameter	Value	Subclause
pMifsTime	2 µs	11.2.7.5
pSifsTime	10 µs ± 0.5 µs	11.2.7.3
pCcaDetectTime	5 × 16/11 µs	11.6.5
pChannelSwitchTime	500 µs	11.2.7.6

11.2.7.2 Interframe space

A conformant implementation shall support the IFS parameters, as described in 7.6.2, given in Table 11-4.

Table 11-4—IFS parameters

MAC parameter	Corresponding PHY parameter	Definition
MIFS	*pMifsTime*	11.2.7.5
SIFS	*pSifsTime*	11.2.7.3
pBackoffSlot	*pSifsTime* + *pCcaDetectTime*	11.6.5
BIFS	*pSifsTime* + *pCcaDetectTime*	11.2.7.3, 11.6.5
RIFS	2 × *pSifsTime* + *pCcaDetectTime*	11.2.7.3, 11.6.5

11.2.7.3 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be *pSifsTime*, including the power-up ramp specified in 11.5.7.

The RX-to-TX turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

11.2.7.4 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than *pSifsTime*, including the power-down ramp specified in 11.5.7.

The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol until the receiver is ready to begin the reception of the next PHY frame.

11.2.7.5 Time between successive transmissions

The minimum time between successive transmissions shall be *pMifsTime*, including the power-up ramp specified in 11.5.7.

The *pMifsTime* shall be measured at the air interface from the trailing edge of the last symbol transmitted until the first symbol of the PHY preamble is present at the air interface.

11.2.7.6 Channel switch time

The channel switch time is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pChannelSwitchTime*.

11.2.8 Data size restrictions

11.2.8.1 General

The PHY definitions create restrictions on the maximum frame size, maximum transfer unit size, and minimum fragmentation size that will be supported.

11.2.8.2 Maximum frame length

The maximum frame length allowed, *pMaxFrameBodySize*, shall be 2048 octets. This total includes the frame payload and FCS but not the PHY preamble, PHY header, MAC header, or HCS. The maximum frame length also does not include the tail symbols, as described in 11.4.7, or the stuff bits, as described in 11.4.6.

11.2.8.3 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 2044 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2044 octets minus the security overhead, as defined in 6.3.5.2.

11.2.8.4 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, that is allowed with the 2.4 GHz PHY shall be 64 octets.

11.2.9 Header check sequence

The combined PHY and MAC headers shall be protected with a CCITT CRC-16 header check sequence (HCS). The value *pLengthHcs* shall be 2 for this PHY. The CCITT CRC-16 HCS shall be the one's complement of the remainder generated by the modulo-2 division of the protected combined PHY and MAC headers by the polynomial shown in Equation (11-1):

$$x^{16}+x^{12}+x^5+1 \tag{11-1}$$

The protected bits shall be processed in transmit order. All HCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 11-1.

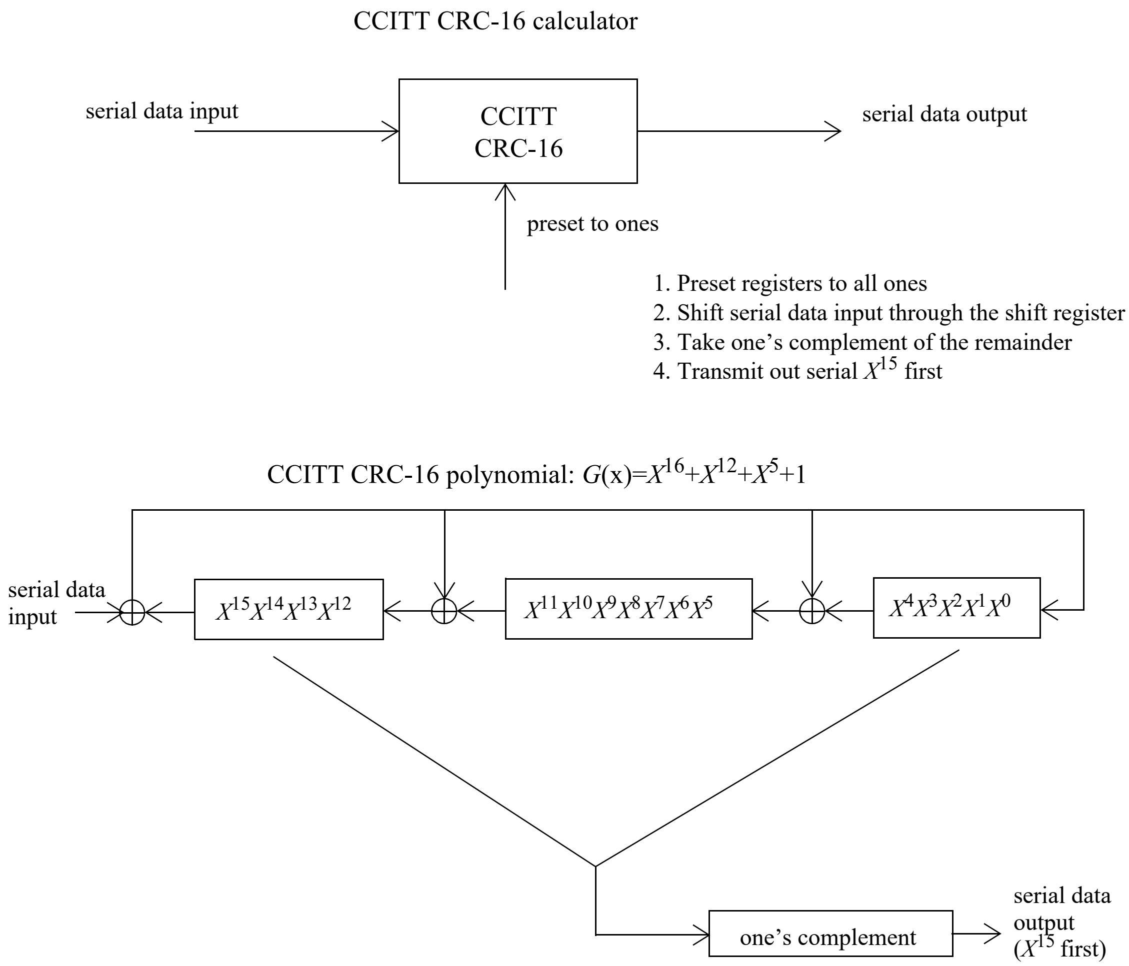

Figure 11-1—CCITT CRC-16 Implementation

As an example, consider the following 32-bit length sequence to be protected by the CRC-16:

0101 0000 0000 0000 0000 0011 0000 0000
b0...b31

The leftmost bit (b0) is transmitted first in time.

The one's complement HCS for this sequence would be the following:

0101 1011 0101 0111
b0.............................b15

The leftmost bit (b0) is transmitted first in time. Bit b0 corresponds to X^{15} in the Figure 11-1.

An illustrative example of the CCITT CRC-16 HCS using the information from Figure 11-1 is shown in Figure 11-2.

Data	CRC Registers MSB LSB	
	1111111111111111	; Initialize preset to ones
0	1110111111011111	
1	1101111110111110	
0	1010111101011101	
1	0101111010111010	
0	1011110101110100	
0	0110101011001001	
0	1101010110010010	
0	1011101100000101	
0	0110011000101011	
0	1100110001010110	
0	1000100010001101	
0	0000000100111011	
0	0000001001110110	
0	0000010011101100	
0	0000100111011000	
0	0001001110110000	
0	0010011101100000	
0	0100111011000000	
0	1001110110000000	
0	0010101100100001	
0	0101011001000010	
0	1010110010000100	
1	0101100100001000	
1	1010001000110001	
0	0101010001000011	
0	1010100010000110	
0	0100000100101101	
0	1000001001011010	
0	0001010010010101	
0	0010100100101010	
0	0101001001010100	
0	1010010010101000	

Figure 11-2—Example of CRC calculation

The CRC-16 described in this subclause is the same one used in IEEE Std 802.11b™-1999 [B3].

11.2.10 Channel access methods

A PNC-capable DEV compliant to this standard shall allow the use of the CAP for contention based access for association, data, and commands, as described in 6.3.1, when using the 2.4 GHz PHY. A DEV that is compliant to this standard shall support the use of the CAP when using the 2.4 GHz PHY.

11.3 Modulation and coding

11.3.1 General

The 2.4 GHz PHY uses uncoded DQPSK, uncoded QPSK, and 16/32/64-QAM with trellis coding (see Ungerboeck [B10]). A 2.4 GHz DEV shall, at a minimum, support DQPSK modulation. In addition, if a 2.4 GHz DEV supports a given modulation format other than DQPSK, it shall also support all of the lower modulation formats. For example, if an IEEE 802.15.3 implementation supports 32-QAM, it shall also support 16-QAM and QPSK-TCM as well as the DQPSK modulation formats.

The symbol rate for all modulations shall be 11 Mbaud. Based on this symbol rate and the coding, the raw PHY data rates supported are 11 Mb/s, 22 Mb/s, 33 Mb/s, 44 Mb/s, 55 Mb/s (QPSK-TCM, DQPSK, 16/32/64-QAM –TCM, respectively) as shown in Table 11-1. The data rates are respectively the entries to the *phyDataRateVector*, as described in 11.7.

11.3.2 Base data rate

The base data rate of the 2.4 GHz PHY shall be 22 Mb/s operating in the uncoded DQPSK mode.

The DQPSK mode is used as a base rate instead of the 11 Mb/s QPSK-TCM mode to reduce the overhead due to the duration of the PHY and MAC headers. Also, DQPSK capability is necessary to implement the PHY preamble, 11.4.2.

The QPSK-TCM mode is implemented in assigned CTAs to help maintain connections of DEVs that are in range of the PNC, but which may be more distant from each other.

11.3.3 Signal constellations

Figure 11-3 illustrates the signal constellations used in encoding bit streams into discrete signal levels sent through the common air interface.

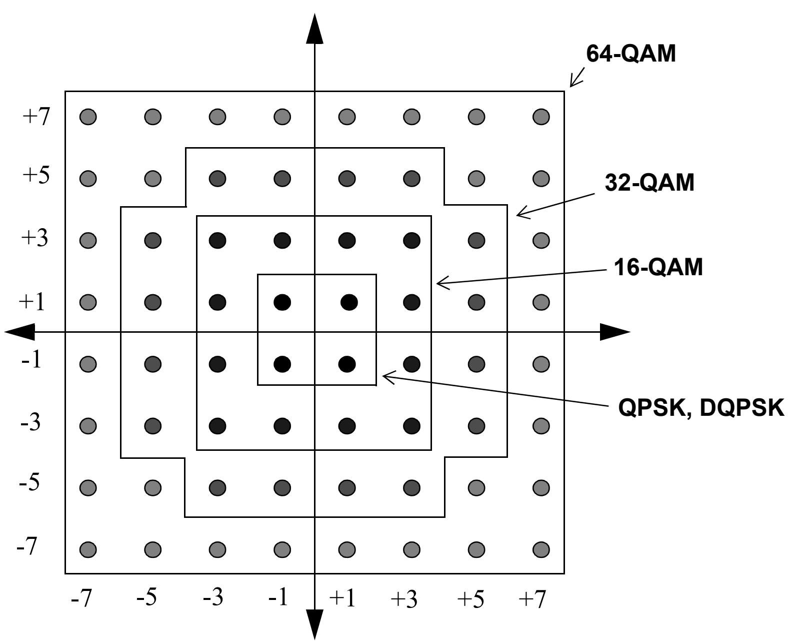

Figure 11-3—DQPSK, QPSK, 16/32/64-QAM signal constellations

The average power within a frame including the PHY preamble and header is required to be a constant, regardless of the modulation format. Thus, a conformant implementation shall scale the constellation such that the PHY header and the MPDU have the same average power. One method to calculate the normalization is as follows: The output values, d, are formed by multiplying the resulting $(I + jQ)$ value by a normalization factor, K_{MOD}, as shown in Equation (11-2):

$$d = (I + jQ) \times K_{MOD} \tag{11-2}$$

The normalization factor, K_{MOD}, depends on the base modulation mode and is given in for each of the modulation formats in Table 11-5. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor may be used, as long as the DEV conforms with the modulation accuracy requirements described in 11.5.2.

Table 11-5—Normalization factor for PHY modulation formats

Modulation	K_{MOD}
DQPSK	1
QPSK-TCM	1
16-QAM-TCM	$1/(\sqrt{5})$
32-QAM-TCM	$1/(\sqrt{10})$
64-QAM-TCM	$1/(\sqrt{21})$

11.3.4 DQPSK modulation

No coding shall be applied to the DQPSK modulation. The mapping of the bit pairs to DQPSK symbols shall be implemented as specified in Table 11-6. In Table 11-6, a "+jω" phase change shall be defined as a counterclockwise rotation. The differential encoding shall apply only to the DQPSK mode. In this mode the entire frame, with the exception of the PHY preamble, shall be encoded differentially. The phase change of the first symbol is determined relative to the phase of the last symbol in the CAZAC sequence, as described in 11.4.2.

The differential encoding is provided to allow for non-coherent receiver implementations.

Table 11-6—DQPSK encoding table

Bit pattern (d0,d1) d0 is first in time	Phase change (+jω)
0,0	0
0,1	π/2
1,1	π
1,0	3π/2 (–π/2)

11.3.5 QPSK and 16/32/64-QAM with trellis coding

The QPSK and 16/32/64-QAM formats shall use an 8-state two-dimensional (2-D) trellis code. The encoder shown in Figure 11-4 shall be used in implementing the 8-state 2-D trellis code.

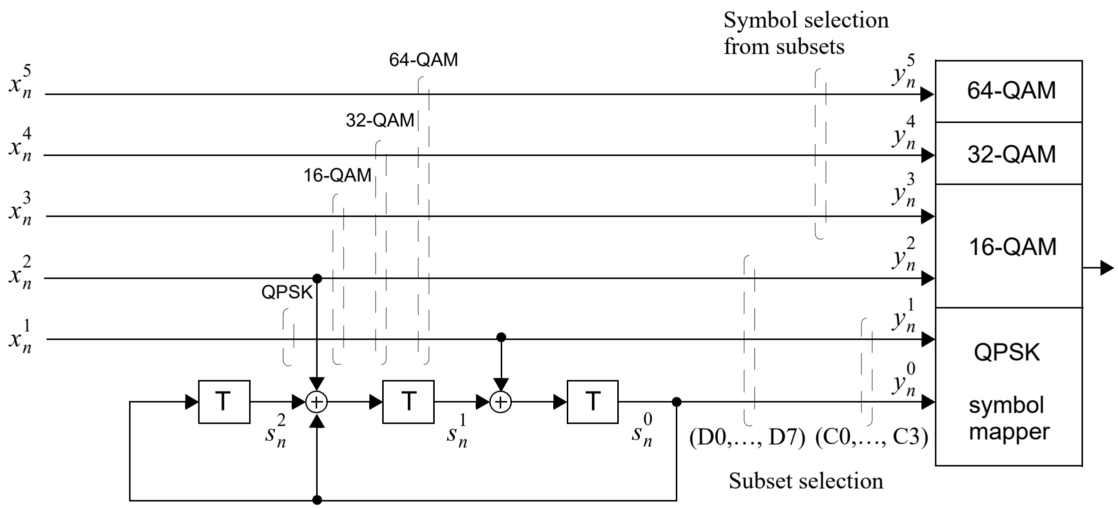

Figure 11-4—QPSK, 16/32/64-QAM 8-state trellis encoder

The pair $([x_n^5, x_n^4, x_n^3, x_n^2, x_n^1],[y_n^5, y_n^4, y_n^3, y_n^2, y_n^1, y_n^0])$ represents the input and output relationship of the trellis encoder, where 5 input bits per symbol interval are encoded into 6 output bits for 64-QAM symbol mapping. Likewise, the input/output pair $([x_n^4, x_n^3, x_n^2, x_n^1],[y_n^4, y_n^3, y_n^2, y_n^1, y_n^0])$ represents the 32-QAM symbol mapping case, where 4 input bits per symbol interval are encoded into 5 output bits. Similarly, for the case of 16-QAM symbol mapping, 3 input bits per symbol interval are encoded into 4 output bits as denoted by the input/output pair $([x_n^3, x_n^2, x_n^1],[y_n^3, y_n^2, y_n^1, y_n^0])$. Finally, for the QPSK symbol mapping, 1 input bit per symbol interval is encoded into 2 output bits as denoted by the input/output pair $([x_n^1],[y_n^1, y_n^0])$. The 8 states generated by the trellis encoder shown in Figure 11-4 are denoted as S0, S1, S2,…, S7, which correspond to the binary representations of $s_n^2 s_n^1 s_n^0$. The input-output relations and state transitions of the trellis encoder shown above is given in Table 11-7 for the QPSK modulation format and in Table 11-8 for the 16/32/64-QAM formats. Trellis symbol subsets C0, C1, C2, C3 for the QPSK constellation and D0, D1, D2,…,D7 for the 16/32/64-QAM constellations will be described next in the context of “mapping-by-set-partitioning” concept.

Table 11-7—Input-output relations and state transitions of QPSK trellis encoder

Current state $s_n^2\ s_n^1\ s_n^0$	**Input bit** x_n^1	**Output bits**		**Next state** $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^1\ y_n^0$	**Subset number**	
0 0 0 (**S0**)	0	0 0	**C0**	0 0 0 (**S0**)
	1	1 0	**C2**	0 0 1 (**S1**)
0 0 1 (**S1**)	0	0 1	**C1**	1 1 0 (**S6**)
	1	1 1	**C3**	1 1 1 (**S7**)
0 10 (**S2**)	0	0 0	**C0**	0 0 1 (**S1**)
	1	1 0	**C2**	0 0 0 (**S0**)

Table 11-7—Input-output relations and state transitions of QPSK trellis encoder (*continued*)

Current state $s_n^2\ s_n^1\ s_n^0$	Input bit x_n^1	Output bits		Next state $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^1\ y_n^0$	Subset number	
0 1 1 (**S3**)	0	0 1	**C1**	1 1 1 (**S7**)
	1	1 1	**C3**	1 1 0 (**S6**)
1 0 0 (**S4**)	0	0 0	**C0**	0 1 0 (**S2**)
	1	1 0	**C2**	0 1 1 (**S3**)
1 0 1 (**S5**)	0	0 1	**C1**	1 0 0 (**S4**)
	1	1 1	**C3**	1 0 1 (**S5**)
1 1 0 (**S6**)	0	0 0	**C0**	0 1 1 (**S3**)
	1	1 0	**C2**	0 1 0 (**S2**)
1 1 1 (**S7**)	0	0 1	**C1**	1 0 1 (**S5**)
	1	1 1	**C3**	1 0 0 (**S4**)

Table 11-8—Input-output relations and state transitions of 16/32/64-QAM trellis encoder

Current state $s_n^2\ s_n^1\ s_n^0$	Input bits $x_n^2\ x_n^1$	Output bits		Next state $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^2\ y_n^1\ y_n^0$	Subset number	
0 0 0 (**S0**)	0 0	0 0 0	D0	0 0 0 (**S0**)
	0 1	0 1 0	D2	0 0 1 (**S1**)
	1 0	1 0 0	D4	0 1 0 (**S2**)
	1 1	1 1 0	D6	0 1 1 (**S3**)
0 0 1 (**S1**)	0 0	0 0 1	D1	1 1 0 (**S6**)
	0 1	0 1 1	D3	1 1 1 (**S7**)
	1 0	1 0 1	D5	1 0 0 (**S4**)
	1 1	1 1 1	D7	1 0 1 (**S5**)
0 1 0 (**S2**)	0 0	0 0 0	D0	0 0 1 (**S1**)
	0 1	0 1 0	D2	0 0 0 (**S0**)
	1 0	1 0 0	D4	0 1 1 (**S3**)
	1 1	1 1 0	D6	0 1 0 (**S2**)
0 1 1 (**S3**)	0 0	0 0 1	D1	1 1 1 (**S7**)
	0 1	0 1 1	D3	1 1 0 (**S6**)
	1 0	1 0 1	D5	1 0 1 (**S5**)
	1 1	1 1 1	D7	1 0 0 (**S4**)

Table 11-8—Input-output relations and state transitions of 16/32/64-QAM trellis encoder (*continued*)

Current state $s_n^2\ s_n^1\ s_n^0$	Input bits $x_n^2\ x_n^1$	Output bits		Next state $s_{n+1}^2\ s_{n+1}^1\ s_{n+1}^0$
		$y_n^2\ y_n^1\ y_n^0$	Subset number	
1 0 0 (**S4**)	0 0	0 0 0	D0	0 1 0 (**S2**)
	0 1	0 1 0	D2	0 1 1 (**S3**)
	1 0	1 0 0	D4	0 0 0 (**S0**)
	1 1	1 1 0	D6	0 0 1 (**S1**)
1 0 1 (**S5**)	0 0	0 0 1	D1	1 0 0 (**S4**)
	0 1	0 1 1	D3	1 0 1 (**S5**)
	1 0	1 0 1	D5	1 1 0 (**S6**)
	1 1	1 1 1	D7	1 1 1 (**S7**)
1 1 0 (**S6**)	0 0	0 0 0	D0	0 1 1 (**S3**)
	0 1	0 1 0	D2	0 1 0 (**S2**)
	1 0	1 0 0	D4	0 0 1 (**S1**)
	1 1	1 1 0	D6	0 0 0 (**S0**)
1 1 1 (**S7**)	0 0	0 0 1	D1	1 0 1 (**S5**)
	0 1	0 1 1	D3	1 0 0 (**S4**)
	1 0	1 0 1	D5	1 1 1 (**S7**)
	1 1	1 1 1	D7	1 1 0 (**S6**)

The symbol mapper shown in Figure 11-4 provides a one-to-one mapping between an output bit vector of the trellis encoder and a two-dimensional signal point of the signal constellation given in Figure 11-3. For a given output bit vector of the trellis encoder, a QPSK or 16/32/64-QAM constellation point is selected based on the set partitioning rule illustrated in Figure 11-5 and Figure 11-6, respectively. The lower order output bits y_n^1, y_n^0 and y_n^2, y_n^1, y_n^0 are used in determining the symbol subsets C0, C1, C2, C3 and D0, D1, D2,…,D7 for QPSK and 16/32/64-QAM constellations, respectively. The subsets C0, C1, C2, C3 each contain 1 symbol for the QPSK modulation, whereas the subsets D0, D1, D2,…,D7 each contain 2, 4, and 8 symbols for 16-QAM, 32-QAM, and 64-QAM cases, respectively. Therefore, as shown in Figure 11-4, the remaining output bits y_n^5, y_n^4, y_n^3 select one of the 8 symbols from the subsets D0,…,D7 in the 64-QAM case, and the output bits y_n^4, y_n^3 select one of the 4 symbols from the subsets D0,…,D7 in the 32-QAM case, and finally the output bit y_n^3 selects one of the 2 symbols from the subsets D0,…,D7 in the 16-QAM case. Figure 11-7 and Figure 11-8 show the assignment of signal subsets to the QPSK and 16/32/64-QAM constellations, respectively. Furthermore, specific bit mappings to constellation points are given in Figure 11-9 and Figure 11-10 for the respective constellations. Binary representations below the subset numbers correspond to $y_n^1 y_n^0$ in the QPSK case and to $y_n^5 y_n^4 y_n^3 y_n^2 y_n^1 y_n^0$ in the 16/32/64-QAM cases. The lower order 2 bits correspond to the subset numbers for the QPSK modulation. Likewise, the lower order 3 bits correspond to the subset numbers for the 16/32/64-QAM. The higher order 3 bits for the 16/32/64-QAM cases are assigned within each signal subset (D0,…,D7) such that decimal representation of the bit mapping goes from low to high as the constellation points are traced from center outward. This rule ensures that the decimal representations of the bit mappings from 0 to 15 belong to 16-QAM constellation, and 0 to 31 belong to 32-QAM constellation, and 0 to 63 belong to 64-QAM constellation.

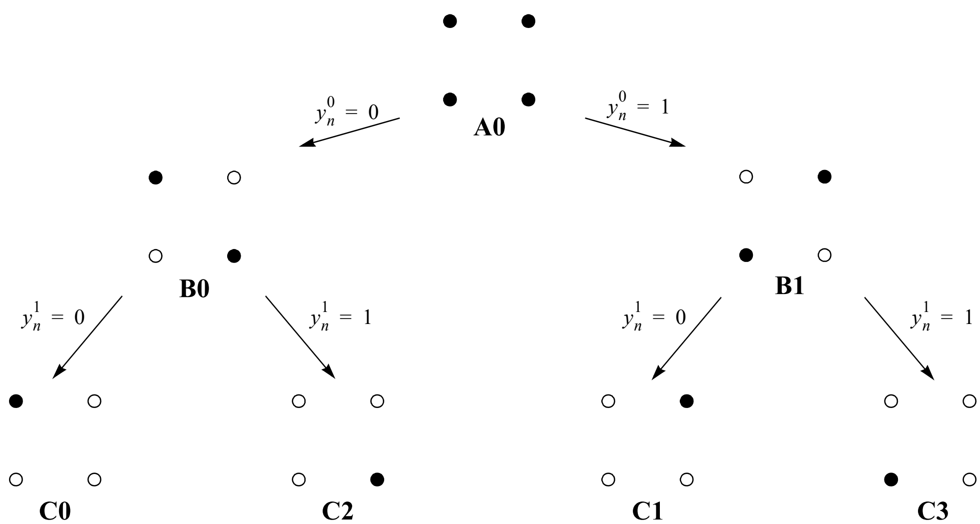

Figure 11-5—QPSK set partitioning

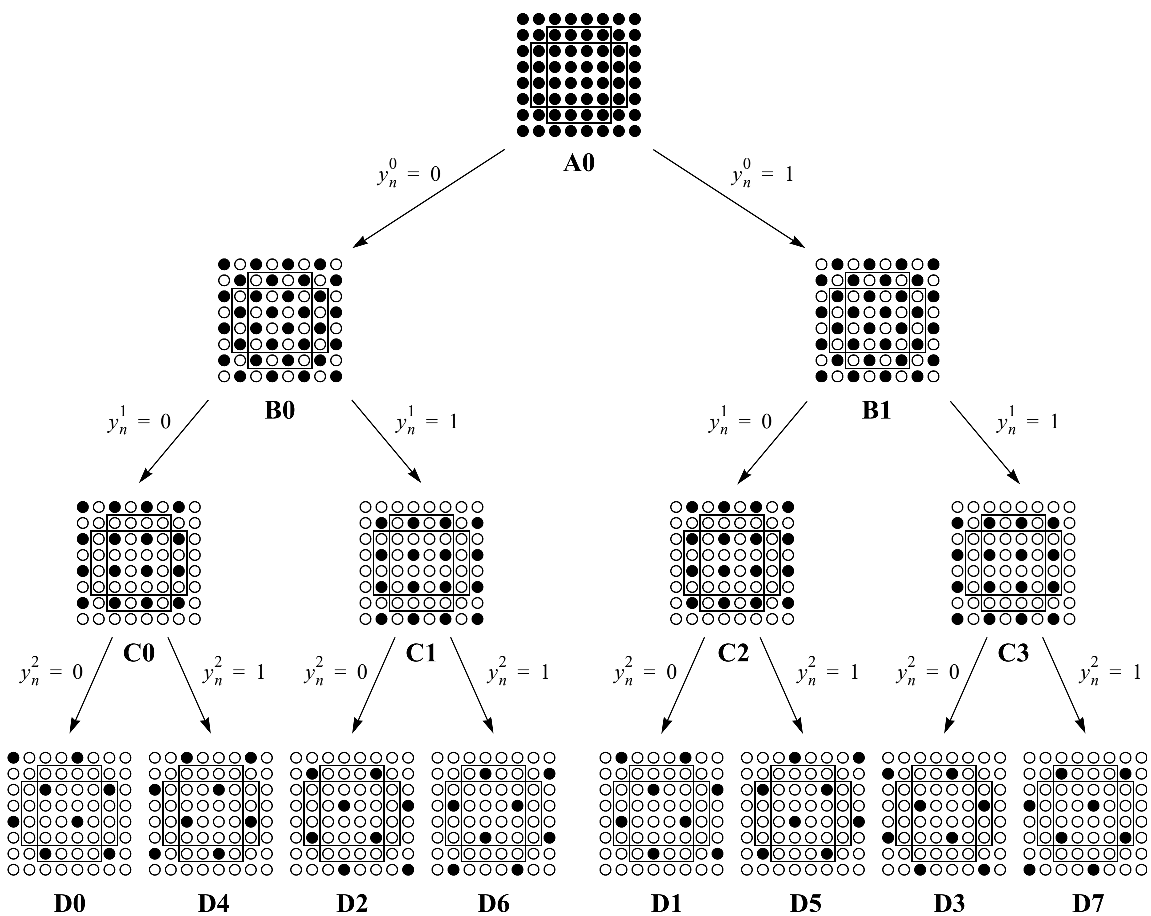

Figure 11-6—16/32/64-QAM set partitioning

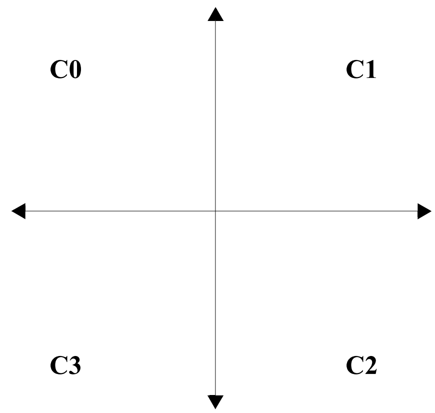

Figure 11-7—Assignment of subsets to QPSK constellation symbols

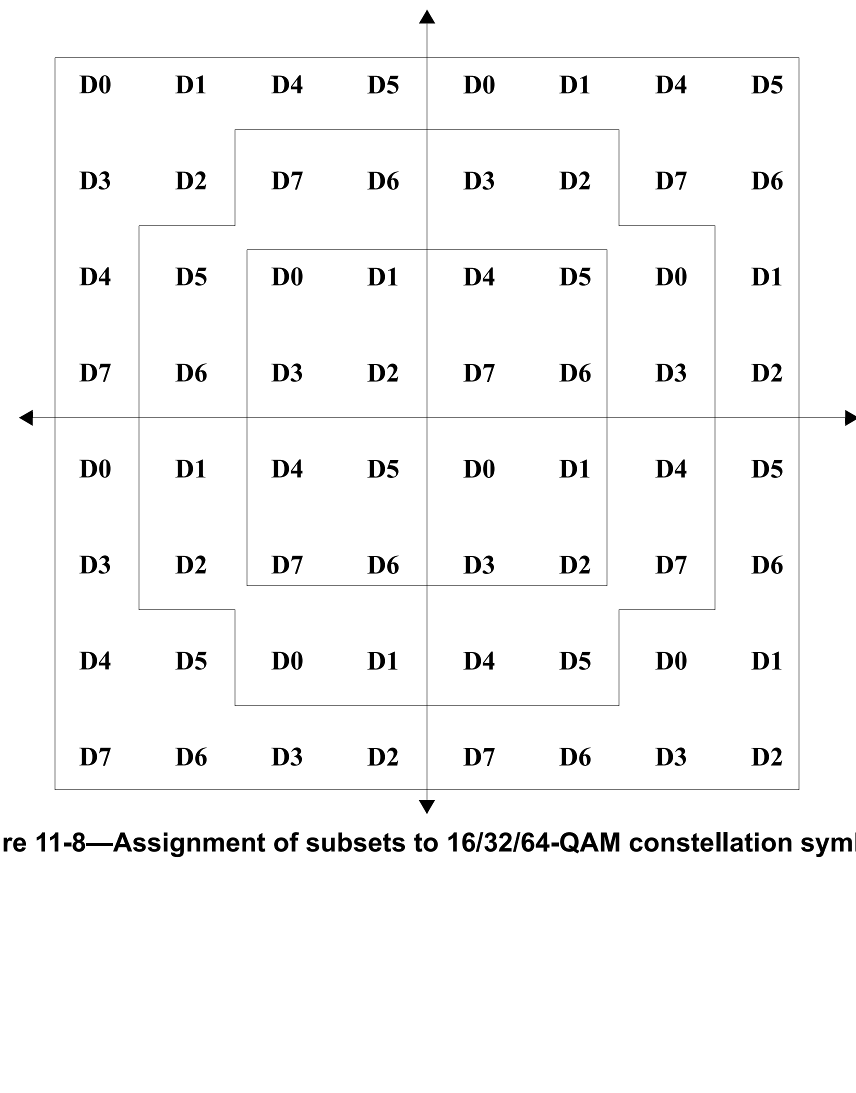

Figure 11-8—Assignment of subsets to 16/32/64-QAM constellation symbols

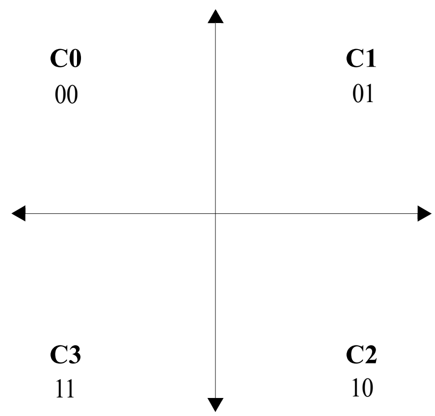

Figure 11-9—QPSK constellation bit mappings

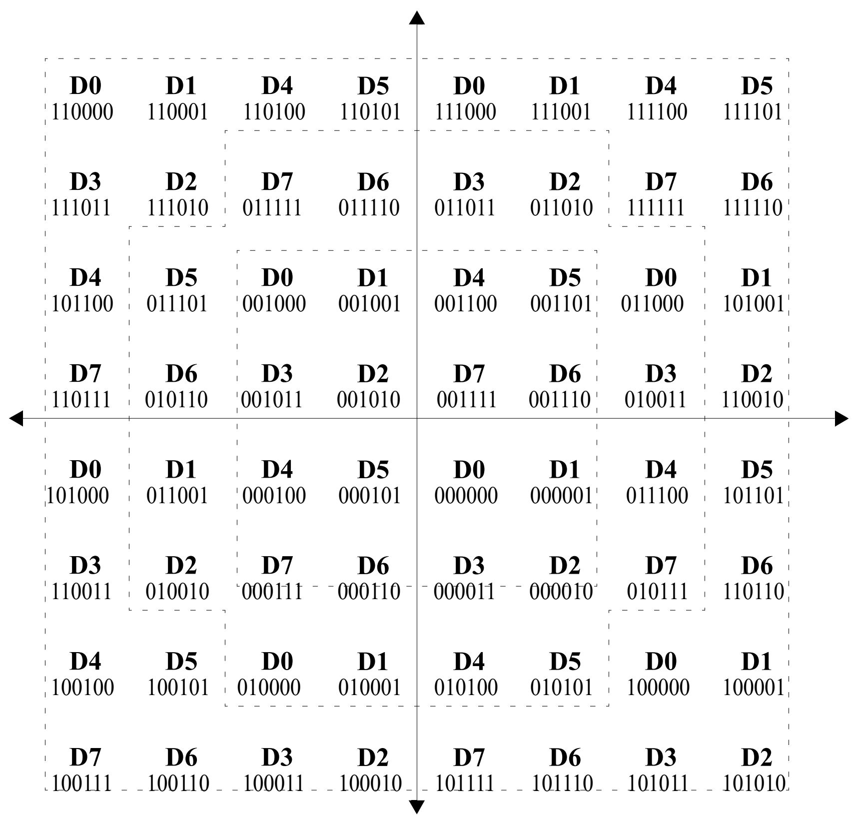

Figure 11-10—16/32/64-QAM constellation bit mappings

Finally, based on Table 11-7 and Table 11-8, the state-transition diagram of the 8-state trellis code is shown for QPSK modulation in Figure 11-11 and for 16/32/64-QAM in Figure 11-12

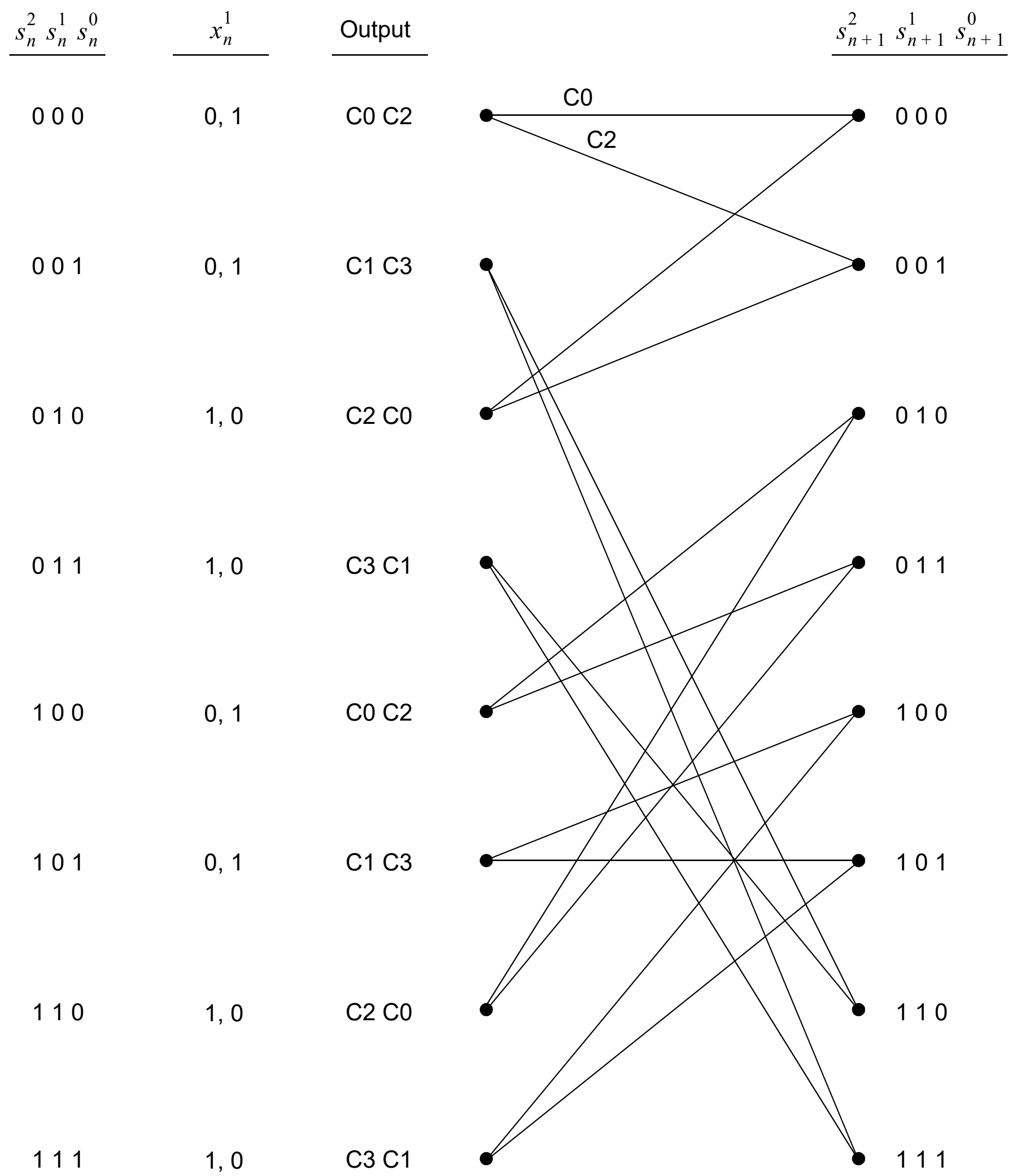

Figure 11-11—State transition diagram of 8-state QPSK trellis code

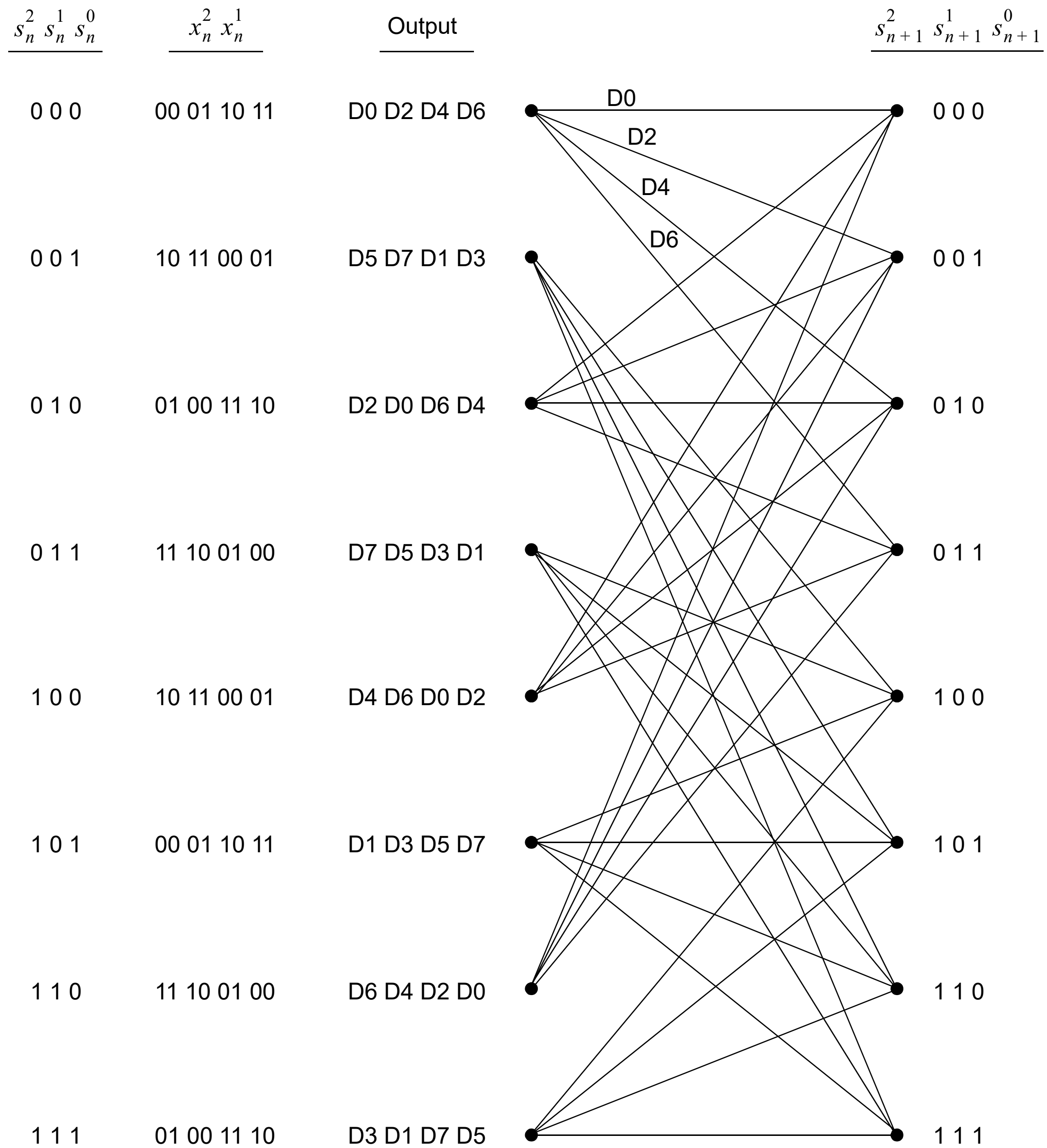

Figure 11-12—State transition diagram of 8-state 16/32/64-QAM trellis code

11.4 PHY frame format

11.4.1 Frame format

The PHY frame format for the 22 Mb/s, 33 Mb/s, 44 Mb/s, and 55 Mb/s modes is illustrated in Figure 11-13. The PHY layer prepends the PHY header, as described in 11.4.5, to the MAC header, as described in 6.2, calculates the HCS, as described in 11.2.9, over the combined PHY and MAC headers, and appends the HCS to the end of the MAC header. If the size of the MAC Frame Body field, as defined in 6.2, is not an integer multiple of the bits/symbol, then stuff bits are added following the MAC Frame Body field, as described in 11.4.6. The PHY preamble, as described in 11.4.2, is sent first, followed by the PHY header, MAC header and HCS, followed by the frame payload, the FCS, the SBs, if necessary, and finally the tail symbols, as described in 11.4.7. As shown in Figure 11-13, for the 22 Mb/s, 33 Mb/s, 44 Mb/s, and 55 Mb/s transmission modes, the PHY preamble, as described in 11.4.2, is modulated with the 22 Mb/s QPSK mode. The PHY header, MAC header, and the HCS is modulated in the 22 Mb/s DQPSK mode. Finally, the frame payload, FCS, stuff bits (if necessary) and the tail symbols are modulated at the desired rates of either 22 Mb/s, 33 Mb/s, 44 Mb/s, or 55 Mb/s.

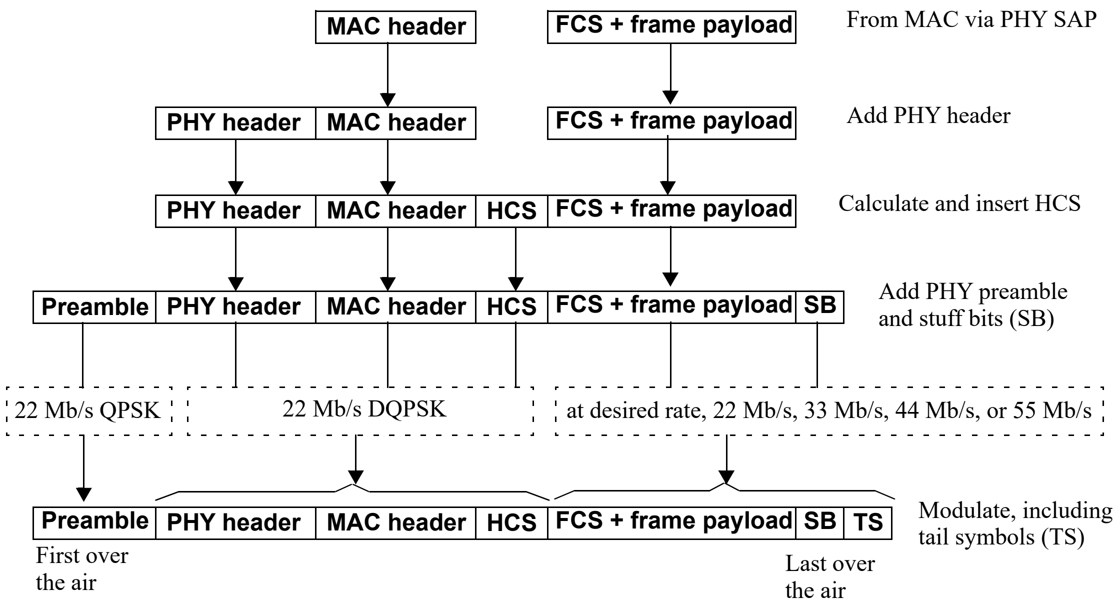

Figure 11-13—PHY frame formatting for 22 Mb/s, 33 Mb/s, 44 Mb/s, and 55 Mb/s modes

The PHY frame format for the 11 Mb/s mode is slightly different from the other modes and is illustrated in Figure 11-14. As in the other modes, the PHY layer prepends the PHY header, as described in 11.4.5, to the MAC header, as described in 6.2, calculates the HCS, as described in 11.2.9, over the combined PHY and MAC headers, and appends the HCS to the end of the MAC header. Since the 11 Mb/s mode is modulated at 1 bit per symbol, stuff bits are not needed for the MAC Frame Body field in this mode. The concatenation of the PHY header, MAC header and HCS is repeated in the 11 Mb/s transmission mode. That is, the PHY preamble, as described in 11.4.2, is sent first in the frame, followed by two repetitions of the combined PHY header, MAC header and HCS, followed by the frame payload, the FCS and finally the tail symbols, as described in 11.4.7.

As illustrated in Figure 11-14, for the 11 Mb/s transmission mode, the PHY preamble, as described in 11.4.2, is modulated in the 22 Mb/s QPSK mode since the CAZAC sequence used, as described in 11.4.2, is based on 4-phase symbols. The first repetition of the combined PHY header, MAC header and the HCS, is modulated in the 22 Mb/s DQPSK mode. The second repetition of the combined PHY header, MAC header and HCS, is modulated in the 11 Mb/s QPSK-TCM mode. The repetition process ensures that the header error rate is significantly lower than the frame error rate. Finally, the Frame Payload field, FCS, and the tail symbols are all modulated with the 11 Mb/s QPSK-TCM mode. Note that although the concatenation of the PHY header, MAC header and HCS is repeated twice in the PHY frame format, the PHY layer shall only provide one copy of the correctly decoded MAC header to the MAC sublayer at the receiver side.

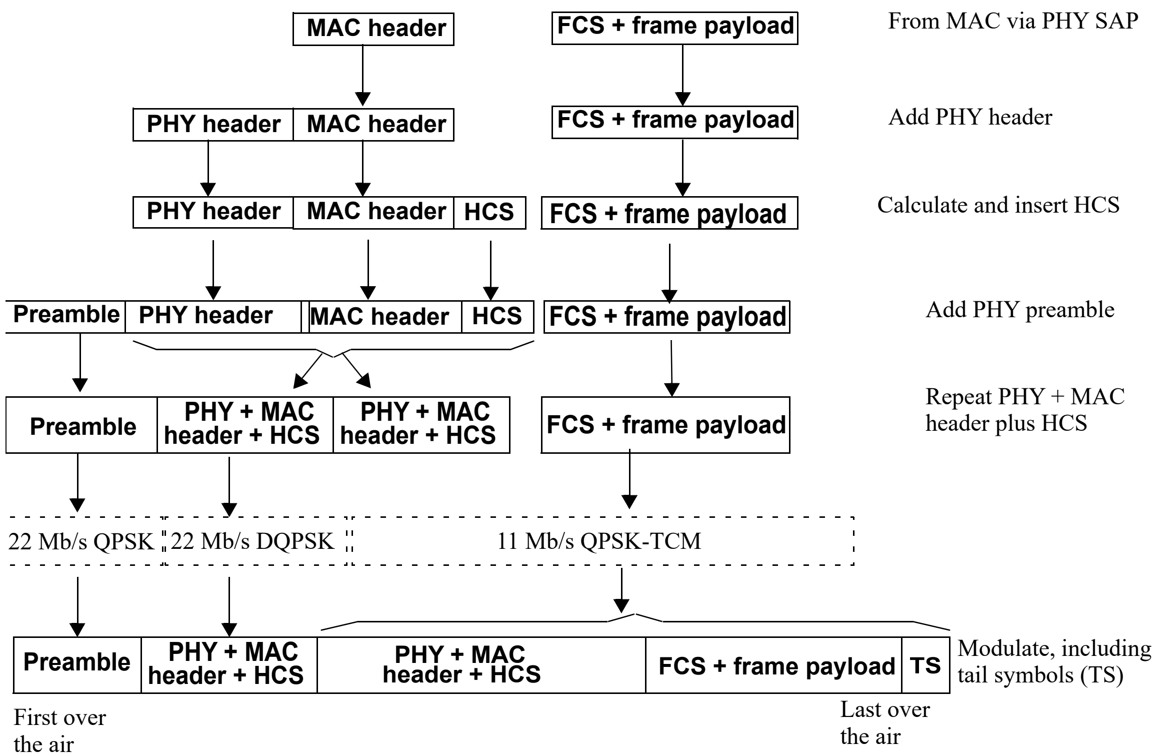

Figure 11-14—PHY frame formatting for 11 Mb/s mode

11.4.2 PHY preamble

A PHY preamble shall be added prior to the PHY header to aid receiver algorithms related to synchronization, carrier-offset recovery, and signal equalization. The preamble shall consist of multiple periods of a special sequence of 16 QPSK symbols called a CAZAC sequence (see Milewski [B8]), which demonstrates a constant amplitude zero auto-correlation property. The CAZAC sequence shall be denoted as $\{C_0, C_1, C_2, \ldots, C_{15}\}$. Each element, C_i, of the CAZAC sequence shall have a complex value representing the in-phase and quadrature components of a QPSK-type sequence, as shown in Table 11-9.

Table 11-9—CAZAC sequence

CAZAC sequence element	Value
C_0	$1 + j$
C_1	$1 + j$
C_2	$1 + j$
C_3	$1 + j$
C_4	$-1 + j$
C_5	$-1 - j$
C_6	$1 - j$
C_7	$1 + j$
C_8	$-1 - j$

Table 11-9—CAZAC sequence (*continued*)

CAZAC sequence element	Value
C_9	$1 + j$
C_{10}	$-1 - j$
C_{11}	$1 + j$
C_{12}	$1 - j$
C_{13}	$-1 - j$
C_{14}	$-1 + j$
C_{15}	$1 + j$

The PHY preamble shall be constructed by successively appending 12 periods, denoted as $\{P_0, P_1, P_2, ..., P_{10}, E\}$, of the CAZAC sequence defined in Table 11-9, except for the 12th period where each element of the CAZAC sequence shall be negated, or equivalently, rotated by 180 degrees. $P_0, P_1, ..., P_{10}$ are all identical vectors containing the CAZAC symbols defined in Table 11-9, and the vector E denotes the end-of-preamble delimiter that is the 180 degrees rotated version of the CAZAC sequence given in Table 11-9. The complete 2.4 GHz PHY layer preamble is shown in Figure 11-15.

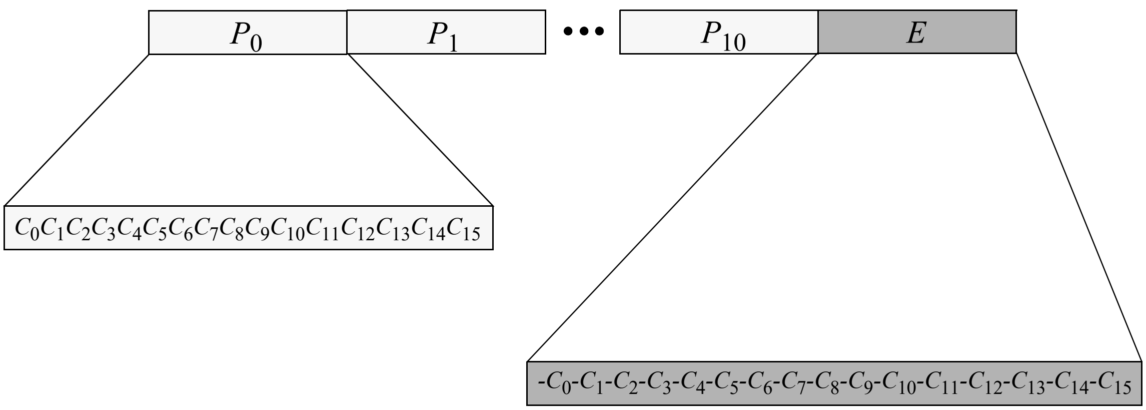

Figure 11-15—2.4 GHz PHY preamble format

11.4.3 Header modulation

The 2.4 GHz PHY header and MAC header shall be modulated using DQPSK modulation for all modulation formats except the 11-Mb/s QPSK-TCM mode.

The header is usually much shorter than the MAC Frame Body field. Consequently, it is more probable to correctly receive the header than the MAC Frame Body field even without a FEC for the header that is modulated in the DQPSK format. This is true for all modulation formats other than the 11-Mb/s QPSK-TCM.

For the QPSK-TCM mode, the combined PHY header, MAC header and the HCS shall be repeated twice as shown in Figure 11-14 to ensure that the header error rate is lower than the frame payload error rate. The first occurrence of the combined PHY header, MAC header and HCS shall be encoded using the DQPSK modulation. The second occurrence of the combined PHY header, MAC header, and HCS shall be encoded using the 11-Mb/s QPSK-TCM format.

11.4.4 Scrambling

A side-stream scrambler (see Figure 11-16) shall be used for the MAC Header field, HCS field, MAC Frame Body field and, if present, the stuff bits. The PHY Preamble field, PHY Header field and tail symbols shall not be scrambled. The polynomial generator, *g(D)*, for the pseudo-random binary sequence (PRBS) generator shall be as shown in Equation (11-3):

$$g(D) = 1 + D^{14} + D^{15} \tag{11-3}$$

where D is a single bit delay element. The polynomial forms not only a maximal length sequence, but also is a primitive polynomial (see Peterson, et al. [B9]). By the given generator polynomial, the corresponding PRBS, x_n, is generated as shown in Equation (11-4):

$$x_n = x_{n-14} \oplus x_{n-15} \tag{11-4}$$

where "" $\oplus$ "" denotes modulo-2 addition.

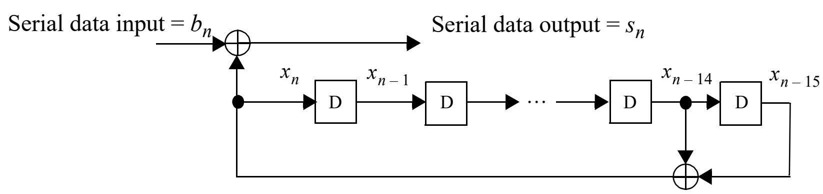

Figure 11-16—Realization of side-stream scrambler by linear feedback shift registers

The sequence shown in Equation (11-5) defines the initialization sequence, x_{init}, which is specified by the parameter "seed value" in Table 11-10.

$$x_{\text{init}} = \begin{bmatrix} x^i_{n-1} & x^i_{n-2} & x^i_{n-3} & x^i_{n-4} & x^i_{n-5} & x^i_{n-6} & x^i_{n-7} & x^i_{n-8} & x^i_{n-9} & x^i_{n-10} & x^i_{n-11} & x^i_{n-12} & x^i_{n-13} & x^i_{n-14} & x^i_{n-15} \end{bmatrix} \tag{11-5}$$

where x^i_{n-k} represents the binary initial value at the output of the k^{th} delay element.

The scrambled data bits, s_n, are obtained as shown in Equation (11-6):

$$s_n = b_n \oplus x_n \tag{11-6}$$

where b_n represents the unscrambled data bits. The side-stream de-scrambler at the receiver shall be initialized with the same initialization vector, x_{init}, used in the transmitter scrambler. The initialization vector is determined from the seed identifier contained in the PHY header of the received frame.

The 15-bit seed value chosen shall correspond to the seed identifier, as described in 11.4.5, and as shown in Table 11-10. The seed identifier value is set to 00 when the PHY is initialized and is incremented in a 2-bit rollover counter for each frame that is sent by the PHY. The value of the seed identifier that is used for the frame is sent in the PHY header, as described in 11.4.5.

The 15-bit seed value is configured as follows. At the beginning of each PHY frame, the register is cleared, the seed value is loaded in using x_{init}, where n is set to 15, and the first scrambler bit is calculated. The PHY preamble and PHY header shall not be scrambled. The first bit of data of the MAC header, is modulo-2 added with the first scrambler bit, followed by the rest of the bits in the MAC Header field and the MAC Frame Body field. In the 11 Mb/s mode, the PHY header of the second repetition of the PHY + MAC

Table 11-10—Scrambler seed selection

Seed identifier b1, b0	Seed value $x_{14} \dots x_0$
0, 0	0011 1111 1111 111
0, 1	0111 1111 1111 111
1, 0	1011 1111 1111 111
1, 1	1111 1111 1111 111

header + HCS shall not be scrambled. In this mode, the scrambler shall be re-initialized with the same seed used for the first header when it begins scrambling the second header. The scrambler shall continue its operation as normal for the MAC Frame Body field following the second header structure.

11.4.5 PHY header

The PHY header consists of two octets that identify the number of octets in the frame payload (which does not include the FCS, as described in 6.2), the data rate of the MAC Frame Body field, and the seed identifier for the data scrambler. The frame payload length does not include the tail symbols, as described in 11.4.7, or the stuff bits, as described in 11.4.6. The fields for the PHY header are shown in Table 11-11. Bit b0 is sent over the air first and the other bits follow sequentially.

Table 11-11—PHY header

Bits	Content	Description
b1–b0	Seed identifier	A field that selects the seed for the data scrambler, defined in Table 11-10.
b4–b2	MAC Frame Body field data rate	A field that indicates the data rate at which the MAC Frame Body field is sent. The data rate encodings are defined in Table 11-12.
b15–b5	Payload length	A field that contains the length of the frame payload, in octets, MSB is b15, LSB is b5, e.g., 5 octets of data, is encoded as 0b00000000101 (MSB on left, LSB on right). A zero-length frame payload is encoded as 0b00000000000, and there is no FCS for this frame.

The encoding for the MAC Frame Body field data rate is defined in Table 11-12.

Table 11-12—MAC Frame Body field data rate encoding

Modulation	Data rate	b4	b3	b2
QPSK-TCM	11 Mb/s	0	0	0
DQPSK	22 Mb/s	0	0	1
16-QAM-TCM	33 Mb/s	0	1	0

Table 11-12—MAC Frame Body field data rate encoding (*continued*)

Modulation	Data rate	b4	b3	b2
32-QAM-TCM	44 Mb/s	0	1	1
64-QAM-TCM	55 Mb/s	1	0	0

11.4.6 Stuff bits

If the total length of the MPDU is not an integer multiple of the number of bits/symbol that will be used to modulate the MPDU, then stuff bits shall be added to the end of the MPDU prior to modulation. The stuff bits may be set to either 0 or 1 and shall be ignored when the frame is received. Note that the stuff bits are not a part of either the HCS or FCS calculation. A compliant PHY shall add enough stuff bits so that the MPDU plus stuff bits is an integer multiple of the bits/symbol that is to be used to modulate the MPDU. A compliant PHY shall add less than the number of bits than are contained in a single symbol, i.e., less than 3 for the 33-Mb/s mode and less than 5 for the 55-Mb/s mode.

Since the MPDU is an integer number of octets, this requirement applies only to the 33- and 55-Mb/s modes. For the 11-, 22-, and 44-Mb/s modes, the bits/symbol (1, 2, and 4, respectively) allow the MPDU to be directly mapped into an integer number of symbols.

11.4.7 Tail symbols

Tail symbols shall be added to the end of the MAC Frame Body field, i.e., after either the FCS or the stuff bits, if they are present, for all modulation formats. The tail symbols are used for trellis coded modulation formats in order to terminate the encoded trellis sequence in a known state to aid the decoding process.

For 11-Mb/s QPSK-TCM format, three tail symbols, each containing 1 bit, shall be appended to the end of the MAC Frame Body field. As shown in Figure 11-11, the lowest order input bit x_n^1 solely determines the state transitions. Encoded trellis sequences shall be terminated in state 0, i.e., S0 at the end of each transmission frame as illustrated in Figure 11-17. Table 11-13 shows the assignment of trellis bits based on the last state visited at the end of the MAC Frame Body field.

Table 11-13—Assignment of trellis tail symbol for QPSK-TCM

Last state	1st tail bit x_n^1	2nd tail bit x_n^1	3nd tail bit x_n^1
S0	0	0	0
S1	0	1	1
S2	1	0	0
S3	1	1	1
S4	0	1	0
S5	0	0	1
S6	1	1	0
S7	1	0	1

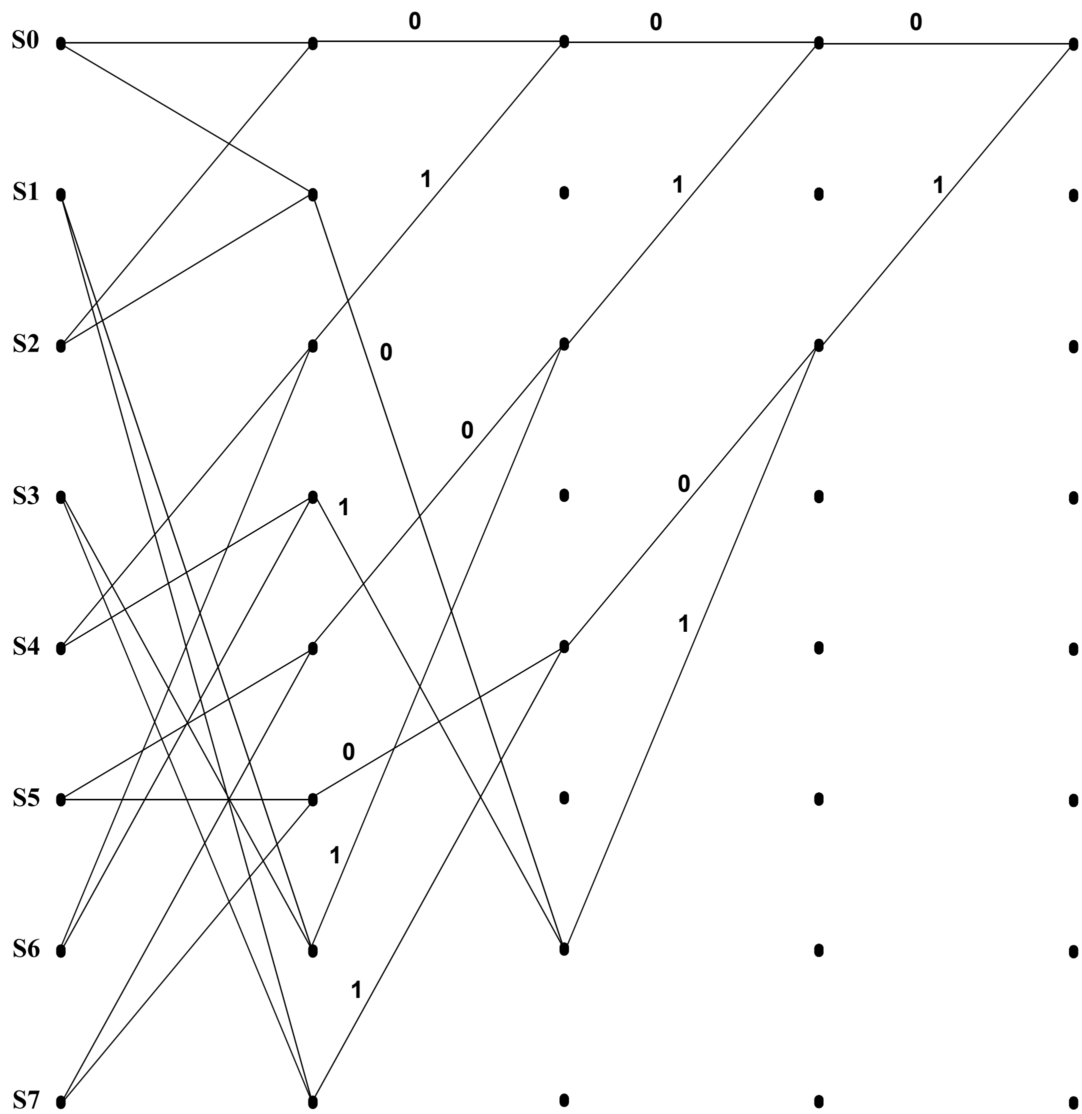

Figure 11-17—Assignment of tail symbols for QPSK-TCM

For the 16/32/64-QAM formats, the lower order two input bits x_n^2, x_n^1 to the trellis encoder determine the trellis code state transitions as illustrated in Figure 11-12. In this case, only two trellis symbols shall be appended, as shown in Figure 11-18, to the end of the MAC Frame Body field in order to terminate the trellis sequence in state 0, i.e., S0. Consequently, the one higher order bit, x_n^3, for 16-QAM does not affect the outcome. Likewise, the higher order bits, x_n^4, x_n^3 and $x_n^5, ,x_n^4, x_n^3$, for 32-QAM and 64-QAM, respectively, are irrelevant in determining the final state. The resultant bit assignments to trellis tail symbols shall take on the form given in Table 11-14, for which x_n^3, x_n^2, x_n^1 refer to 16-QAM, $x_n^4, x_n^3, x_n^2, x_n^1$ refer to 32-QAM, and $x_n^5, x_n^4, x_n^3, x_n^2, x_n^1$ refer to 64-QAM.

For the DQPSK modulation format, the two tail symbols shall be constructed from the binary sequence 0101 where 0 is the first bit of the bit pair.

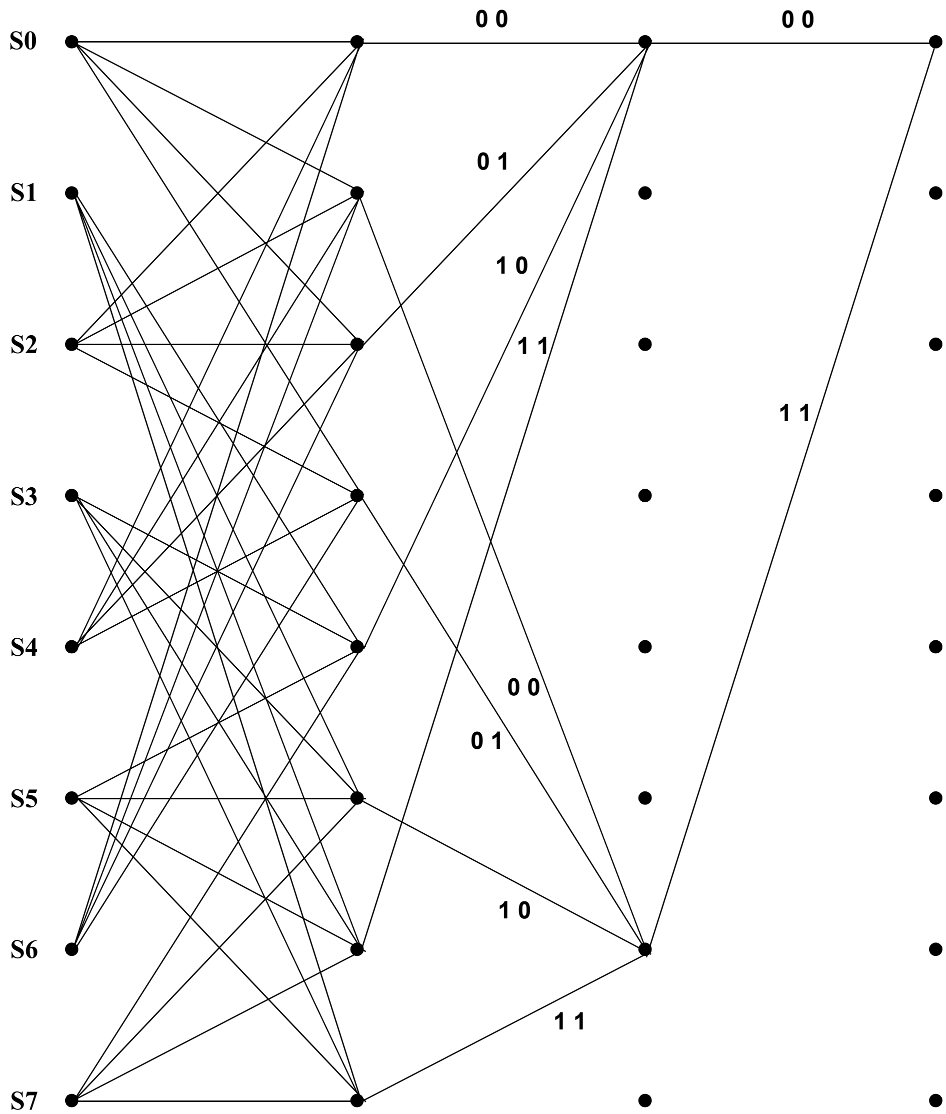

Figure 11-18—Assignment of tail symbols for 16/32/64-QAM-TCM

Table 11-14—Assignment of trellis tail symbols for 16/32/64-QAM

Last State	1st Tail symbol $x_n^5, x_n^4, x_n^3, x_n^2, x_n^1$	2nd Tail symbol $x_n^5, x_n^4, x_n^3, x_n^2, x_n^1$
S0	XXX00	XXX00
S1	XXX00	XXX11
S2	XXX01	XXX00
S3	XXX01	XXX11
S4	XXX10	XXX00
S5	XXX10	XXX11
S6	XXX11	XXX00
S7	XXX11	XXX11

11.5 Transmitter specifications

11.5.1 Error vector magnitude definition

The modulation accuracy of a compliant IEEE 802.15.3 transmitter is determined with an error vector magnitude (EVM) measurement (see Figure 11-19). In order to calculate this measurement, a time record of N received signal co-ordinate pairs $(\tilde{I}_j, \tilde{Q}_j)$ is captured. For each received symbol, a decision is made as to which symbol was transmitted. The ideal position of the chosen symbol (the center of the decision box) is represented by the vector (I_j, Q_j). The error vector $(\delta I_j, \delta Q_j)$ is defined as the distance from this ideal position to the actual position of the received symbol.

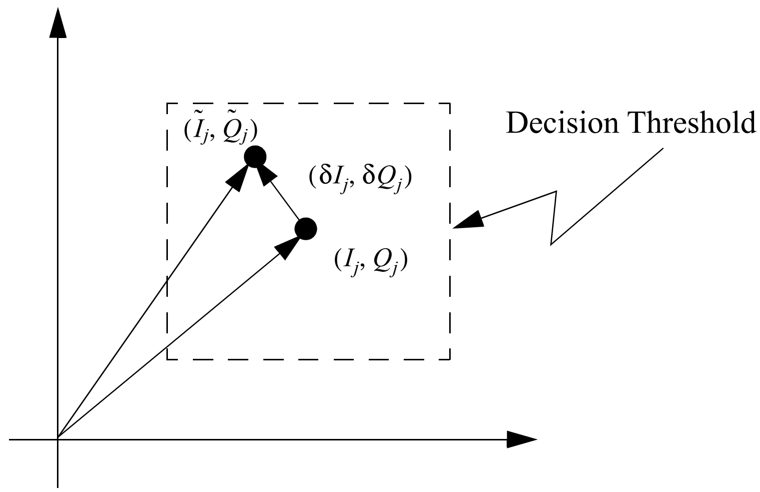

Figure 11-19—Error vector calculation

Thus, the received vector is the sum of the ideal vector and the error vector, as shown in Equation (11-7).

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q_j) \tag{11-7}$$

The EVM for this standard is defined as shown in Equation (11-8):

$$EVM \equiv \sqrt{\frac{\frac{1}{N}\sum_{j=1}^{N}(\delta I_j^{\,2} + \delta Q_j^{\,2})}{S^2_{max}}} \tag{11-8}$$

where S_{max} is the magnitude of the vector to the outermost constellation point and $(\delta I_j, \delta Q_j)$ is the error vector.

11.5.2 EVM calculated values

A compliant transmitter shall have EVM values of less than those given in Table 11-15 for all of the modulation levels supported by the PHY when measured for 1000 symbols. The error vector measurement shall be made on baseband I and Q data after recovery through a ideal reference receiver system. The ideal reference receiver shall perform carrier lock, symbol timing recovery and amplitude adjustment while making the measurements. The ideal reference receiver shall have a data filter impulse response that approximates that of an ideal root raised cosine filter with 30% excess bandwidth.

Table 11-15—EVM values for various modulations

Modulation	EVM (%)
64 QAM	3.3
32 QAM	4.8
16 QAM	7.5
DQPSK	9.2
QPSK-TCM	20.0

11.5.3 Transmit power spectral density (PSD) mask

The transmitted spectral products shall be less than the limits specified in Table 11-16. The power shall be measured in a 100 kHz bandwidth relative to the highest average power in a 100 kHz bandwidth measured within ±6 MHz of the center frequency.

Table 11-16—Transmit PSD limits

Frequency	Relative limit
7.5 MHz < $\lvert f–f_c \rvert$ < 15 MHz	–30 dBr
15 MHz < $\lvert f–f_c \rvert$ < 22 MHz	–10/7[$\lvert f–f_c$ (MHz)$\rvert$ + 13] dBr
22 MHz < $\lvert f–f_c \rvert$	–50 dBr

The transmitter may also have one in-band image, i.e., within 2.4–2.4835 GHz, with a relaxed transmit power spectral density (PSD) requirement of –40 dBr over a 15 MHz bandwidth.

A graphical (informative) representation of the transmit PSD is shown in Figure 11-20.

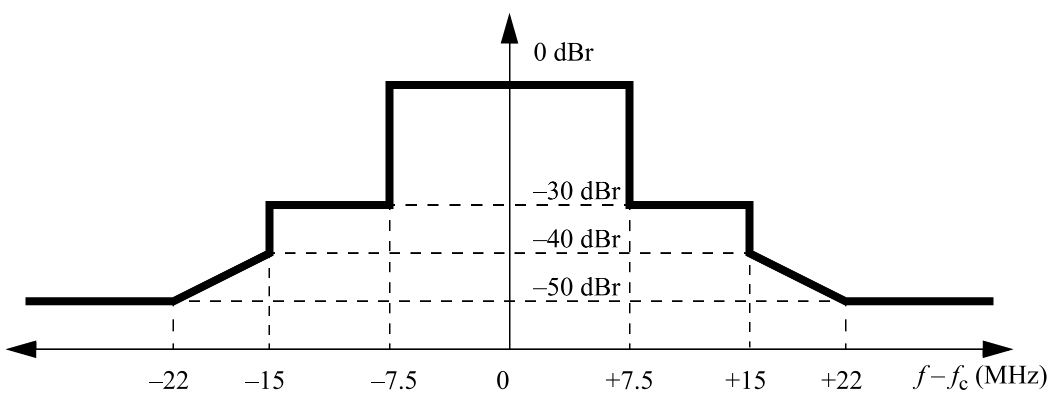

Figure 11-20—Transmit power spectral density mask

11.5.4 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be ± 25 × $10^{–6}$ maximum.

11.5.5 Symbol rate

The PHY shall be capable of transmitting at a symbol rate of 11-Mbaud ± 25 × $10^{–6}$.

The MAC parameter *pClockAccuracy* shall be ± 25 × $10^{–6}$.

11.5.6 Clock synchronization

The transmit center frequency and the symbol rate shall be derived from the same reference oscillator.

11.5.7 Transmit power-on and power-down ramp

The transmit power-on ramp is defined as the time it takes for the RF power emitted by the compliant DEV to rise from less than 10% to greater than 90% of the maximum power to be transmitted in the frame.

The transmit power-on ramp shall be less than 2 µs.

The transmit power-down ramp is defined as the time it takes for the RF power emitted by the compliant DEV to fall from greater than 90% to less than 10% of the maximum power to be transmitted in the frame.

The transmit power-down ramp shall be less than 2 µs.

The transmit power ramps shall be constructed such that the emissions conform with the unwanted emissions specification defined in 11.2.5.

11.5.8 RF carrier suppression

The RF carrier suppression, measured at the channel center frequency, shall be at least 15 dB below the peak sin(x)/x power spectrum. The RF carrier suppression shall be measured while transmitting a repetitive 01 data sequence with the scrambler disabled using DQPSK modulation. A 100 kHz resolution bandwidth shall be used to perform this measurement.

11.5.9 Transmit power

The maximum allowable output power, as measured in accordance with practices specified by the appropriate regulatory bodies, is shown in Table 11-17. A compliant DEV may use any transmit power level up to the applicable limits in the geographical region.

Table 11-17—Maximum transmit power levels

Geographical region	Power limit	Regulatory document
Japan	10 mW	ARIB STD-T66
Europe (except Spain and France)	100 mW EIRP 10 mW/MHz peak power density	ETS 300-328 [B1]
USA	50 mV/m at 3 m in at least a 1 MHz resolution bandwidth[a]	47 CFR 15.249

[a]Electric field strength measurement rather than conducted power measurement.

A compliant transmitter that is capable of transmitting more than 0 dBm shall be capable of reducing its power to less than 0 dBm in monotonic steps no smaller than 3 dB and no larger than 5 dB. The steps shall form a monotonically decreasing sequence of transmit power levels. A compliant DEV shall have its supported power levels indicated in its PHY PIB based on its maximum transmit power and power level step size.

The minimum TX power level required to support TPC, *pMinTpcLevel*, shall be 0 dBm.

11.6 Receiver specifications

11.6.1 Error rate criterion

The error rate criterion shall be a FER of less than 8% with a frame payload length of 1024 octets of pseudo-random data generated with a PN23 sequence as defined by $x_{n+1} = x_n^{23} + x_n^5 + 1$.

Note that the frames used for measuring the error rate criterion include not only the frame payload of 1024 octets, but also the PHY preamble, PHY header, MAC header, HCS, and the FCS.

11.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 11.6.1 is met. The error ratio shall be determined after any error correction has been applied. Compliant systems may have a lower actual sensitivity than the reference sensitivity. A compliant DEV shall achieve at least the reference sensitivity listed in Table 11-18 for each of the modulation formats that the DEV supports.

Table 11-18—Reference sensitivity levels for modulation formats

Modulation	Reference sensitivity
QPSK-TCM	–82 dBm
DQPSK	–75 dBm
16-QAM-TCM	–74 dBm
32-QAM-TCM	–71 dBm
64-QAM-TCM	–68 dBm

11.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal present at the input of the receiver for which the error rate criterion in 11.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

11.6.4 Receiver jamming resistance

The jamming resistance levels are given in Table 11-19. The high-density channel plan shall be used for this test. The adjacent channel is the one on either side of the desired channel that is closest in frequency to the desired channel for the high-density channel mode defined in 11.2.3. The alternate channel is one removed from the adjacent channel in that channel mode. For example, when channel 1 is the desired channel, channel 2 is the adjacent channel and channels 4 and 5 are the alternate channels. When channel 2 is the desired channel, channels 1 and 4 are the adjacent channels and channel 5 is the alternate channel. Channel 3 is not used for the purposes of this test.

Table 11-19—Receiver jamming resistance requirements

Modulation format	Adjacent channel rejection	Alternate channel rejection
QPSK-TCM	33 dB	48 dB
DQPSK	26 dB	41 dB
16-QAM-TCM	25 dB	40 dB
32-QAM-TCM	22 dB	37 dB
64-QAM-TCM	19 dB	34 dB

The list of adjacent and alternate channels is given in Table 11-20.

Table 11-20—Adjacent and alternate channels for receiver jamming resistance test

Desired channel number	Adjacent channel number	Alternate channel number
1	2	4, 5
2	1, 4	5
4	2, 5	1
5	4	1, 2

The adjacent channel rejection shall be measured as follows. The desired signal shall be a conformant IEEE 802.15.3 2.4 GHz PHY signal of pseudo-random data modulated with one of the five modulation types. The desired signal is input to the receiver at a level 6 dB above the reference sensitivity for that modulation as given in Table 11-18. In either the adjacent or alternate channel a conformant IEEE 802.15.3 signal is input at the level specified in Table 11-19 relative to the reference sensitivity for that modulation as given in Table 11-18. For example, for QPSK-TCM the desired signal is input at –76 dBm while the adjacent channel interferer would be input at a level of –49 dBm. The interfering signal shall be DQPSK modulated with pseudo-random data uncorrelated in time with the desired signal. The receiver shall meet the error rate criteria defined in 11.6.1 under these conditions.

A compliant implementation shall satisfy the receiver jamming test for all of the modulations types supported by the DEV.

The desired and interfering signals shall conform to the transmit PSD mask specified in 11.5.3. In addition, the test shall be performed for only one interfering signal at a time.

11.6.5 Receiver CCA performance

A compliant receiver provides CCA capability by performing energy detection in the received signal bandwidth. The start of a valid preamble sequence at a receive level equal to or greater than the minimum sensitivity for the DQPSK base rate, as described in 11.6.2, shall cause CCA to indicate medium busy with a probability of >90% within five CAZAC periods as described in 11.4.2. The receiver CCA function shall in all circumstances report medium busy with any signal 20 dB above the minimum sensitivity for the DQPSK base rate.

The CCA detection time shall be equal *pCcaDetectTime*, which is five CAZAC periods. The CCA shall be maintained as busy until the end of the frame for which the inverted CAZAC sequence was detected.

11.6.6 Receiver RSSI

RSSI is defined as the power relative to the maximum receiver input power level, as described in 11.6.3, in 8 steps of 8 dB with ± 4 dB step size accuracy. The range covered shall be a minimum of 40 dB. The steps shall be monotonic. The RSSI power shall be the average power measured in the last CAZAC sequence of the PHY preamble, as described in 11.4.2.

11.6.7 Link quality indication (LQI)

The LQI shall be reported for the trellis coded modulation (TCM) coded QAM modes using a SNR estimation. The SNR shall be measured at the decision point in the receiver. The SNR includes the thermal noise, distortion, uncorrected interference, and other signal impairments at the decision point in the receiver. The receiver shall report the SNR as a 5-bit number that covers a range from 6 dB to 21.5 dB of SNR. The value 0x00000 shall correspond to less than or equal to 6 dB SNR and 0x11111 shall correspond to more than or equal to 21.5 dB SNR with equal steps in between. The LQI SNR shall be measured during the TCM MAC Frame Body field and shall be reported after the last FCS symbol.

11.7 PHY management

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the PHY layer of a DEV. The encoding of the PHY data rates used in the Supported Data Rates field in the Capability IE, as described in 6.4.12, is given in Table 11-21.

Table 11-21—2.4 GHz PHY supported data rate encoding

Rates supported (Mb/s)	b0	b1	b2	b3	b4
22	0	0	0	0	0
11, 22	1	0	0	0	0
11, 22, 33	0	1	0	0	0
11, 22, 33, 44	1	1	0	0	0
11, 22, 33, 44, 55	0	0	1	0	0

The encoding of the supported PHY data rates into an octet number is accomplished by adding bits b5–b7, all set to zero, to the encoding given in Table 11-21. Bit b0 is the LSB while bit 7 is the MSB. Thus a DEV that supports 11 Mb/s, 22 Mb/s, and 33 Mb/s would have a Supported Data Rates field 01000 (LSB to MSB) and an octet encoding of 0x2.

The encoding of the preferred fragment size used in the Capability IE, as described in 6.4.12, is given in Table 11-22.

Table 11-22—2.4 GHz PHY preferred fragment size encoding

Field value	Preferred fragment size (octets)
0	*pMaxFrameBodySize*
1	1792
2	1536
3	1280
4	1024
5	512
6	256
7	*pMinFragmentSize*

The Fragmentation Control field, 6.2.7, in an Imm-ACK or Dly-ACK frame may be used by the DEV to report the status of the frame that is being ACKed. In the case of a Dly-ACK frame, the status is valid only for the frame that prompted the Dly-ACK frame and not for other frames indicated in the Dly-ACK frame, 6.3.2.2. DEVs are neither required to send receive status information in Imm-ACK and Dly-ACK frames nor required to decode the field in Imm-ACK or Dly-ACK frames. If the Fragmentation Control field is used to send receive status, for the 2.4 GHz PHY it shall be formatted as illustrated in Figure 11-21.

Bits: b0–b2	b3–b7	b8–b23
RSSI	LQI	Reserved

Figure 11-21—Format of fragmentation control field when used for receive status information

The RSSI field and LQI field are set to the values determined for the frame that is being ACKed. The values for RSSI and LQI are defined in 11.6.6 and 11.6.7, respectively.

The PHY-dependent PIB values for the 2.4 GHz PHY are given in Table 11-23 and Table 11-24. The PHY PIB characteristics group, Table 11-23, contains information that is common to most 2.4 GHz implementations.

Table 11-23—PHY PIB characteristics group parameters

Managed object	Octets	Definition	Access
phyType	1	0x00=2.4 GHz	Read/write
phyRegDomainsSupported	Variable	One octet for each regulatory domain supported, as defined for *phyCurrentRegDomain*	Read/write
phyCurrentRegDomain	1	0x00 = European Telecommunications Standards Institute 0x01 = Federal Communications Commission 0x02 = Industry Canada 0x03 = Association of Radio Industries and Businesses 0x04–0xFF = Reserved	Read/write
phyDataRateVector	1	Encodes the data rates, defined in Table 11-21 and 11.7.	Read/write
phyNumChannelsSupported	1	Value = 0x05, see 11.2.3.	Read/write
phyCurrentChannel	1	Indicates the channel that is currently being used, see 11.2.3.	Read/write
phyCcaThreshold	1	The CCA threshold in dBm, encoded in two's complement format. The value is implementation dependent but no larger than the value listed in 11.6.5.	Read/write
phyMaxFrameLength	2	*pMaxFrameBodySize*, see 11.2.8.2.	Read/write
phyPreferredFragmentSize	1	The preferred fragment size of a DEV. The encoding is defined in Table 11-22.	Read/write

The PHY PIB implementation group, Table 11-24, contains information that is more characteristic of a particular PHY implementation than of the PHY as a whole.

Table 11-24—PHY PIB implementation group parameters

Managed object	Octets	Definition	Access
phyDiversitySupported	1	Numeric entry that indicates the number of antennas that are available.	Read/write
phyMaxTXPower	1	The maximum TX power that the DEV is capable of using, 6.4.12, implementation dependent.	Read/write
phyTXPowerStepSize	1	The step size for power control supported by the DEV, 6.4.19, implementation dependent.	Read/write
phyNumPMLevels	1	Number of power management levels supported. The range is 1 to 8 and the value is implementation dependent.	Read/write
phyPMLevelReturn	Variable	Table of vectors with number of entries given by *phyNumPMLevels*. Each vector is the time required to change between power saving states of the PHY. Vector number 0 is the time required to change the PHY from the off state to a state where it is ready to receive commands. Other values are implementation dependent.	Read/write

12. PHY specification for millimeter wave

12.1 General requirements

12.1.1 Overview

A compliant millimeter wave (mmWave) PHY shall implement at least one of the following PHY modes:

a) Single Carrier mode in mmWave PHY (SC PHY), as defined in 12.2.
b) High Speed Interface mode in mmWave PHY (HSI PHY), as defined in 12.3.
c) Audio/Visual mode in mmWave PHY (AV PHY), as defined in 12.4.

Unless otherwise specified, all reserved fields shall be set to zero on transmission and shall be ignored upon reception.

Unless otherwise stated, in all figures in this clause the ordering of the octets and bits as they are presented to the PHY for modulation is the same, as defined in 6.1.

12.1.2 Regulatory information

The mmWave PHY operating frequency is within the 57.0 GHz to 71.0 GHz range as allocated by the regulatory agencies in Europe, Japan, Canada, and the United States. This band will also be available in other areas where allocated by the regulatory bodies.

The documents listed in Table 12-1 are provided as a reference for various geographic regulatory regions. The list is neither exhaustive nor complete. It is the responsibility of the implementer to verify that the DEV complies with all regulatory requirements in the geographic region where the device is deployed or sold.

Table 12-1—Documents for selected geographic regulatory regions

Region	Regulatory document
Australia	Radio communications class license 2000
Canada	RSS-210, Issue 6, September 2005
Japan	Radio Regulatory Commission Rules, No. 14 of 1950, Article 6 (4) (iv)
USA	47 CFR 15.255

The maximum allowable output power, as measured in accordance with practices specified by the appropriate regulatory bodies, is shown in Table 12-2. A compliant DEV may use any transmit power level up to the applicable limits in the geographical region.

Table 12-2—Maximum transmit power levels in selected geographical regions

Region	Power limit	EIRP limit	Regulatory document
USA	—	Maximum indoor EIRP: 27 dBi Maximum outdoor EIRP: 40 dBi	47 CFR 15.255
Japan	Maximum output power: 10dBm Maximum bandwidth: 2.5 GHz	Maximum EIRP: 57 dBi	ARIB STD-T69, ARIB STD-T74
Australia	Maximum output power: 10dBm	Maximum EIRP: 51.8 dBi	Radiocommunications Class License 2000

12.1.3 RF power measurements

Unless otherwise stated, all RF power measurements for the purpose of this standard, either transmit or receive, shall be made based on EIRP and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation. The gain of the antenna is the maximum estimated gain by the manufacturer.

12.1.4 Unwanted emissions

Conformant implementations shall comply with the in-band and out-of-band emissions for all operational modes as set by the applicable regulatory bodies.

12.1.5 Operating temperature range

A conformant implementation shall meet all of the specifications in this standard for ambient temperatures from 0 °C to 40 °C.

12.1.6 RF channelization

The mmWave PHYs use the channels defined in Table 12-3.

Table 12-3—mmWave PHY channelization

CHNL_ID	Start frequency[a]	Center frequency	Stop frequency[a]
1	57.240 GHz	58.320 GHz	59.400 GHz
2	59.400 GHz	60.480 GHz	61.560 GHz
3	61.560 GHz	62.640 GHz	63.720 GHz
4	63.720 GHz	64.800 GHz	65.880 GHz
5	65.880 GHz	66.960 GHz	68.040 GHz
6	68.040 GHz	69.120 GHz	70.200 GHz

[a]The start and stop frequencies are nominal values. The frequency spectrum of the transmitted signal needs to conform to the transmit PSD mask for the PHY mode as well as any regulatory requirement

12.1.7 Transmit PSD mask

The transmitted spectrum shall adhere to the transmit spectrum mask shown in Figure 12-1. For the transmit mask measurements, the resolution bandwidth is set to 3 MHz and video bandwidth to 300 kHz. During OOK modulation, transmitters shall meet the same PSD mask, except for the single line spectra of 40 dB above the 0 dB line in Figure 12-1 within the frequency band of [–6 MHz,+6 MHz] from the carrier frequency.

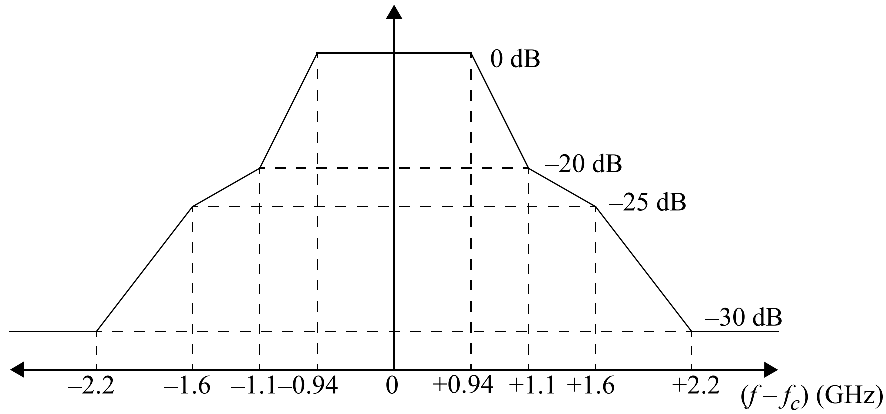

Figure 12-1—Transmit spectral mask

12.1.8 Error vector magnitude calculation

12.1.8.1 SC PHY

The error vector measurement shall be made on baseband I and Q data after recovery through an ideal reference receiver system. An ideal receiver is a receiver that is capable of converting the transmitted signal into a stream of complex samples at sufficient rate or more, with sufficient accuracy in terms of I/Q arm amplitude and phase balance, DC offsets, and phase noise. It shall perform carrier lock, symbol timing recovery and amplitude adjustment while making the measurements. For SC PHY EVM, measuring 1000 chips at the chip rate is recommended. The EVM is given by Equation (12-1):

$$EVM \equiv \sqrt{\frac{1}{1000P_{avg}}\sum_{i=1}^{1000}[(I_i - I^*{}_i)^2 + (Q_i - Q^*{}_i)^2]} \tag{12-1}$$

where

P_{avg}	is the average power of the constellation
(I_i^*, Q_i^*)	is the complex coordinates of the i^{th} measured chip
(I_i, Q_i)	is the complex coordinates of the nearest constellation point for the i^{th} measured chip

The measuring device should have the accuracy of at least 20 dB better than the EVM value to be measured. The test equipment should use a root-raised cosine filter with roll-off factor of 0.25 for the pulse shaping filter when conducting EVM measurement.

12.1.8.2 AV and HSI PHYs

The EVM of an OFDM mmWave transmitter is measured by calculating the Euclidean distance between the received symbol and the closest constellation point. The EVM measurement has the following requirements:

a) The measurement shall be made over the OFDM data symbols starting with the PHY header.
b) The phase and frequency for the data subcarriers shall be corrected for using the pilot subcarriers.
c) Channel estimation shall have been performed for each of the data subcarriers.
d) The error vector magnitude calculation for the HSI and AV PHY modes shall be averaged over a minimum of 20 frames.

The EVM is then the average the RMS errors in the frame over a number of frames. In equation form, it is given by Equation (12-2):

$$EVM_{rms} \equiv \frac{1}{N_p}\left[\sum_{i=1}^{N_p}\sqrt{\frac{1}{P_{avg}N_sN_{dsc}}\sum_{j=1}^{N_s}\left\{\sum_{k=1}^{N_{dsc}}[(I_{ijk}-I^*_{ijk})^2+(Q_{ijk}-Q^*_{ijk})^2]\right\}}\right] \tag{12-2}$$

where

N_p is the number of frames in the measurement
N_s is the number of symbols per frame in the measurement
N_{dsc} is the number of data subcarriers in the OFDM modulation
P_{avg} is the average power of the constellation
(I^*_{ijk}, Q^*_{ijk}) is the complex coordinates of the j^{th} measured symbol in the k^{th} subcarrier of the i^{th} frame
(I_{ijk}, Q_{ijk}) is the complex coordinates of the nearest constellation point for the j^{th} measured symbol of the k^{th} subcarrier of the i^{th} frame

12.1.9 Common PHY management for mmWave PHY modes

12.1.9.1 Supported MCSs

The Supported Data Rates field in the Capability IE, as described in 6.4.12, shall be formatted as illustrated in Figure 12-2.

Bits: 0	b1	b2	b3	b4	b5	b6	b7
SC spreading	SC QPSK	SC 8-PSK	SC 16-QAM	SC LDPC 1	SC LDPC 2	AV 16-QAM	HSI QAM

Figure 12-2—Supported data rates field format for mmWave PHY modes

The SC spreading field shall be set to one if spreading factors 2, 4, and 6 are supported by the SC PHY DEV and shall be set to zero otherwise.

The SC QPSK field shall be set to one if QPSK modulation is supported by the SC PHY DEV and shall be set to zero otherwise.

The SC 8-PSK field shall be set to one if 8-PSK modulation is supported by the SC PHY DEV and shall be set to zero otherwise.

The SC 16-QAM field shall be set to one if 16-QAM modulation is supported by the SC PHY DEV and shall be set to zero otherwise.

The SC LDPC 1 field shall be set to one if the optional FECs LDPC(672, 336), LDPC(672, 504), and LDPC(672, 588) are supported by the SC PHY DEV and shall be set to zero otherwise.

The SC LDPC 2 field shall be set to one if the optional FEC LDPC(1440, 1344) is supported by the SC PHY DEV and shall be set to zero otherwise.

The AV 16-QAM field shall be set to one if 16-QAM modulation is supported by the AV PHY DEV and shall be set to zero otherwise.

The HSI QAM field shall be set to one if the 16-QAM and 64-QAM modulations are supported by the HSI PHY DEV and shall be set to zero otherwise.

12.1.9.2 Preferred fragment size

The encoding of the Preferred Fragment Size field, as described in 6.4.12, is given in Table 12-4.

Table 12-4—Preferred fragment size field encoding

Field value	Preferred fragment size (octets)
0b000	1 048 576
0b001	262 144
0b010	65 536
0b011	16 384
0b100	4096
0b101	2048
0b110	512
0b111	Reserved

12.1.9.3 Receive Status field

The Receive Status field is used to send information about the received frame to the transmitter, as described in 6.2.7. The Receive Status field shall be formatted as illustrated in Figure 12-3.

Bits: 0	b1–b4	b5–b8	b9–b10	b11–b14	b15–b16	b17	b18–b19	b20–b23
Valid	RSSIR	SINR	Suggested Preamble Type	FER	Suggested Pilot Word Type	Suggested PCES	Suggested MCS Status	Reserved

Figure 12-3—Receive Status field format for mmWave PHY

The Valid field shall be set to one if the information in the Receive Status field is to be accessed by the receiver and shall be set to zero otherwise.

The RSSIR field contains the difference of the received signal power in dBm and the sensitivity in dBm of the selected MCS. The range of the RSSIR field is from 0 dB to 28 dB in 2 dB steps with 0b0000 corresponding to less than or equal to 0 dB, and 0b1111 corresponding to greater than 28 dB. For example, an RSSIR value that is greater than or equal to 8 dB but less than 10 dB would be encoded as 0b0101.

The SINR field contains the estimated signal-to-interference-plus-noise ratio of the received frame, and is defined to be $E_b/(I + N)$ in dB where E_b is the energy per bit, I is the interference power. and N is the noise power over the bandwidth. The range of the SINR field is from 2 dB to 28 dB in 2 dB steps with 0b0001 corresponding to less than or equal to 2 dB, 0b1111 corresponding to greater than 28 dB, and 0b0000 reserved. For example, an SINR value that is greater than or equal to 18 dB but less than 20 dB would be encoded as 0b1010.

Valid values of the Suggested Preamble Type field are as follows:

0b00 → Suggest CMS preamble
0b01 → Suggest SC preamble
0b10–0b11 → Reserved

Valid values of the Suggested Pilot Word Type field are as follows:

0b00 → Suggest SC pilot word length of 64
0b01 → Suggest SC pilot word length of 8
0b10 → Suggest SC pilot word length of 0
0b11 → Reserved

The Suggested PCES field shall be set to zero to suggest that no PCES should be sent and shall be set to one to suggest the use of PCES.

The FER field contains the exponent of the estimate of the FER ranging from 10^{-1} to 10^{-10} in steps of 10^{-1} with 0b0000 corresponding to an FER exponent of less than or equal to –1 (i.e., an FER greater than or equal to 10^{-1}), 0b1010 corresponding to an FER exponent of greater than –10 (i.e., an FER of less than 10^{-10}), and 0b1011–0b1111 reserved. For example, an FER field value of 0b0110 indicates an exponent of –6 and that the FER was less than or equal to 10^{-6} but was greater than 10^{-7}.

Valid values of the Suggested MCS Status field are as follows:

0b00 → Use an MCS with lower required SNR
0b01 → Use current MCS
0b10 → Use an MCS with higher required SNR
0b11 → Reserved

12.1.10 Requirements for mmWave PNCs

In order to promote coexistence among DEVs using different PHYs, the following MAC rules have defined:

— An AV PNC-capable DEV, when operating as a PNC, shall transmit an AV beacon and a Sync frame using CMS in every superframe.
— An HSI PNC-capable DEV, when operating as a PNC, shall transmit an HSI beacon and a Sync frame using CMS in every superframe.
— An AV PNC-capable DEV shall be able to receive Sync frames and MAC Command frames that use CMS.
— An HSI PNC-capable DEV shall be able to receive Sync frames and MAC Command frames that use CMS.
— A DEV capable of transmitting a Sync frame may do so in the first granted CTA in a superframe, and in every pre-defined number of superframes for the directions indicated in the Sync Frame Frequency IE, as indicated in 6.4.42.

A DEV is defined as being able to receive a CMS frame if it can at least successfully perform detection of the CMS preamble when the signal power is greater than the receiver sensitivity.

12.1.11 CP operation

A PNC-capable DEV compliant to this standard shall allow the use of the CAP for contention-based access for association, data, and commands when using the mmWave PHY, as described in 6.3.1. A DEV compliant to this standard shall support the use of the CAP. CSMA/CA shall be used for all CPs in a mmWave PHY piconet.

The CPs in a piconet shall all be conducted using the same mmWave PHY mode, one of SC, HSI or AV, as the beacon. Any MCS that is supported by both the source and destination and that is in the same PHY mode as the beacon may be used in a CP, with the exception of the AV HRP modes, which shall not be used in a CP. The usage of directional CAPs is described in 7.8.6.3.

For medium access in the CP, the following rules apply:

— A DEV attempting to access the medium in an SC CAP for data transmission, shall always use frames consisting of the CMS preamble, as described in 12.1.13.5, the MR header, as described in 12.2.3.2, and any SC MCS, as described in 12.2.3.1, using the best TX direction over regular CAP and S-CAP.
— A DEV attempting to access the medium in an HSI CAP for data transmission shall always do so by using frames consisting of the HSI MCS0 long preamble, as described in 12.3.4.2, and any HSI MCS, as described in 12.3.3.1, using the best TX direction over regular CAP and S-CAP.
— A DEV attempting to access the medium in an AV CP shall always do so by using omni low-rate protocol data unit (LRPDU), as described in 12.4.4.

12.1.12 mmWave PHY mode usage in CTA

In a CTA, any PHY mode may be used that is supported by both the source and destination DEVs.

Examples of how this is used in a mmWave PHY piconet are given in F.3.

12.1.13 Common mode signaling (CMS)

12.1.13.1 General

CMS is a low data rate SC PHY mode specified to enable interoperability among different PHY modes. The CMS is used for transmission of the Beacon frame defined in 7.8.3, and, if supported, the Sync frame defined in 6.3.9 and 12.1.10. The CMS is also used for transmission of command frame and training sequence in the beam forming procedure, as defined in Clause 15, for the SC and HSI PHYs.

The CMS frame shall be formatted as illustrated in Figure 12-4.

CMS PHY Preamble	CMS Frame Header	CMS PHY Payload

Figure 12-4—CMS frame format

The CMS PHY preamble is defined in 12.1.13.5.

The CMS Frame Header shall be formatted as illustrated in Figure 12-5. The CMS frame header shall be constructed as described in 12.1.13.6 and 12.1.13.7.

PHY Header	MAC Header	HCS	RS Parity

Figure 12-5—CMS Frame Header format

The CMS PHY Payload field of a shall be constructed as described in 12.1.13.8.

The chip rate of CMS is 1760 Mchip/s. The entire CMS frame shall be modulated with $\pi/2$ BPSK, as specified in 12.2.3.5.2. The FEC for the CMS frame header and MAC Frame Body field shall be as specified in 12.1.13.2. The CMS Frame Header field and MAC Frame Body field shall be spread as specified in 12.1.13.3. The CMS preamble shall be excluded from the spreading process. The scrambling process shall be specified in 12.1.13.4.

The header rate shall be as defined in Table 12-12 while the PHY payload rate shall be as defined in Table 12-10.

12.1.13.2 Forward error correction for CMS

The FEC scheme for CMS shall be RS coding. The RS(255,239), which is the mother code, shall be used for encoding the MAC Frame Body field of CMS. The RS(n +16, n), a shortened version of RS(255,239) where n is the number of octets in the frame header, shall be used for encoding the frame header of CMS. Details of the coding are provided in 12.2.3.6.2.

12.1.13.3 Code spreading for CMS

To increase robustness in the frame header and MAC Frame Body field of the CMS, code spreading shall be applied using Golay sequences. The code spreading factor shall be 64, and the Golay sequence specified in Table 12-5 shall be used. The frame header and the MAC Frame Body field shall be spread according to Figure 12-6. Note that in each hexadecimal-equivalent 4-binary-digit group, the leftmost bit shall be the MSB, and the rightmost bit, the LSB. For example, 3 is denoted as 0011. The order of the octets and bits over the air is the same as defined in 6.1.

Table 12-5—Golay sequences with length 64

Sequence name	Sequence value
$\mathbf{a}_{64}$	0x63AF05C963500536
$\mathbf{b}_{64}$	0x6CA00AC66C5F0A39

As shown in Figure 12-6, a PRBS generated using an LFSR is used to vary the spreading code from one bit to another, i.e., each bit is spread with Golay code $\mathbf{a}_{64}$ or Golay code $\mathbf{b}_{64}$ depending on the LFSR output. The LFSR shall be run at a rate equals to 1/64 the bit rate and shall be initialized with the following 15-bit seed value: $[x_{-1}, x_{-2}, \ldots, x_{-15}]$ =[0101 0000 0011 111]. Each input bit shall be held for 64 chips as it is XORed with Golay code $\mathbf{a}_{64}$ or Golay code $\mathbf{b}_{64}$ operating at the chip rate.

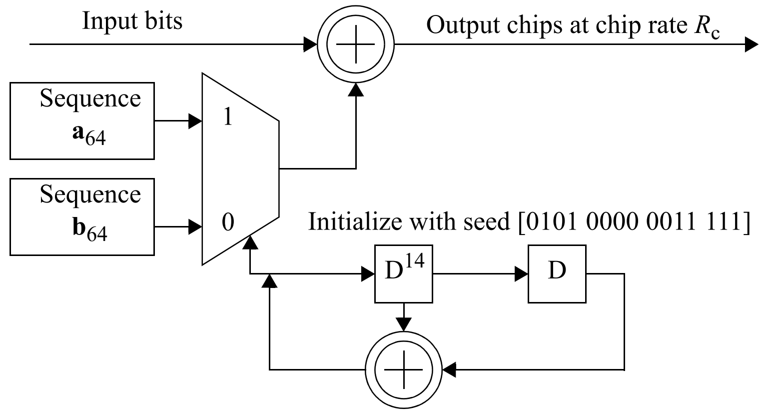

Figure 12-6—Realization of the CMS code spreading

12.1.13.4 Scrambling for CMS

Scrambling shall be employed to whiten the CMS frames. Scrambling shall be used for the MAC header, HCS, and the entire PHY Payload field as detailed in 12.2.3.10.

12.1.13.5 PHY preamble for CMS

A PHY preamble is used to aid receiver algorithms related to AGC setting, timing acquisition, frame synchronization, and channel estimation. The CMS preamble shall be formatted as illustrated in Figure 12-7.

CMS preamble		
Sync	SFD	CES

Figure 12-7—PHY preamble structure for CMS

The Sync field is used primarily for frame detection and shall consist of 48 repetitions of $\mathbf{b}_{128}$.

The SFD field is used to establish frame timing and shall consist of [+1 –1 +1 +1 –1 –1 –1] spread by $\mathbf{b}_{128}$.

The CES field is used for channel estimation and shall consist of $\mathbf{b}_{128}$ $\mathbf{b}_{256}$ $\mathbf{a}_{256}$ $\mathbf{b}_{256}$ $\mathbf{a}_{256}$ $\mathbf{a}_{128}$ with $\mathbf{a}_{128}$ first in time.

The Golay complimentary sequences of length 128, denoted by $\mathbf{a}_{128}$ and $\mathbf{b}_{128}$, are shown in Table 12-6. The Golay code complimentary sequences of length 256, denoted by $\mathbf{a}_{256}$ and $\mathbf{b}_{256}$, are defined as follows:

$$\mathbf{a}_{256} = [\mathbf{a}_{128}\ \mathbf{b}_{128}]$$

$$\mathbf{b}_{256} = [\mathbf{a}_{128}\ \overline{\mathbf{b}}_{128}]$$

where the number on the right ($\mathbf{b}_{128}$ and $\overline{\mathbf{b}}_{128}$) are the first in time and the binary complement of a sequence x is denoted by an overline on x (i.e., $\overline{x}$).

Alternatively, the CMS preamble may be used by the SC PHY for data transmission in the CAP, as specified in 12.2.4.2.

Table 12-6—Golay sequences with length 128

Sequence name	Sequence value
$\mathbf{a}_{128}$	0x0536635005C963AFFAC99CAF05C963AF
$\mathbf{b}_{128}$	0x0A396C5F0AC66CA0F5C693A00AC66CA0

12.1.13.6 Frame header for CMS

The frame header conveys information in the PHY and MAC headers necessary for successfully decoding the frame. The construction of the CMS header is shown in Figure 12-8.

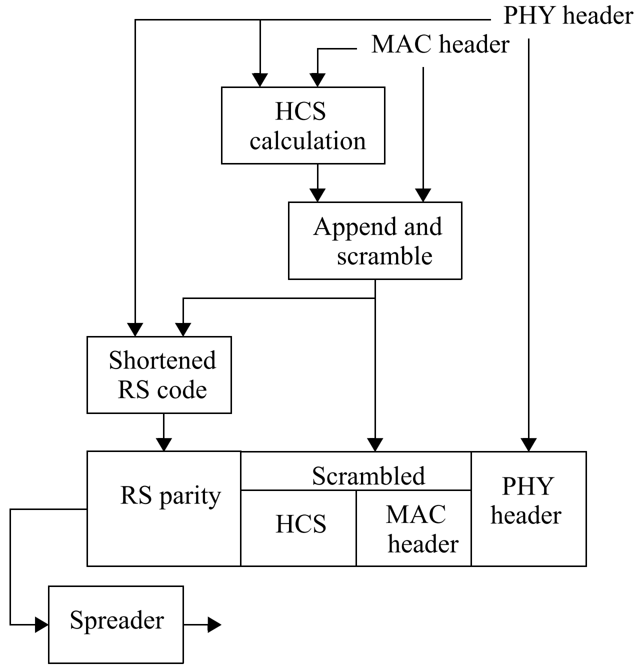

Figure 12-8—Frame header construction process for CMS

The detailed process of the construction is as follows:

a) Construct the PHY header, as described in 12.1.13.7.
b) Compute the HCS as described in 12.2.4.3.3 over the combined PHY and MAC headers.
c) Append the HCS to the MAC header.
d) Scramble the combined MAC header and HCS, as described in 12.1.13.4.
e) Compute the RS Parity field by encoding the concatenation of the PHY Header field, scrambled MAC Header field, and scrambled HCS field into a shortened RS block code, as described in 12.1.13.2.
f) Form the base frame header by concatenating the PHY Header field, scrambled MAC Header field, scrambled HCS field, and RS Parity field.
g) Spread the frame header, as described in 12.1.13.3.

12.1.13.7 PHY header for CMS

The CMS PHY header shall be formatted as illustrated in Figure 12-22. The description of each field is provided in 12.2.4.3.2. In this subclause, the field values for the CMS PHY header are specified.

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 12.1.13.4.

The Aggregation field shall be set to zero.

The UEP field shall be set to zero.

The MCS field shall be set to 0b00000.

The Frame Length field shall be an unsigned integer that indicates the number of octets in the MAC Frame Body field, excluding the FCS.

The Preamble Type field shall be set to 0b00.

The Beam Tracking field shall be set to one if the training sequence for beam tracking is following the current frame, and shall be set to zero otherwise.

The Low-latency mode field shall be set to zero.

The Pilot Word Length field shall be set to 0b10.

The PCES field shall be set to zero.

12.1.13.8 PHY Payload field for CMS

The PHY Payload field is the last component of the CMS frame, and is constructed as shown in Figure 12-9.

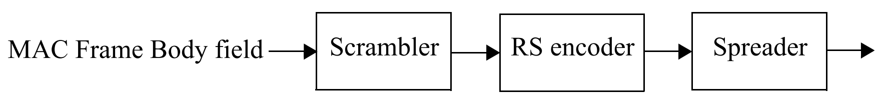

Figure 12-9—PHY Payload field construction process for CMS

The PHY Payload field of the CMS shall be constructed as follows:

a) Scramble the MAC Frame Body field according to 12.1.13.4.
b) Encode the scrambled MAC Frame Body field, as specified in 12.1.13.2.
c) Spread the encoded and scrambled MAC Frame Body field using the spreading code, as detailed in 12.1.13.3.

12.1.13.9 Receiver clear channel assessment performance for CMS

The start of a valid preamble sequence at a receive level equal to or greater than the minimum sensitivity for the CMS, as described in 12.2.6.2, shall cause CCA to indicate medium busy with a probability of > 90% within *pCcaDetectTime* for the SC PHY mode. The receiver CCA function shall in all circumstances report medium busy with any signal 20 dB above the minimum sensitivity for the CMS. The CCA shall be maintained as busy until the end of the frame.

12.1.14 mmWave PHY PIB

The PHY-dependent PIB values for the mmWave PHY are given in Table 12-7 and Table 12-8. The PHY PIB characteristics group, Table 12-7, contains information that is common to most implementations.

The PHY PIB implementation group, Table 12-8, contains information that is more characteristic of a particular PHY implementation than of the PHY as a whole.

Table 12-7—PHY PIB characteristics group parameters

Managed Object	Octets	Definition	Access
phyType	1	0x01 = mmWave PHY	Read/write
phyMode	1	bit 1 = SC PHY bit 2 = HSI PHY bit 3 = AV PHY bit 4 = OOK mode bit 5 = DAMI mode bit 6–8 = Reserved A bit is set to one if the associated PHY is supported, and is set to zero otherwise.	Read/write
phyRegDomainsSupported	Variable	One octet for each regulatory domain supported, as defined for *phyCurrentRegDomain*.	Read/write
phyCurrentRegDomain	1	0x00 = European Telecommunications Standards Institute 0x01 = Federal Communications Commission 0x02 = Industry Canada 0x03 = Association of Radio Industries and Businesses	Read/write
phyDataRateVector	Variable	One octet for each supported MCS. The three MSBs indicate the mmWave PHY mode, as in *phyMode*, and the last five LSBs contain the MCS supported for that mode using the encoding for that PHY mode.	Read/write
phyNumChannelsSupported	Variable	As defined in 12.1.6.	Read/write
phyCurrentChannel	1	Indicates the channel that is currently being used, as defined in 12.1.6.	Read/write
phyCcaThreshold		The CCA threshold in dBm, encoded in two's complement format. The value is implementation dependent but no larger than the value listed in 12.1.13.9.	Read/write
phyFrameLengthMax	2	The value of *pMaxFrameBodySize*	Read/write

Table 12-8—PHY PIB Implementation group parameters

Managed Object	Octets	Definition	Access
phyDiversitySupported	1	Numeric entry that indicates the number of antennas that are available.	Read/write
phyMaxTXPower	1	The maximum TX power that the DEV is capable of using, 6.4.12, implementation dependent.	Read/write
phyTXPowerStepSize	1	The step size for power control supported by the DEV, 6.4.19, implementation dependent.	Read/write
phyNumPMLevels	1	Number of power management levels supported. The range is 1 to 8 and the value is implementation dependent.	Read/write
phyPMLevelReturn	Variable	Table of vectors with number of entries given by *phyNumPMLevels*. Each vector is the time required to change between power saving states of the PHY. Vector number 0 is the time required to change the PHY from the off state to a state where it is ready to receive commands. Other values are implementation dependent.	Read/write

12.2 Single Carrier Mode of mmWave PHY (SC PHY)

12.2.1 General

The SC PHY provides three classes of MCSs targeting different wireless connectivity applications. Class 1 is specified to address the low-power low-cost mobile market while maintaining a relatively high data rate of up to 1.5 Gb/s. Class 2 is specified to achieve data rates up to 3 Gb/s. Class 3 is specified to support high performance applications with data rates in excess of 5 Gb/s. Table 12-9 summarizes the MCS classes. There are two mandatory MCSs for all SC DEVs except for the optional OOK/DAMI mode DEVs, the CMS, and the mandatory PHY rate (MPR). The optional OOK/DAMI modes, as described in 12.2.9, are allowed for low complexity SC DEVs. The CMS shall be the base rate for the SC PHY. The CMS is used for transmission of beacon and command frame in the association procedure as well as for transmission of command frame and training sequence in the beam forming procedure.

Table 12-9—MCS categorization for the SC PHY

Class	Categorization
Class 1	Data rates < 1.5 Gb/s[a]
Class 2	1.5 Gb/s < Data rates < 3 Gb/s
Class 3	Data rates >3 Gb/s

[a]The data rates are based on subblock length of 512 and pilot word length of 64.

The SC PHY is specified with a high degree of flexibility in order to allow implementers the ability to optimize for different applications. The SC PHY supports NLOS, as well as line-of-sight (LOS), operation, with or without equalization. The data is divided into blocks, each block is divided into subblocks, as described in 12.2.4.5.1, and each subblock may be equalized in time, frequency or in hybrid time-frequency domain. A subblock consists of pilot word and data.

The SC PHY supports π/2 BPSK, π/2 QPSK, π/2 8-PSK, and π/2 16-QAM modulations. Two main FEC schemes are specified: RS block codes and LDPC block codes. The RS(255,239) and the shortened RS(33,17) block codes are mandatory, whereas all the other FECs are optional.

12.2.2 PHY operating specifications of SC PHY

12.2.2.1 Channelization

The RF channels are defined in Table 12-3. A compliant implementation shall support at least one channel from the channels allocated for operation by its corresponding regulatory body.

The *phyCurrentChannel* is the CHNL_ID of the current channel. For the purpose of the Remote Scan Request and Remote Scan Response commands, as described in 6.5.8.4 and 6.5.8.5, respectively, the Channel Index field is the CHNL_ID in Table 12-3.

12.2.2.2 Scanning channels

When a DEV is scanning to start a piconet, it should scan all channels it supports in respective regions to decrease the probability of choosing an occupied channel.

12.2.3 Modulation, forward error correction, and spreading

12.2.3.1 MCS-dependent parameters

The chip rate for all SC PHY MCS is given in Table 12-14. The MCS-dependent parameters shall be set according to Table 12-10. The data rates in the table are approximate and are calculated to three significant figures. The CMS and MPR are part of Class 1, and are listed as MCS identifier 0 and 3, respectively.

Table 12-10—MCS-dependent parameters

MCS class	MCS identifier	Data rate (Mb/s) with pilot word length = 0	Data rate (Mb/s) with pilot word length = 64	Modulation	Spreading factor, L_{SF}	FEC type
Class1	0	25.8 (CMS)	—	π/2 BPSK	64	RS(255,239)
	1	412	361		4	
	2	825	722		2	
	3	1650 (MPR)	1440		1	
	4	1320	1160	π/2 BPSK	1	LDPC(672,504)
	5	440	385	π/2 BPSK	2	LDPC(672,336)
	6	880	770		1	

Table 12-10—MCS-dependent parameters (*continued*)

MCS class	MCS identifier	Data rate (Mb/s) with pilot word length = 0	Data rate (Mb/s) with pilot word length = 64	Modulation	Spreading factor, L_{SF}	FEC type
Class2	7	1760	1540	π/2 QPSK	1	LDPC(672,336)
	8	2640	2310	π/2 QPSK	1	LDPC(672,504)
	9	3080	2700	π/2 QPSK	1	LDPC(672,588)
	10	3290	2870	π/2 QPSK	1	LDPC(1440,1344)
	11	3300	2890	π/2 QPSK	1	RS(255,239)
Class3	12	3960	3470	π/2 8-PSK	1	LDPC(672,504)
	13	5280	4620	π/2 16-QAM	1	LDPC(672,504)

The FEC rate, R_{FEC}, is for an RS(n, m) code or an LDPC(n, m) code is m/n. The subblock length $L_{subblock}$ shall be 512 chips. The pilot word length L_{PW} shall be 0, 8, or 64 chips. For subblocks with L_{PW} = 8, the effective length of the pilot word and data is equivalent to that with L_{PW} = 64, as described in 12.2.4.5.1.

The MCS-dependent parameters for the optional OOK/DAMI modes are given in Table 12-11.

Table 12-11—MCS-dependent parameters for optional OOK/DAMI modes

Device Type	MCS identifier	Data rate[a] (Mb/s)	Modulation	Spreading factor	FEC type	Support for CMS
PNC-capable DEVs	OOK	25.8 (CMS)	π/2 BPSK	64	RS(255,239)	Mandatory
		818	OOK	2		
		1640		1		
	DAMI	25.8 (CMS)	π/2 BPSK	64		
		3270	DAMI	1		
Non PNC-capable DEVs	OOK	818	OOK	2	RS(255,239)	Not mandatory
		1640		1		
	DAMI	3270	DAMI	1		

[a]The data rate of CMS is calculated in a manner similar to Table 12-10. The data rates of two OOK modes are calculated based on pilot word design as in G.5. The data rate of DAMI mode is calculated based on subblock length of 512 and pilot word length of 4. The data rates are approximate and are rounded to three significant figures.

12.2.3.2 Header rate-dependent parameters

The base header rate-dependent parameters shall be set according to Table 12-12. The base headers use a shortened RS code, as defined in 12.2.4.3.4.

For the MR and HR headers, the header subblock is divided into eight sub-subblocks. The first sub-subblock is a pilot word with L_{PW} = 64. This is followed by six sub-subblocks, each with a prepended pilot word L_{PW} = 8. The last sub-subblock consists of eight pilot words of L_{PW} = 8. This gives the effective pilot word length of L_{PW} = 176.

Table 12-12—Base header rate-dependent parameters

Header class	Header rate (Mb/s)	Modulation scheme	Spreading factor, L_{SF}	Pilot word length (chips), L_{PW}	Coded bits per subblock, L_{CBPS}	Number of occupied subblocks, $N_{subblock\_hdr}$	Number of stuff bits, L_{STUFF}
CMS rate	12.5	π/2 BPSK	64	0	8	33	0
Medium rate (MR)	82.5	π/2 BPSK	6	176 (effective)	56	5	16
High rate (HR)	206	π/2 BPSK	2		168	2	72

There are three types of MAC subheaders: the standard aggregation subheader, the low-latency EEP subheader, and the low-latency UEP subheader, as defined in 6.2.11. When the MAC subheader is present, the subheader shall be modulated with π/2 BPSK and the header rate-dependent parameters shall be set according to Table 12-13. The subheader subblock is the same as for the base header, giving the same effective pilot word length of 176 chips. The MAC subheaders for standard aggregation use a shortened RS code for the FEC, as defined in the 12.2.4.3.6. If the base header uses either the CMS or MR, the MAC subheader shall be sent using the subheader MR. If the base header uses HR, the MAC subheader shall be sent using subheader HR. For the standard aggregation frames, each subframe may use a different MCS. For the low-latency frames, the MAC subheaders and the MSDU subheaders shall use the same MCS for the MAC Frame Body field, thus the MCS remains the same within a low-latency aggregation frame.

Table 12-13—MAC subheader rate-dependent parameters for standard aggregation

Frame type	Subheader class	Header rate[a] (Mb/s)	Spreading factor, L_{SF}	Coded bits per subblock, L_{CBPS}	Number of occupied subblocks	Number of stuff bits, L_{STUFF}
Non-secure frame	Subheader MR	131	6	56	9	16
	Subheader HR	394	2	168	3	16
Secure frame	Subheader MR	132	6	56	10	32
	Subheader HR	330	2	168	4	144

[a]Only the parameters for the MAC subheader for standard aggregation is presented in the table. The MAC subheader for low-latency aggregation uses the same MCS as for the MAC Frame Body field and therefore is not listed in this table.

12.2.3.3 Timing-related parameters

Table 12-14 lists the general timing parameters associated with the SC PHY.

12.2.3.4 Frame-related parameters

The frame parameters associated with the PHY are listed in Table 12-15 where CEIL is the ceiling function, which returns the smallest integer value greater than or equal to its argument. The maximum frame duration occurs when the number of octets in the PHY Payload field is 1 048 576.

Table 12-14—Timing-related parameters

Parameter	Description	Value			Unit	Formula
R_c	Chip rate	1760			Mchip/s	
T_c	Chip duration	~0.568			ns	$1/R_c$
$L_{subblock}$	Subblock length	512			chips	
L_{PW}	Pilot word length	0	8[a]	64	chips	
T_{PW}	Pilot word duration	0	4.5	~37.0	ns	
L_{DC}	Length of data chips per subblock	512	56	448	chips	
$T_{subblock}$	Subblock duration	~290.9			ns	$L_{subblock} \times T_c$
$R_{subblock}$	Subblock rate	~3.44			MHz	$1/T_{subblock}$

[a]Details of the subblock with pilot word length 8 are given in 12.2.3.1 and 12.2.3.2.

Table 12-15—Frame-related parameters

Parameter	Description	Value
N_{SYNC}	Number of code repetitions in the Sync sequence[a]	14
T_{SYNC}	Duration of the Sync sequence	~1 µs
N_{SFD}	Number of code repetitions in the SFD	4
T_{SFD}	Duration of the SFD	~0.29 µs
N_{CES}	Number of code repetitions in the CES	9
T_{CES}	Duration of the CES	~0.65 µs
N_{pre}	Number of code repetitions in the PHY preamble	27
T_{pre}	Duration of the PHY preamble	~1.96 µs
L_{hdr}	Length of the base header in octets	33
$N_{subblock\_hdr}$	Number of subblocks in the base frame header[b]	$\mathrm{CEIL}[L_{hdr} \times 8 \times L_{SF} / (L_{subblock} - L_{PW})]$
T_{hdr}	Duration of the base frame header	$N_{subblock\_hdr} \times T_{subblock}$
$L_{payload}$	Length of frame payload in octets	variable
L_{FCS}	Length of FCS in octets	4
L_{MFB}	Length of the MAC Frame Body field in octets	$L_{payload} + L_{FCS}$
N_{PCES}	Number of code repetitions in the PCES	9
T_{PCES}	PCES duration	0.65 µs
N_{PCES_frame}	Number of PCESs per frame	$\mathrm{CEIL}[(L_{MFB} \times 8) / (L_{subblock} \times 64)]–1$
$T_{PCES\_interval}$	Interval of PCES insertion	$T_{subblock} \times 64 + T_{PW}$
L_{CBPS}	Number of coded bits per subblock in the MAC Frame Body field [c]	$(L_{subblock} - L_{PW}) / L_{SF}$
$N_{subblock\_MFB}$	Number of subblocks in the MAC Frame Body field	$\mathrm{CEIL}[(L_{MFB} \times 8) / (R_{FEC} \times L_{CBPS})]$

Table 12-15—Frame-related parameters (*continued*)

Parameter	Description	Value
T_{MFB}	Duration of the MAC Frame Body field	$N_{\mathrm{subblock\_MFB}} \times T_{\mathrm{subblock}}$
$T_{\mathrm{datafield}}$	Duration of the PHY Payload field	$T_{\mathrm{MFB}} + (N_{\mathrm{PCES\_frame}} + 1) \times T_{\mathrm{PW}} + N_{\mathrm{PCES\_frame}} \times T_{\mathrm{PCES}}$
T_{frame}	Duration of the frame	$T_{\mathrm{pre}} + T_{\mathrm{hdr}} + T_{\mathrm{datafield}}$

[a]A *code repetition* is defined as a code with length of 128 chips.
[b]The value for L_{PW} given here is the effective pilot word length, as defined in 12.2.3.2.
[c]The value for L_{PW} given here is the effective pilot word length, as defined in 12.2.3.1.

NOTE—Regardless of the maximum frame length, no frame is allowed to exceed the timing boundaries, e.g., CAP end time or CTA end time.

12.2.3.5 Modulation

12.2.3.5.1 General

After channel encoding and spreading, the bits shall be inserted into the constellation mapper. The constellations used for the SC PHY are illustrated in Figure 12-10. The mapping rules and diagrams for skewed constellation are given in 12.3.3.6.

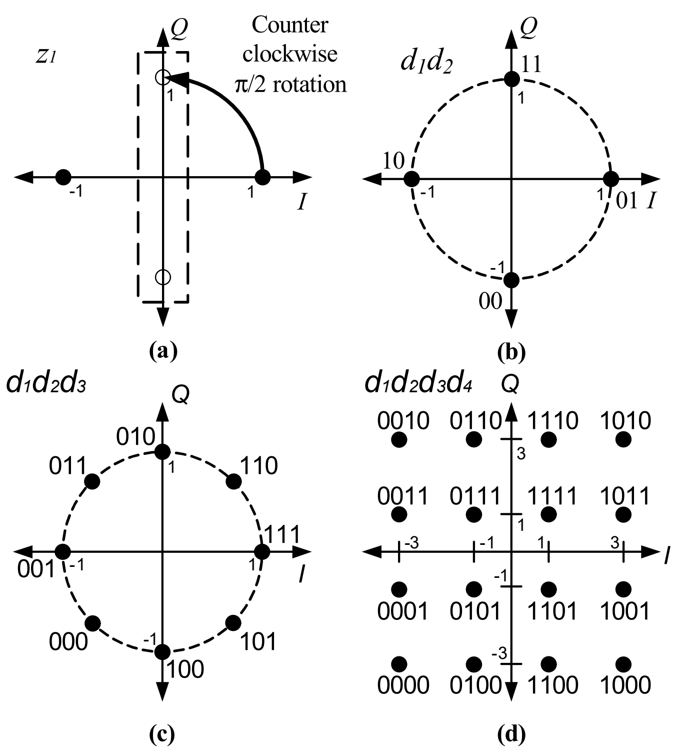

Figure 12-10—Constellation maps for modulations: (a) π/2 BPSK, (b) π/2 QPSK, (c) π/2 8-PSK, (d) π/2 16-QAM

12.2.3.5.2 π/2 BPSK

The π/2 BPSK modulation is a binary phase modulation with π/2 phase shift counter-clockwise. Figure 12-10(a) shows the signal constellation of π/2 BPSK signals, and Figure 12-11 shows the π/2 BPSK modulator. The data chips, g_n, at the output of PCES inserter shown in Figure 12-23 are mapped to $d_1, d_2, \ldots, d_N$, where $d_N = 2 \times g_n$ 1. The d_n values are mapped onto constellation points z_n as shown in Equation (12-3):

$$z_n = j^n \times d_n,\ n = 1, 2, \ldots, N \qquad (12\text{-}3)$$

where j denotes π/2 phase rotation. In Figure 12-11, c_n denotes a complex envelope of filtered π/2 BPSK signal.

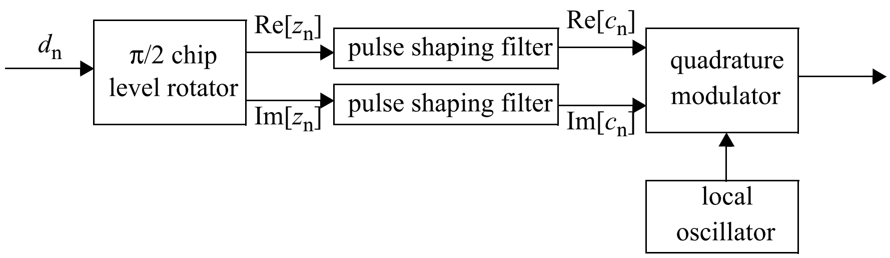

Figure 12-11—Possible π/2 BPSK modulator realizations

The filtered waveform of the signal shall satisfy transmit PSD mask as in 12.1.7.

12.2.3.5.3 π/2 QPSK

The π/2 QPSK constellation diagram is shown in Figure 12-10(b), with four points equally spaced on a circle of radius one, representing four phases. QPSK shall encode 2 bits per symbol, with input bit d_1 being the earliest in the stream. The π/2 shift is employed to obtain a simple implementation aligning with the π/2 BPSK. The π/2 rotation is performed in the same manner as in 12.2.3.5.2. As illustrated in Figure 12-10(c), Gray encoding shall be employed. The normalization factor, K_{MOD}, is 1.

12.2.3.5.4 π/2 8-PSK

The π/2 8-PSK constellation diagram is shown in Figure 12-10(c), equally spaced on a circle of radius one, representing eight phases. The π/2 8-PSK shall encode 3 bits per symbol, with input bit d_1 being the earliest in the stream. The π/2 rotation is performed in the same manner as in 12.2.3.5.2. Gray encoding shall be employed in the mapping of π/2 8-PSK. The normalization factor, K_{MOD}, is 1.

12.2.3.5.5 π/2 16-QAM

The π/2 16-QAM constellation diagram is shown in Figure 12-10(d). The serial bit stream shall be divided into groups of four bits with input bit d_1 being the earliest in the stream. The π/2 rotation is performed in the same manner as in 12.2.3.5.2. The normalization factor for π/2 16-QAM constellation is $1/\sqrt{10}$. An approximate value of the normalization factor may be used, as long as the device conforms to the modulation accuracy requirements.

12.2.3.6 FEC

12.2.3.6.1 General

The FEC schemes are specified in the following subclauses. Support for RS block codes is mandatory, whereas support for LDPC block codes is optional.

12.2.3.6.2 Reed-Solomon block codes in GF(2^8)

The RS(255,239), which is the mother code, is used in payloads of CMS, MPR and MCS identifier 1 MCSs in Table 12-10. A shortened version of RS(255,239) is used for the base frame header and MAC subheader, as defined in 12.2.3.2.

The systematic RS code shall use the generator polynomial shown in Equation (12-4):

$$g(x) = \prod_{k=1}^{16} (x + \alpha^k) \tag{12-4}$$

where $\alpha = 0x02$ is a root of the binary primitive polynomial $p(x)=1+x^2+x^3+x^4+x^8$. As notation, the element $M = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$, has the binary representation $b_7b_6b_5b_4b_3b_2b_1b_0$, where b_7 is the MSB and b_0 is the LSB.

The mapping of the information octets $\boldsymbol{m} = (m_{238}, m_{237}, \ldots, m_0)$ to codeword octets $\boldsymbol{c} = (m_{238}, m_{237}, \ldots, m_0, r_{15}, r_{14}, \ldots, r_0)$ is achieved by computing the remainder polynomial $r(x)$, as shown in Equation (12-5):

$$r(x) = \sum_{k=0}^{15} r_k x^k = x^{16} m(x) \bmod g(x) \tag{12-5}$$

where $m(x)$ is the information polynomial:

$$m(x) = \sum_{k=0}^{238} m_k x^k$$

and r_k, $k = 0, \ldots, 15$, and m_k, $k = 0, \ldots, 238$, are elements of GF(2^8). The message order is as follows: m_{238} is the first octet of the message and m_0 is the last octet of the message.

For a shortened RS(L_{inf} + 16, L_{inf}), 239-L_{inf} zero elements are appended to the incoming L_{inf} octet message as follows:

$m_k = 0$, $k = L_{inf}, \ldots, 238$

These inserted zero elements are not transmitted. A shift-register implementation of the RS encoder RS(L_{inf}+16, L_{inf}) is shown in Figure 12-12, with additions and multiplications over GF(2^8). After m_0 has been inserted into the shift register, the switch shall be moved from the message polynomial input connection to the shift register output connection (right-to-left).

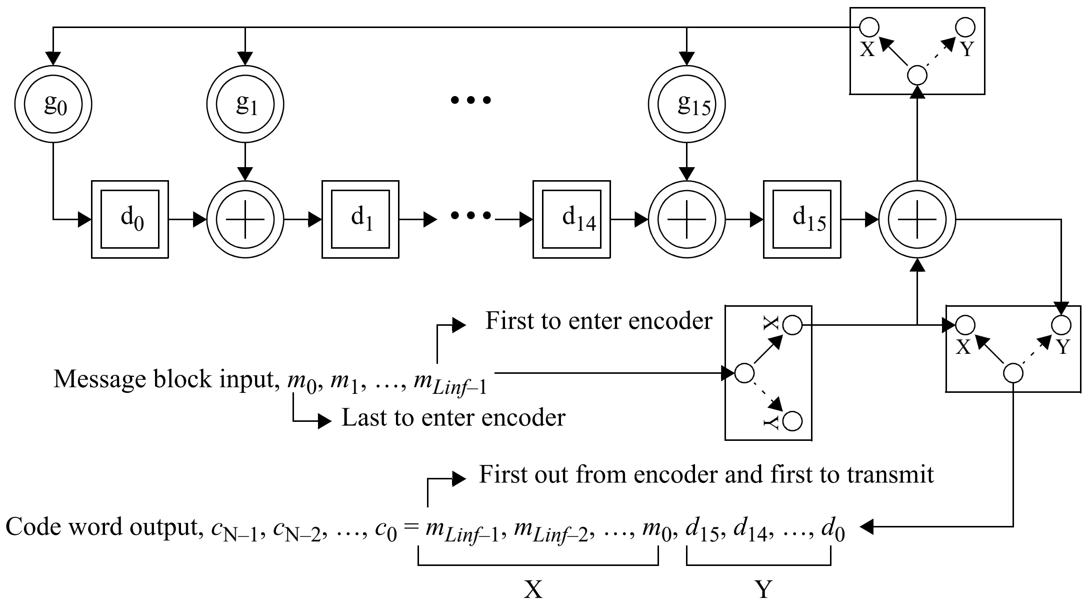

Figure 12-12—Reed Solomon encoder GF(2^8)

12.2.3.6.3 Irregular LDPC codes

The irregular LDPC codes are used as a high-performance error correction coding technique. The supported FEC rates, information block lengths, and codeword block lengths are described in Table 12-16.

Table 12-16—Irregular LDPC parameters

FEC rate, R_{FEC}	LDPC information block length (bits), L_{inf}	LDPC codeword block length (bits), L_{FEC}
1/2	336	672
3/4	504	672
7/8	588	672

The LDPC encoder is systematic, i.e., it encodes an information block of size k, $i = (i_0, i_1, \ldots, i_{(k-1)})$, into a codeword $\mathbf{c}$ of size n, $\mathbf{c} = (i_0, i_1, \ldots, i_{(k-1)}, p_0, p_1, \ldots, p_{(n-k-1)})$, by adding n-k parity bits obtained so that $\mathbf{Hc}^{\mathbf{T}} = 0$, where $\mathbf{H}$ is an $(n - k) \times n$ parity check matrix.

Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $z \times z$ ($z = 21$). These submatrices are either cyclic-permutations of the identity matrix or null (all-zero) submatrices.

The cyclic-permutation matrix $\mathbf{p}^i$ is obtained from the $z \times z$ identity matrix by cyclically shifting the columns to the left by i elements. The matrix $\mathbf{p}^0$ is the $z \times z$ identity matrix.

In the following, an example of cyclic-permutation matrices with $z = 21$ is shown. The matrix $\mathbf{p}^1$ and $\mathbf{p}^2$ are produced by cyclically shifting the columns of the identity matrix $\mathbf{I}_{21\times21}$ to the left by 1 and 2 places, respectively.

$$p^0 = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & 0 & \dots & 0 & \dots \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}, p^1 = \begin{bmatrix} 0 & \dots & \dots & 0 & 1 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \dots & 0 & 1 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}, p^2 = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & 0 & 1 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

where all the above matrices have dimension 21 × 21.

Due to the cyclic permutation, $\mathbf{p}^{21} = \mathbf{p}^0 = \mathbf{I}_{21\times21}$.

Figure 12-13 displays the matrix permutation indices of parity-check matrices for all three FEC rates at block length $n = 672$ bits. The integer i denotes the cyclic-permutation matrix $\mathbf{p}^i$, as explained in the preceding example. The "–" entries in the table denote null (all zero) submatrices.

(672,336), Code rate: 1/2

	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	-	-	-	5	-	18	-	-	-	-	3	-	10	-	-	-	-	-	-	5	-	-	-	-	-	-	-	5	-	7	-	-
2	0	-	-	-	-	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	-	-	10	-	-	-	-	-	19
3	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	-	-	-	-	-	-	19	-	10	-	-	-
4	-	18	-	-	-	-	-	0	10	-	-	-	-	16	-	-	-	-	9	-	-	-	-	-	4	-	-	-	-	-	17	-
5	5	-	-	-	-	-	18	-	-	-	-	3	-	10	-	-	5	-	-	-	-	-	-	-	-	-	-	-	-	-	7	-
6	-	0	-	-	-	-	-	16	6	-	-	-	0	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	19	-	-	-
7	-	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	-	-	-	-	-	-	-	-	10	-	-
8	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	17
9	-	5	-	-	-	-	-	18	3	-	-	-	-	-	10	-	-	5	-	-	4	-	-	-	-	5	-	-	-	-	-	7
10	-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	-	-	-	-	7	-	4	-	-	-	-	-	10	-	19	-	-
11	6	-	-	-	-	-	7	-	-	-	-	2	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10	-
12	-	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	9	-	-	-	-	-	-	12	-	-	4	-	17	-	-	-
13	-	-	5	-	18	-	-	-	-	3	-	-	-	-	-	10	-	-	5	-	-	-	-	-	-	-	5	-	-	-	-	-
14	-	-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	-	-	10	-	-	-	-	-	-	-
15	-	6	-	-	-	-	-	7	2	-	-	-	-	9	-	-	-	-	-	20	-	-	-	-	-	19	-	-	-	-	-	-
16	18	-	-	-	-	-	-	0	-	-	-	10	16	-	-	-	-	9	-	-	-	-	-	-	-	-	-	4	-	-	-	-

(672,504), Code rate: 3/4

	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	0	-	-	5	-	18	16	-	-	-	3	6	10	-	-	0	-	7	-	5	-	-	4	4	-	10	-	5	-	-	-	-
2	-	18	6	-	7	-	-	0	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	-	-	-	-
3	5	0	-	-	-	-	18	16	6	-	-	3	0	10	-	-	5	-	7	-	4	-	-	4	5	-	10	-	19	-	-	-
4	-	-	18	6	0	7	-	-	-	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	10	-	-
5	-	5	0	-	16	-	-	18	3	6	-	-	-	0	10	-	-	5	-	7	4	4	-	-	-	5	-	-	-	-	-	-
6	6	-	-	18	-	0	7	-	-	-	10	2	9	-	-	16	9	-	20	-	-	-	4	12	19	-	-	-	-	-	-	-
7	-	-	5	0	18	16	-	-	-	3	6	-	-	-	0	10	7	-	5	-	-	4	4	-	10	-	5	-	7	-	19	-
8	18	6	-	-	-	-	0	7	2	-	-	10	16	9	-	-	-	9	-	20	12	-	-	4	-	19	-	4	-	17	-	10

(672,588), Code rate: 7/8

	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	0	18	6	5	7	18	16	0	10	2	3	6	10	16	9	0	20	7	9	5	4	12	4	4	4	10	19	5	10	-	-	-
2	5	0	18	6	0	7	18	16	6	10	2	3	0	10	16	9	5	20	7	9	4	4	12	4	5	4	10	19	19	10	-	-
3	6	5	0	18	16	0	7	18	3	6	10	2	9	0	10	16	9	5	20	7	4	4	4	12	19	5	4	10	17	19	10	-
4	18	6	5	0	18	16	0	7	2	3	6	10	16	9	0	10	7	9	5	20	12	4	4	4	10	19	5	4	7	17	19	10

Figure 12-13—Matrix permutation indices of structured parity check matrices

For shortened LDPC operation, the k–l zero elements are appended to the incoming l message bits as follows: $r_i = 0$ for $i = l, l+1, \ldots, k-1$. These inserted zero elements are not transmitted.

12.2.3.6.4 Rate 14/15 LDPC code

The rate 14/15 LDPC(1440,1344) code is also systematic, i.e., the LDPC encoder encodes an information block of size k, $\mathbf{i} = (i_0, i_1, \ldots, i_{(k-1)})$, into a codeword $\mathbf{c}$ of size n, $\mathbf{c} = (i_0, i_1, \ldots, i_{(k-1)}, p_0, p_1, \ldots, p_{(n-k-1)})$, by adding $n - k$ parity bits obtained so that $\mathbf{Hc}^\mathrm{T} = 0$, where $\mathbf{H}$ is an $(n - k) \times n$ parity check matrix. Denote the 96×1440 parity check matrix as $\mathbf{H} = (h_{i,j})$, where $h_{i,j}$ consists of $\{0,1\}$, $0 \le i < 96$ and $0 \le j < 1440$. Table 12-17 shows the matrix elements whose values are "1" in the first 15 columns of parity check matrix.

Table 12-17—Positions of 1's in the first 15 columns of parity check matrix H (codeword block length L_{FEC} = 1440)

$h_{0,0}$ $h_{1,0}$ $h_{4,0}$
$h_{32,1}$ $h_{34,1}$ $h_{39,1}$
$h_{64,2}$ $h_{70,2}$ $h_{78,2}$
$h_{8,3}$ $h_{18,3}$ $h_{95,3}$
$h_{31,4}$ $h_{42,4}$ $h_{54,4}$
$h_{63,5}$ $h_{76,5}$ $h_{91,5}$
$h_{14,6}$ $h_{45,6}$ $h_{94,6}$
$h_{30,7}$ $h_{47,7}$ $h_{83,7}$
$h_{17,8}$ $h_{62,8}$ $h_{80,8}$
$h_{28,9}$ $h_{48,9}$ $h_{82,9}$
$h_{22,10}$ $h_{60,10}$ $h_{81,10}$
$h_{27,11}$ $h_{49,11}$ $h_{84,11}$
$h_{7,12}$ $h_{53,12}$ $h_{77,12}$
$h_{19,13}$ $h_{44,13}$ $h_{85,13}$
$h_{6,14}$ $h_{46,14}$ $h_{75,14}$

For $15 \le j$, the matrix element can be obtained by using Equation (12-6):

$$h_{i,j} = h_{\mathrm{mod}(i + \lfloor j/15 \rfloor,\, 96),\, \mathrm{mod}(j,\, 15)} \tag{12-6}$$

where mod(x, y) is the modulo function and is defined as $x - n \times y$ where n is the nearest integer less than or equal to x/y.

The LDPC(1440, 1344) code is a quasi-cyclic code such that every cyclic shift of a codeword by 15 symbols yields another codeword.

For shortened LDPC operation, the k–l zero elements are appended to the incoming l message bits as follows: $r_i = 0$ for $i = l, l+1, \ldots, k-1$. The message order is r_{k-1} as the first bit of the message with r_0 as the last bit of the message. These inserted zero elements are not transmitted.

12.2.3.7 Stuff bits

Stuff bits shall be added to the end of the encoded MAC Frame Body field if the number of the encoded data bits is not an integer multiple of the length of the data portion in the subblock. The number of stuff bits is computed for each subframe if standard aggregation is employed. The calculation of stuff bits is as follows.

In the encoded MAC Frame Body field, the number of FEC codewords, N_{FEC} is given by Equation (12-7):

$$N_{\mathrm{FEC}} = \mathrm{CEIL}[(L_{\mathrm{MFB}} \times 8)/(L_{\mathrm{FEC}} \times R_{\mathrm{FEC}})] \tag{12-7}$$

where

L_{FEC} is the FEC codeword length
L_{MFB} is the length of the MAC Frame Body field in octets
R_{FEC} is the FEC rate

The FEC codeword length, L_{FEC}, is 2040 for the RS FEC specified in 12.2.3.6.2, 672 for the irregular LDPC specified in 12.2.3.6.3, and 1440 for the rate 14/15 LDPC specified in 12.2.3.6.4.

The encoded MAC Frame Body field shall be concatenated with stuff bits of length L_{STUFF} so that the resulting MAC Frame Body field is aligned on the subblock symbol boundary. The stuff bits shall be set to zero and then scrambled using the continuation of the scrambler sequence that scrambled the MAC Frame Body field in 12.2.3.10. The length of bits in the encoded MAC Frame Body field, L_{EBITS}, is given by Equation (12-8):

$$L_{\mathrm{EBITS}} = 8 \times L_{\mathrm{MFB}} + N_{\mathrm{FEC}} \times (1 - R_{\mathrm{FEC}}) \times L_{\mathrm{FEC}} \tag{12-8}$$

The number of subblocks in the encoded MAC Frame Body field, $N_{\mathrm{subblock\text{-}encMFB}}$, and the length of stuff bits, L_{STUFF}, are given by Equation (12-9) and Equation (12-10):

$$N_{\mathrm{subblock–encMFB}} = \mathrm{CEIL}(L_{\mathrm{EBITS}}/L_{\mathrm{CBPS}}) \tag{12-9}$$

$$L_{\mathrm{STUFF}} = N_{\mathrm{subblock–encMFB}} \times L_{\mathrm{CBPS}} - L_{\mathrm{ebits}} \tag{12-10}$$

where L_{CBPS} is the number of coded bits per subblock as given in Table 12-18 for each MCS.

Table 12-18—Rate-dependent bits per symbol

MCS identifier	Coded bits per subblock, L_{CBPS} (pilot word length = 0)	Coded bits per subblock, L_{CBPS} (pilot word length = 64)
0	8	—
1	128	112
2	256	224
3	512	448
4	512	448
5	256	224
6	512	448
7	1024	896

Table 12-18—Rate-dependent bits per symbol (*continued*)

MCS identifier	Coded bits per subblock, L_{CBPS} (pilot word length = 0)	Coded bits per subblock, L_{CBPS} (pilot word length = 64)
8	1024	896
9	1024	896
10	1024	896
11	1024	896
12	1536	1344
13	2048	1792

For the stuff bits in the frame headers, the values are given in Table 12-12 and Table 12-13.

12.2.3.8 Code spreading

12.2.3.8.1 General

To increase robustness in header and MAC Frame Body field, Golay and PRBS codes by LFSR are applied for code spreading. The following two categories of spreading are defined:

a) For spreading factor of 64, Golay sequences shall be used.
b) For Class 1 MCSs with spreading factor of 2 and 4 and for headers with spreading factor of 2 and 6, the LFSR shall be used.

12.2.3.8.2 Golay sequences

In the base rate mode, the frame header and MAC Frame Body field shall be spread as shown in Figure 12-6. The Golay sequences for spreading factor 64 are given in Table 12-5.

12.2.3.8.3 PRBS generation with LFSR

For a spreading factor of length 2, 4, or 6, the data bits shall be spread with a PRBS generated using an LFSR, as shown in Figure 12-14. Since the output of the spreader is a factor of L_{SF} larger than the input, the input shall hold while the feedback and output clock.

The 15-bit seed value of the LFSR shall be $[x_{-1}, x_{-2}, \ldots, x_{-15}]$ = [0101 0000 0011 111].

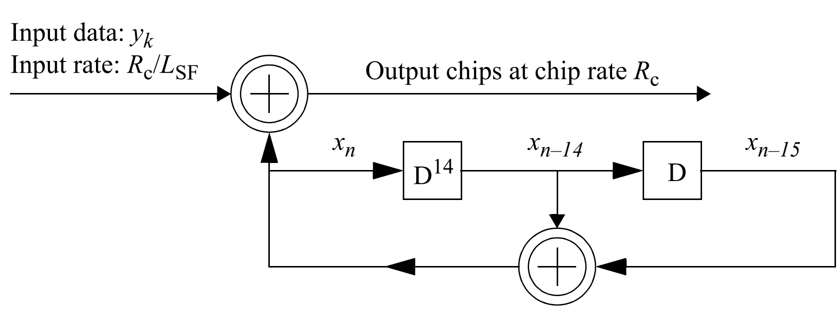

Figure 12-14—PRBS generation by LFSR

12.2.3.9 Unequal error protection

The UEP MCSs for SC PHY are shown in Table 12-19. UEP is an optional function. The data rate listed in the table is approximate and is rounded to three significant figures.

Table 12-19—UEP MCS for SC PHY

UEP MCS	Modulation	FEC		Data rate (Mb/s)	Supported UEP Type
		MSB	LSB		
0b00001	π/2 BPSK	RS (255,239)		1420	UEP Type 1 and UEP Type 2
0b00010		LDPC (672,336)		756	UEP Type 1 and UEP Type 2
0b00011		LDPC (672,504)		1130	UEP Type 1 and UEP Type 2
0b00100	π/2 QPSK	LDPC (672,336)		1510	UEP Type 1 and UEP Type 2
0b00101		LDPC (672,504)		2270	UEP Type 1 and UEP Type 2
0b00110		LDPC (672,588)		2650	UEP Type 1 and UEP Type 2
0b00111	π/2 QPSK	LDPC(672,336)	LDPC(672,504)	2040	UEP Type 3
0b01000		LDPC(672,504)	LDPC(672,588)	2650	UEP Type 3

In UEP Type 3, when the transmitter chooses one of the UEP MCS types, the transmitted data is divided into two groups in octet, MSB and LSB, as shown in Figure 12-15. The MSB and LSB groups are processed in parallel to adopt different LDPC coding rates. The outputs of the two LDPC encoders are multiplexed to generate a single bit stream for the symbol mapping.

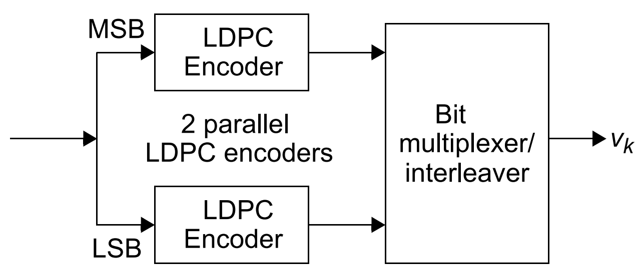

Figure 12-15—Parallel encoders for UEP

The bit multiplexing/interleaving method depends on the MSB and LSB coding rates as illustrated in Figure 12-16. When LDPC(672,336) is used for MSB group and LDPC(672,504) is used for LSB group, the encoded bits shall be multiplexed every 10 bits with 6 bits as the encoded MSBs and 4 bits as the encoded LSBs. A1, A2, A3, A4, A5, A6 are used in an increasing order in time to label the 6 encoded MSBs from encoder LDPC(672,336), while B1, B2, B3, B4 are used in an increasing order in time to label the four encoded LSBs from encoder LDPC(672,504). The bit multiplexing/interleaving shall be performed such that the output pattern is A1, B1, A2, B2, A3, A4, B3, A5, B4, A6 with A1 being the earliest bit at the input of the symbol mapper while A6 being the latest, as illustrated in the upper part of Figure 12-16.

When LDPC(672,504) is used for MSB group and LDPC(672,588) is used for LSB group, the encoded bits shall be multiplexed every 13 bits with 7 bits as the encoded MSBs and 6 bits as the encoded LSBs. A1, A2, A3, A4, A5, A6, A7 are used in an increasing order in time to label the seven encoded MSBs from encoder

LDPC(672,504), while B1, B2, B3, B4, B5, B6 are used in an increasing order in time to label the six encoded LSBs from encoder LDPC(672,588). The bit multiplexing/interleaving shall be performed such that the output pattern is A1, B1, A2, B2, A3, B3, A4, B4, A5, B5, A6, B6, A7 with A1 being the earliest bit at the input of the symbol mapper while A7 being the latest, as illustrated in the lower part of Figure 12-16.

A6	B4	A5	B3	A4	A3	B2	A2	B1	A1	→ to constellation mapper

LDPC(672,336), LDPC(672,504)

A7	B6	A6	B5	A5	B4	A4	B3	A3	B2	A2	B1	A1	→ to constellation mapper

LDPC(672,504), LDPC(672,588)

Figure 12-16—UEP bit multiplexing/interleaving

The effect of unequal error protection can also be obtained by using a skewed constellation, which is shown in Figure 12-29. A longer distance in x-axis than in y-axis is given between the two symbols in the skewed constellation, so that more energy is given to the MSB group. The mapping rules and the constellation are given in 12.3.3.6.

To apply the skewed constellation in the symbol mapping, both LDPC encoders in Figure 12-15 shall have the same coding rates among LDPC (672,336), LDPC (672,504), and LDPC (672,588), and the bit multiplexing/interleaving is 1:1 alternating, as illustrated in Figure 12-17.

B6	A6	B5	A5	B4	A4	B3	A3	B2	A2	B1	A1	→ to constellation mapper

Figure 12-17—UEP bit multiplexing/interleaving for skewed constellations

12.2.3.10 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as illustrated in Figure 12-14 with $L_{SF} = 1$.

The scrambler shall be used for the MAC header, HCS, MAC subheader, HCS (for subheader), and MAC Frame Body field. The PHY preamble, PHY Header field, and RS field shall not be scrambled. The polynomial for the PRBS generator used by the scrambler shall be as shown in Equation (12-11):

$$g(D) = 1 + D^{14} + D^{15} \tag{12-11}$$

where D is a single bit delay element. The polynomial forms not only a maximal length sequence, but also is a primitive polynomial. By the given generator polynomial, the corresponding PRBS, is generated as shown in Equation (12-12):

$$x_n = x_{n-14} \oplus x_{n-15},\ n = 0, 1, 2, \ldots \tag{12-12}$$

The initialization sequence is defined by Equation (12-13):

$$x_{init} = [x_{-1}x_{-2}x_{-3}x_{-4}x_{-5}x_{-6}x_{-7}x_{-8}x_{-9}x_{-10}x_{-11}x_{-12}x_{-13}x_{-14}x_{-15}] \tag{12-13}$$

The scrambled data bits, s_n, are obtained as shown in Equation (12-14):

$$s_n = b_n \oplus x_n \tag{12-14}$$

where b_n represents the unscrambled data bits. The side-stream descrambler at the receiver shall be initialized with the same initialization vector, x_{init}, used in the transmitter scrambler. The initialization vector is determined from the Scrambler Seed ID field contained in the PHY header of the received frame.

The 15-bit seed value chosen shall be computed from the Scrambler Seed ID field as shown in Equation (12-15):

$$[x_{-1}x_{-2}\ldots x_{-15}] = [11010000101\ S1\ S2\ S3\ S4] \tag{12-15}$$

The seed identifier value is set to 0000 when the PHY is initialized and is incremented in a 4-bit rollover counter for each frame that is sent by the PHY. The value of the seed identifier that is used for the frame is sent in the PHY header.

For a Scrambler Seed ID field set to all zero, the first 16 bits should be as shown in Equation (12-16):

$$[x_0x_1\ldots x_{15}] = [0001111000111010] \tag{12-16}$$

The 15-bit seed value is configured as follows. At the beginning of each PHY frame, the register is cleared, the seed value is loaded, and the first scrambler bit is calculated. The first bit of the data of the MAC header is modulo-2 added with the first scrambler bit, followed by the rest of the bits in the MAC header, MAC subheader, and MAC Frame Body field. The pilot word and pilot channel estimation sequences shall be excluded from the scrambling process.

12.2.4 SC PHY frame format

12.2.4.1 Overview

The SC PHY frame shall be formatted as illustrated in Figure 12-18.

PHY preamble	Frame header	PHY Payload field

Figure 12-18—SC PHY frame format

The Frame Header field for the PHY frame shall be formatted as illustrated in Figure 12-19.

Base frame header					Optional frame header			
PHY Header	MAC Header	Base HCS	RS Parity	Stuff bits	MAC Subheader	HCS	RS Parity	Stuff bits

Figure 12-19—Frame header format

The PHY preamble is described in 12.1.13.5. The MAC header is defined in 6.2 and the MAC subheader is defined in 6.2.11. The PHY header is defined in 12.2.4.3.2, and the HCS is defined in 12.2.4.3.3. The PHY Payload field consisting of the MAC Frame Body field, the PCES, and stuff bits, is described in 12.2.4.4. The PCES is described in 12.2.4.5.2. The stuff bits are described in 12.2.3.7.

When transmitting a frame, the PHY preamble is sent first, followed by the base frame header, and then followed by the optional frame header, and finally the PHY Payload field.

12.2.4.2 PHY preamble

12.2.4.2.1 General

A PHY preamble shall be added prior to the frame header to aid receiver algorithms related to AGC setting, antenna diversity selection, timing acquisition, frequency offset estimation, frame synchronization, and channel estimation.

The PHY preamble shall be transmitted at the chip rate defined in Table 12-14.

In the CTAP, the CMS may be used. Transmission using the CMS in a CTAP shall be done using the frame consisting of the CMS preamble described in 12.1.13.5, the CMS header described in 12.2.3.2, and the CMS payload described in 12.2.3.1.

In the CAP, the CMS may be used. For this purpose, the SC data frame shall be specified as in 12.1.13, except that a different pattern, [+1 +1 +1 –1 –1 +1 –1] spread by $\mathbf{b}_{128}$, shall be used in the SFD field.

Figure 12-20 shows the structure of the PHY preamble.

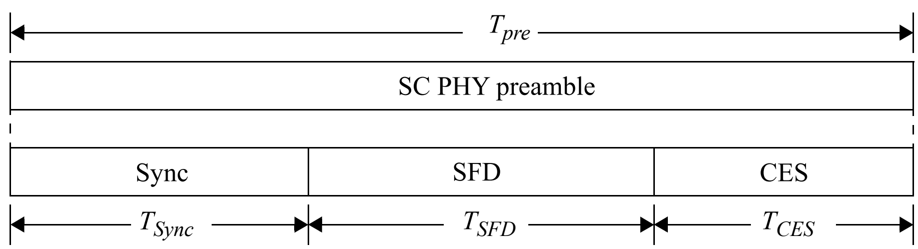

Figure 12-20—SC PHY preamble structure

12.2.4.2.2 Sync field

The Sync field is used for frame detection and uses a repetition of codes for a higher of robustness. The Sync field shall consist of 14 code repetitions of $\mathbf{a}_{128}$. Table 12-6 shows the sequence for $\mathbf{a}_{128}$ used for the Sync field.

12.2.4.2.3 SFD field

The SFD is used to establish frame timing as well as the header rate, either MR or HR. The SFD for the two header rates are as follows:

— The MR header shall use an SFD with [+1 –1 +1 –1] spread by $\mathbf{a}_{128}$
— The HR header shall use an SFD with [+1 +1 –1 –1] spread by $\mathbf{a}_{128}$

12.2.4.2.4 Channel estimation sequence (CES)

The CES field, used for channel estimation, shall consist of [$\mathbf{b}_{128}$ $\mathbf{b}_{256}$ $\mathbf{a}_{256}$ $\mathbf{b}_{256}$ $\mathbf{a}_{256}$] where the right most sequence, $\mathbf{a}_{256}$, is first in time. The sequences $\mathbf{a}_{128}$, $\mathbf{a}_{256}$, and $\mathbf{b}_{256}$ are specified in 12.1.13.5.

12.2.4.3 Frame header

12.2.4.3.1 General

A frame header shall be added after the PHY preamble. The frame header conveys information in the PHY and MAC headers necessary for successfully decoding the frame. The frame header consists of a base frame header followed by an optional frame header. The construction of the frame header is shown in Figure 12-21.

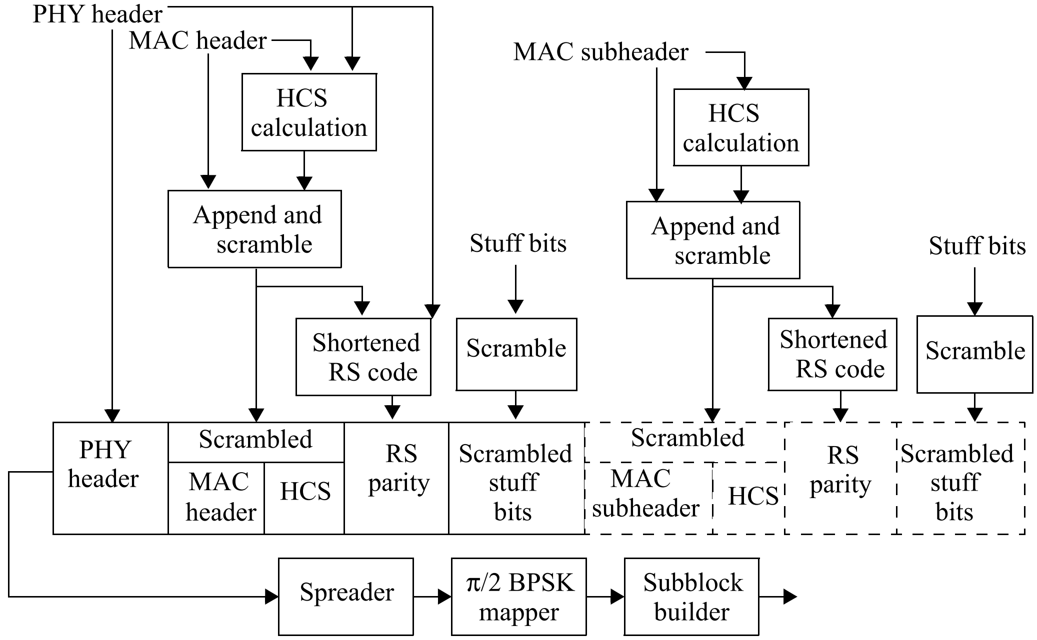

Figure 12-21—Frame header construction process

The detailed process of the construction is as follows:

a) Form the base frame header as follows:
   1) Construct the PHY header based on information provided by the MAC.
   2) Compute the HCS over the combined PHY and MAC headers.
   3) Append the HCS to the MAC header.
   4) Scramble the combined MAC header and HCS, as described in 12.2.3.10.
   5) Compute the RS Parity field by encoding the concatenation of the Header field, scrambled MAC Header, and scrambled HCS into a shortened RS block code, as described in 12.2.3.6.2.
   6) Form the base frame header by concatenating the PHY Header field, scrambled MAC Header, field, scrambled HCS, RS Parity field, and scrambled stuff bits.
b) Form the optional frame header as follows:
   1) Compute the HCS over the MAC subheader.
   2) Append the HCS to the MAC subheader.
   3) Scramble the combined MAC subheader and HCS, as described in 12.2.3.10.
   4) Compute the RS Parity field by encoding the concatenation of scrambled MAC Subheader field and scrambled HCS into a shortened RS block code, as defined in 12.2.3.6.2.
   5) Form the optional frame header by concatenating the scrambled MAC subheader, scrambled HCS, RS Parity field, and scrambled stuff bits.
c) Form the frame header by concatenating the base frame header and optional frame header. The resulting frame header shall be modulated as shown in Figure 12-21.
d) Spread the frame header, as described in 12.1.13.3.
e) Map the frame header onto π/2 BPSK, as described in 12.2.3.5.2.
f) Build subblocks from the resulting frame header, as described in 12.2.4.5.1.

The LFSR for the spreader is reset between the header and payload.

12.2.4.3.2 SC PHY header

The SC PHY header shall be formatted as illustrated in Figure 12-22.

Bits: b0–b3	b4	b5	b6–b10	b11–b30	b31–b32	b33	b34	b35–b36	b37	b38–b39
Scrambler seed ID	Aggregation	UEP	MCS	Frame length	Preamble type	Beam tracking	Low-latency mode	Pilot word length	PCES	Reserved

Figure 12-22—PHY header format for CMS and SC PHY

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 12.2.3.10.

The Aggregation field shall be set to one if aggregation is used; it shall be set to zero otherwise.

The UEP field shall be set to one if UEP is used; it shall be set to zero if otherwise.

The MCS field shall be set according to the values in Table 12-20.

Table 12-20—Modulation and coding scheme (MCS)

MCS	MCS identifier
0b00000	0
0b00001	1
0b00010	2
0b00011	3
0b00100	4
0b00101	5
0b00110	6
0b00111	7
0b01000	8
0b01001	9
0b01010	10
0b01011	11
0b01100	12
0b01101	13
0b01110–0b11111	Reserved

The Frame Length field shall be an unsigned integer equal to the number of octets in the MAC Frame Body field of a regular frame, excluding the FCS; it shall be set to zero for an aggregated frame.

The Preamble Type field indicates the type of the PHY preamble used in the next frame, as defined in Table 12-21.

Table 12-21—Preamble type field definition

Preamble type	Type of preamble used for next frame
0b00	CMS preamble
0b01	SC preamble
0b10–0b11	Reserved

The Beam Tracking field shall be set to one if training sequences for beam tracking are present following the current frame; it shall be set to zero otherwise.

The Low-latency Mode field shall be set to one if the frame is using the low-latency aggregation mode, and it shall be set to zero otherwise. If the Low-latency Mode field is set to one, then the Aggregation field shall also be set to one.

The Pilot Word Length field indicates the length of the pilot word used in the current frame and shall be encoded as defined Table 12-22.

Table 12-22—Pilot word length field definition

Pilot word length	Pilot word length in subblock
0b00	64 (mandatory)
0b01	8
0b10	0 (mandatory)
0b11	Reserved

The PCES field shall be set to one if the frame includes PCES; it shall be set to zero if otherwise.

12.2.4.3.3 Base header HCS

The combination of the PHY header and MAC header shall be protected with an ITU-T CRC-16 base HCS. The ITU-T CRC-16 is described in 11.2.9.

12.2.4.3.4 Base header FEC

The concatenation of the PHY header, scrambled MAC header, and scrambled HCS shall use shortened systematic RS(n+16,n) for the FEC, where n is the number of octets in the combined PHY header, MAC header and HCS. The 128 RS parity bits are appended after the scrambled HCS as shown in Figure 12-21.

12.2.4.3.5 MAC subheader HCS

The MAC subheader shall be protected with an ITU-T CRC-16 HCS. This shall be computed using the method specified in 12.2.4.3.3.

12.2.4.3.6 MAC subheader FEC

The scrambled MAC subheader and scrambled HCS shall be encoded using a shortened RS(n+16,n) for the FEC, where n is the number of octets in the MAC subheader and HCS. The 128 RS parity bits are appended after the scrambled HCS, as shown in Figure 12-21.

NOTE—The length of the MAC subheader is different for secure and non-secure frames and hence the length of the shortened RS code will be different as well.

12.2.4.4 SC PHY Payload field

12.2.4.4.1 General

The SC PHY Payload field is the last component of the frame and is constructed as shown in Figure 12-23.

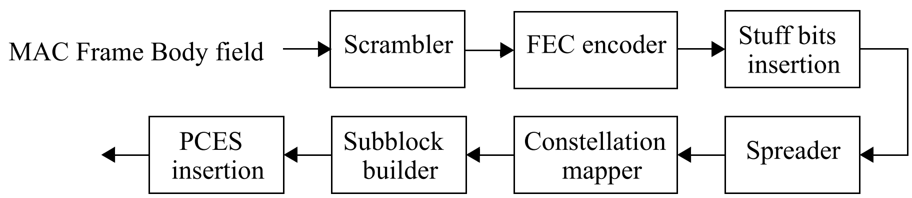

Figure 12-23—SC PHY Payload field construction process

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC Frame Body field according to 12.2.3.10.
b) Encode the scrambled MAC Frame Body field, as specified in 12.2.3.6.
c) Add stuff bits to the encoded and scrambled MAC Frame Body field according to 12.2.3.7.
d) Spread the encoded and scrambled MAC Frame Body field using the spreading code, as detailed in 12.1.13.3.
e) Map the resulting MAC Frame Body field onto the appropriate constellation, as described in 12.2.3.5.
f) Build subblocks from the resulting MAC Frame Body field according to 12.2.4.5.1.
g) Insert PCES periodically, as described in 12.2.4.5.2.
h) Apply a chip-level $\pi/2$ continuous rotation to the resulting MAC Frame Body field, as described in 12.2.3.5.

12.2.4.4.2 SC PHY Payload scrambling

The SC PHY payload shall use the scrambling process defined in 12.2.3.10.

12.2.4.4.3 Modulation

Modulation for the MAC Frame Body field is defined in 12.2.3.5.

12.2.4.4.4 FEC

FEC for the MAC Frame Body field is defined in 12.2.3.6.

12.2.4.4.5 Code spreading

Code spreading for the MAC Frame Body field is defined in 12.1.13.3.

12.2.4.5 Pilot word and PCES

12.2.4.5.1 Subblocks and pilot word

Pilot words are used in SC PHY for timing tracking, compensation for clock drift, and compensation for frequency offset error. Furthermore, pilot words act as a known cyclic prefix and enables frequency domain equalization if desired. In frequency domain equalization, the data is handled in the unit of subblocks. The building of the data blocks and subblocks is illustrated by Figure 12-24.

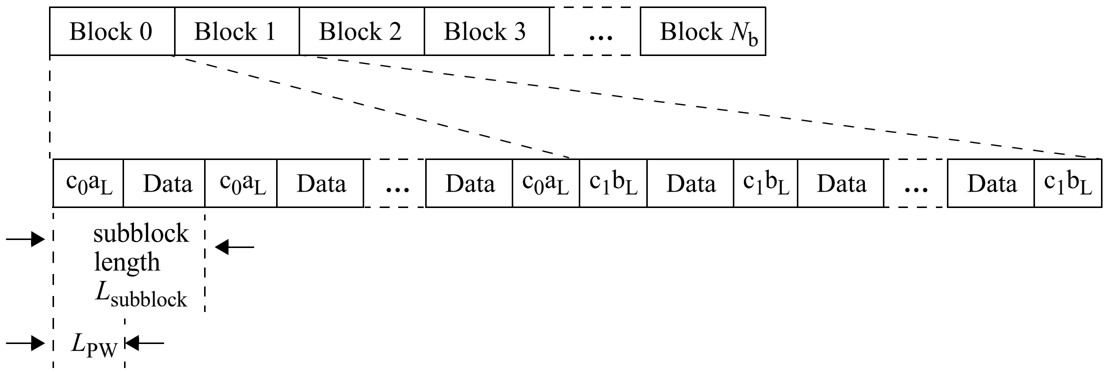

Figure 12-24—Frame format with pilot word

A block shall contain 64 subblocks with the exception of the last block (i.e., the N_b-th block). A subblock is formed by appending a pilot word to the data. The possible pilot word lengths are 0, 8, and 64. For pilot word lengths 0 and 64, the length of the data is $L_{DC} = L_{subblock}–L_{PW}$ symbols. For $L_{PW} = 8$, every subblock consists of 8 sub-subblocks each with $L_{PW} = 8$ and $L_{DC} = L_{subblock}/8–L_{PW}$, thus giving the effective length of pilot word and data that is equivalent to that with $L_{PW} = 64$.

The Golay sequences for pilot word of length 8 are given in Table 12-23, while pilot word sequences of length 64 are shown in Table 12-5.

Table 12-23—Golay sequences of length 8

Sequence name	Sequence value
$\mathbf{a}_8$	0xEB
$\mathbf{b}_8$	0xD8

Even number blocks shall use pilot word sequences of type **a**. Odd number blocks shall use pilot word sequences of type **b**. Furthermore, an LFSR shall be used to change the polarity of pilot word from one block to another. The LFSR used shall be the same as the LFSR described in 12.2.3.8.3 with the same initial state, but shall be run at the appropriate rate, i.e., one LFSR output per block. The last subblock of the block shall be followed by a pilot word as well. The pilot word is modulated with π/2 BPSK.

12.2.4.5.2 Pilot channel estimation sequence (PCES)

PCES insertion is an optional feature that allows a DEV to periodically re-acquire the channel. To add the PCES, the scrambled, encoded, spread, and modulated MAC Frame Body field is divided into data blocks. Each data block, as shown in Figure 12-24, shall be preceded by a PCES, with the exception of the first block. The duration and insertion interval of PCES are specified in Table 12-15. The PCES field shall be the CES field in the preamble defined in 12.2.4.2.4. Similarly, the PCES is modulated with π/2 BPSK.

12.2.5 Transmitter specifications

12.2.5.1 Error vector magnitude

A compliant transmitter shall have EVM values of less than those given in Table 12-24 for the MCS classes listed.

Table 12-24—EVM for SC PHY MCS classes

MCS	EVM (dB)
Class 1	–7
Class 2	–14
Class 3	–21

12.2.5.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 25 \times 10^{-6}$ maximum.

12.2.5.3 Symbol rate

The SC PHY shall be capable of transmitting at the chip rate defined in Table 12-14 to within $\pm 25 \times 10^{-6}$. The MAC parameter, *pClockAccuracy*, shall be $\pm 25 \times 10^{-6}$. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked).

12.2.5.4 Transmit power-on and power-down ramp

The transmit power-on ramp is defined as the time it takes for the RF power emitted by the compliant DEV to rise from less than 10% to greater than 90% of the maximum power to be transmitted in the frame.

The transmit power-on ramp shall be less than 9.3 ns.

The *transmit power-down ramp* is defined as the time it takes for the RF power emitted by the compliant DEV to fall from greater than 90% to less than 10% of the maximum power to be transmitted in the frame.

The transmit power-down ramp shall be less than 9.3 ns.

The transmit power ramps shall be constructed such that the emissions conform to the unwanted emissions specification defined in 12.1.4.

12.2.6 Receiver specifications

12.2.6.1 Error rate criterion

The error rate criterion shall be a FER of less than 8% with a frame payload length of 2048 octets. The error rate should be determined at the PHY SAP interface after any error correction methods (excluding retransmission) required in the proposed device has been applied. The measurement shall be performed in additive white Gaussian noise (AWGN) channel.

12.2.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 12.2.6.1 is met. The error ratio shall be determined after any error correction has been applied. A compliant DEV that implements the SC PHY shall achieve at least the reference sensitivity listed in Table 12-25.

Table 12-25—Reference sensitivity levels for MCS

MCS Identifier	Receiver sensitivity
0	–70 dBm
1	–61 dBm
2	–58 dBm
3	–55 dBm
4	–59 dBm
5	–65 dBm
6	–62 dBm
7	–58 dBm
8	–56 dBm
9	–54 dBm
10	–53 dBm
11	–52 dBm
12	–50 dBm
13	–46 dBm

12.2.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 12.2.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

12.2.6.4 Receiver clear channel assessment performance

A compliant receiver provides CCA capability by performing energy detection in the received signal bandwidth. The start of a valid preamble sequence at a receive level equal to or greater than the minimum sensitivity for the CMS, as described in 12.2.6.2, shall cause CCA to indicate medium busy with a probability of >90% within *pCcaDetectTime*. The receiver CCA function shall in all circumstances report medium busy with any signal 20 dB above the minimum sensitivity for the CMS.

The CCA detection time shall be equal to *pCcaDetectTime*. The CCA shall be maintained as busy until the end of the frame.

12.2.7 PHY layer timing

12.2.7.1 General

The values for the PHY layer timing parameters are defined Table 12-26.

Table 12-26—PHY layer timing parameters

PHY parameter	Value	Subclause
pMifsTime	0.2 µs, 0.5 µs (default), 2.0 µs	12.2.7.5
pSifsTime	0.2 µs, 2.0 µs, 2.5 µs (default)	12.2.7.4
pCcaDetectTime	4 µs	12.2.6.4
pChannelSwitchTime	100 µs	12.2.7.6

12.2.7.2 Interframe space

A conformant implementation shall support the IFS parameters, as described in 7.6.2, given in Table 12-27.

Table 12-27—IFS parameters

MAC parameter	Corresponding PHY parameter	Definition
MIFS	*pMifsTime*	12.2.7.5
SIFS	*pSifsTime*	12.2.7.4
pBackoffSlot	*pSifsTime* + *pCcaDetectTime*	12.2.7.4
BIFS	*pSifsTime* + *pCcaDetectTime*	12.2.7.4, 12.2.6.4
RIFS	2 × *pSifsTime* + *pCcaDetectTime*	12.2.7.4, 12.2.6.4

12.2.7.3 Receive-to-transmit turnaround time

The receive-to-transmit turnaround time shall be *pSifsTime*, including the power-up ramp specified in 12.2.5.4. The receive-to-transmit turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

12.2.7.4 Transmit-to-receive turnaround-time

The transmit-to-receive turnaround time shall be less than *pSifsTime*, including the power-down ramp specified in 12.2.5.4.

12.2.7.5 Time between successive transmissions

The minimum time between successive transmissions shall be *pMifsTime*, including the power-up ramp specified in 12.2.5.4. The *pMifsTime* shall be measured at the air interface from the trailing edge of the last symbol transmitted until the first symbol of the PHY preamble is present at the air interface.

12.2.7.6 Channel switch

The channel switch time is defined as the time from the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pChannelSwitchTime*.

12.2.8 PHY management for SC PHY

12.2.8.1 General

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the SC PHY layer of a DEV.

12.2.8.2 Maximum frame size

The maximum frame length allowed, *pMaxFrameBodySize*, shall be 8 388 608 octets. This total includes the MAC subheader and the MAC Frame Body field, but not the PHY preamble, base header (PHY header, MAC header, and HCS). The maximum frame length also does not include the stuff bits.

12.2.8.3 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 8 388 576 octets. If security is enabled for the data connection, the upper layers should limit data frames to 8 388 576 octets minus the security overhead, as defined in 6.3.5.2, 6.2.11.2.2, or 6.2.11.3.2.

12.2.8.4 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed with the SC PHY shall be 512 octets.

12.2.9 Optional OOK/DAMI modes

12.2.9.1 General

Besides the MCS classes in 12.2.3.1, optional low complexity and low power consumption MCSs, which are important for SC applications, may be employed within child piconets. OOK and DAMI MCSs may be employed as optional modes for these applications.

All PNC-capable OOK/DAMI DEVs shall be able to transmit and receive CMS signals.

If a PNC-capable OOK/DAMI DEV starts an independent piconet, it shall start the piconet in SC mode. OOK/DAMI PNC-capable DEVs may use OOK/DAMI signals in CTAs allocated for child piconet to support communication with non-PNC-capable OOK/DAMI DEVs. Details on child piconet creation and usage are described in G.2.

The summary of the MCS for OOK and DAMI is given in Table 12-11. All PNC-capable OOK/DAMI DEVs shall transmit CMS beacons and conduct CP in CMS, with $\pi/2$ BPSK and RS(255,239). These PNC-capable OOK/DAMI DEVs may create child piconet for respective non-PNC-capable DEVs by using respective MCS-formatted signals in Table 12-11. For OOK non-PNC-capable DEVs, OOK modulation and RS(255,239) shall be used. For DAMI non-PNC-capable devices, DAMI modulation and RS(255,239) shall be used.

12.2.9.2 OOK

The OOK modulation shall use variable amplitudes to represent the data. As shown in Figure 12-25(a), OOK shall be represented by two points in the constellation map. The simplest form of OOK represents a binary “1” with the presence of the signal, and a binary “0” with the absence of it. The normalization factor, K_{MOD}, shall be sqrt(2).

12.2.9.3 DAMI

DAMI modulation is shown in Figure 12-25(b). The transmitted RF signal for a DAMI system shall be a single-sideband (SSB) modulated signal accompanied by two low-power pilot tones, as described in G.3.

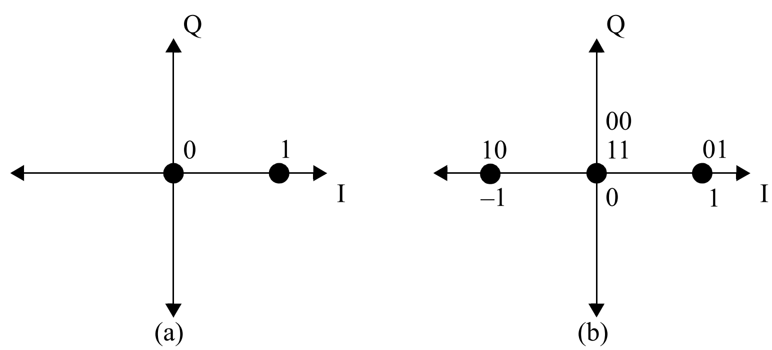

Figure 12-25—Constellation diagram for (a) OOK and (b) DAMI

12.2.9.4 FEC

The forward error correction scheme for OOK and DAMI shall be RS(255,239), as described in 12.2.3.6.

12.2.9.5 OOK/DAMI spreading

The spreading scheme for OOK with spreading factor of 2 shall use a simple bit repetition in which each bit shall be repeated twice.

12.3 High Speed Interface mode of mmWave PHY

12.3.1 General

The High Speed Interface mode of mmWave PHY (HSI PHY) is designed for devices with low-latency, bidirectional high-speed data and uses OFDM. HSI PHY supports a variety of MCSs using different frequency-domain spreading factors, modulations, and LDPC block codes.

12.3.2 Operating frequency bands

The set of operating channels is defined in Table 12-3. A compliant IEEE 802.15.3 implementation that implements the HSI PHY shall support at least CHNL_ID 2 or CHNL_ID 3.

12.3.3 HSI PHY modulation, forward error correction, and spreading

12.3.3.1 MCS-dependent parameters

The HSI PHY MCS-dependent parameters are listed in Table 12-28. For the FEC rates of 1/2, 5/8, 3/4, and 7/8, LDPC(672,336), LDPC(672,504), LDPC(672,420), and LDPC(672,588) codes are used, respectively. For the UEP MCSs (MCS index 8-11), different coding schemes are applied to the MSB octets and LSB octets. The data rates specified in Table 12-28 assume a cyclic prefix length of 64 chips.

Table 12-28—HSI PHY MCS-dependent parameters

MCS index	Data rate (Mb/s)	Modulation scheme	Spreading factor (L_f)	Coding mode	FEC rate (R_{FEC})	
					MSB 8b	LSB 8b
0	32.1	QPSK	48	EEP	1/2	
1	1540	QPSK	1		1/2	
2	2310	QPSK	1		3/4	
3	2695	QPSK	1		7/8	
4	3080	16-QAM	1		1/2	
5	4620	16-QAM	1		3/4	
6	5390	16-QAM	1		7/8	
7	5775	64-QAM	1		5/8	
8	1925	QPSK	1	UEP	1/2	3/4
9	2503	QPSK	1		3/4	7/8
10	3850	16-QAM	1		1/2	3/4
11	5005	16-QAM	1		3/4	7/8

The number of spread, coded, and data information bits per OFDM symbol are listed in Table 12-29 for the various MCSs.

Table 12-29—HSI PHY MCS-dependent bits per OFDM symbol

MCS index	Spread and coded bits/OFDM symbol (N_{SCBPOS})	Coded bits/OFDM symbol (N_{CBPOS})	Information bits/OFDM symbol (N_{IBPOS})	
			MSB 8b	LSB 8b
0	672	14	7	
1	672	672	336	
2	672	672	504	
3	672	672	588	
4	1344	1344	672	
5	1344	1344	1008	

Table 12-29—HSI PHY MCS-dependent bits per OFDM symbol (*continued*)

MCS index	Spread and coded bits/OFDM symbol (N_{SCBPOS})	Coded bits/OFDM symbol (N_{CBPOS})	Information bits/OFDM symbol (N_{IBPOS})	
			MSB 8b	LSB 8b
6	1344	1344	1176	
7	2016	2016	1260	
8	672	672	336	504
9	672	672	504	588
10	1344	1344	672	1008
11	1344	1344	1008	1176

The HSI PHY frame header rate-dependent modulation parameters are listed in Table 12-30. The frame header is QPSK modulated and encoded with shortened LDPC(672,336) code.

Table 12-30—HSI PHY frame header rate-dependent parameters

Header rate type	Header rate (Mb/s)		Spreading factor	Coded bits /OFDM symbol	Number of occupied OFDM symbols		Number of stuff bits[a]	
	Main	Optional			Main	Optional	Main	Optional
MCS 0	16.8	29.6	48	14	35	47	10	2
MCS 1–11	587	1363	1	672	1	1	192	16

[a]Stuff bits are inserted after the LDPC encoding and before constellation mapping.

The base rate shall be MCS index 0 for HSI PHY and the corresponding header rates shall be as indicated in Table 12-30.

A DEV that supports the HSI PHY shall support HSI MCS index 1 and either CMS or HSI MCS index 0, as defined in 12.1.13.

12.3.3.2 HSI PHY timing-related parameters

Table 12-31 lists the timing-related parameters.

Table 12-31—Timing-related parameters

Parameters	Description	Value	Formula
f_s	Reference sampling rate/chip rate	2640 MHz	
T_C	Sample/chip duration	~0.38 ns	$1/f_s$
N_{sc}	Number of subcarriers/FFT size	512	
N_{dsc}	Number of data subcarriers	336	

Table 12-31—Timing-related parameters (*continued*)

Parameters	Description	Value	Formula
N_P	Number of pilot subcarriers	16	
N_G	Number of guard subcarriers	141	
N_{DC}	Number of DC subcarriers	3	
N_R	Number of reserved subcarriers	16	
N_U	Number of used subcarriers	352	$N_{dsc} + N_P$
N_{GI}	Guard interval length in samples	64	
Δf_{sc}	Subcarrier frequency spacing	5.15625 MHz	f_s/N_{sc}
BW	Nominal used bandwidth	1815 MHz	$N_U \times \Delta f_{sc}$
T_{FFT}	IFFT and FFT period	~193.94 ns	$1/\Delta f_{sc}$
T_{GI}	Guard interval duration	~24.24 ns	$N_{GI} \times T_C$
T_S	OFDM Symbol duration	~218.18 ns	$T_{FFT} + T_{GI}$
F_S	OFDM Symbol rate	~4.583 MHz	$1/T_S$
N_{CPS}	Number of samples per OFDM symbol	576	$N_{sc} + N_{GI}$

12.3.3.3 HSI PHY frame-related parameters

The frame-related parameters are listed in Table 12-32.

Table 12-32—OFDM frame-related parameters

Parameter	Description	Value	
N_{pre}	Number of symbols in the PHY preamble (A preamble symbol is 512 chips long)	Long Preamble	16
		Short Preamble	6.75
T_{pre}	Duration of the PHY preamble	Long Preamble	~3.15 µs
		Short Preamble	~1.31 µs
T_{HDR}	Duration of the header	Main header only for MCS 0	~7.64 µs
		Main header only for MCS 1–11	~0.22 µs
		Main header and optional header for MCS 0	~17.89 µs
		Main header and optional header for MCS 1–11	~0.44 µs
N_{OSMF}	Number of OFDM symbols in the MAC Frame Body field	Variable	
T_{OSMF}	Duration of the MAC Frame Body field	$N_{OSMF} \times T_S$	

Table 12-32—OFDM frame-related parameters (*continued*)

Parameter	Description	Value
N_{frame}	Number of OFDM symbols in the frame	$N_{pre} + N_{HDR} + N_{OSMF}$
T_{frame}	Duration of the frame	$T_{pre} + T_{HDR} + T_{OSMF}$

12.3.3.4 HSI PHY FEC

12.3.3.4.1 Overview

The HSI PHY FEC process is illustrated in Figure 12-26. In the figure, the bit interleaver block is drawn with a dashed line because it is an optional part of the HSI PHY FEC process.

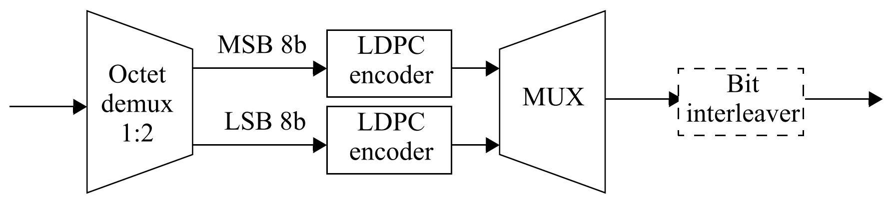

Figure 12-26—FEC process for the HSI PHY

12.3.3.4.2 LDPC block code

The supported LDPC block FEC rates, information block lengths, L_{INF}, and codeword block lengths, L_{FEC}, are described in Table 12-33.

Table 12-33—LDPC parameters

R_{FEC}	L_{INF} (bits)	L_{FEC} (bits)
1/2	336	672
5/8	420	
3/4	504	
7/8	588	

The LDPC encoder with rates 1/2, 3/4, and 7/8 is described in 12.2.3.6.3. For rate 5/8, the matrix permutation indices of parity-check matrix is given in Figure 12-27.

	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	0	-	-	5	-	18	16	-	-	-	3	6	10	-	-	0	-	7	-	5	-	-	4	4	-	10	-	5	-	-	-	-
2	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	4	-	-	-	-	-	19	-	-	-	-	-
3	-	18	-	-	-	-	-	0	10	-	-	-	-	16	-	-	-	-	9	-	-	12	-	-	4	-	-	-	-	-	17	-
4	5	0	-	-	-	-	18	16	6	-	-	3	0	10	-	-	5	-	7	-	4	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	4	-	-	-	-	-	-	-	-	-	-
6	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	-	-	-	-	9	-	-	12	-	-	-	-	-	-	-	-	-
7	-	5	0	-	16	-	-	18	3	6	-	-	-	0	10	-	-	5	-	7	4	4	-	-	-	5	-	-	-	-	-	-
8	6	-	-	-	-	-	7	-	-	-	-	2	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	-	-
9	-	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	9	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
10	-	-	5	0	18	16	-	-	-	3	6	-	-	-	0	10	7	-	5	-	-	4	4	-	10	-	5	-	7	-	-	-
11	-	6	-	-	-	-	-	7	2	-	-	-	-	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10
12	18	-	-	-	-	-	0	-	-	-	-	10	16	-	-	-	-	9	-	-	12	-	-	-	-	-	-	4	-	17	-	-

Figure 12-27—Matrix permutation indices of parity check matrix of rate-5/8 LDPC code

12.3.3.4.3 EEP data multiplexer

For the EEP MCSs, as defined in Table 12-28, the two LDPC encoders use the same rate, and the outputs of the LDPC encoders shall be multiplexed to form a single data stream. Let a_n and b_n be the outputs bits of the MSB and LSB encoders, respectively. The multiplexer output shall be $a_0,b_0,a_1,b_1\ldots$

12.3.3.4.4 UEP data multiplexer

For UEP Type 3 MCSs, as defined in Table 12-28, the two LDPC encoders have different rates. The method used to multiplex the encoded bits is dependent on the LDPC MSB and LSB rates.

For UEP MCSs with MSB encoder rate 1/2 and LSB encoder rate 3/4, the encoded bits shall be multiplexed every 5 bits. During the length 5 multiplexing cycle, a group multiplexer shall be used with group size three for the MSB encoder and group size two for the LSB encoder. a_0,a_1,a_2 are used to label the three encoded bits (in increasing order in time) from the MSB encoder, and b_0,b_1 are used to label the two encoded bits from the LSB encoder. The multiplexer output shall be a_0,b_0,a_1,b_1,a_2.

For UEP MCSs with MSB encoder rate 3/4 and LSB encoder rate 7/8, the encoded bits shall be multiplexed every 13 bits. During the length 13 multiplexing cycle, a group multiplexer shall be used with group size 7 for the MSB encoder and group size 6 for the LSB encoder. $a_0,a_1,a_2,\ldots,a_6$ are used to label the 7 encoded bits (in increasing order in time) from the MSB encoder, and $b_0,b_1,b_2,\ldots,b_5$ are used to label the 6 encoded bits from the LSB encoder. The multiplexer output shall be $a_0,b_0,a_1,b_1,\ldots,a_5,b_5,a_6$.

12.3.3.4.5 Bit interleaver

After the data multiplexer, the bits shall be interleaved by a block interleaver if the Bit Interleaver field in the PHY header is set to one. The interleaving is performed upon encoded bits included within an interleaving depth covering 4 LDPC codewords, i.e., more than 2688 bits.

The block interleaving process is performed using a permutation rule $L(k)$. That is, the k^{th} output, written to location k in the output vector, is read from location $L(k)$ in the input vector.

The block interleaving algorithm $L(k) = I^j_{p,q}(k)$ is described by four parameters: the block size $K_B = 2688$, an integer parameter p setting the partition size, an integer parameter q and the iteration j governing the interleaving spreading. The relationship between the block of K_B coded bits, $a_0, a_1, \ldots, a_{K\text{-}1}$, and the block of K_B interleaved bits, $b_0, b_1, \ldots, b_{K\text{-}1}$, is given Equation (12-17):

$$b(k) = a[I^j_{p,q}(k)] \tag{12-17}$$

To realize the interleaver stage, it is recommended to implement a lookup table that contains the interleaving rule.

The interleaving rule is based on an iterative structure, as illustrated in Figure 12-28, in order to increase the scalability of the interleaver.

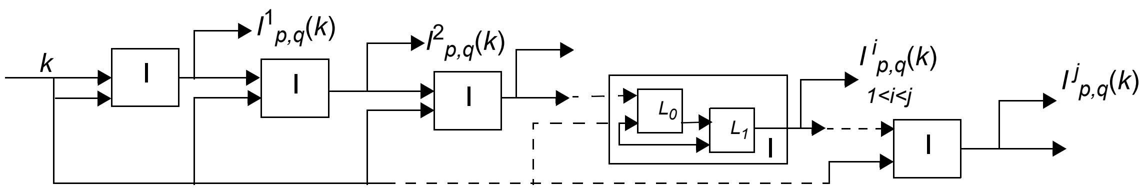

Figure 12-28—Turbo-based bit interleaver structure

The interleaving rules for the 1st and the j^{th} iteration are defined as shown in Equation (12-18) and Equation (12-19):

$$I^1_{p,q}(k) = \mathrm{mod}[K_B - p + k + q \times p \times \mathrm{mod}(-k - p \times k, K_B), K_B] \tag{12-18}$$

$$I^j_{p,q}(k) = \mathrm{mod}[K_B - p + k + q \times p \times \mathrm{mod}(-k - p \times I^{j-1}_{p,q}(k), K_B), K_B] \tag{12-19}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

The interleaver parameters are selected in order to optimize the interleaving spreading between successive samples. The interleaving spreading $\Delta L(s)$ is defined as the minimum distance between interleaved bits separated by a distance $s - 1$, and is expressed as shown in Equation (12-20)

$$\Delta L(s) = \min_k[\ |I^j_{p,q}(k + s) - I^j_{p,q}(k)|\ ] \tag{12-20}$$

The interleaving spreading is calculated in an algebraic way and allows the selection of interleaving parameters $\{p,q,j\}$ for each interleaving block size K_B. The binary interleaving parameters shall be $p = 24$, $q = 2$, $j = 1$, and $\Delta L(s = 1, 2, 4, 6) = \{1199, 290, 580, 870\}$.

12.3.3.5 Stuff bits

Stuff bits shall be appended in the MAC Frame Body field after scrambling, encoding, and interleaving so that the resulting MAC Frame Body field is aligned with the boundaries of an OFDM symbol. The stuff bits shall be set to zero and then scrambled using the continuation of the scrambler sequence that scrambled the MAC Frame Body field. To calculate the number of stuff bits, L_{STUFF}, that shall be inserted, first the number of codewords should be calculated as shown in Equation (12-21):

$$N_{\mathrm{FEC}} = \mathrm{CEIL}[(L_{\mathrm{MFB}} \times 8)/(L_{\mathrm{FEC}} \times R_{\mathrm{FEC}})] \tag{12-21}$$

where L_{MFB} is the length of the uncoded MAC Frame Body field in octets. FEC codeword length is denoted by L_{FEC} and R_{FEC} is the coding rate. Number of the encoded bits L_{EBITS} in the MAC Frame Body field is given byEquation (12-22):

$$L_{\mathrm{EBITS}} = L_{\mathrm{MFB}} \times 8 + N_{\mathrm{FEC}} \times (1 - R_{\mathrm{FEC}}) \times L_{\mathrm{FEC}} \tag{12-22}$$

Number of OFDM symbols N_{OSMF} in MAC Frame Body field is equal to Equation (12-23):

$$N_{\mathrm{OSMF}} = \mathrm{CEIL}(N_{\mathrm{EBITS}}/N_{\mathrm{SCBPOS}}) \tag{12-23}$$

where N_{SCBPOS} is the number of spread and coded bits per OFDM symbol. N_{SCBPOS} for different MCSs is given in Table 12-29. The number of stuff bits for a MAC Frame Body field is equal to Equation (12-24):

$$L_{\mathrm{PAD}} = N_{\mathrm{sOSMF}} \times N_{\mathrm{SCBPOS}} - N_{\mathrm{EBITS}} \tag{12-24}$$

Stuff bits shall be added to main and optional headers as well. The number of stuff bits for each header is given in Table 12-30.

12.3.3.6 Constellation mapping

The coded and interleaved binary serial input data, b_i, where $i = 0, 1, 2, \ldots$, shall be modulated using QPSK, 16-QAM or 64-QAM modulation, depending on the MCS requested. The binary serial stream shall be divided into groups of N_{BPSC} (2, 4, or 6) bits and converted into complex numbers representing QPSK, 16-QAM or 64-QAM constellation points. The conversion shall be performed according to Gray-coded constellation mappings, illustrated in Figure 12-29 with the input bit, b_0, being the earliest in the stream. The output values, a_k, where $k = 0, 1, 2, \ldots$, are formed by multiplying the resulting value $(I_k + \mathrm{j}Q_k)$ by a normalization factor, K_{MOD}, as shown in Equation (12-25):

$$a_k = (I_k + \mathrm{j}Q_k) \times K_{MOD} \tag{12-25}$$

The normalization factor, K_{MOD}, depends on the modulation, as prescribed in Table 12-34. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor can be used, as long as the device conforms to the modulation accuracy requirements described in 12.3.5.1.

Table 12-34—Modulation-dependent normalization factor

Modulation	K_{MOD}
QPSK	$1/\sqrt{1+d^2}$
16-QAM	$1/\sqrt{5(1+d^2)}$
64-QAM	$1/\sqrt{21(1+d^2)}$

An optional skewed constellation is also specified in Figure 12-29. A parameter d is introduced to distinguish between normal and skewed constellation. Its value is given byEquation (12-26):

$$d = \begin{cases} 1 & \text{normal constellation} \\ 1.25 & \text{skewed constellation} \end{cases} \tag{12-26}$$

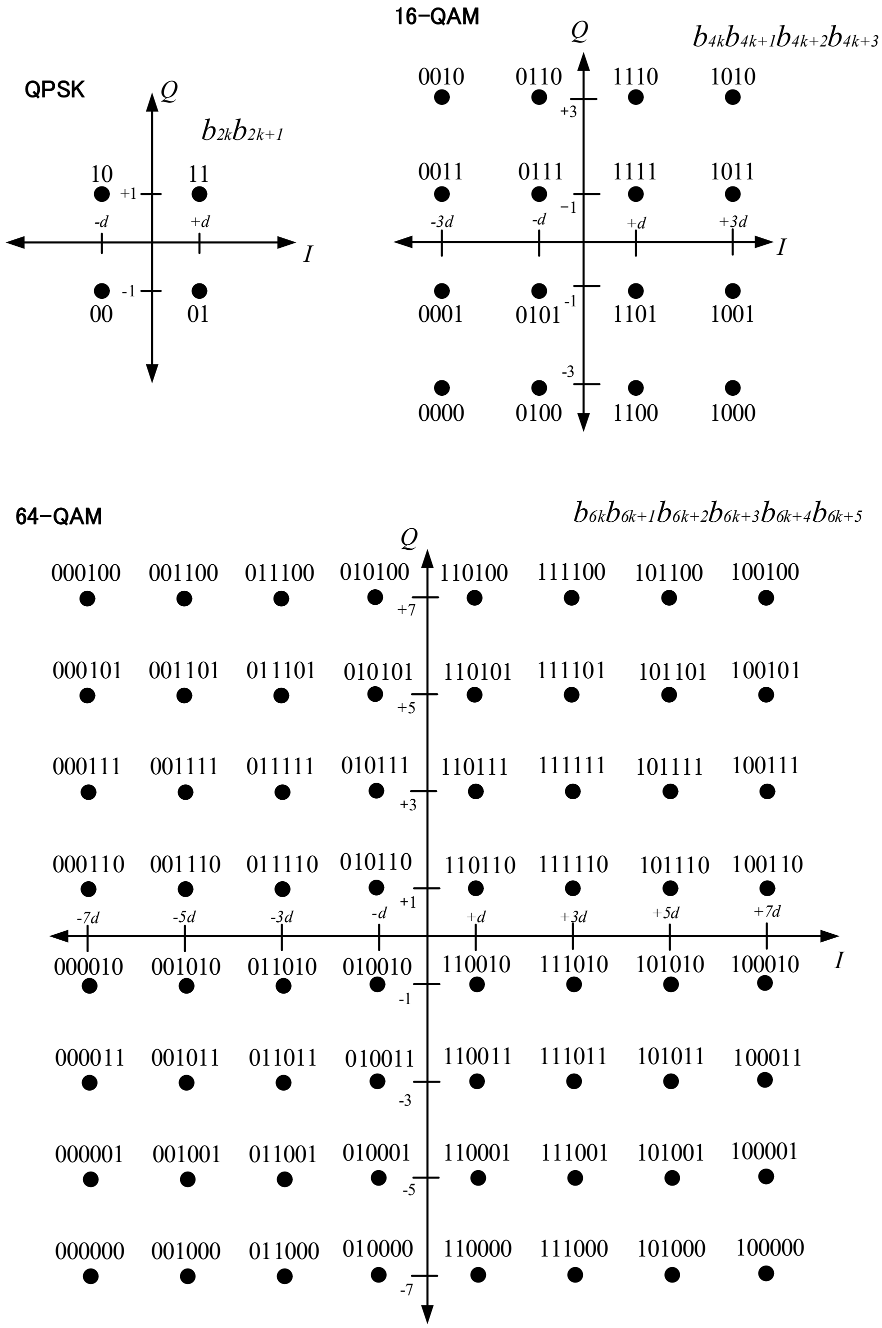

Figure 12-29—QPSK,16-QAM and 64-QAM normal and skewed constellation bit encoding

12.3.3.7 HSI spreader

12.3.3.7.1 General

The spreading factors in relationship to the data rate in the PHY Payload field and Frame Header field are specified in Table 12-28 and Table 12-30, respectively. The spreading rules shall be the same for all three fields.

12.3.3.7.2 Spreader for spreading factor of 1

For a spreading factor of 1, the modulated QPSK and QAM complex values a_k, where $k = 0, 1, 2, \ldots$ at the output of the constellation mapper shall be grouped into sets of 336 complex numbers. Each group shall be assigned to an OFDM symbol. This is denoted by writing the complex number $b_{k,n}$. See Equation (12-27):

$$b_{k,n} = a_{k+n\times 336}, \text{ for } k = 0,1,\ldots,335,\ n = 0, 1, 2, \ldots \tag{12-27}$$

where n is the OFDM symbol number. Each group shall be passed to the tone interleaver before being modulated by the OFDM modulator into an OFDM symbol.

12.3.3.7.3 Spreader for spreading factor of 48

For a spreading factor of 48, the modulated QPSK complex values $a(k)$, where $k = 0, 1, 2, \ldots$ at the output of the constellation mapper shall be grouped into sets of seven complex numbers. This is denoted by writing the complex number $a_{k,n}$. See Equation (12-28)

$$a_{k,n} = a_{k+n\times 28}, \text{ for } k = 0{:}6,\ n = 0, 1, 2, \ldots \tag{12-28}$$

where n is the group number. Each group shall be spread by a factor of 48 to generate a block of 336 complex numbers given by Equation (12-29):

$$b_{k,n} = q_{\text{floor}(k/28)} a_{k,n}, \text{ for } k = 0{:}167, \text{ and} \tag{12-29}$$

$$b_{k,n} = b^*_{335-k,n}, \text{ for } k = 168{:}335$$

where q is a length 12 sequence given by Equation (12-30):

$$q = [+1\ +j\ -1\ +j\ +j\ +1\ -1\ +j\ -j\ +j\ -1\ -j\ -1\ +1\ +1\ +1\ +j\ -j\ -1\ -1\ -1\ +j\ -j\ +j] \tag{12-30}$$

and the function floor() rounds its argument to the nearest integer toward minus infinity. Each spread group shall be passed to the tone interleaver before being modulated into an OFDM symbol.

12.3.3.8 Tone interleaver

All bits shall be interleaved by a block interleaver with a block size corresponding to the size of FFT in a single OFDM symbol, N_{sc}. The interleaver is used so that the adjacent data symbols are mapped onto separate subcarriers.

At the transmitter side, the interleaver permutation shall be defined as follows: Let k be the index of the tones (including data tones, pilot tones, DC tones, and null tones) before permutation ranging between 0 and $N_{sc}-1$. Let i be the index of the interleaved tones over the same range (including data tones, pilot tones, DC tones and null tones) after permutation. Let, as shown in Equation (12-31):

$$k = \sum_{j=0}^{L} a_j 2^j \tag{12-31}$$

where $L = \log_2(N_{sc}) - 1$, with $[a_L, \ldots, a_0]$ being the binary representation of integer k. Then the binary representation of integer i can be written as $[a_0, \ldots, a_L]$:

$$i = \sum_{j=0}^{L} a_j 2^{L-j}$$

DC, null, and pilot tones shall be inserted in the bit-reversal position before the tone interleaver. This ensures that after permutation, the DC, null, and pilot tones appear in the pre-specified positions.

12.3.3.9 HSI PHY OFDM modulator

12.3.3.9.1 General

The stream of interleaved complex numbers is divided into groups of N_{dsc} data complex numbers. This is denoted by writing the complex number $d_{k,n}$, which corresponds to data subcarrier k of OFDM symbol n, as shown in Equation (12-32):

$$d_{k,n} = d_{k+n\times N_D}, \quad k = 0, 1, 2, \ldots, N_D - 1, \quad n = 0, 1, \ldots, N_{hdr} + N_{OSMF} - 1 \tag{12-32}$$

where $N_{hdr} + N_{OSMF}$ is the number of OFDM symbols occupied by the header and PHY Payload field.

The discrete-time signal during the n^{th} OFDM symbol is given by Equation (12-33):

$$s_{k,n} = \frac{1}{\sqrt{N_{sc}}}\left[\sum_{m=0}^{N_D-1} d_{m,n} e^{j2\pi\frac{k\times M_D(m)}{N_{sc}}} + x_n \sum_{m=0}^{N_P-1} p_{m,n} e^{j2\pi\frac{k\times M_P(m)}{N_{sc}}} + \sum_{m=0}^{N_D-1} g_{m,n} e^{j2\pi\frac{k\times M_G(m)}{N_{sc}}}\right] \tag{12-33}$$

where

$k \in [0{:}N_{FFT} - 1]$
N_{dsc} is the number of data subcarriers
N_P is the number of pilot subcarriers
N_R is the number of reserved subcarriers
N_G is the number of guard subcarriers
N_{sc} is the number of total subcarriers
$d_{m,n}$, $p_{m,n}$, and $g_{m,n}$ are the complex numbers placed on the m^{th} data, pilot, and guard subcarriers of the n^{th} OFDM symbol, respectively

The functions $M_{dsc}(m)$, $M_P(m)$, and $M_G(m)$ define a mapping between the indices $[0{:}N_{dsc} - 1]$, $[0{:}N_P - 1]$, $[0{:}N_R - 1]$, and $[0{:}N_G - 1]$ into the logical frequency offset index $[-N_{sc}/2{:}N_{sc}/2 - 1]$. The definition for the mapping functions are given by the following:

$$M_{dsc}(m) = \begin{cases} m - 177 + round(m/21) & 0 \le m \le 167 \\ m - 174 + round[(m+1)/21] & 168 \le m \le 335 \end{cases}$$

$$M_P(m) = \begin{cases} -166 + m \times 22 & 0 \le m \le 7 \\ 12 + (m - 8 \times 22) & 8 \le m \le 15 \end{cases}$$

$$M_G(m) = \begin{cases} -185 + m & 0 \le m \le 7 \\ 170 + m & 8 \le m \le 15 \end{cases}$$

where the function round(), rounds the input argument to the nearest integer.

The mapping of data and pilot subcarriers within an OFDM symbol is illustrated in Figure 12-30. The mapping is further summarized in Table 12-35. As shown in Figure 12-30, there are 16 groups of subcarriers where each group is constituted of 21 data subcarriers and one pilot subcarrier.

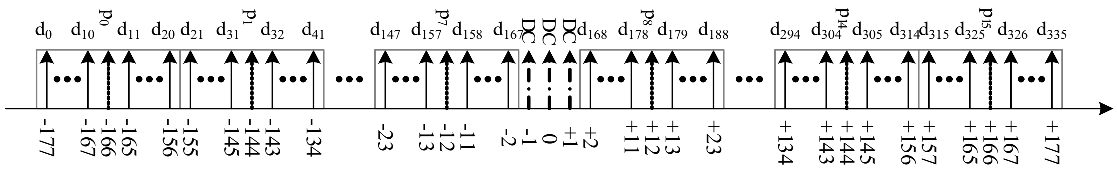

Figure 12-30—Subcarrier frequency allocation

Table 12-35—Subcarrier frequency allocation

Subcarriers type	Number of subcarriers	Logical subcarriers indexes
Null subcarriers	141	$[-256:-186] \cup [186:255]$
DC subcarriers	3	$-1, 0, 1$
Pilot subcarriers	16	$[-166:22:-12] \cup [12:22:166]$
Guard subcarriers	16	$[-185:-178] \cup [178:185]$
Data subcarriers	336	All others

The common way to implement the inverse Fourier transform is by an inverse Fast Fourier Transform (IFFT) algorithm. If, for example, a 512-point IFFT is used, the logical subcarriers 2 to 185 are mapped to the same numbered IFFT inputs, while the logical frequency subcarriers –185 to –2 are copied into IFFT inputs 327 to 510. The rest of the inputs, 186 to 326 and the 0 (DC) input, 1, and 511 are set to zero. The subcarriers 0, 1, and 511 are set to zero to avoid difficulties in D/A and A/D converter offsets and carrier feed through in the RF system. This mapping is illustrated in Figure 12-31. After performing an IFFT, the output is cyclically extended to the desired length.

12.3.3.9.2 Pilot subcarriers

In all OFDM symbols following the frame preamble, 16 of the subcarriers shall be dedicated to pilot signals in order to allow for coherent detection and to provide robustness against frequency offsets and phase noise. These pilot signals shall be placed into logical frequency subcarriers –166, –144, –122, –100, –78, –56, –34, –12, 12, 34, 56, 78, 100, 122, 144, and 166. The information for the m^{th} pilot subcarrier of the n^{th} OFDM symbol shall be defined as shown in Equation (12-34):

$$p_m = \begin{cases} (1+j)/\sqrt{2} & \text{for} \quad m = 0, 3, 5, 7, 9, 13, 15 \\ (1-j)/\sqrt{2} & \text{for} \quad m = 1, 2, 4, 6, 8, 10, 11, 12, 14 \end{cases} \tag{12-34}$$

The polarity of the pilot subcarriers is controlled by the sequence, x_n, generated by the linear feedback shift register described in 12.3.3.11.

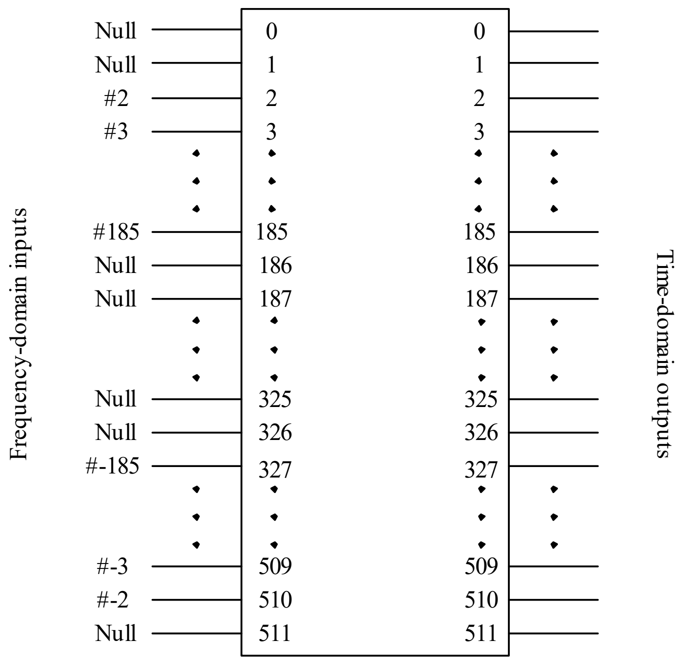

Figure 12-31—Input and output relationship of the IFFT

The 15-bit seed value chosen is $x_{-1:-15}$ = [1101 0000 1010 000]. The first 16 output values should be $x_{0:15}$ = [0001 1110 0011 1010].

12.3.3.9.3 Guard subcarriers

In all OFDM symbols following the frame preamble, there shall be 16 guard subcarriers, 8 on each edge of the occupied frequency band, at logical frequency subcarriers –185, –184, …, –178 and 178, 179, …, 185. The data on these subcarriers shall be left to the implementer. Individual implementations may exploit these guard subcarriers for various purposes, including relaxing the specifications on analog transmit and analog receive filters, and possibly peak to average power ratio reduction.

12.3.3.10 Pilot channel estimation sequence (PCES) insertion

The PCES symbols are used for channel re-acquisition or tracking. This is an optional field whose content is identical to the CES of the short preamble prepended by $\mathbf{c}_{128}$.

If PCES is used then the PCES symbols shall be inserted periodically in the PHY payload field as shown in Figure 12-32. The value of the exact period N_{PCES} shall be 96 OFDM symbols.

N_{PCES} OFDM symbols	PCES	N_{PCES} OFDM symbols	PCES	…	N_{PCES} OFDM symbols	PCES	N_{PCES} OFDM symbols

Figure 12-32—PCES positions in the PHY Payload field

12.3.3.11 HSI PHY scrambling

HSI PHY uses same scrambling method of SC PHY, which is explained in 12.2.3.10.

12.3.4 HSI PHY frame format

12.3.4.1 Overview

The HSI PHY frame shall be formatted as illustrated in Figure 12-33.

PHY preamble	Frame header	PHY payload

Figure 12-33—HSI PHY frame format

The PHY preamble is defined in 12.3.4.2.

The Frame Header field for the HSI PHY frame shall be formatted as illustrated in Figure 12-34. The goal of the frame header is to convey necessary information about both the PHY and the MAC to aid decoding of the PHY Payload at the receiver.

Main frame header					Optional frame header			
PHY header	MAC header	HCS	Parity	Stuff bits	MAC subheader	HCS	Parity	Stuff bits

Figure 12-34—HSI frame header format

The construction of main frame header and optional frame header is explained in 12.3.4.3. PHY header is defined in 12.3.4.4. MAC header is defined in 6.2, and the MAC subheader is defined in 6.2.11.

The PHY Payload field is formed by adding any necessary stuff bits to the MAC Frame Body field, as defined in 6.2, to align the data stream on the boundary of an OFDM symbol. The optional PCES may be added to the PHY Payload field.

The frame header is sent at the appropriate header rate, as defined in Table 12-30. The PHY Payload field is transmitted at the desired data rate, as defined in Table 12-28.

12.3.4.2 PHY preamble

A PHY preamble shall be added prior to the frame header to aid receiver algorithms related to frame detection, AGC setting, timing acquisition, frequency recovery, frame synchronization, and channel estimation.

The PHY preamble shall be transmitted at the rate equal to the subcarrier frequency spacing, as defined in Table 12-31, i.e., $R_S = \Delta f_{SC}$. A preamble symbol is defined as a sequence of length 512 chips that corresponds to the FFT length.

Two preambles are defined for the HSI PHY mode: the long preamble and optional short preamble. The Preamble Type field in the PHY header, as described in 12.3.4.4, indicates the type of preamble that shall be used in the next frame. The long preamble is used with MCS index 0 and is the same structure as defined for the CMS in Figure 12-7 using the chip rate of the HSI PHY mode. The short preamble is the same structure as defined for the SC PHY mode in Figure 12-20 using the chip rate of the HSI PHY mode and the Sync defined in 12.2.4.2.2, the SFD defined in 12.2.4.2.3, and the CES defined in 12.2.4.2.4. The durations of the parts of the preambles are provided in Table 12-31.

12.3.4.3 Frame header

A frame header shall be added after the PHY preamble. It conveys information in the PHY and the MAC headers necessary for a successful decoding of the frame. The frame header is constructed from a main frame header followed by an optional frame header.

The frame header shall be constructed as follows:

a) Form the main frame header as follows:
   1) Construct the PHY header based on information provided by the MAC.
   2) Compute the HCS over the combined PHY and MAC headers, as described in 12.3.4.5.1.
   3) Append the HCS to the MAC header.
   4) Scramble the combined MAC header and HCS, as described in 12.3.3.5.
   5) Form the main frame header by concatenating the PHY header, scrambled MAC header, scrambled HCS.
   6) The resulting main frame header shall be modulated as follows:
      i) The main frame header is further encoded with shortened LDPC encoder of rate 1/2, as described in 12.3.3.4.2.
      ii) The resulting main frame header has stuff bits added, as indicated in Table 12-30.
      iii) The resulting main frame header is then mapped to a QPSK constellation, as described in 12.3.3.6.
      iv) The resulting complex values are passed to the OFDM modulator, as described in 12.3.3.9. The cyclic prefix during the header is fixed to 128 chips.
b) Form the optional frame header as follows:
   1) Compute the HCS over the MAC subheader, as described in 12.3.4.5.1.
   2) Append the HCS to the MAC subheader.
   3) Scramble the combined MAC subheader and HCS, as described in 12.3.3.5.
   4) Form the optional frame header by concatenating the scrambled MAC subheader, scrambled HCS.
   5) The resulting optional frame header shall be modulated as follows:
      i) The optional frame header is further encoded with shortened LDPC encoder of rate 1/2, as described in 12.3.3.4.2.
      ii) The resulting optional frame header has stuff bits added, as indicated in Table 12-30.
      iii) The resulting optional frame header is then mapped to a QPSK constellation, as described in 12.3.3.6.
      iv) The resulting complex values are passed to the OFDM modulator, as described in 12.3.3.9. The cyclic prefix during the header is fixed to 128 chips.
c) Form the frame header by concatenating the main frame header and optional frame header.

The construction and modulation of the main frame header for HSI PHY is illustrated in Figure 12-35.

The construction and modulation of the optional header for HSI is illustrated in Figure 12-36.

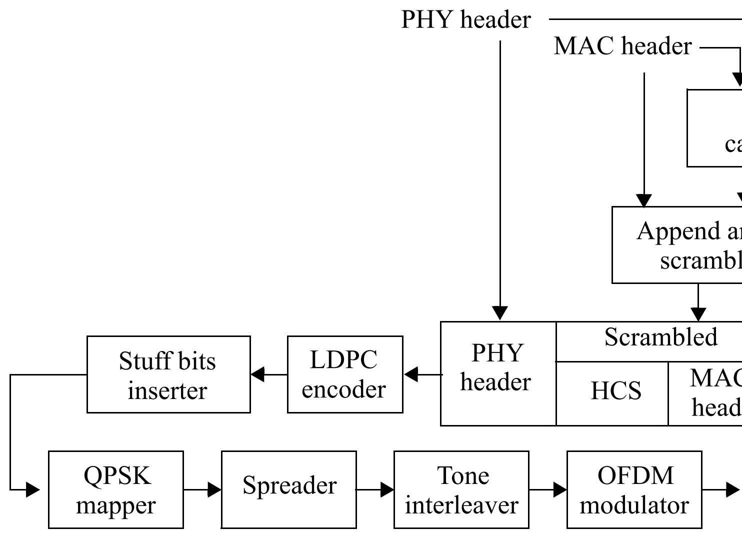

Figure 12-35—HSI main frame header encoding process

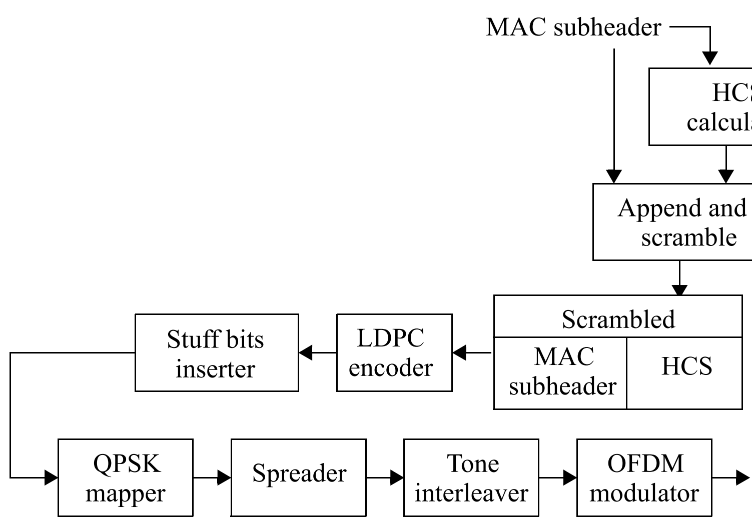

Figure 12-36—HSI optional frame header encoding process

12.3.4.4 HSI PHY header

The HSI PHY header field shall be formatted as illustrated in Figure 12-37.

Bits: b0–b3	b4	b5	b6–b10	b11–b30	b31–b32
Scrambler ID	Aggregation	UEP	MCS	Frame Length	Preamble Type

b33	b34	b35	b36	b37	b38–b47
Beam Tracking	Low-latency Mode	Bit Interleaver	PCES Mode	Skewed Constellation	Reserved

Figure 12-37—HSI PHY header format

The scrambling process is described in 12.3.3.5. The MAC shall set the bits defined in 12.3.3.11 according to the scrambler seed identifier value.

The Aggregation field shall be set to one if aggregation is used; it shall be set to zero otherwise.

The UEP field shall be set to one if any type of UEP is used; it shall be set to zero otherwise.

The MCS field shall be set to the MCS used in the MAC Frame Body field, according to the values in Table 12-36. MCS field also identifies which header rate should be used, as defined in Table 12-30. The MCS identifiers in relation to the data rate, modulation, spreading code length, FEC, pilot word, and burst length are provided in Table 12-28.

Table 12-36—MCS field

MCS identifier	MCS index
0b00000	0
0b00001	1
0b00010	2
0b00011	3
0b00100	4
0b00101	5
0b00110	6
0b00111	7
0b01000	8
0b01001	9
0b01010	10
0b01011	11
0b01100–0b11111	Reserved

The Frame Length field shall be an unsigned integer number that indicates the number of octets in the Frame Payload field (which does not include the FCS).

The Preamble Type field indicates the type of the PHY preamble (long or short) used in the next frame, as defined in Table 12-37.

Table 12-37—Preamble type field

Preamble type	Type of preamble used for next frame
0b00	Long preamble
0b01	Short preamble
0b10–0b11	Reserved

The Beam Tracking field shall be set to one if training sequences for beam tracking are present following the current frame. It shall be set to zero otherwise.

The Low-latency Mode field shall be set to one if the frame is using the low-latency aggregation mode, and shall be set to zero otherwise.

If the Low-latency Mode field is set to one, then the Aggregation field shall also be set to one.

The Bit Interleaver field shall be set to one to indicate that bit interleaving is used in the payload of the frame and shall be set to zero otherwise.

The PCES Mode field shall be set to one to indicate that PCES symbols are added periodically to the PHY payload field, and it shall be set to zero otherwise.

The Skewed Constellation field is optional. When used, this field shall be set to one if UEP uses constellation mapping and shall be set to zero to indicate that UEP uses coding. If the field is not used, it shall be set to zero.

12.3.4.5 Header check sequences (HCSs)

12.3.4.5.1 Main Header HCS

The combination of the PHY header and the MAC header shall be protected with an ITU-T CRC-16 HCS. The ITU-T CRC-16 is described in 11.2.9.

12.3.4.5.2 MAC Subheader HCS

The MAC subheader shall be protected with an ITU-T CRC-16 HCS. This shall be computed in the same way, as specified in 11.2.9.

12.3.4.6 PHY Payload field

The PHY Payload field is the last component of the frame, and is constructed as shown in Figure 12-38.

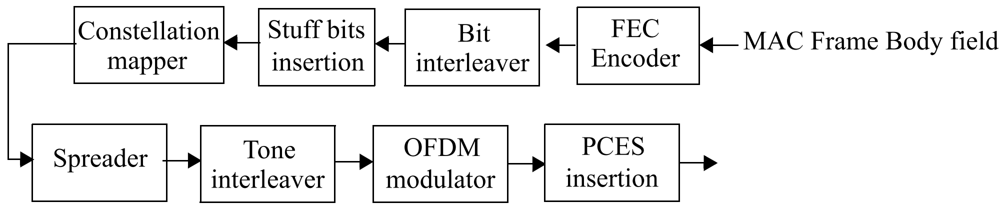

Figure 12-38—PHY Payload field construction process

The PHY Payload field shall be encoded as follows:

a) Scramble the MAC Frame Body field according to 12.3.3.11.
b) Encode and interleave the scrambled MAC Frame Body field, as specified in 12.3.3.4.
c) If necessary, add stuff bits, as defined in 12.3.3.5.
d) Map the MAC Frame Body field onto the appropriate constellation, as described in 12.3.3.6.
e) Spread the modulated signal, as described in 12.3.3.7.
f) Interleave the MAC Frame Body field with the tone interleaver, as described in 12.3.3.8.
g) Modulate the resulting complex values with the OFDM modulator described in 12.3.3.9.
h) Insert PCES periodically, as described in 12.3.3.10.

Note that when the frame payload length is zero, the length of the PHY Payload field shall also be zero.

12.3.5 Transmitter specifications

12.3.5.1 EVM requirement

The EVM of a compliant transmitter shall be measured and calculated as defined in 12.1.8 and shall not exceed the values given in Table 12-38.

Table 12-38—Allowed relative constellation error versus data rate

Data rate	Relative constellation RMS error
Up to 1.5 Gb/s	–7 dB
2.1 Gb/s to 2.7 Gb/s	–14 dB
2.8 Gb/s to 5.3 Gb/s	–21 dB
Above 5.4 Gb/s	–23 dB

The relative constellation RMS error calculation shall be performed using a device capable of converting the transmitted signal into a stream of complex samples at the sampling rate, f_s, as defined in Table 12-31, or higher, with sufficient accuracy in the I/Q imbalance, dc offset, phase noise, etc. The sampled signal shall then be processed in a manner similar to that of an ideal receiver including adding the 64 samples of the cyclic prefix to the received OFDM symbol.

12.3.5.2 Chip rate and clock alignment

The transmitted center frequency and chip clock frequency tolerances shall be ±20 10^{-6} maximum. The transmit center frequency and symbol clock frequency shall be derived from the same reference oscillator (locked).

12.3.6 Receiver specifications

12.3.6.1 Receiver sensitivity

For a bit error rate (BER) of 10^{-6}, the minimum receiver sensitivity shall be –50 dBm for MCS index 1 and shall be –70 dBm for MCS index 0. The BER shall be measured at the MAC/PHY interface after PHY level FEC. A pseudo noise (PN) sequence defined by $x^{55} + x^{24} + 1$ shall be used for the BER measurement.

12.3.6.2 Receiver CCA performance

The start of a valid transmission at a receiver equal to or greater than the minimum sensitivity for MCS index 0, if MCS index 0 is implemented, shall cause CCA to indicate busy with a probability > 90% within *pCcaDetectTime*, as defined in Table 12-26. The receiver CCA function shall in all circumstances report the medium busy with any signal 20 dB above the minimum sensitivity for MCS index 0.

If CMS is supported, then the receiver CCA performance is described in 12.1.13.9.

12.3.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal present at the input of the receiver for which the error rate criterion is met. A compliant receiver shall have a receiver maximum input level of at least –25 dBm for each of the modulation formats that the device supports.

12.3.6.4 PHY layer timing

The values for the PHY layer timing parameters are given in Table 12-39.

Table 12-39—PHY layer timing parameters

PHY parameter	Value	Subclause
pMifsTime	0.2 µs, 0.5 µs (default), 2.0 µs	12.3.6.8
pSifsTime	0.2 µs, 2.0 µs, 2.5 µs (default)	12.3.6.6
pCcaDetectTime	2.5 µs	12.3.6.2
pChannelSwitchTime	100 µs	12.3.6.9

12.3.6.5 Interframe spacing

The interframe spacing parameters are given in Table 12-40.

Table 12-40—HSI interframe spacing parameters

MAC parameter	PHY parameter	Definition
MIFS	*pMifsTime*	11.2.7.5
SIFS	*pSifsTime*	11.2.7.3
pBackoffSlot	*pSifsTime* + *pCcaDetectTime*	11.2.7.2
BIFS	*pSifsTime* + *pCcaDetectTime*	11.2.7.2
RIFS	2 × *pSifsTime* + *pCcaDetectTime*	11.2.7.2

12.3.6.6 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be less than *pSifsTime*. The RX-to-TX turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

12.3.6.7 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than *pSifsTime*. The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol until the receiver is ready to begin the reception of the next PHY frame.

12.3.6.8 Time between successive transmissions

The time between successive transmissions shall be *pMifsTime*, including the power-up ramp specified in 11.5.7. The *pMifsTime* shall be measured at the air interface from the trailing edge of the last symbol transmitted until the first symbol of the PHY preamble is present at the air interface.

12.3.6.9 Channel switch time

The channel switch time is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pChannelSwitchTime*.

12.3.7 HSI PHY management

12.3.7.1 General

The PHY PIB comprises the managed objects, attributes and notifications required to manage the PHY layer of a DEV.

12.3.7.2 PHY supported data rate encoding

The encoding of the PHY data rates used in the Supported Data Rates field in the Capability IE, as described in 6.4.12, is given in 12.1.9.1.

12.3.7.3 HSI PHY fragment size encoding

The encoding of the preferred fragment size used in the Capability IE, as described in 6.4.12, is given in Table 12-4.

The PHY definitions create restrictions on the maximum frame size, maximum transfer unit size, and minimum fragmentation size that will be supported.

12.3.7.4 Maximum frame length

The maximum frame length allowed, *pMaxFrameBodySize*, shall be $2^{20} - 1$ octets. This total includes the Frame Body field and FCS but not the PHY preamble, PHY header, or MAC header. The maximum frame length also does not include the tail symbols or the stuff bits.

12.3.7.5 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be the same as *pMaxFrameBodySize*, as defined in 12.3.7.4. If security is enabled for the data connection, the upper layers should limit data frames to *pMaxFrameBodySize* minus the security overhead, as defined in 6.3.5.2.

12.3.7.6 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, shall be 512 octets.

12.4 Audio/Visual (AV) mode of mmWave PHY

12.4.1 General

The AV PHY is implemented with two PHY modes: the HRP and LRP, both of which use OFDM. The data rates supported by the HRP are defined in Table 12-41.

Table 12-41—HRP data rates and coding

HRP mode index	Coding mode	Modulation	Inner code rate		Data rate (Gb/s)
			MSB	LSB	
0	EEP	QPSK	1/3		0.952
1		QPSK	2/3		1.904
2		16-QAM	2/3		3.807
3	UEP	QPSK	4/7	4/5	1.904
4		16-QAM	4/7	4/5	3.807
5	MSB-only retransmission	QPSK	1/3	N/A	0.952
6		QPSK	2/3	N/A	1.904

If a DEV supports the use of the HRP, it shall support the use of HRP mode index 0 and HRP mode index 1.

HRP mode index 5 and HRP mode index 6 transmit only the four MSBs of each octet. In this case the data rate refers to MSBs only. The four LSBs in each octet are discarded by the transmitter and only the MSB portions of the baseband, as shown in Figure 12-39, shall be used.

Typical video and audio consumer electronics are configured either as a source of data, e.g., a video disc player, or as a sink, e.g., a display. For these applications, the data flow is highly asymmetric in the same direction. Thus, AV PHY DEVs shall implement one of the following configurations:

— HR0: The DEV implements LRP transmit and receive functions.
— HRRX: The DEV implements HRP receive, LRP transmit, and LRP receive functions.
— HRTX: The DEV implements HRP transmit, LRP transmit, and LRP receive functions.
— HRTR: The DEV implements HRP transmit, HRP receive, LRP transmit, and LRP receive functions.

The data rates supported by the LRP are defined in Table 12-42.

Table 12-42—LRP data rates and coding

LRP mode index	Modulation	FEC	Data rate (Mb/s)	Repetition
0	BPSK	1/3	2.5	8×
1		1/2	3.8	8×
2		2/3	5.1	8×
3		2/3	10.2	4×

A DEV that supports the AV PHY mode shall support LRP modes 0, 1, and 2. All broadcast and multi-cast frames using the AV PHY shall be sent with an LRP mode.

12.4.2 General requirements

12.4.2.1 AV PHY channelization

The HRP mode uses the channels defined in Table 12-3. A compliant IEEE 802.15.3 implementation that implements the AV PHY shall support at least channel number 2.

In each of the HRP channels, three LRP channels are defined. In a piconet, only one HRP channel and one LRP channel is used at a time. Each of the LRP channels is defined relative to the center frequency of the current HRP channel, $f_{c(HRP)}$, as defined in Table 12-3. The LRP channels are defined in Table 12-43.

Table 12-43—LRP channelization

LRP channel index	Start frequency[a]	Center frequency	Stop frequency[a]
1	$f_{c(HRP)}$ – 207.625 MHz	$f_{c(HRP)}$ – 158.625 MHz	$f_{c(HRP)}$ – 109.625 MHz
2	$f_{c(HRP)}$ – 49 MHz	$f_{c(HRP)}$	$f_{c(HRP)}$ + 49 MHz
3	$f_{c(HRP)}$ + 109.625 MHz	$f_{c(HRP)}$ + 158.625 MHz	$f_{c(HRP)}$ + 207.625 MHz

[a]The start and stop frequencies are nominal values. The TX mask requirements for the LRP is defined in 12.4.5.1.

The *phyCurrentChannel* is the CHNL_ID of the current channel. For the purpose of the Remote Scan Request and Remote Scan Response commands, 6.5.8.4 and 6.5.8.5, respectively, the Channel Index field is the CHNL_ID in Table 12-44.

Table 12-44—Mapping HRP/LRP channel index to CHNL_ID

CHNL_ID	HRP channel index	LRP channel index
1	1	1
2	2	1
3	3	1
4	4	1
5	1	2
6	2	2
7	3	2
8	4	2
9	1	3
10	2	3
11	3	3
12	4	3
13	5	1
14	6	1
15	5	2
16	6	2
17	5	3
18	6	3

12.4.2.2 PHY layer timing

12.4.2.2.1 General

The values for the AV PHY layer timing are defined in Table 12-45.

Table 12-45—AV PHY layer timing parameters

PHY parameter	Value	Subclause
pMifsTime	*pSifsTime*	12.4.2.2.5
pSifsTime	2 µs ± 24 samples at LRP reference sampling rate	12.4.2.2.3
pCcaDetectTime	9 µs	12.4.2.2.7
pChannelSwitchTime	1000 µs	12.4.2.2.6

12.4.2.2.2 Interframe space

A conformant AV PHY implementation shall support the IFS parameters, as described in 7.6.2, given in Table 12-46.

Table 12-46—AV PHY IFS parameters

IEEE 802.15.3 MAC parameter	Corresponding AV PHY parameter	Definition
MIFS	*pMifsTime*	12.4.2.2.5
SIFS	*pSifsTime*	12.4.2.2.3
pBackoffSlot	*pSifsTime* + *pCcaDetectTime*	12.4.2.2.3, 12.4.2.2.7
BIFS	*pSifsTime* + *pCcaDetectTime*	12.4.2.2.3, 12.4.2.2.7
RIFS	2 × *pSifsTime* + *pCcaDetectTime*	12.4.2.2.3, 12.4.2.2.7

12.4.2.2.3 Receive-to-transmit turnaround time

The RX-to-TX turnaround time shall be *pSifsTime*. The RX-to-TX turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

12.4.2.2.4 Transmit-to-receive turnaround time

The TX-to-RX turnaround time shall be less than *pSifsTime*. The TX-to-RX turnaround time shall be measured at the air interface from the trailing edge of the last transmitted symbol until the receiver is ready to begin the reception of the next PHY frame.

12.4.2.2.5 Time between successive transmissions

The time between successive transmissions shall be *pMifsTime*. The *pMifsTime* shall be measured at the air interface from the trailing edge of the last symbol transmitted until the first symbol of the PHY preamble is present at the air interface.

12.4.2.2.6 Channel switch time

The channel switch time is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pChannelSwitchTime*.

12.4.2.2.7 CCA detect time

The LRP shall be able to detect the presence of an LRP preamble at an input power level equivalent to the sensitivity of the LRP mode 1 within *pCcaDetectTime*. CCA detection is not used for the HRP mode because only the LRP is used in CPs.

12.4.2.3 Data size restrictions

12.4.2.3.1 General

The PHY definitions creates restrictions on the maximum frame size, maximum transfer unit size, and minimum fragmentation size that will be supported. These parameters are defined in the subclauses that follow.

12.4.2.3.2 Maximum frame length

The maximum frame length allowed, *pMaxFrameBodySize*, shall be $2^{20} – 1$ octets for HRP subframes and $2^{12} – 1$ octets for LRP frames. This total includes the frame payload and FCS but not the PHY preamble, PHY header, MAC header, or HCS. The maximum frame length also does not include the tail symbols or the stuff bits.

12.4.2.3.3 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be the same as *pMaxFrameBodySize*, as defined in 12.4.2.3.2. If security is enabled for the data connection, the upper layers should limit data frames to *pMaxFrameBodySize* minus the security overhead, as defined in 6.3.5.2.

12.4.2.3.4 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, that is allowed with the AV PHY shall be 512 octets.

12.4.2.4 Header check sequence

The HCS shall be a 32-bit CRC that is equivalent to the one used for the FCS, as defined 6.2.10.6. For the AV PHY, the MAC parameter *pLengthHcs* shall be four. The HCS shall be transmitted in the order specified in 6.1. At both the receiver and transmitter, the initial state of the remainder shall be set to all ones.

12.4.3 AV PHY modulation and forward error correction

12.4.3.1 General

The modulation parameters for the HRP are given in Table 12-47.

The modulation parameters for the LRP are given in Table 12-48.

Unless otherwise specified, the terms *sampling rate* or *sample* are in terms of the reference sampling rate for that mode.

Table 12-47—HRP modulation parameters

Parameter	Value	Symbol
Occupied bandwidth	1.76 GHz	N/A
Reference sampling rate	2.538 GHz	$f_{s(HR)}$
Number of subcarriers	512	$N_{sc(HR)}$
FFT period	$N_{sc(HR)}/f_{s(HR)}$ ~ 202 ns	$T_{FFT(HR)}$
Subcarrier spacing	$1/T_{FFT(HR)}$ ~ 4.96 MHz	$\Delta f_{SC(HR)}$
Guard interval	$64/f_{s(HR)}$ ~ 25.2 ns	$T_{GI(HR)}$
Symbol duration	$T_{FFT(HR)} + T_{GI(HR)}$ ~ 227 ns	$T_{S(HR)}$
Number of data subcarriers	336	$N_{dsc(HR)}$

Table 12-48—LRP modulation parameters

Parameter	Value	Symbol
Occupied bandwidth	92 MHz	N/A
Reference sampling rate	317.25 MHz	$f_{s(LR)}$
Number of subcarriers	128	$N_{sc(LR)}$
FFT period	$N_{sc(LR)}/f_{s(LR)}$ ~ 403 ns	$T_{FFT(LR)}$
Subcarrier spacing	$1/T_{FFT(LR)}$ ~ 2.48 MHz	$\Delta f_{SC(LR)}$
Guard interval	$28/f_{s(LR)}$ ~ 88.3 ns	$T_{GI(LR)}$
Symbol duration	$T_{FFT(LR)} + T_{GI(LR)}$ ~ 492 ns	$T_{S(LR)}$
Number of data subcarriers	30	$N_{dsc(LR)}$

A reference implementation of the HRP baseband is illustrated in Figure 12-39.

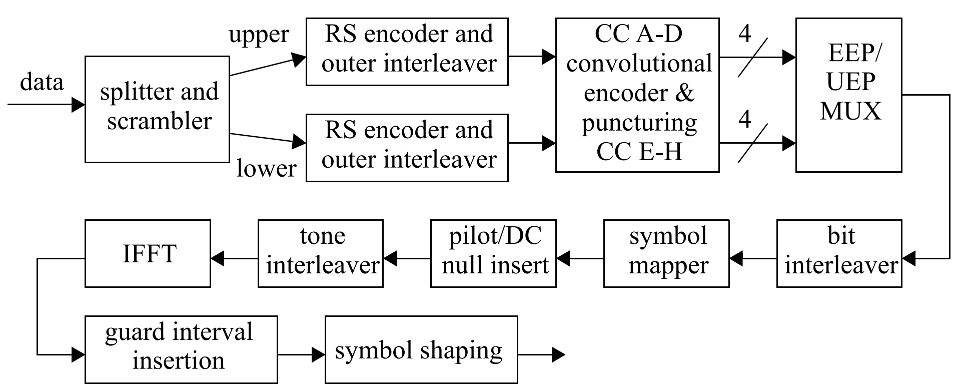

Figure 12-39—HRP reference implementation block diagram

A reference implementation of the LRP baseband is illustrated in Figure 12-40.

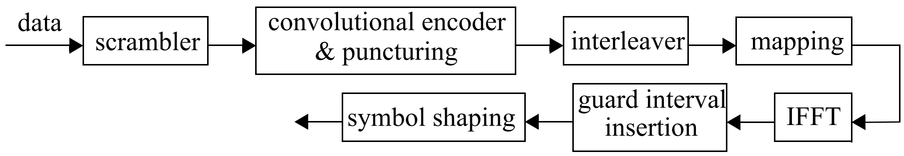

Figure 12-40—LRP reference implementation block diagram

The purpose of the reference implementation is to provide a reference for defining the encoding of the incoming bit stream into the appropriate RF signal. Implementations may use different architectures as long as the resulting RF signal matches the one that would be generated by a reference implementation.

12.4.3.2 AV PHY base rate

The base data rate of the AV PHY shall be LRP mode 0, 1, or 2.

12.4.3.3 Repetition coding and spatial diversity

The LRP transmitter utilizes spatial diversity and repetition coding for the omni-directional and directional modes. Up to eight antenna directions or phased array patterns may be used that shall be identified by indices zero through seven.

For omni LRPDUs, the repetition coding is implemented by repeating the OFDM symbol and its associated cyclic prefix four or eight times as indicated by the LRP mode index and sending each repetition with a different TX antenna direction or pattern. For the case of eight times repetition coding, each OFDM symbol in the Channel Estimation field and the OFDM symbols that follow in the frame are repeated using patterns zero through seven. For the four times repetition coding, each OFDM symbol in the MAC header, HCS, and MAC Frame Body field is repeated using patterns zero through three, where patterns zero through three are the first four patterns used in the Channel Estimation field of the omni LRP preamble. The assignment of indices to the patterns is implementation dependent and arbitrary. However, for all omni LRPDUs that use the short preamble, with the exception of LRP mode 3, the assignment of indices shall remain unchanged after the system is powered on. For omni LRPDUs that use the long omni LRP preamble, or omni LRPDUs that use the short preamble with LRP mode 3, the assignment of indices may be different from frame to frame. However, in these cases the assignment of indices shall remain unchanged for the duration of a frame. These TX antenna directions or patterns are selected such that the transmission covers the region of space that is of interest. The patterns that are used do not need to be unique as long as the same set of patterns is used for all omni LRPDUs. For example, for implementations with less than eight independently controllable elements, some of the patterns are repeated during the eight times repetition. In the case of a single TX antenna, a single direction or pattern is used for all of the repetitions.

In the Directional mode, the preamble is used to perform channel estimation and receiver training for only one optimum TX antenna direction, or phase array pattern. The same TX optimum pattern is also used for the header and payload. This optimum direction is selected while an omni LRPDU with short preamble is received. The Directional LRPDU Header field shall use the eight times directional repetition coding while the directional repetition coding of the data symbols is either eight times or four times as determined by the LRP mode index. The directional repetition coding for Directional LRPDUs is performed by adding one additional OFDM symbol as cyclic prefix to the eight or four times repeated 128-sample OFDM symbols depending on the mode index. Therefore, for eight times directional repetition, the 128-sample OFDM symbol is repeated nine times, while it is repeated five times for four times directional repetition. In this case, no other cyclic prefix is used.

12.4.3.4 Stuff bits

In order that an integer number of OFDM symbols are created and, for the HRP, complete the outer interleaver units, the PHY adds additional bits to the bit stream, called *stuff bits*, prior to performing any operations on the incoming data. Stuff bits for the HRP shall be set to zero prior to adding them to the end of the bit stream. Stuff bits for the LRP shall be set equal to a sequence generated by the scrambler, defined in 12.4.3.6, initialized to an arbitrary state. In LRPDUs, stuff bits only occur at the end of a frame, i.e., after the Frame Body field or the Directional LRP Payload field.

The PHY shall add the minimum number of stuff bits necessary to create an integer number of OFDM symbols and, for HRP modes, complete the outer interleaver unit for the combination of the PHY Header field, MAC Header field, and HCS field. In addition, the HRP PHY shall add the minimum number of stuff bits necessary to create an integer number of OFDM symbols and complete the outer interleaver unit for each combination of subframes that have the same HRP mode index and for the last subframe in the frame. These additional bits shall be discarded by the receiver upon reception. The last outer interleaver unit for each of the subframes that end on a HRP mode change and for the last subframe may be shortened, as specified in 12.4.3.8.

12.4.3.5 HRP splitter and scrambler

For the HRP header and MSB only mode, no re-ordering is applied to the input octet stream before it is sent to the scrambler. For all other HRP modes, prior to scrambling, the input octets are re-ordered into upper and lower branches. If the input to the splitter is a an array of octets, $in(i, b)$, where i is the index of the input octets, $i = 0, 1, \ldots, N - 1$, where N is the number of octets in the input stream, and b is the index of the bit in the octet where $b = 0$ is the LSB and $b = 7$ is the MSB. The output of the scrambler is two arrays of octets, $upper(n, b)$ and $lower(n, b)$, where $n = 0, 1, \ldots, N - 1$. Note that adding the stuff bits guarantees that the input octet stream is an even number of octets.

In EEP mode, the output arrays are constructed according to Equation (12-35) and Equation (12-36):

$$lower(n, b) = in(\{\mathrm{floor}[b/4] + \mathrm{floor}[n/2] \times 4\}, \{\mathrm{mod}[b, 4] + \mathrm{mod}[n, 2] \times 4\}) \tag{12-35}$$

$$upper(n, b) = in(\{\mathrm{floor}[b/4] + \mathrm{floor}[n/2] \times 4 + 2\}, \{\mathrm{mod}[b, 4] + \mathrm{mod}[n, 2] \times 4\}) \tag{12-36}$$

where $\mathrm{mod}(x, y)$ is the modulo function and is defined in 12.2.3.6.4.

In UEP mode, the output arrays are constructed according to Equation (12-37) and Equation (12-38):

$$lower(n, b) = in(\{\mathrm{floor}[b/4] + n \times 2\}, \mathrm{mod}[b, 4]) \tag{12-37}$$

$$upper(n, b) = in(\{\mathrm{floor}[b/4] + n \times 2\}, \{\mathrm{mod}[b, 4] + 4\}) \tag{12-38}$$

When the octets are re-ordered, the first 8 bits of the scrambler are applied to the lower branch and the next 8 bits to the upper branch, switching every 8 bits between the two branches.

The HRP scrambler shall use the generator polynomial $P(x) = x^{15} + x^{14} + 1$. A reference implementation is illustrated in Figure 12-14.

The initial value of scrambler shall be set by the four variable seeds, S_0, S_1, S_2 and S_3, and the 11 fixed seeds as $[x_{-1}, x_{-2}, \ldots, x_{-15}] = [1101\ 0000\ 101\ S_3\ S_2\ S_1\ S_0]$. All fields in the HRP header, MAC header, and HCS, as illustrated in 12.4.4, shall be scrambled as with the variable seeds set to $S_0 = 0$, $S_1 = 1$, $S_2 = 0$, $S_3 = 1$. The entire data stream following the HCS, including the stuff bits, is scrambled using the seed bits specified in the PHY Control field, as illustrated in Figure 12-55.

12.4.3.6 LRP scrambler

Scrambling applies to the MAC header, HCS, and Frame Body field only. A scrambler with 6 state bits, s0 through s5, and wired according to the polynomial $x^6 + x + 1$ is initialized with a 4-bit random value combined with 0b01 as illustrated in Figure 12-41. Bit s5 from successive states of the scrambler are exclusive-ORed with each data bit in sequence to create the scrambled data. The 4-bit random value that is used to initialize the scrambler is placed in the fields S_0, S_1, S_2, and S_3 in Figure 12-60.

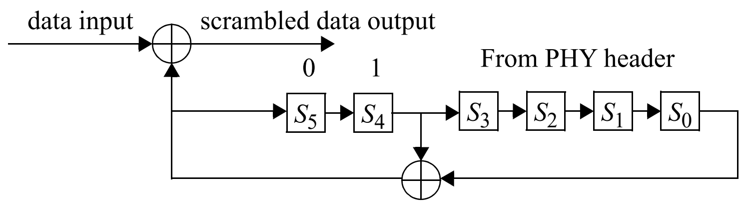

Figure 12-41—LRP scrambler reference implementation

NOTE—The directional LRPDU, which does not contain the MAC Header, HCS, or MAC Frame Body field, is not scrambled.

12.4.3.7 HRP outer code

For HRP modes, the output from the scrambler is split into two branches, MSB and LSB. The data in each branch shall have an outer code using a Reed-Solomon code with parameters (224, 216, t = 4). The last portion of the data shall be coded by a further shortened Reed-Solomon code. The tail bits for the inner convolutional code are inserted in the outer interleaver by further shortening of the Reed-Solomon code. The parameters of these shortened codes are defined in 12.4.3.8.

The field generator polynomial is shown in Equation (12-39)

$$g_{field}(x) = x^8 + x^4 + x^3 + x^2 + 1 \tag{12-39}$$

and the code generator polynomial is shown in Equation (12-40):

$$g_{RS}(x) = (x+\lambda)(x+\lambda^2)(x+\lambda^3)(x+\lambda^4)(x+\lambda^5)(x+\lambda^6)(x+\lambda^7)(x+\lambda^8) \tag{12-40}$$

where

λ = 0x02

The combination of PHY Header field, MAC Header field, and HCS shall use only the MSB branch.

The input and output order for the Reed-Solomon encoder is specified in 12.4.3.8 in combination with the outer interleaver.

12.4.3.8 HRP outer interleaver

The outer interleaver shall output the octets from $i = 0$, $k = 0$ first to $i = depth–1$, $k = N–1$ last, where *depth* is the depth of the outer interleaver and N is the length of RS code. With M parallel convolutional inner encoders for each RS codeword and $b(n, m)$ is the output of the RS encoder, the outer interleaver shall give the RS octets of b(0,0), …, b(*depth*–1,0) to the first convolutional encoder with LSB first. All octets of $b(i,k \times M + m)$, $i = 0, …, depth–1$, $k = 0, 1, …, N/M–1$, shall be output to the m^{th} convolutional encoder. The number of parallel convolutional encoders is specified in 12.4.3.9.

The outer block interleaver shall be operated with a *depth* = 4 for HRP data and a *depth* = 2 for the combination of the HRP Header, MAC Header, and HCS fields. LRP modes do not use an outer block interleaver.

The combination of HRP header, MAC header and HCS field has 92 octets that are encoded into 112 octets by adding 16 parity octets for error protection and 4 tail octets to terminate the convolutional code.

For the combination of the HRP Header, MAC Header, and HCS fields, the first 48 octets are encoded using RS(56, 48, $t = 4$) while the next 44 octets are encoded using RS(52, 44, $t = 4$). The second codeword is followed by 4 tail octets set to zero. The transmitted order of those octets, the method to insert the tail octets, and the method of interleaving for the outer interleaver are the same as those used for data.

The outer interleaver inserts the tail bits for the convolutional encoder. For the outer interleaver with tail bits, to improve the efficiency, the number of rows of the outer interleaver may be reduced to a minimum number that is an integer multiple of 28. At the columns of $i = 0$ to $i = depth - 2$, a shortened RS($28 \times n$, $28 \times n - 8$, $t = 4$) code may be used, where $n = 1$ to 8.

12.4.3.9 Convolutional encoder

Scrambled data, tail bits, and stuff bits shall be encoded with a convolutional encoder of code rate 1/3. The convolutional encoder shall use constraint length $K = 7$, delay memory 6, generator polynomial $g0 = 133o$, $g1 = 171o$, $g2 = 165o$, mother code rate 1/3. A detailed schematic diagram of the convolutional encoder is shown in Figure 12-42.

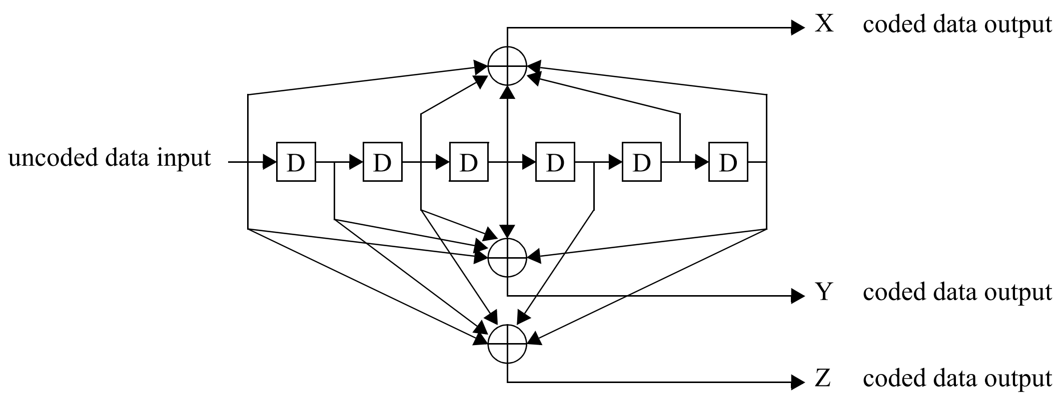

Figure 12-42—Convolutional encoder reference implementation

The initial value of the delay register shall be zero at the beginning of every HRP header and at the beginning of the Frame Body field.

The HRP transmitter uses eight parallel convolutional encoders, labeled A through H. The first four encoders, labeled A through D, are for the first outer Reed-Solomon coding branch, and the last four encoders, labeled E through H are for the second outer Reed-Solomon coding branch.

The HRP header shall use the same convolutional encoder with a code rate of 1/3. However, only four parallel convolutional encoders, labeled A through D, are used.

For all LRP Frame Body fields other than with the directional LRP payload, six tail bits with value zero are added. For LRP PHY header, the 1/2 and 1/3 rate coding with tail-biting are used. In the tail-biting method, the initial encoder state is set equal to the last six information bits, and no tail bits are appended. Therefore, the initial and final encoder states are equal. The directional LRPDU payload shall use either tail-biting or tail bits, as specified in 12.4.4.9.

12.4.3.10 Puncturing

Convolutional encoded data is punctured to make the desired code rate using the puncturing pattern indicated in Table 12-49. In Table 12-49, puncturing pattern “1” means to send the bit while a “0” means to omit (or do not transmit) the corresponding bit.

The output of puncturing block shall be serialized with sequential order as shown in Table 12-49.

Table 12-49—Puncturing table

Code rate	Puncturing pattern	Transmitted sequence
1/3	X: 1 Y: 1 Z: 1	X1 Y1 Z1
1/2	X: 1 Y: 1 Z: 0	X1 Y1
4/7	X: 1 1 1 1 Y: 1 0 1 1 Z: 0 0 0 0	X1 Y1 X2 X3 Y3 X4 Y4
2/3	X: 1 1 Y: 1 0 Z: 0 0	X1 Y1 X2
4/5	X: 1 1 1 1 Y: 1 0 0 0 Z: 0 0 0 0	X1 Y1 X2 X3 X4

12.4.3.11 HRP data multiplexer and bit interleaver

12.4.3.11.1 General

The output of the 8 encoders, labeled A through H, shall be multiplexed to form a single data stream prior to the bit interleaver, defined in 12.4.3.12, as illustrated in Figure 12-39. The method used to multiplex the encoded bits is dependent on the type of HRP mode, either EEP or UEP.

12.4.3.11.2 EEP data multiplexer

In the EEP mode, all the 8 encoders shall use the same inner code rate. The encoded bits shall be multiplexed and bit-interleaved every 48 bits.

During the length 48 multiplexing/interleaving cycle, a group multiplexer shall be used first with fixed group size 6 for all eight encoders. A1, A2, A3, A4, A5, A6 are used to label the 6 encoded bits (in increasing order in time) from encoder A, and similarly for B1 through B6, C1 through C6, D1 through D6, E1 through E6, F1 through F6, G1 through G6, and H1 through H6 from encoders B, C, D, E, F, G, and H, respectively. At the output of the multiplexer, the 48 encoded bits shall be ordered and numbered as illustrated in Table 12-50.

Table 12-50—Multiplexing scheme for EEP mode

numbering	0	1	2	3	4	5	6	7	8	9	10	11
labeling	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6
numbering	12	13	14	15	16	17	18	19	20	21	22	23
labeling	C1	C2	C3	C4	C5	C6	D1	D2	D3	D4	D5	D6
numbering	24	25	26	27	28	29	30	31	32	33	34	35
labeling	E1	E2	E3	E4	E5	E6	F1	F2	F3	F4	F5	F6
numbering	36	37	38	39	40	41	42	43	44	45	46	47
labeling	G1	G2	G3	G4	G5	G6	H1	H2	H3	H4	H5	H6

The multiplexed 48 bits are then sent to the bit interleaver. Let x = 0, …, 47 and y = 0, …, 47 be the index at the input and output of the bit interleaver, respectively. The bit interleaver in the EEP mode shall implement the relation given by Equation (12-41):

$$y = \mathrm{mod}((6 \times \mathrm{floor}(x/6) - 5 \times \mathrm{mod}(x,6)), 48) \tag{12-41}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

The overall read out order of the multiplexer and interleaver for EEP is illustrated in Figure 12-43.

column-wise readout (from top to bottom) in this direction

A1	E5	D3	C1	G5	F3	E1	A5	H3	G1	C5	B3
B2	F6	E4	D2	H6	G4	F2	B6	A4	H2	D6	C4
C3	B1	F5	E3	D1	H5	G3	F1	B5	A3	H1	D5
D4	C2	G6	F4	E2	A6	H4	G2	C6	B4	A2	E6

Figure 12-43—EEP multiplexing and bit interleaving pattern

At the receiver side, after demodulation, the received bits shall be deinterleaved. Let y = 0, …, 47 and z = 0, …, 47 be the index at the input and output of the bit deinterleaver, respectively. The bit deinterleaver in the EEP mode shall implement the relation shown in Equation (12-42):

$$z = \mathrm{mod}((6 \times \mathrm{floor}(y/6) - 7 \times \mathrm{mod}(y,6)), 48) \tag{12-42}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

12.4.3.11.3 UEP coding data multiplexer

The UEP coding data multiplexer shall be implemented by DEVs that support UEP HRP modes. It shall not be implemented by DEVs that do not support any UEP HRP modes.

In the UEP coding mode, top 4 encoders (or encoders A, B, C, D) shall use rate 4/7 convolutional codes, and bottom 4 encoders (or encoders E, F, G, H) shall use rate 4/5 convolutional codes. The encoded bits shall be multiplexed and bit interleaved every 96 bits. The total length of 96 bits is divided into two half cycles, with each half cycle multiplexing and interleaving 48 bits, in a slightly different manner.

In the first half cycle, a group multiplexer with group size 7, 7, 7, 7, 5, 5, 5, 5 for all eight encoders, A through G, respectively, shall be used. In this half cycle, A1, A2, A3, A4, A5, A6, A7 are used to label the

7 encoded bits (in increasing order in time) from encoder A, and similarly B1 through B7, C1 through C7, D1 through D7, E1 through E5, F1 through F5, G1 through G5, and H1 through H5 from encoders B, C, D, E, F, G, and H, respectively. At the output of the multiplexer, the 48 encoded bits shall be ordered and numbered as illustrated in Table 12-51.

Table 12-51—Multiplexing scheme in the first half cycle for UEP coding mode

numbering	0	1	2	3	4	5	6	7	8	9	10	11
labeling	A1	A2	A3	A4	A5	A6	A7	B1	B2	B3	B4	B5
numbering	12	13	14	15	16	17	18	19	20	21	22	23
labeling	B6	B7	C1	C2	C3	C4	C5	C6	C7	D1	D2	D3
numbering	24	25	26	27	28	29	30	31	32	33	34	35
labeling	D4	D5	D6	D7	E1	E2	E3	E4	E5	F1	F2	F3
numbering	36	37	38	39	40	41	42	43	44	45	46	47
labeling	F4	F5	G1	G2	G3	G4	G5	H1	H2	H3	H4	H5

The multiplexed 48 bits are then sent to the bit interleaver. Let $x = 0, \ldots, 47$ be the index at the input of the bit interleaver, and $y = 0, \ldots, 47$ be the index at the output of the bit interleaver. The bit interleaver in the first half cycle of the UEP coding mode shall implement the relation shown in Equation (12-43):

$$y = \mathrm{mod}((6 \times \mathrm{floor}(x/6) - 5 \times \mathrm{mod}(x,6)), 48) \tag{12-43}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

The overall read out order of the multiplexer and interleaver in the first half cycle for UEP is illustrated in Figure 12-44.

column-wise readout (from top to bottom) in this direction

A1	E1	C7	B6	G3	E5	D4	A5	H2	F4	C3	B2	I MSB
B1	F1	D7	C6	H5	G2	E4	B5	A4	H1	D3	C2	I LSB
C1	A7	F2	D6	C5	H4	G1	E3	B4	A3	G5	D2	Q MSB
D1	B7	G4	F1	D5	A6	H3	F5	C4	B3	A2	E2	Q LSB

Figure 12-44—UEP coding mode first half cycle multiplexing and bit interleaving pattern

At the receiver side, after demodulation, the received bits shall be deinterleaved. Let $y = 0, \ldots, 47$ and $z = 0, \ldots, 47$ be the index at the input and output of the bit deinterleaver, respectively. The bit deinterleaver in the first half cycle of the UEP coding mode shall implement the relation shown in Equation (12-44):

$$z = \mathrm{mod}((6 \times \mathrm{floor}(y/6) - 7 \times \mathrm{mod}(y,6)), 48) \tag{12-44}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

In the second half cycle, a group multiplexer with group size 7, 7, 7, 7, 5, 5, 5, 5 for all eight encoders shall be used. In this half cycle, A8, A9, A10, A11, A12, A13, A14 are used to label the 7 encoded bits (in increasing order in time) from encoder A, and similarly B8 through B14, C8 through C14, D8 through D14, E6 through E10, F6 through F10, G6 through G10, and H6 through H10 from encoders B, C, D, E, F, G, and

H, respectively. At the output of the multiplexer, the 48 encoded bits shall be ordered and numbered as illustrated in Table 12-52.

Table 12-52—Multiplexing scheme in the second half cycle for UEP coding mode

numbering	0	1	2	3	4	5	6	7	8	9	10	11
labeling	B8	B9	B10	B11	B12	B13	B14	C8	C9	C10	C11	C12
numbering	12	13	14	15	16	17	18	19	20	21	22	23
labeling	C13	C14	D8	D9	D10	D11	D12	D13	D14	A8	A9	A10
numbering	24	25	26	27	28	29	30	31	32	33	34	35
labeling	A11	A12	A13	A14	F6	F7	F8	F9	F10	G6	G7	G8
numbering	36	37	38	39	40	41	42	43	44	45	46	47
labeling	G9	G10	H6	H7	H8	H9	H10	E6	E7	E8	E9	E10

The multiplexed 48 bits are then sent to the bit interleaver. Let $x = 0, \ldots, 47$ be the index at the input of the bit interleaver, and $y = 0, \ldots, 47$ be the index at the output of the bit interleaver. The bit interleaver in the first half cycle of the UEP coding mode shall implement the relation shown in Equation (12-45):

$$y = \mathrm{mod}((6 \times \mathrm{floor}(x/6) - 5 \times \mathrm{mod}(x,6)), 48) \tag{12-45}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

The overall read out order of the multiplexer and interleaver in the second half cycle for UEP is illustrated in Figure 12-45.

column-wise readout (from top to bottom) in this direction

B8	F6	D14	C13	H8	F10	A11	B12	E7	G9	D10	C9
C8	G8	A14	D13	E10	H7	F9	C12	B11	E6	A10	D9
D8	B14	G7	A13	D12	E9	H6	F8	C11	B10	H10	A9
A8	C14	H9	G6	A12	B13	E8	G10	D11	C10	B9	F7

Figure 12-45—UEP coding mode second half cycle multiplexing and bit interleaving pattern

At the receiver side, after demodulation, the received bits shall be deinterleaved. Let $y = 0, \ldots, 47$ and $z = 0, \ldots, 47$ be the index at the input and output of the bit deinterleaver, respectively. The bit deinterleaver in the second half cycle of the UEP coding mode shall implement the relation shown in Equation (12-46):

$$z = \mathrm{mod}((6 \times \mathrm{floor}(y/6) - 7 \times \mathrm{mod}(y,6)), 48) \tag{12-46}$$

where mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

12.4.3.11.4 UEP mapping data multiplexer

The UEP mapping data multiplexer shall be implemented by DEVs that support UEP HRP modes. It shall not be implemented by DEVs that do not support any UEP HRP modes.

In the UEP mapping mode, all eight encoders shall use the same coding rate and the encoded bits shall be multiplexed and bit interleaved every 48 bits.

During the length 48 multiplexing/interleaving cycle, a group multiplexer shall be used with fixed group size 6 for all eight encoders. A1, A2, A3, A4, A5, A6 are used to label the 6 encoded bits (in increasing order in time) from encoder A, and similarly B1 through B6, C1 through C6, D1 through D6, E1 through E6, F1 through F6, G1 through G6, and H1 through H6 from encoders B, C, D, E, F, G, H, respectively. At the output of the multiplexer, the 48 encoded bits shall be ordered and numbered as illustrated in Table 12-53.

Table 12-53—Multiplexing scheme for UEP mapping mode

numbering	0	1	2	3	4	5	6	7	8	9	10	11
labeling	A1	A2	A3	A4	A5	A6	B1	B2	B3	B4	B5	B6
numbering	12	13	14	15	16	17	18	19	20	21	22	23
labeling	C1	C2	C3	C4	C5	C6	D1	D2	D3	D4	D5	D6
numbering	24	25	26	27	28	29	30	31	32	33	34	35
labeling	E1	E2	E3	E4	E5	E6	F1	F2	F3	F4	F5	F6
numbering	36	37	38	39	40	41	42	43	44	45	46	47
labeling	G1	G2	G3	G4	G5	G6	H1	H2	H3	H4	H5	H6

The multiplexed 48 bits are then sent to the bit interleaver. Let $x = 0, \ldots, 47$ be the index at the input of the bit interleaver. At the output of the bit interleaver, 24 bits are mapped to the I branch of the constellation, and the other 24 bits are mapped to the Q branch of the constellation. Let $yI = 0, \ldots, 23$ be the index of those bits mapped to the I branch, and let $yQ = 0, \ldots, 23$ be the index of those bits mapped to the Q branch. The bit interleaver in the UEP mapping mode shall implement the relations shown in Equation (12-47) and Equation (12-48):

$$yI = \mathrm{mod}((6 \times \mathrm{floor}(x/6) - 5 \times \mathrm{mod}(x,6), 24), \text{ if } 0 \le x \le 23 \tag{12-47}$$

$$yQ = \mathrm{mod}((6 \times \mathrm{floor}(x/6) - 5 \times \mathrm{mod}(x,6), 24), \text{ if } 24 \le x \le 47 \tag{12-48}$$

The overall readout order of the multiplexer and bit interleaver for UEP mapping is illustrated in Figure 12-46. Columns are read out from left to right, while the read out order inside each column depends on the constellation used.

column-wise readout (from top to bottom) in this direction always mapped to the I branch

A1	C3	A5	B1	D3	B5	C1	A3	C5	D1	B3	D5
B2	D4	B6	C2	A4	C6	D2	B4	D6	A2	C4	A6

column-wise readout (from top to bottom) in this direction always mapped to the Q branch

E1	G3	E5	F1	H3	F5	G1	E3	G5	H1	F3	H5
F2	H4	F6	G2	E4	G6	H2	F4	H6	E2	G4	E6

Figure 12-46—UEP mapping mode overall multiplexing and bit interleaving pattern

In the QPSK case, the read out order inside each column is (row 1, row 3, row 2, row 4). Take the first column for example, and the readout is ordered as A1, E1, B2, F2, where A1, B2 are mapped to the Q branch, and E1, F2 are mapped to the I branch.

In the 16-QAM case, the read out order inside each column is (row 1, row 2, row 3, row 4). Take the first column for example, and the readout is ordered as A1, B2, E1, F2, where A1, B2 are mapped to the I branch, and E1, F2 are mapped to the Q branch.

At the receiver side, after demodulation, the received bits shall be deinterleaved. Let $yI = 0, \ldots, 23$, $yQ = 0, \ldots, 23$ be the index at the inputs of the deinterleaver I–branch and Q–branch, respectively, and $z = 0, \ldots, 47$ be the index at the output of the bit deinterleaver. The bit deinterleaver in the UEP mapping mode shall implement the relations shown in Equation (12-49) and Equation (12-50):

$$z = \mathrm{mod}((6 \times \mathrm{floor}((yI)/6) - 5 \times \mathrm{mod}(\mathrm{yI},6), 24), \text{ if } 0 \le z \le 23 \quad (12\text{-}49)$$

$$z = \mathrm{mod}((6 \times \mathrm{floor}(yI/6) - 7 \times \mathrm{mod}(\mathrm{yQ},6), 24), \text{ if } 24 \le z \le 47 \quad (12\text{-}50)$$

12.4.3.11.5 HRP header data multiplexer

The data multiplexer and bit interleaver for the HRP header is similar to that for EEP mode described in 12.4.3.11.2. At the output of the multiplexer, the 48 encoded bits shall be ordered and numbered as illustrated in Table 12-54.

Table 12-54—HRP header multiplexing scheme

numbering	0	1	2	3	4	5	6	7	8	9	10	11
labeling	A1	A3	A5	A7	A9	A11	B1	B3	B5	B7	B9	B11
numbering	12	13	14	15	16	17	18	19	20	21	22	23
labeling	C1	C3	C5	C7	C9	C11	D1	D3	D5	D7	D9	D11
numbering	24	25	26	27	28	29	30	31	32	33	34	35
labeling	A2	A4	A6	A8	A10	A12	B2	B4	B6	B8	B10	B12
numbering	36	37	38	39	40	41	42	43	44	45	46	47
labeling	C2	C4	C6	C8	C10	C12	D2	D4	D6	D8	D10	D12

After the data multiplexer, the bit interleaver, tone interleaver, and other operations for the PHY followed that for HRP mode index 0.

12.4.3.12 Bit reversal tone interleaver

All bits shall be interleaved by a block interleaver with a block size corresponding to the size of FFT in a single OFDM symbol, $N_{sc(HR)}$ or $N_{sc(LR)}$. The interleaver is used so that the adjacent data symbols are mapped onto separate subcarriers.

At the transmitter side, the interleaver permutation shall be defined as follows: Let k be the index of the tones (including data tones, pilot tones, DC tones and null tones) before permutation ranging between 0 and $N_{sc(HR)} - 1$ for HRP modes and between 0 and $N_{sc(LR)} - 1$ for LRP modes. Let i be the index of the interleaved tones over the same range (including data tones, pilot tones, DC tones and null tones) after permutation. Let, as shown in Equation (12-51),

$$k = \sum_{j=0}^{L} a_j 2^j \qquad (12\text{-}51)$$

where $L = \log_2(N_{sc(HR)}) - 1$ for HRP modes and $L = \log_2(N_{sc(LR)}) - 1$ for LRP modes, with $[a_L, \ldots, a_0]$ being the binary representation of integer k. Then the binary representation of integer i can be written as $[a_0, \ldots, a_L]$, as shown in Equation (12-52),

$$i = \sum_{j=0}^{L} a_j 2^{L-j} \qquad (12\text{-}52)$$

DC, null, and pilot tones shall be inserted in the bit-reversal position before the tone interleaver. This makes sure that after permutation, the DC, null, and pilot tones appear in the pre-specified positions.

12.4.3.13 Signal constellations

The bits for the LRP mode for each subcarrier shall be mapped to –1 when the bit is zero and shall be mapped to +1 when the bit is one.

For QPSK modulation, serial bits shall be divided into groups of $N_{BPSC} = 2$ bits and converted into complex numbers representing QPSK constellation points. The subcarrier mapping for the HRP mode for QPSK modulation is illustrated in Figure 12-47 where the input bit b_0 is the earliest in time.

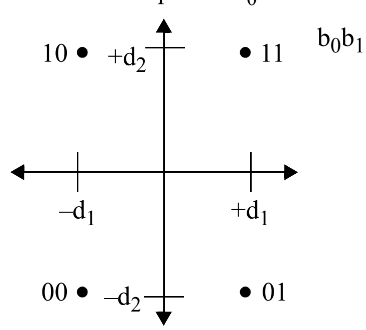

Figure 12-47—HRP QPSK constellation mapping

For 16-QAM modulation, serial bits shall be divided into groups of $N_{BPSC} = 4$ bits and converted into complex numbers representing 16-QAM constellation points. The subcarrier mapping for the HRP mode for 16-QAM modulation is illustrated in Figure 12-48 where the input bit b_0 is the earliest in time.

The constellation parameters for EEP shall be $d_1 = d_2$, while for UEP modulation mapping modes it shall be $d_1 = 1.25 \times d_2$.

The output values, d, are formed by multiplying the resulting $(I + jQ)$ value by a normalization factor, K_{MOD}, calculated as $d = (I + jQ) \times K_{MOD}$. The normalization factor, K_{MOD}, is used to achieve the same average power for all mappings. The value of K_{MOD} depends on the modulation and is defined in Table 12-55.

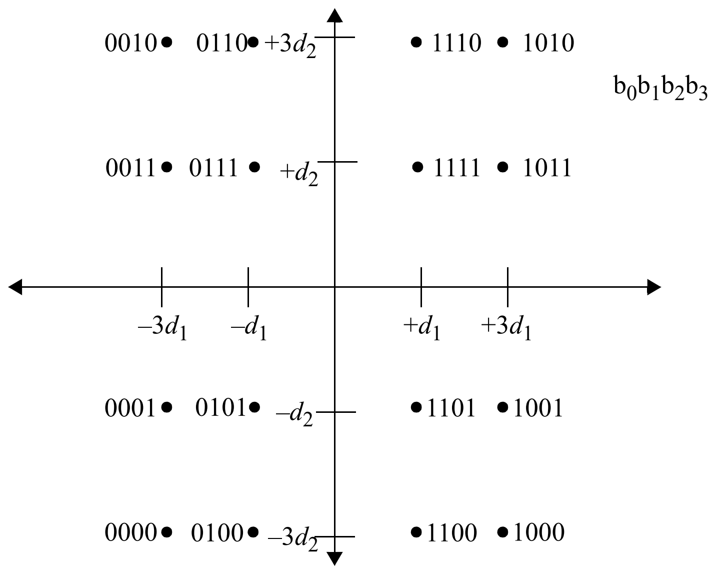

Figure 12-48—HRP 16-QAM constellation mapping

Table 12-55—Normalization factor for PHY modulation formats

Modulation	K_{MOD}
QPSK	$1/(\sqrt{(d_1^2 + d_2^2)})$
16-QAM	$1/(\sqrt{5 \times (d_1^2 + d_2^2)})$

12.4.3.14 AV PHY non-data subcarriers

The pilot subcarriers for the AV PHY mode shall be modulated with BPSK modulation. For HRP frames, the pilot subcarriers shall be set according to the corresponding subcarrier in the first channel estimation symbol, preamble symbol #5. For the LRP frames, the pilot subcarriers shall be set according to the corresponding subcarrier in the LRP training symbol of Table 12-62.

A compliant AV PHY mode receiver may ignore the null and DC subcarriers.

12.4.3.15 AV PHY OFDM modulation

The subcarriers are numbered from $-N_{SC(\{HR,LR\})}/2$ to $N_{SC(\{HR,LR\})}/2 - 1$ where $N_{SC(HR)}$ is defined in Table 12-47 and $N_{SC(LR)}$ is defined in Table 12-48. The frequency offset of a subcarrier is determined by Equation (12-53):

$$f_{offset(sc)}(n) = n \times \Delta f_{SC(\{HR,LR\})} \tag{12-53}$$

where n is the number of the subcarrier, $\Delta f_{SC(HR)}$ is the HRP subcarrier spacing defined in Table 12-47, and $\Delta f_{SC(LR)}$ is the LRP subcarrier spacing defined in Table 12-48.

The HRP subcarriers shall be arranged as indicated in Table 12-56. In Table 12-56, symbol 0 is the first channel estimating symbol, symbol #5 in the preamble, as defined in 12.4.4.2. The function mod(x, y) is the modulo function and is defined in 12.2.3.6.4.

Table 12-56—HRP subcarrier assignment

Subcarrier type	Subcarrier number, *k*
Null	k = (–256:1:–178) and (178:1:255)

Pilots	for *sym* in 0:*Nsymbol* – 1
	{
	k = (–177 + mod(3×*sym*,22)):22:177
	k != (–1,0,1)
	}

DC	k = (–1,0,1)
Data	All remaining

The LRP subcarriers shall be arranged as indicated in Table 12-57.

Table 12-57—LRP subcarrier assignment

Subcarrier type	Subcarrier number
Null	–64 to –19 and 19 to 63
Pilots	–14, –6, 6, 14
DC	–1,0,1
Data	–18 to –15, –13 to –7, –5 to –2, 2 to 5, 7 to 13, 15 to 18

The stream of complex symbols from the modulation mapping is divided into groups of $N_{dsc(\{HR,LR\})}$ complex numbers, numbered from $n = 0$ to $n = N_{dsc(\{HR,LR\})} - 1$ where $n = 0$ corresponds to the first complex number received in time. Each of the complex numbers are mapped sequentially to the subcarriers, skipping the pilots and DC subcarriers. For HRP modes, the complex numbers are mapped beginning with $n = 0$ mapped to $k = -177$ and $n = N_{dsc(HR)} - 1$ mapped to $k = 177$. For LRP modes, the complex numbers are mapped beginning with $n = 0$ mapped to $k = n - 18$ and $n = N_{dsc(LR)} - 1$ mapped to $k = 18$.

12.4.4 AV PHY frame formats

12.4.4.1 General

Unless otherwise specified, all fields, values, bits, or octets defined as reserved shall be set to zero on transmission and shall be ignored on reception.

The high-rate protocol data unit (HRPDU) shall be formatted as illustrated in Figure 12-49.

HRP preamble	HRP header	MAC header	HCS	MAC Frame Body

Figure 12-49—HRPDU frame format

The HRP Header field, MAC Header field, and HCS field, as defined in 12.4.2.4, shall be sent using HRP mode 0 modulation, as defined in Table 12-41.

The MAC Frame Body field, as defined in 6.2, for HRPDUs is made up of one or more subframes. Each subframe may use a different HRP mode. The HRP mode used in a subframe shall be one that is supported by both the source and destination. Higher efficiency is available if subframes with the same HRP mode index are adjacent to each other in the MAC Frame Body field.

The AV PHY supports the use of the beam forming protocol defined in Clause 15. However, because all AV PHY DEVs are required to support an omni-directional mode, AV PHY piconets shall not use extended beacons, as described in 7.8.3, or the superframe support for directional PHYs described in 7.8.6.

HRPDUs shall be used only in directed CTAs and not in CPs, broadcast CTAs, or multicast CTAs.

Two types of LRPDUs are defined based on the transmitter antenna setting: omni LRPDU and directional LRPDU. Omni LRPDUs are used for broadcast/multicast LRP frames, such as beacons and frames sent during contention periods, and may also be used for sending MAC commands and data. The directional LRPDU, with and without payload, shall be used to acknowledge HRP frames.

The first omni LRP frame in a CTA shall be sent using the short omni LRP preamble. Subsequent LRP frames sent in a CTA shall use the long omni LRP preamble. All omni ACK frames shall use the short preamble. The short omni LRP preamble shall be used for frames sent in a CP. The Beacon frame shall use the long omni LRP preamble.

The omni LRPDU shall be formatted as illustrated in Figure 12-50.

Omni LRP preamble (long or short)	Omni LRP header	MAC header	HCS	MAC Frame Body

Figure 12-50—Omni LRPDU frame format

The directional LRPDU shall be formatted as illustrated in Figure 12-51.

Directional LRP preamble	Directional LRP header	Directional LRP payload

Figure 12-51—Directional LRPDU frame format

The MAC Frame Body field for LRPDUs is made up of only one subframe.

12.4.4.2 HRP preamble

The first four symbols of the HRP preamble shall be derived from an 8th-order M-sequence after it is resampled 3/2 times. The M-sequence is generated by the following polynomial:

$$x^8 + x^7 + x^2 + x + 1$$

as shown in Figure 12-52 with a period of 255. The resampled M-sequence has a reference sample rate of $f_{s(HR)}$, as specified in Table 12-47.

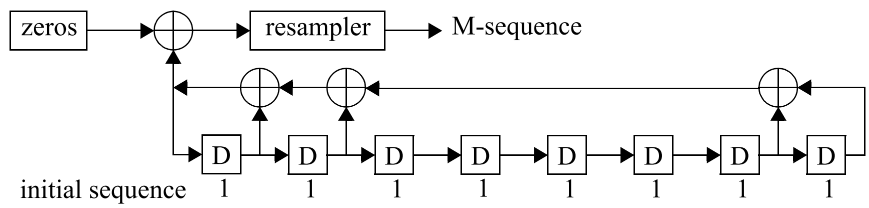

Figure 12-52—Reference diagram for HRP preamble M-sequence

The time-domain preamble shall occupy the time interval corresponding to 4 OFDM symbols by resampling a sequence comprised of five repeated resampled M-sequences, followed by a sign flipped M-sequence, and filled with sufficient number of zeros. The beginning of the first M-sequence shall be aligned with the OFDM boundary. The initial state of the shift registers shall be all ones. The DC components of the time-domain preamble shall be zero, and the first four symbols shall have 3 dB more TX power than the remaining OFDM symbols. The dc component may be canceled by shifting the dc-level of the M-sequence before the resample. Both I and Q branches of the signal shall have the same value.

The spectrum of the time-domain preamble shall conform to the spectrum mask of Figure 12-1. The 3/2 time resampling is a suggestion and any method may be used as long as the transmit PSD mask is met.

The next four symbols of the preamble, 5 to 8, are defined in the frequency domain, as listed in Table 12-63. Before converting to time-domain samples, the frequency domain values are multiplied by a constant whose value is 1 for symbols 5 and 6 and –1 for symbols 7 and 8. The time-domain samples are obtained by taking a 512–point IFFT of the corresponding frequency-domain values. The time-domain samples for symbols 5 to 6 and symbols 7 to 8 symbols are connected with continuous phase, then repeating the last 2×64 samples in the IFFT output for that symbol pair before the first sample, to form a 2×576 sample symbol.

12.4.4.3 HRP header

The HRP header shall be formatted as illustrated in Figure 12-53.

Octets: 1	**3**	**…**	**3**
PHY control	Subframe header 1	…	Subframe header 7

Figure 12-53—HRP header format

The Subframe Header field shall be formatted as illustrated in Figure 12-54. Every HRP header shall contain all seven Subframe Header fields.

Bits: b0–b3	**b4–b23**
HRP mode index	Length

Figure 12-54—Subframe header field format

The Length field contains the length, in octets, of the subframe. If a subframe is not present in an HRP frame, then its Length field shall be set to zero.

The HRP Mode Index field shall be set to the HRP mode that is used for that subframe. The indices are defined in Table 12-41, all other values shall be reserved.

The PHY control field shall be formatted as illustrated in Figure 12-55.

Bits: b0	b1	b2	b3	b4	b5	b6–b7
Reserved	UEP mapping	S0	S1	S2	S3	Reserved

Figure 12-55—PHY control field format

The bits S0, S1, S2, and S3 are the scrambler initialization seeds, as described in 12.4.3.5.

The UEP mapping field shall be set to one if the UEP modes that are used in the subframes use the UEP mapping mode, as defined in 12.4.3.11.4 and 12.4.3.13. It shall be set to zero if the UEP modes in the subframes use the UEP coding mode, as defined in 12.4.3.11.3.

12.4.4.4 LRP preamble sequences

The sequence used in the LRP preamble is created with PN chip sequence with a chip rate of one-half of the LRP reference sampling rate, as defined in Table 12-48. The PN chip sequence is converted to QPSK by using the same binary PN sequence for the in-phase (*I*) and quadrature (*Q*) branches with bipolar mapping, as defined in 12.4.3.13. For the OQPSK mapping, the *Q*-branch signal is delayed by one-half chip, with respect to the *I*-branch. The sequence shall be filtered to comply with the LRP TX mask defined in Figure 12-65.

The Barker-13 chip sequence shall be defined as (–1 –1 –1 –1 –1 1 1 –1 –1 1 –1 1 –1) with the first number listed as the first chip in time.

The 6th-order M-sequences shall be generated using the polynomial $x^6 + x^5 + 1$. The LFSR that generates this 6th-order M-sequence is illustrated in Figure 12-56, where the seed shall be set equal to 0b010111 with the LSB representing the last register.

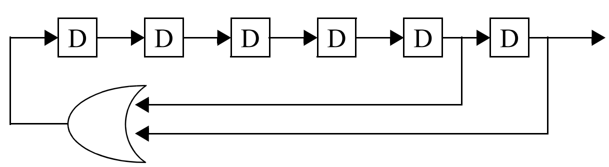

Figure 12-56—LFSFR for generating 6th-order M-sequence

The 12th-order M-sequence shall be generated using the polynomial $x^{12} + x^{11} + x^8 + x^6 + 1$. The LFSR that generates this 12th-order M-sequence is illustrated in Figure 12-57, where the seed shall be set equal to 0xB50 with the LSB representing the last register.

.

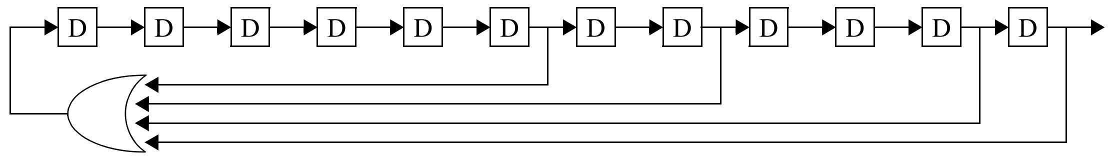

Figure 12-57—LFSFR for generating 12th-order M-sequence

12.4.4.5 Long omni LRP preamble

The long omni LRP preamble is used for frequency synchronization and blind timing recovery. It also allows DEVs receiving the beacon to adjust their transmit frequencies and symbol rates to the accuracy defined in 12.4.5.3 and 12.4.5.4, respectively. The long LRP preamble shall be formatted as illustrated in Figure 12-58.

AGC and signal detect	Coarse FOE and timing recovery	Fine FOE and timing recovery	RX diversity training	AGC2	Channel estimation

Figure 12-58—Long omni LRP preamble format

The AGC and Signal Detect field is composed of 78 symbols, with each symbol being spread by the Barker-13 chip sequence, as defined in 12.4.4.4. The symbol sequence is constructed by repeating the 3-symbol sequence (–1, –1, 1) 26 times. The symbols are modulated using the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The Coarse FOE (frequency offset estimation) and Timing Recovery field is composed of 81 symbols, with each symbols being spread by the Barker-13 chip sequence, as defined in 12.4.4.4. The symbols sequence is constructed by repeating the 9-symbol sequence (–1, 1, –1, 1, 1, 1, –1, –1, 1) 9 times. The symbols are modulated using the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The Fine FOE and Timing Recovery field consists of a 1440 chip PN sequence generated by the 12th-order M-sequence defined in 12.4.4.4 that is modulated using the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The RX Diversity Training field consists of a 2560 chip PN sequence generated by the 6th-order M-sequences defined in 12.4.4.4 that is modulated with the OQPSK sequence defined in 12.4.4.4.

The AGC2 field is an 20 times repetition of a 32-sample long OFDM training symbol equal to the IFFT of the 32-frequency tone vector described by Table 12-61.

The Channel Estimation field consists of 32 156-sample OFDM training symbols, where each is equal to the IFFT of the 128-frequency tone vector defined in Table 12-62 and preceded by a 28-sample cyclic prefix.

The TX antenna direction or phased array pattern changes during the long omni preamble fields every N_{switch} samples, where N_{switch} is defined in Table 12-58.

The first four fields of the long omni LRP preamble shall be transmitted with the same average transmit power, P_1. On the other hand, the last two fields of the long omni LRP preamble shall be transmitted with the same average transmit power used in the transmission of the omni LRP header and payload, P_2, which shall be within 3 dB of the P_1.

Table 12-58—Long omni LRP preamble antenna direction/pattern switching

Field	N_{switch}
AGC and signal detect	78
Coarse FOE	234
Fine FOE	80
RX diversity training	640
AGC2	64
Channel estimation	156

12.4.4.6 Short omni LRP preamble

The short omni LRP preamble is used for limited timing adjustment and shall be formatted as illustrated in Figure 12-59.

AGC 1	AGC 2	Signal Detection And Time Sync	RX Diversity Training	AGC	Channel Estimation

Figure 12-59—Short omni LRP preamble format

The AGC 1 field is 368 chips long and uses the 6th-order M-sequences defined in 12.4.4.4 that is modulated with the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The AGC 2 field is 264 chips long and uses the 6th-order M-sequences defined in 12.4.4.4 that is modulated with the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The Signal Detection and Time Sync field is 720 chips long and uses the 6th-order M-sequences defined in 12.4.4.4 that is modulated with the π/4 rotated QPSK-mapped or OQPSK-mapped sequence defined in 12.4.4.4.

The RX Diversity Training field is defined in 12.4.4.5 and is used to select the best RX antenna or combination of antennas.

The AGC field is defined in 12.4.4.5.

The Channel estimation field is defined in 12.4.4.5.

The TX antenna direction or phased array pattern changes during the short omni preamble fields every N_{switch} samples, where N_{switch} is defined in Table 12-59

The first four fields of the short omni LRP preamble shall be transmitted with the same average transmit power, P_1. On the other hand, the last two fields of the short omni LRP preamble shall be transmitted with the same average transmit power used in the transmission of the omni LRP header and payload, P_2, which shall be within 3 dB of the P_1.

.

Table 12-59—Short omni LRP preamble antenna direction/pattern switching

Field	N_{switch}
AGC and signal detect	32
Coarse FOE	48
Fine FOE	80
RX diversity training	640
AGC2	64
Channel estimation	156

12.4.4.7 Omni LRP header

The Omni LRP header shall use rate 1/3 tail-biting convolutional code, as defined in 12.4.3.9, with eight times repetition coding, defined in 12.4.3.3. The Omni LRP header shall be formatted as illustrated in Figure 12-60.

Bits: b0–b1	b2–b3	b4–b15	b16	b17	b18	b19
LRP mode index	Reserved	Length	S0	S1	S2	S3

Figure 12-60—Omni LRP header format

The fields S0, S1, S2, and S3 are used to set the initial state of the scrambler, as defined in 12.4.3.6.

The Length field contains the length, in octets, of the MAC header, HCS and MAC Frame Body field. This number shall not include the number of stuff bits that are used to create an integer number of symbols for the LRPDU, nor shall it include the number of tail bits.

The LRP Mode Index field indicates the LRP mode, as defined in Table 12-42, that is used for the MAC Header, HCS, and Frame Body field. The MAC Header, HCS and Frame Body field shall be sent using one of the LRP modes that is supported by both the source and destination.

12.4.4.8 Directional LRP preamble

The Directional LRP preamble consists of five repetitions of the 128-sample OFDM training symbol defined in the frequency domain by Table 12-62.

12.4.4.9 Directional LRP header and payload

The Directional LRPDU Header field shall be coded into one OFDM symbol using the rate 1/2 tail-biting convolutional code, as defined in 12.4.3.9, and repeated, as defined in 12.4.3.3. The directional LRPDU uses only one antenna direction or phased array pattern for the preamble, header, and payload. This optimum TX antenna direction or pattern is regularly tracked by the LRP receiver while receiving omni-directional LRPDUs with short omni LRP preambles. This information is fed back to the LRP receiver in the header of the next HRP frame. Omni-directional LRPDUs with long omni LRP preamble shall not be used to track the optimum TX antenna direction or pattern.

The Directional LRP Payload field shall be encoded in either LRP mode index 2 or LRP mode index 3, where the mode indexes are defined in Table 12-42.

The Directional ACK frame uses the directional LRPDU with no Directional LRP Payload field. The Directional ACK header shall be formatted as illustrated in Figure 12-61.

Bits: b0	b1	b2–b6	b7–b14
0	Reserved	ACK group	SCS

Figure 12-61—Directional ACK header format

The first bit of the Directional ACK LRP header is set to zero to identify it as an LRPDU without any payload.

The ACK Group field shall be formatted as illustrated in Figure 12-62.

Bits: b2	b3	b4	b5	b6
ACK group 1	ACK group 2	ACK group 3	ACK group 4	ACK group 5

Figure 12-62—ACK group field format

The ACK Group *n* field shall be set to one if the n^{th} ACK group was correctly received and shall be set to zero otherwise. The ACK Group *n* field shall be set to zero if the subframe was not present in the frame that is being acknowledged.

The Short Check Sequence (SCS) field shall contain the bit-wise inverse of an 8 bit CRC calculated over the first seven bits of the Imm-ACK or short LRP header and is defined by the polynomial given by $x^8 + x^2 + x + 1$.

The SCS field is the same as defined for the Directional ACK frame. The CRC calculation is equivalent to the one defined in ANSI X3.66-1979. Mathematically, the CRC for the SCS is defined by the following procedure:

a) All 7 bits of the header are complemented and become the coefficients of a polynomial, $M(x)$, of degree 6.
b) The remainder, $R(x)$, is calculated from $[(M(x) \times x^8 + x^7)/G(x)]$.
c) $R(x)$ is complemented to become the CRC.

The directional LRP header for an LRPDU with a payload shall be formatted as illustrated in Figure 12-63.

Bits: b0	b1	b2	b3–b6	b7–b14
1	Mode	Reserved	Length-1	SCS

Figure 12-63—Short LRP header format for directional data frames

The first bit of the directional LRP header is set to one to identify the frame as a directional LRPDU with a payload.

The Mode field shall be set to zero if the LRP payload is transmitted with LRP mode index 2, and it shall be set to one if the LRP payload is transmitted with LRP mode index 3, as defined Table 12-42.

The Length-1 field contains one less than the length, in octets, of the Frame Body field. This number shall not include the number of tail bits or stuff bits that are used to create an integer number of symbols for the LRPDU.

The Directional LRP Payload field shall be formatted as illustrated in Figure 12-64.

Octets: 3	**2**
Receive Status field	HCS

Figure 12-64—Directional LRP Payload field format

The HCS field contains the HCS, as defined in 11.2.9, calculated over the Receive Status Field.

The Receive Status field is defined in 12.1.9.3.

The number of encoded bits in the Directional LRP Payload field determines if the standard code with tail bits or the tail-biting code is used on the field. If using the tail-biting code would result in a shorter transmission (one OFDM symbol less than the full convolutional code), then the tail-biting code shall be used; otherwise, the standard code with tail bits shall be used.

12.4.5 AV PHY transmitter requirements

12.4.5.1 TX mask

The transmit PSD mask shall be measured with a 3 MHz resolution bandwidth and a 300 kHz video bandwidth. The transmit PSD mask requirement does not include any carrier leakage.

The HRP transmit PSD mask shall conform to the values illustrated in Figure 12-1.

The LRP transmit PSD mask shall conform to the values illustrated in Figure 12-65 with the following exceptions:

— For LRP channel 1, the spectral mask shall exclude the interval from f_0 + 150.625 MHz to f_0 + 166.625 MHz.
— For LRP channel 2, the spectral mask shall exclude the interval from f_0 – 4 MHz to f_0 + 4 MHz.
— For LRP channel 3, the spectral mask shall exclude the interval from f_0 – 150.625 MHz to f_0 – 166.625 MHz.

The LRP transmit spectral mask shall be measured with a fixed transmit antenna direction or phase pattern, meaning that for omni-directional LRP frames, only one antenna direction or phase pattern is used. This fixed antenna direction or phase pattern may be any of the eight transmit antenna directions or phase patterns described in 12.4.3.3.

12.4.5.2 EVM requirement

The EVM of a compliant transmitter shall be measured and calculated as defined in 12.1.8 and shall be less than or equal to the values given in Table 12-60 for the indicated mode.

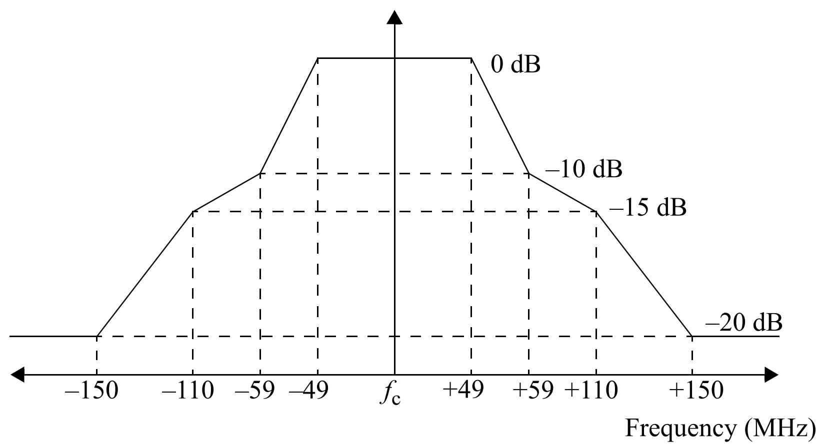

Figure 12-65—LRP transmit PSD mask

Table 12-60—Maximum allowed EVM for AV PHY transmitters

Mode index	Maximum EVM (dB)
LRP 0	–10
LRP 1	–10
LRP 2	–12
LRP 3	–12
HRP 0	–10
HRP 1	–14
HRP 2	–19

In order to measure EVM of omni-directional LRP frames, a fixed antenna direction or phase pattern may be used, where it may be any of the 8 transmit antenna directions or phase patterns described in 12.4.3.3.

12.4.5.3 Symbol timing

The HRP symbol rate shall be $1/T_{S(HR)}$, as defined in Table 12-47, with an accuracy of better than $\pm 20 \times 10^{-6}$.

The LRP symbol rate shall be $1/T_{S(LR)}$, as defined in Table 12-48, with an accuracy of better than $\pm 20 \times 10^{-6}$.

In addition, each DEVs in the piconet shall adjust the symbol rate of its LRP and HRP transmitter so that it is within $\pm 1.5 \times 10^{-6}$ of the symbol timing of the PNC. The MAC parameter *pClockAccuracy* shall be $\pm 1.5 \times 10^{-6}$ for the AV PHY.

12.4.5.4 TX frequency accuracy

The frequency accuracy of the AV PHY shall be less than $\pm 20 \times 10^{-6}$. In addition, all DEVs in a piconet shall adjust their TX frequency so that it is within $\pm 1.5 \times 10^{-6}$ of the frequency of the PNC. The preamble of the beacon is designed to be used by the DEVs to accurately determine the transmit frequency of the PNC. The improved frequency accuracy allows the use of shorter preambles for the LRP and HRP.

12.4.5.5 TX power ramp on and off

The TX power ramp on and off time is implementation dependent and shall be controlled to meet both the timing requirements in 12.4.2.2 and the regulatory restrictions in the appropriate geographical region.

12.4.6 AV PHY Receiver characteristics

12.4.6.1 Error rate criterion

The error rate criterion shall be a BER of less than 1×10^{-7} with data generated by a PN23 sequence as defined by $x^{n+1} = x^{n23} + x^{n18} + 1$, which conforms to CCITT O.151/ITU-T O.151. The error rate is measured at the MAC/PHY interface after any PHY level error correction has taken place.

12.4.6.2 Sensitivity

The receiver sensitivity is determined by measuring, either directly or indirectly, the minimum power input to a single receiver such that the error criterion, as defined in 12.4.6.1, is met.

A compliant LRP receiver shall have a sensitivity that is less than –70 dBm when measured with omni-directional LRP frames with short preamble sent with LRP mode index 1.

A compliant HRP receiver shall have a sensitivity that is less than –50 dBm for HRP mode index 0.

12.4.6.3 Maximum input level

The maximum input level is defined as the highest input power for a single receiver for which the error rate criterion, as defined in 12.4.6.1, is met.

A compliant LRP receiver shall have a maximum input level that is greater than or equal to –30 dBm.

A compliant HRP receiver shall have a maximum input level that is greater than or equal to –24 dBm.

12.4.7 Preambles and training symbols

The 32-sample LRP OFDM training symbol shall have the frequency domain values defined in Table 12-61. There are 32 tones total; all tones not listed have zero value.

Table 12-61—32-sample LRP OFDM training symbol

OFDM tone number	Value
–4	1
–3	1
–2	1
–1	–1
0	0
1	–1

Table 12-61—32-sample LRP OFDM training symbol (*continued*)

OFDM tone number	Value
2	–1
3	1
4	–1

The 128-sample LRP OFDM training symbol shall have the frequency domain values defined in Table 12-62. There are 128 tones total, and the tones not listed have zero value.

Table 12-62—128-sample LRP OFDM training symbol

OFDM tone number	Value	OFDM tone number	Value
–18	–1	1	0
–17	1	2	1
–16	1	3	–1
–15	1	4	1
–14	1	5	–1
–13	1	6	–1
–12	–1	7	1
–11	–1	8	–1
–10	1	9	1
–9	–1	10	1
–8	1	11	1
–7	–1	12	1
–6	–1	13	1
–5	–1	14	1
–4	1	15	–1
–3	–1	16	–1
–2	1	17	1
–1	0	18	1
0	0	—	—

The subcarrier numbers and data values for the HRP preamble for symbols 5–8 is listed in Table 12-63.

Table 12-63—HRP preamble for symbols 5–8

Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value
–178	–1	–89	–1	2	–1	91	–1
–177	–1	–88	–1	3	–1	92	–1
–176	1	–87	–1	4	1	93	–1
–175	1	–86	1	5	–1	94	1
–174	–1	–85	–1	6	–1	95	–1
–173	1	–84	–1	7	–1	96	1
–172	1	–83	1	8	1	97	–1
–171	–1	–82	–1	9	1	98	–1
–170	1	–81	1	10	1	99	1
–169	–1	–80	1	11	–1	100	1
–168	–1	–79	1	12	–1	101	–1
–167	–1	–78	1	13	1	102	1
–166	–1	–77	1	14	–1	103	1
–165	1	–76	–1	15	1	104	1
–164	1	–75	–1	16	–1	105	1
–163	–1	–74	1	17	1	106	–1
–162	–1	–73	1	18	–1	107	1
–161	–1	–72	1	19	1	108	1
–160	–1	–71	–1	20	1	109	–1
–159	1	–70	1	21	–1	110	1
–158	–1	–69	–1	22	–1	111	–1
–157	1	–68	–1	23	1	112	1
–156	1	–67	1	24	–1	113	–1
–155	–1	–66	–1	25	–1	114	1
–154	–1	–65	1	26	1	115	1
–153	–1	–64	1	27	1	116	1
–152	1	–63	1	28	–1	117	1
–151	1	–62	–1	29	–1	118	1
–150	–1	–61	–1	30	–1	119	1
–149	1	–60	–1	31	–1	120	–1
–148	–1	–59	–1	32	–1	121	–1
–147	–1	–58	1	33	1	122	–1
–146	1	–57	1	34	1	123	–1
–145	–1	–56	–1	35	–1	124	–1

Table 12-63—HRP preamble for symbols 5–8 (*continued*)

Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value
–144	–1	–55	–1	36	–1	125	1
–143	1	–54	–1	37	1	126	–1
–142	1	–53	–1	38	1	127	–1
–141	1	–52	1	39	–1	128	1
–140	1	–51	1	40	–1	129	1
–139	–1	–50	–1	41	–1	130	1
–138	–1	–49	–1	42	1	131	1
–137	1	–48	–1	43	1	132	1
–136	–1	–47	1	44	1	133	–1
–135	1	–46	1	45	–1	134	1
–134	–1	–45	1	46	1	135	1
–133	–1	–44	–1	47	1	136	–1
–132	1	–43	1	48	1	137	1
–131	–1	–42	–1	49	1	138	–1
–130	–1	–41	1	50	1	139	1
–129	–1	–40	–1	51	–1	140	1
–128	–1	–39	1	52	1	141	–1
–127	–1	–38	1	53	–1	142	1
–126	1	–37	1	54	1	143	–1
–125	1	–36	–1	55	1	144	1
–124	–1	–35	–1	56	–1	145	1
–123	1	–34	1	57	–1	146	–1
–122	1	–33	–1	58	1	147	–1
–121	1	–32	1	59	1	148	1
–120	1	–31	1	60	–1	149	–1
–119	–1	–30	–1	61	–1	150	1
–118	1	–29	1	62	1	151	–1
–117	1	–28	1	63	–1	152	1
–116	1	–27	1	64	–1	153	1
–115	–1	–26	1	65	–1	154	1
–114	–1	–25	1	66	1	155	–1
–113	–1	–24	1	67	–1	156	1
–112	1	–23	–1	68	–1	157	1
–111	1	–22	1	69	1	158	–1

Table 12-63—HRP preamble for symbols 5–8 (*continued*)

Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value	Subcarrier	Data value
–110	1	–21	–1	70	1	159	1
–109	1	–20	1	71	1	160	–1
–108	1	–19	1	72	1	161	1
–107	–1	–18	1	73	1	162	1
–106	1	–17	–1	74	1	163	1
–105	–1	–16	–1	75	–1	164	–1
–104	–1	–15	1	76	–1	165	1
–103	–1	–14	1	77	1	166	–1
–102	1	–13	1	78	–1	167	1
–101	1	–12	1	79	1	168	1
–100	1	–11	–1	80	1	169	1
–99	1	–10	–1	81	1	170	–1
–98	1	–9	1	82	–1	171	–1
–97	–1	–8	–1	83	–1	172	1
–96	1	–7	–1	84	–1	173	–1
–95	1	–6	–1	85	1	174	1
–94	1	–5	1	86	–1	175	–1
–93	1	–4	–1	87	–1	176	1
–92	1	–3	–1	88	–1	177	–1
–91	–1	–2	–1	89	1	178	–1
–90	1			90	–1	—	—

13. PHY specification for HRCP

13.1 General requirements

13.1.1 Overview

A compliant HRCP PHY shall implement at least one of the following PHY modes:

a) High-rate single carrier mode PHY (HR-SC PHY), as defined in 13.2.
b) On-off keying mode PHY (OOK PHY), as defined in 13.3.

Unless otherwise stated, in all figures in this clause the ordering of the octets and bits as they are presented to the PHY for modulation is the same as defined in 6.1.

13.1.2 Regulatory Information

The HRCP PHY operating frequency is the same as that described in 12.1.2.

13.1.3 RF power measurements

Unless otherwise stated, all RF power measurements for the purpose of this standard, either transmit or receive, shall be made based on EIRP and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation. The gain of the antenna is the maximum estimated gain by the manufacturer.

13.1.4 Unwanted emissions

Conformant implementations shall comply with the in-band and out-of-band emissions for all operational modes as set by the applicable regulatory bodies.

13.1.5 RF channelization

The HRCP PHY uses the channels defined in Table 13-1. CHNL_ID from 1 to 4 are the same as those in Table 12-3.

Table 13-1—HRCP PHY channelization

CHNL_ID	Start frequency[a]	Center frequency	Stop frequency[a]
1	57.240 GHz	58.320 GHz	59.400 GHz
2	59.400 GHz	60.480 GHz	61.560 GHz
3	61.560 GHz	62.640 GHz	63.720 GHz
4	63.720 GHz	64.800 GHz	65.880 GHz
5	65.880 GHz	66.960 GHz	68.040 GHz
6	68.040 GHz	69.120 GHz	70.200 GHz
7	57.240 GHz	59.400 GHz	61.560 GHz
8	59.400 GHz	61.560 GHz	63.720 GHz
9	61.560 GHz	63.720 GHz	65.880 GHz

Table 13-1—HRCP PHY channelization (*continued*)

CHNL_ID	Start frequency[a]	Center frequency	Stop frequency[a]
10	63.720 GHz	65.880 GHz	68.040 GHz
11	65.880 GHz	68.040 GHz	70.200 GHz
12	57.240 GHz	60.480 GHz	63.720 GHz
13	59.400 GHz	62.640 GHz	65.880 GHz
14	63.720 GHz	66.960 GHz	70.200 GHz
15	57.240 GHz	61.560 GHz	65.880 GHz
16	59.400 GHz	63.720 GHz	68.040 GHz

[a]The start and stop frequencies are nominal values. The frequency spectrum of the signal needs to conform to the transmit PSD mask for the PHY mode as well as any regulatory requirement.

The channel whose CHNL_ID is 2 shall be defined as the default channel. CHNL_ID 9, 10, 11, and 14 are only used for channel aggregation.

Channel aggregation is defined in 13.2.9.2.

13.1.6 Transmit PSD mask

The transmitted spectrum for HRCP PHY shall adhere to the transmit spectrum masks for a single channel shown in Figure 13-1 and for channel bonding shown in Figure 13-2.

The transmit spectral mask for HRCP PHY that supports channel bonding shall conform to the values indicated in Table 13-2 and Table 13-3.

For the transmit mask measurements, the resolution bandwidth is set to 3 MHz and video bandwidth to 300 kHz. During OOK modulation, transmitters shall meet the same PSD mask, except for the single line spectra of 40 dB above the 0 dBr line in Figure 13-1 and Figure 13-2 within the frequency band of [–6 MHz,+6 MHz] from the carrier frequency.

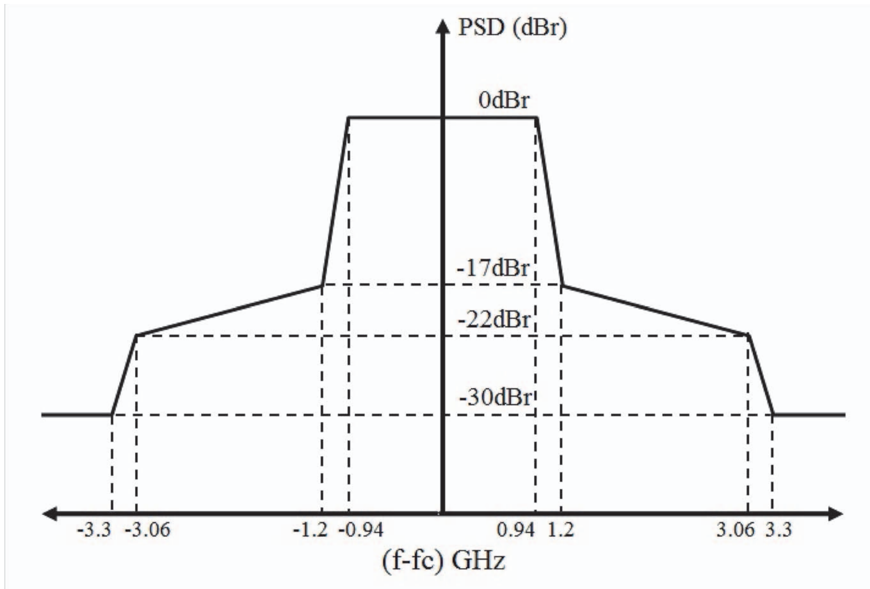

Figure 13-1—Transmit spectral mask for a single channel

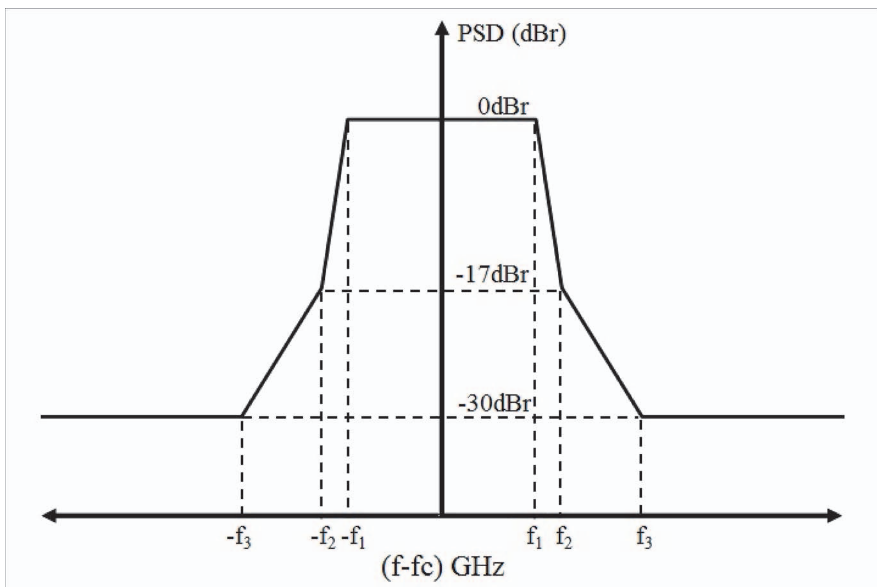

Figure 13-2—Transmit spectral mask for channel bonding

Table 13-2—Transmit spectral mask limit for channel bonding

Frequency	Relative limit (dBr)
$\lvert f - f_c \rvert < f_1$	0
$f_2 \le \lvert f - f_c \rvert < f_1$	$-17(\lvert f - f_c \rvert - f_1)/(f_2 - f_1)$
$f_3 \le \lvert f - f_c \rvert < f_2$	$-17 - 13(\lvert f - f_c \rvert - f_2)/(f_3 - f_2)$
$f_3 \le \lvert f - f_c \rvert$	-30

Table 13-3—Transmit spectral mask parameters for channel bonding

Channel bonding	f_1 (GHz)	f_2 (GHz)	f_3 (GHz)
Two-bonded channel transmission	1.880	2.400	4.000
Three-bonded channel transmission	2.820	3.600	6.000
Four-bonded channel transmission	3.760	4.800	8.000

13.1.7 Error vector magnitude calculation

The error vector magnitude (EVM) shall be measured and calculated using the method defined in 12.1.8.1.

13.1.8 HRCP-PHY management

13.1.8.1 Supported MCSs

The Supported data rates field in the DEV capabilities field is defined as described in 6.4.13.

13.1.8.2 HRCP-PHY PIB

The PHY dependent PIB values for the HRCP PHY are given in Table 13-4 and Table 13-5. The PHY PIB characteristics group, Table 13-4, contains information that is common to most implementations. The PHY PIB implementation group, Table 13-5, contains information that is more characteristic of a particular PHY implementation than of the PHY as a whole.

Table 13-4—PHY PIB characteristics group parameters

Managed object	Octets	Definition	Access
phyType	1	0x02 = HRCP PHY	Read/ Write
phyMode	1	bit 1 = HRCP-SC PHY bit 2 = HRCP-OOK PHY bit 3–8 = Reserved A bit is set to one if the associated PHY is supported, and is set to zero otherwise.	Read/ Write
phyRegDomainsSupported	Variable	One octet for each regulatory domain supported, as defined for phyCurrentRegDomain.	Read/ Write
phyCurrentRegDomain	1	0x00 = European Telecommunications Standards Institute (ETSI) 0x01 = Federal Communications Commission (FCC) 0x02 = Industry Canada (IC) 0x03 = Association of Radio Industries and Businesses (ARIB)	Read/ Write
phyDataRateVector	Variable	One octet for each supported MCS. The MSB indicates the HRCP PHY mode, as in phyMode, and the last seven LSBs contain the MCS supported for that mode using the encoding for that PHY mode.	Read/ Write
phyNumChannelsSupported	1	Indicates the number of channels supported, as defined in 13.1.5. The range is 1 to 16 and the value is implementation dependent.	Read/ Write
phyCurrentChannel	1	Indicates the channel that is currently being used, as defined in 13.1.5.	Read/ Write
phyFrameLengthMax	2	pMaxFrameBodySize.	Read/ Write

Table 13-5—PHY PIB implementation group parameters

Managed object	Octets	Definition	Access
phyDiversitySupported	1	Numeric entry that indicates the number of antennas that are available.	Read/Write
phyMaxTXPower	1	The maximum TX power that the DEV is capable of transmitting, as defined in 6.4.11, and the value is implementation dependent.	Read/Write
phyTXPowerStepSize	1	The step size for power control supported by the DEV, as defined in 6.4.11, and the value is implementation dependent.	Read/Write
phyNumPMLevels	1	Number of power management levels supported. The range is 1 to 8 and the value is implementation dependent.	Read/Write
phyPMLevelReturn	Variable	Table of vectors with number of entries given by phyNumPMLevels. Each vector is the time required to change between power saving states of the PHY. Vector number 0 is the time required to change the PHY from the off state to a state where it is ready to receive commands. Other values are implementation dependent.	Read/Write

13.2 HRCP-SC PHY

13.2.1 General

The HRCP-SC PHY is designed for extremely high PHY-SAP payload bit rates between 2 Gb/s and 13 Gb/s using a single channel with a band width of 2.16 GHz and the maximum 100 Gb/s using MIMO, channel aggregation, and channel bonding.

The HRCP-SC PHY supports π/2 BPSK, π/2 QPSK, 16-QAM, 64-QAM, and 256-QAM modulations. The modulation of π/2 BPSK is only used for preamble and header sequences, and other modulations are used for payload. The modulations of π/2 BPSK and π/2 QPSK are mandatory for HRCP-SC PHY and other modulations are optional.

FEC includes two LDPC codes with a code rate of 14/15 and a code rate of 11/15. These two LDPC codes are mandatory for HRCP-SC PHY.

The HRCP-SC PHY also supports channel aggregation, channel bonding and MIMO. Channel aggregation, channel bonding, and MIMO are optional.

13.2.2 Channelization of HRCP-SC PHY

The RF channels are defined in Table 13-1. A compliant implementation shall support at least one channel from the channels allocated for operation by its corresponding regulatory body.

CHNL_IDs from 1 to 6 are assigned for single channel operation. The remaining CHNL_IDs are assigned for bonded channel and aggregated single channel. Channel aggregation uses a combination of multiple channels as defined in 6.4.13. Chip rates used for bonded channels are described in 13.2.3.3 and data assignments for aggregated channels are described in 13.2.9.2.

The phyCurrentChannel is the CHNL_ID of the current channel.

13.2.3 Modulation and coding

13.2.3.1 MCS-dependent parameters

The MCS-dependent parameters shall be set according to Table 13-6. The data rates in the table are approximate values for single channel operation.

The chip rate for all HRCP-SC PHY MCS is given in Table 13-8.

Table 13-6—MCS-dependent parameters

MCS identifier	Modulation	FEC rate	Data rate (Gb/s) w/o PW	Data rate (Gb/s) w/ PW
	π/2 QPSK	11/15	2.5813	2.2587
1	π/2 QPSK	14/15	3.2853	2.8747
2	16-QAM	11/15	5.1627	4.5173
3	16-QAM	14/15	6.5707	5.7493
4	64-QAM	11/15	7.7440	6.7760
5	64-QAM	14/15	9.8560	8.6240
6	256-QAM	14/15	13.1413	11.4987

The block length, where a block is defined in 13.2.4.5.1, for HRCP-SC PHY shall be 64 chips. The pilot word (PW) length, where a PW is defined in 13.2.4.5.1, for HRCP-SC PHY shall be 0 chips or 8 chips. The PW length of 0 chips is mandatory and that of 8 chips is optional.

13.2.3.2 Header rate-dependent parameters

The header rate-dependent parameters shall be set according to Table 13-7 for single channel operation. The header rate is proportional to the number of channels bonded. The headers use an extended Hamming (EH) code, as defined in 13.2.4.3.4.

Table 13-7—Header rate-dependent parameters

Header rate (Mb/s)	Modulation scheme	Spreading factor, L_{SF}	FEC	PW length (chips), L_{PW}	Coded bits per block, N_{CBPB}	Number of occupied blocks, N_{block_hdr}	Number of stuff bits, L_{STUFF}
168	π/2 BPSK	4	EH	8	14	19	2

13.2.3.3 Timing-related parameters

Table 13-8 lists the general timing parameters associated with the SC PHY.

Table 13-8—Timing-related parameters

Parameter	Description	Value		Unit	Formula
N_B	Number of bonded channels	Variable integer from 1 to 4			
R_c	Chip rate	$1760 \times N_B$		Mchip/s	
T_C	Chip duration	~0.5682 / N_B		ns	$1/R_C$
L_{block}	Block length	64		chips	
L_{PW}	Pilot word length	0	8	chips	
T_{PW}	Pilot word duration	0	4.5/ N_B	ns	
L_{DC}	Length of data chips per block	64	56	chips	
T_{block}	Block duration	~36.364 / N_B		ns	$L_{block} \times T_c$
R_{block}	Block rate	$27.5 \times N_B$		MHz	$1/ T_{block}$

13.2.3.4 Frame-related parameters

The frame parameters associated with the PHY are listed in Table 13-9 where CEIL is the ceiling function, which returns the smallest integer value greater than or equal to its argument. The maximum frame duration occurs when the number of octets in the PHY Payload field is 2099200.

Table 13-9—Frame-related parameters

Parameter	Description	Value
N_{SYNC}	Number of code repetitions in the SYNC sequence	14 or 28
T_{SYNC}	Duration of the SYNC sequence	~1.019/N_B µs or ~2.036/N_B µs
N_{SFD}	Number of code repetitions in SFD	1
T_{SFD}	Duration of the SFD	~0.073/N_B µs
N_{CES}	Number of code repetitions in the CES	11
T_{CES}	Duration of the CES	~0.800/N_B µs
N_{pre}	Number of code repetitions in the PHY preamble	26 or 40
T_{pre}	Duration of the PHY preamble	~1.891/N_B µs or ~2.909/N_B µs
L_{hdr}	Length of the encoded frame header in octets	33
N_{block_hdr}	Number of blocks in the base frame header	CEIL[$L_{hdr} \times 8 \times L_{SF} / (L_{block} - L_{PW})$] = 19
T_{hdr}	Duration of the frame header	$N_{block\_hdr} \times T_{block\,=}$ ~ 0.691/N_B µs
$L_{payload}$	Length of frame payload in octets	Variable

Table 13-9—Frame-related parameters (*continued*)

Parameter	Description	Value
L_{hds}	Length of the MAC subheader in octets	4
$N_{subframe}$	Number of subframes	Variable between 1 and 256
L_{FCS}	Length of FCS in octets	4
L_{MFB}	Length of MAC frame body in octets	$L_{payload} + (L_{hds} + L_{FCS}) \times N_{subframe}$
N_{PPRE}	Number of code repetitions in the PPRE	26
T_{PPRE}	Duration of the PPRE	~1.891/N_B µs ~2.909/N_B µs
N_{block_PPRE}	Number of blocks between PPRE	Variable between 1024 and 4096
N_{CBPC}	Number of coded bits per chip in the MAC frame body	2, 4, 6, and 8 for QPSK, 16QAM, 64QAM, and 256QAM, respectively
N_{PPRE_frame}	Number of PPREs per frame	CEIL[N_{block_MFB} / N_{block_PPRE}] − 1
$T_{PPRE\_interval}$	Interval of PPRE insertion	$T_{block} \times N_{block\_PPRE} + T_{PW}$
N_{CBPB}	Number of coded bits per block in the MAC frame body	$(L_{block} - L_{PW}) \times N_{CBPC}$
N_{block_MFB}	Number of blocks in the MAC frame body	CEIL[(L_{MFB} × 8) / (R_{FEC} × N_{CBPB})] (R_{FEC}: FEC Rate)
T_{MFB}	Duration of the MAC and PHY frame body	$N_{block\_MFB} \times T_{block}$
$T_{datafield}$	Duration of the PHY data field	$T_{MFB} + (N_{PPRE\_frame} + 1) \times T_{PW} + N_{PPRE\_frame} \times T_{PPRE}$
T_{frame}	Duration of the frame	$T_{pre} + T_{hdr} + T_{datafield}$

13.2.3.5 Modulation

After channel encoding and spreading, the bits shall be inserted into the constellation mapper.

The constellations of π/2 BPSK and π/2 QPSK used for the HRCP-SC PHY are defined as points in Figure 12-10 (a) and (b) with a counter-clockwise phase offset of π/4, respectively, in 12.2.3.5. The constellations of 16-QAM and 64-QAM used for the HRCP-SC PHY are the same as illustrated in Figure 12-29 in 12.3.3.6 with the parameter d set to 1.

The normalization factors for π/2 QPSK, 16-QAM, and 64-QAM constellations are $/1\sqrt{2}$, $/1\sqrt{10}$, and $1\sqrt{42}$, respectively. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor can be used, as long as the DEV conforms to the modulation accuracy requirements described in 13.2.5.1.

The constellation map of 256-QAM used for the HRCP-SC PHY is illustrated in Figure 13-3. The serial bit stream shall be divided into groups of eight bits with input bit d_1 being the earliest in the stream.

The normalization factor for 256-QAM constellation is $1/\sqrt{170}$. An approximate value of the normalization factor may be used, as long as the DEV conforms to the modulation accuracy requirements.

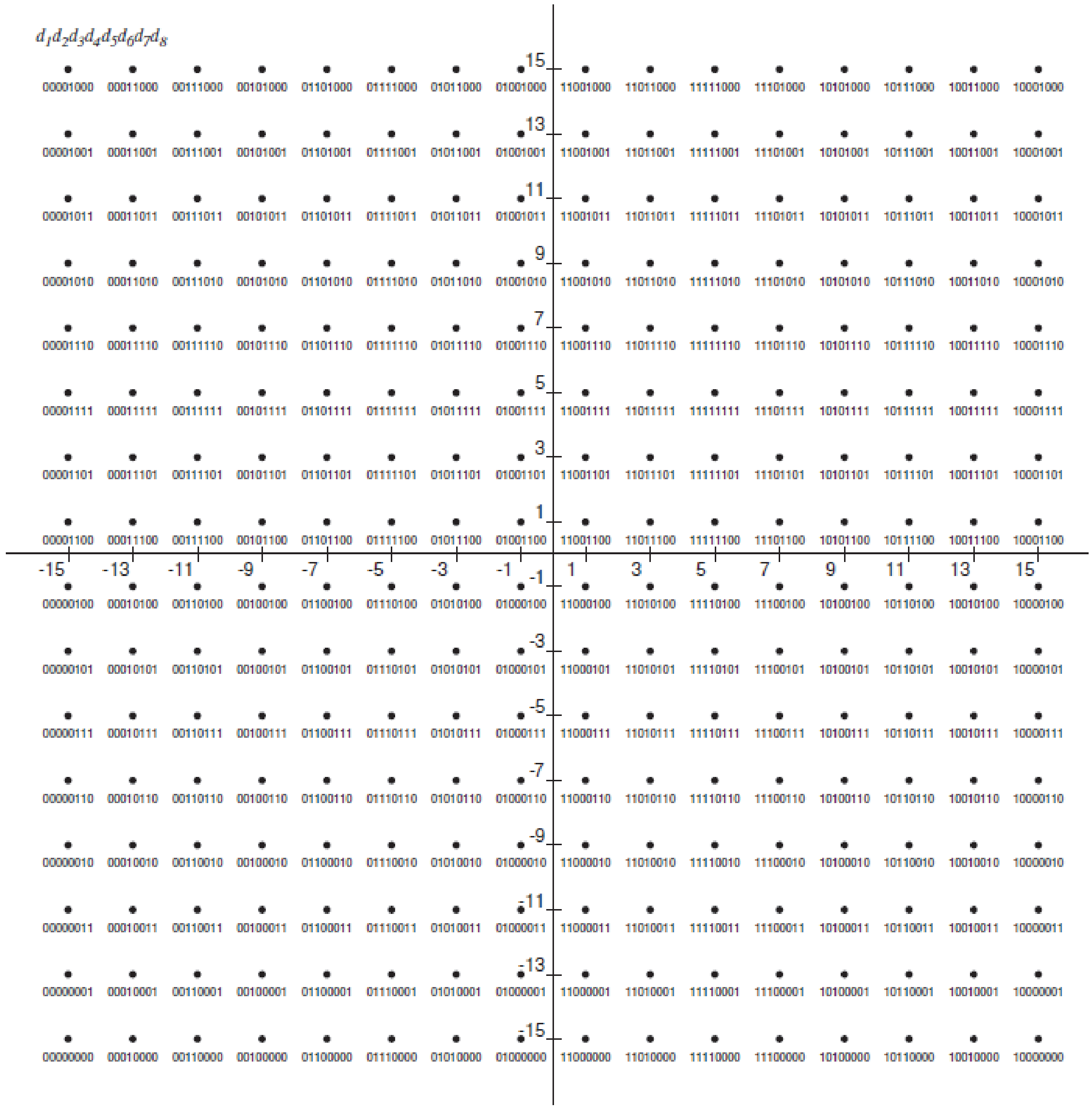

Figure 13-3—Constellation map of 256 QAM

13.2.3.6 Forward Error Correction

The forward error correction (FEC) schemes are specified in this subclause. Supporting the following two rate-compatible LDPC codes, i.e., a rate-14/15 LDPC(1440,1344) code and a rate-11/15 LDPC(1440,1056) code, are mandatory for HRCP-SC PHY, where the rate-14/15 LDPC(1440,1344) code has been defined in 12.2.3.6.4.

The LDPC codes are systematic, i.e. the LDPC encoder encodes an information block of length k, $\mathbf{i} = (i_0, i_1, \ldots, i_{k-1})$, into a codeword $\mathbf{c}$ of length 1440, $\mathbf{c} = (i_0, i_1, \ldots, i_{k-1}, p_0, p_1, \ldots, p_{1440-k-1})$, by adding $(1440 - k)$ parity bits $(p_0, p_1, \ldots, p_{1440-k-1})$ obtained so that $\mathbf{Hc}^T = 0$, where $\mathbf{H}$ is an $(1440 - k) \times 1440$ parity-check matrix and T denotes transverse operation. Denote the parity check matrix as $\mathbf{H} = (h_{i,j})$, where $h_{i,j}$ consists of $\{0,1\}$, $0 \le i < (1440 - k)$ and $0 \le j < 1440$.

Table 13-10 lists the parameters of the LDPC codes with a codeword length of 1440, e.g., supported code rates, information-block lengths k and parity lengths, and the matrix elements whose values are "1" in the first 15 columns of parity check matrix **H** with 1440 columns for the LDPC codes. Matrix elements whose values are "1" in the first 15 columns of parity check matrix **H** for the rate-14/15 LDPC(1440,1344) code has been defined in Table 12-17.

Table 13-10—Parameters of the rate-11/15 LDPC code with a codeword length of 1440

Parameter	Value
code rate	11/15
information-block length, k (bits)	1056
parity length (bits)	384
matrix elements whose values are "1" in the first 15 columns of parity check matrix **H**	$h_{96,0}$ $h_{193,0}$ $h_{4,0}$
	$h_{34,1}$ $h_{320,1}$ $h_{135,1}$
	$h_{352,2}$ $h_{70,2}$ $h_{270,2}$
	$h_{104,3}$ $h_{306,3}$ $h_{287,3}$
	$h_{31,4}$ $h_{234,4}$ $h_{150,4}$
	$h_{159,5}$ $h_{364,5}$ $h_{91,5}$
	$h_{302,6}$ $h_{45,6}$ $h_{286,6}$
	$h_{126,7}$ $h_{239,7}$ $h_{371,7}$
	$h_{17,8}$ $h_{158,8}$ $h_{272,8}$
	$h_{28,9}$ $h_{336,9}$ $h_{178,9}$
	$h_{214,10}$ $h_{60,10}$ $h_{369,10}$
	$h_{219,11}$ $h_{145,11}$ $h_{372,11}$
	$h_{7,12}$ $h_{245,12}$ $h_{173,12}$
	$h_{19,13}$ $h_{140,13}$ $h_{373,13}$
	$h_{6,14}$ $h_{238,14}$ $h_{363,14}$

For $15 \leq j$, the matrix element can be obtained by using the Equation (13-1):

$$h_{i,j} = h_{96*\text{floor}(i/96) + \text{mod}(i + \text{floor}(j/15), 96), \text{mod}(j, 15)} \tag{13-1}$$

where

mod(x, y) is the modulo function and is defined as ($x - n \times y$)
n is the nearest integer less than or equal to x/y

Each LDPC code is a quasi-cyclic code such that every cyclic shift of a codeword by 15 symbols yields another codeword.

For shortened LDPC operation, the k–l zero elements are appended to the incoming l message bits as follows: $r_i = 0$ for $i = l, l+1, \ldots, k-1$. The message order is r_{k-1} as the first bit of the message with r_0 as the last bit of the message. These inserted zero elements are not transmitted.

The last LDPC codeword in a frame shall be shortened when l for the last codeword is less than k, and l of the other LDPC codewords shall be k.

13.2.3.7 Stuff bits

Stuff bits shall be added to the end of the encoded MAC frame body if the number of the encoded data bits is not an integer multiple of the length of the data portion in the block. The calculation of stuff bits is as follows.

In the encoded MAC frame body, the number of FEC codewords, N_{FEC}, is given by Equation (13-2):

$$N_{FEC} = \text{CEIL}[(L_{MFB} \times 8)/(1440 \times R_{FEC})] \tag{13-2}$$

where

L_{MFB} is the length of the MAC frame body in octets
R_{FEC} is the FEC rate

The encoded MAC frame body shall be concatenated with stuff bits of length L_{STUFF} so that the resulting MAC frame body is aligned on the block symbol boundary. The stuff bits shall be set to zero and then scrambled using the continuation of the scrambler sequence that scrambled the MAC frame body in13.2.3.9. The length of bits in the encoded MAC frame body, L_{ebits} is given by Equation (13-3):

$$L_{ebits} = 8 \times L_{MFB} + N_{FEC} \times (1 - R_{FEC}) \times 1440 \tag{13-3}$$

The number of blocks in the encoded MAC frame body, $N_{subblock\text{-}encMFB}$, and the length of stuff bits, L_{STUFF}, are given byEquation (13-4) and Equation (13-5):

$$N_{subblock\text{–}encMFB} = \text{CEIL}(L_{ebits}/N_{CBPB}) \tag{13-4}$$

$$L_{STUFF} = N_{subblock\text{–}encMFB} \times N_{CBPB} - L_{ebits} \tag{13-5}$$

where

N_{CBPB} is the number of coded bits per subblock as given in Table 13-11 for each MCS.

Table 13-11—Rate dependent bits per block

MCS identifier	Coded bits per block, N_{CBPB} (pilot word length = 0)	Coded bits per block, N_{CBPB} (pilot word length = 8)
0	128	112
1	128	112
2	256	224
MCS identifier	**Coded bits per block, N_{CBPB} (pilot word length = 0)**	**Coded bits per block, N_{CBPB} (pilot word length = 8)**
3	256	224
4	384	336
5	384	336
6	512	448

For the stuff bits in the frame headers, the values are given in Table 13-7. The bit pattern of the header stuff bits shall be set to 01, where 0 is transmitted first.

13.2.3.8 Code spreading

Table 13-12 is a spreading table for a frame header. The most significant bit of the output shall be transmitted first in Table 13-12.

Table 13-12—Spreading table

Input	Output
0	1010
1	0101

13.2.3.9 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as illustrated in Figure 12-14 with $L_{SF} = 1$.

13.2.4 HRCP-SC PHY frame format

13.2.4.1 General

The HRCP-SC PHY frame shall be formatted as illustrated in Figure 12-18.

The Frame Header field for the PHY frame shall be formatted as illustrated in Figure 13-4.

<table>
<tr><td colspan="3">Header-FEC encoded</td><td rowspan="3">Stuff bits</td></tr>
<tr><td rowspan="2">PHY Header</td><td colspan="2">Scrambled</td></tr>
<tr><td>MAC Header</td><td>HCS</td></tr>
</table>

Figure 13-4—Frame header format

The PHY preamble is described in 13.2.4.2. The MAC header is defined in 6.2. The PHY header is defined in 13.2.4.3.2, and the HCS is defined in 13.2.4.3.3. The header FEC is defined in 13.2.4.3.4. The PHY Payload field consisting of the MAC frame body, the pilot preamble (PPRE) and stuff bits, is described in 13.2.4.4. The PPRE is described in 13.2.4.5.2. The stuff bits are described in 13.2.3.7.

13.2.4.2 PHY preamble

13.2.4.2.1 Overview

A PHY preamble shall be added prior to the frame header to aid receiver algorithms related to auto-gain control (AGC) setting, frame detection, timing acquisition, frequency offset estimation, frame synchronization, and channel estimation.

The PHY preamble, i.e., PHY-long preamble and PHY-short preamble, shall be transmitted at the chip rate 1760 MHz.

A PHY-long preamble shall be used during PSP, and a PHY-short preamble shall be used during PAP.

Figure 13-5 shows the structure of the PHY-long or PHY-short preambles.

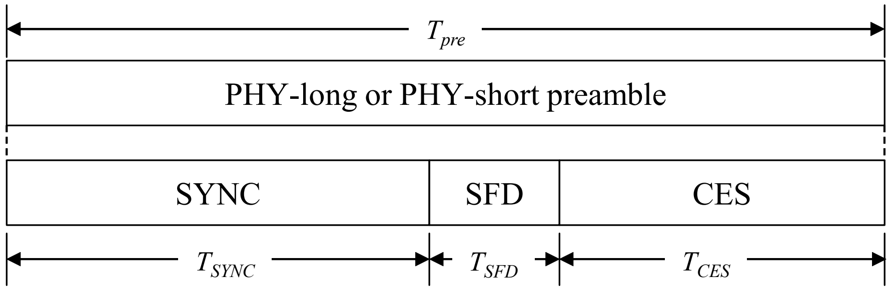

Figure 13-5—HRCP-SC PHY preamble structure

For PHY preamble, T_{SFD} is 0.07 µs and T_{CES} is 0.80 µs. For PHY-long preamble, T_{SYNC} is 2.01 µs and T_{PRE} is 2.91 µs. For PHY-short preamble, T_{SYNC} is 1.02 µs and T_{PRE} is 1.89 µs.

13.2.4.2.2 Frame synchronization (SYNC)

The SYNC field is used for frame detection and uses a repetition of codes for a higher of robustness. The SYNC field for PHY-long preamble shall consist of 28 code repetitions of $\mathbf{a}_{128}$. The SYNC field for PHY-short preamble shall consist of 14 code repetitions of $\mathbf{a}_{128}$.

Table 13-13 shows the sequence $\mathbf{a}_{128}$ used for the SYNC and a sequence $\mathbf{b}_{128}$ used for the CES defined in 13.2.4.2.4. Note that in each hexadecimal-equivalent 4-binary-digit group, the leftmost bit shall be the MSB, and the rightmost bit, the LSB. For example, 3 is denoted as 0011. The order of the octets and bits over the air is the same as defined in 6.1.

Table 13-13—Golay sequences with length 128

Sequence name	Sequence value
$\mathbf{a}_{128}$	5A5599963C33FFF00F00CCC36966AAA5
$\mathbf{b}_{128}$	A5AA6669C3CC000F0F00CCC36966AAA5

13.2.4.2.3 SFD

The SFD field is used to establish frame timing. The SFD field shall consist of the sign inversion sequence of $\mathbf{a}_{128}$.

13.2.4.2.4 CES

The CES field, used for channel estimation, shall consist of [$\mathbf{a}_{256}$ $\mathbf{b}_{512}$ $\mathbf{a}_{512}$ $\mathbf{b}_{128}$] where the right most sequence, $\mathbf{b}_{128}$, is transmitted first.

The Golay complementary sequences of length 512, denoted by $\mathbf{a}_{512}$ $\mathbf{b}_{512}$, are defined in Equation (13-6) and Equation (13-7):

$$\mathbf{a}_{512} = [\mathbf{b}_{256}\ \mathbf{a}_{256}] \tag{13-6}$$

$$\mathbf{b}_{512} = [-\mathbf{b}_{256}\ \mathbf{a}_{256}] \tag{13-7}$$

where the number on the right $\mathbf{a}_{256}$ is transmitted first.

The Golay complementary sequences of length 256, denoted by $\mathbf{a}_{256}$ $\mathbf{b}_{256}$, are defined in Equation (13-8) and Equation (13-9):

$$\mathbf{a}_{256} = [\mathbf{b}_{128}\ \mathbf{a}_{128}] \tag{13-8}$$

$$\mathbf{b}_{256} = [-\mathbf{b}_{128}\ \mathbf{a}_{128}] \tag{13-9}$$

where the number on the right $\mathbf{a}_{128}$ are transmitted first.

13.2.4.3 Frame header

13.2.4.3.1 General

A frame header shall be added after the PHY preamble. The frame header conveys information in the PHY and MAC headers necessary for successfully decoding the frame.

The construction process of the frame header is shown in Figure 13-6.

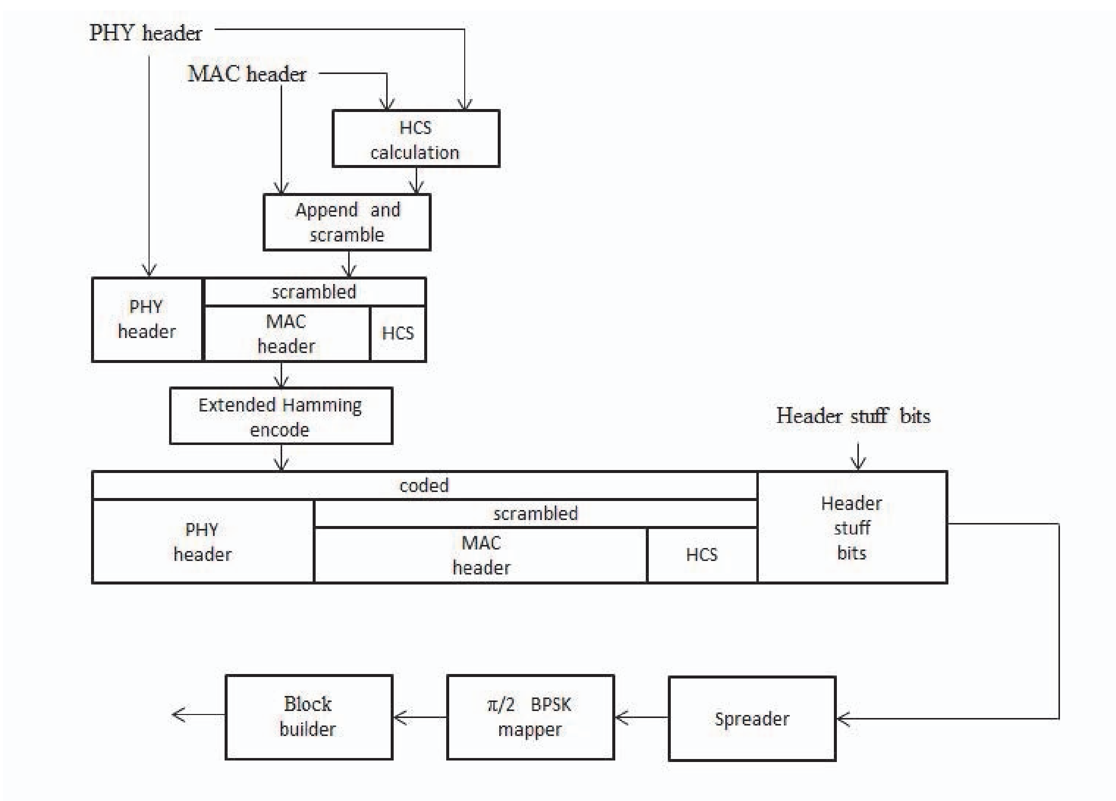

Figure 13-6—Frame header construction process

The detailed process of the construction is as follows:

a) Form the frame header as follows:
   1) Construct the PHY header based on information provided by the MAC
   2) Compute the HCS over the combined PHY and MAC headers
   3) Append the HCS to the MAC header
   4) Scramble the combined MAC header and HCS, as described in 12.2.3.10
   5) Encode the concatenation of the PHY header, scrambled MAC header, and scrambled HCS into a concatenated extended Hamming code, as described in 13.2.4.3.4
   6) Form the base frame header by concatenating the coded PHY header, coded scrambled MAC header, coded scrambled HCS, and header stuff bits

   The resulting frame header shall be modulated as shown in Figure 13-6.
b) Spread the frame header, as described in 13.2.3.8.
c) Map the frame header onto π/2 -shift BPSK, as described in 12.2.3.5.2.
d) Build blocks from the frame header, as described in 13.2.4.5.1.

The LFSR for the spreader is reset between the header and payload.

13.2.4.3.2 HRCP-SC PHY header

The HRCP-SC PHY header shall be formatted as illustrated in Figure 13-7.

Bits: b0–b2	b3	b4–b7	b8–b9	b10-b31	b32–b35
MCS	Pilot word	Scrambler Seed ID	PPRE	Frame Length	Reserved

Figure 13-7—PHY header format for HRCP-SC

The MCS field shall be set according to the values in Table 13-14.

Table 13-14—MCS field values

MCS	MCS identifier
0b000	0
0b001	1
0b010	2
0b011	3
0b100	4
0b101	5
0b110	6
0b111	Reserved

The Pilot Word field shall be set to one if the pilot word used in the current frame and shall be set to zero if otherwise.

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 12.2.3.10.

The PPRE field shall be set according to the values in Table 13-15.

Table 13-15—PPRE field values

PPRE	Number of blocks between PPRE
0b00	No PPRE inserted
0b01	1024
0b10	2048
0b11	4096

The Frame Length field shall be an unsigned integer equal to the number of octets in the MAC frame body including frame payload(s), MAC subheader(s) and padding octets in the aggregated frames, and FCS(s), but not including the frame header and the preamble.

13.2.4.3.3 Header check sequence (HCS)

The combination of the PHY header and MAC header shall be protected with a CRC-16 HCS. The MAC parameter *pLengthHCS* shall be 2 for this PHY. The CRC-16 HCS shall be the ones complement of the remainder generated by the modulo-2 division of the protected combined PHY and MAC headers by the polynomial:

$$x^{16} + x^{15} + x^{13} + x^8 + x^5 + x^3 + x + 1$$

The protected bits shall be processed in transmit order. All HCS calculations shall be made prior to data scrambling. A schematic of the processing is shown in Figure 13-8.

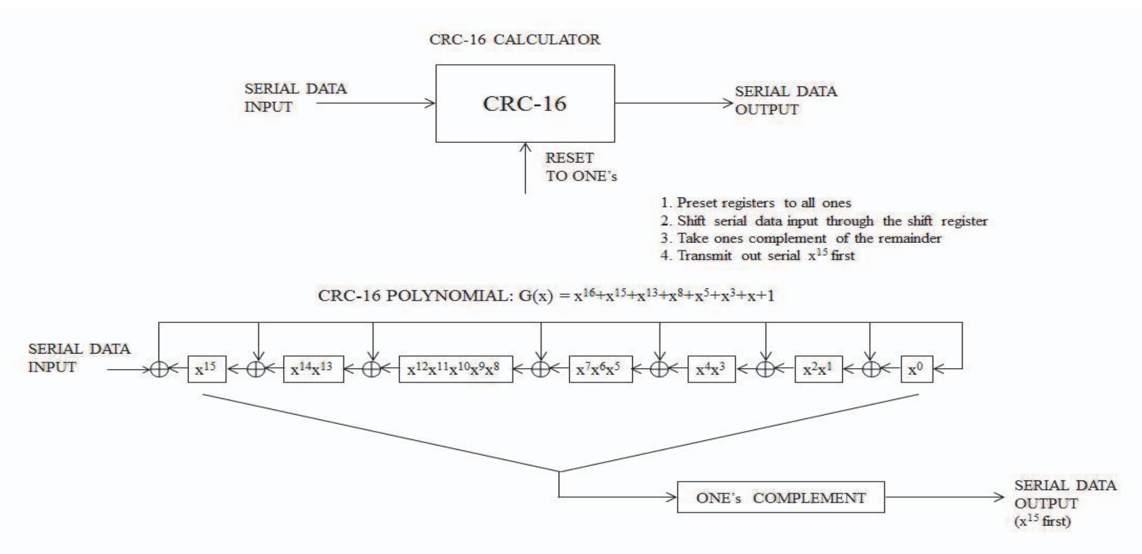

Figure 13-8—CRC-16 Implementation

13.2.4.3.4 Header FEC

To increase robustness in the frame header, the combination of the PHY header, scrambled MAC header, and HCS shall be encoded to concatenated code words of an extended Hamming (EH) code.

For each 4-bit input sequence, denoted as $\{i_0, i_1, i_2, i_3\}$, the encoder shall output the sequence followed by a 4-bit-parity sequence, denoted as $\{p_0, p_1, p_2, p_3\}$ determined using Table 13-16.

Table 13-16—Parity assignment of the Header FEC

i_0	i_1	i_2	i_3	p_0	p_1	p_2	p_3
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0
0	0	1	0	1	0	1	1
0	0	1	1	0	1	0	1
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	1
0	1	1	0	1	1	0	0
0	1	1	1	0	0	1	0
1	0	0	0	1	1	0	1
1	0	0	1	0	0	1	1
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	0
1	1	0	1	0	1	0	0
1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1

13.2.4.4 HRCP-SC PHY Payload field

13.2.4.4.1 General

The HRCP-SC PHY Payload field is the last component of the frame and is constructed as shown in Figure 12-23. This payload is used in single-input, single-output (SISO) operations.

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 12.2.3.10.
b) Encode the scrambled MAC frame body as specified in 13.2.3.6.
c) Add stuff bits to the encoded and scrambled MAC frame body according to 13.2.3.7.
d) Map the resulting MAC frame body onto the appropriate constellation as described in 13.2.3.5.
e) Build blocks from the resulting MAC frame body according to 13.2.4.5.1.
f) Insert PPRE periodically as described in 13.2.4.5.2.

13.2.4.4.2 HRCP-SC PHY Payload scrambling

The HRCP-SC PHY payload shall use the scrambling process defined in 12.2.3.10.

13.2.4.4.3 Modulation

Modulation for the MAC frame body is defined in 13.2.3.5.

13.2.4.4.4 FEC

FEC for the MAC frame body is defined in 13.2.3.6.

13.2.4.5 Pilot word and PPRE

13.2.4.5.1 Block and pilot word

A block is formed by appending a pilot word to the frame data. Building of the blocks is illustrated in Figure 13-9. The possible pilot word lengths are 0 and 8. For the pilot word length 0, the length of the data is 64 symbols. For the pilot word length 8, the length of the data is 56 symbols. The last block of a frame and the block followed by a PPRE shall be followed by a pilot word.

The pilot word with a length of 8 is 0xEB, where the leftmost bit shall be the MSB in each hexadecimal-equivalent 4-binary-digit group. The pilot word shall be modulated with π/2 BPSK.

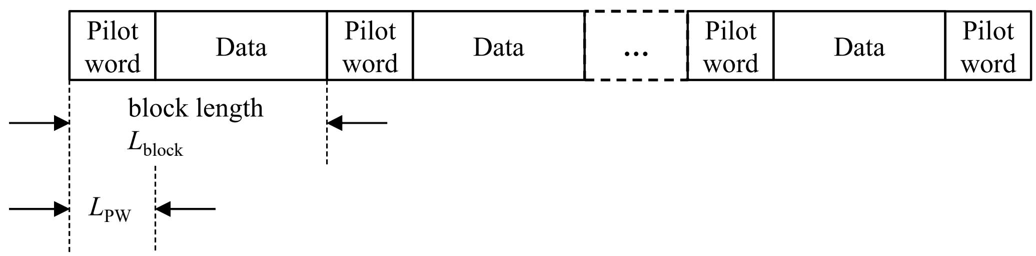

Figure 13-9—Frame format with pilot words

13.2.4.5.2 PPRE

PPRE insertion is an optional feature that allows a DEV to periodically readjust the receiver algorithms as described in 13.2.4.2. PPRE is inserted into the scrambled, encoded, spread, and modulated MAC frame body as illustrated in Figure 13-10 with an interval specified in Table 13-9. The PPRE field shall be the concatenation of SYNC, SFD, and CES field in the preamble defined in 13.2.4.2. PPRE shall be modulated with π/2 BPSK.

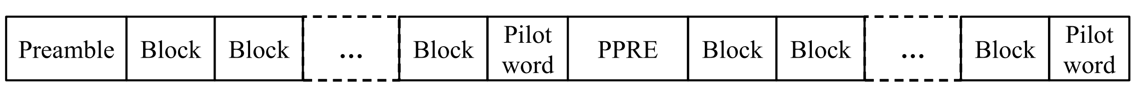

Figure 13-10—Data Blocks with PPRE

13.2.5 Transmitter specifications

13.2.5.1 EVM requirement

The EVM of a compliant transmitter shall be measured and calculated as defined in 12.1.8 and shall not exceed the values given in Table 13-17 for the indicated mode. Note that this requirement assumes a conducted measurement.

13.2.5.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ maximum.

Table 13-17—Maximum allowed EVM for HRCP-SC PHY with a single channel

MCS identifier	Maximum EVM (dB)
0	–12
1	–15
2	–18
MCS identifier	**Maximum EVM (dB)**
3	–22
4	–25
5	–29
6	–36

13.2.5.3 Symbol rate

The SC PHY shall be capable of transmitting at the chip rate, as defined in Table 13-8, to within $\pm 30 \times 10^{-6}$.

The MAC parameter, *pPHYClockAccuracy*, shall be $\pm 30 \times 10^{-6}$.

13.2.5.4 Transmit power-on and power-down ramp

The transmit power-on ramp is defined as the time it takes for the RF power emitted by the compliant DEV to rise from less than 10% to greater than 90% of the maximum power to be transmitted in the frame.

The transmit power-on ramp shall be less than 9.3 ns.

The transmit power-down ramp is defined as the time it takes for the RF power emitted by the compliant DEV to fall from greater than 90% to less than 10% of the maximum power to be transmitted in the frame.

The transmit power-down ramp shall be less than 9.3 ns.

The transmit power ramps shall be constructed such that the emissions conform to the unwanted emissions specification defined in 13.1.4.

13.2.6 Receiver specifications

13.2.6.1 Error rate criterion

The error rate criterion shall be a FER of less than 8% with a frame payload length of 2^{14} octets. The error rate should be determined at the PHY SAP interface after any error correction methods (excluding retransmission) required in the proposed DEV has been applied. The measurement shall be performed in AWGN channel.

13.2.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal present at the input of the receiver for which the error rate criterion in 13.2.6.1 is met. The error ratio shall be determined after any error correction has been applied. A compliant DEV that implements the SC PHY shall achieve at least the reference sensitivity listed in Table 13-18.

Table 13-18—Reference sensitivity levels for MCS

MCS identifier	Receiver sensitivity
0	–61 dBm
1	–58 dBm
2	–55 dBm
3	–51 dBm
4	–49 dBm
5	–45 dBm
6	–39 dBm

13.2.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal present at the input of the receiver for which the error rate criterion in 13.2.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –30 dBm for each of the mandatory modulation formats that the DEV supports.

13.2.7 PHY layer timing

13.2.7.1 General

The values for the PHY layer timing parameters are defined Table 13-19.

Table 13-19—PHY layer timing parameters

PHY parameter	Value	Subclause
pPHYSIFSTime	0.5 µs to 2.5 µs (0.1 µs step), 2.5 µs is default	13.2.7.4
pPHYChannelSwitchTime	100 µs	13.2.7.6

13.2.7.2 Interframe space

A conforming implementation shall support the IFS parameters, as described in 7.6.2, given in Table 13-20. The values described by the Long RIFS MAC parameter shall be applied when the Long RIFS Support field in the THz Pairnet Operation Parameter IE is set to one; the values described by the RIFS MAC parameter shall be applied otherwise.

Table 13-20—IFS parameters

MAC parameter		Corresponding PHY parameter	Definition
SIFS		*pPHYSIFSTime*	13.2.7.4
RIFS	PRC	2*pPHYSIFSTime + 3.01 µs	7.6.2
	DEV	4*pPHYSIFSTime + 9.05 µs	
Long RIFS	PRC	2*pPHYSIFSTime + 69.7 µs	7.6.2
	DEV	4*pPHYSIFSTime + 142 µs	

13.2.7.3 Receive-to-transmit turnaround time

The receive to transmit turnaround time shall be *pPHYSIFSTime*, including the power-up ramp specified in 13.2.5.4. The receive to transmit turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

13.2.7.4 Transmit-to-receive turnaround-time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*, including the power-down ramp specified in 13.2.5.4.

13.2.7.5 Time between transmission

The minimum time between the end of the last transmitted frame and the beginning of the retransmitted frame shall be less than a RIFS time specified in Table 13-20. A PRC shall use the shorter RIFS than that of a DEV in Table 13-20.

13.2.7.6 Channel switch

The *channel switch time* is defined as the time from the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pPHYChannelSwitchTime*.

13.2.8 PHY management for HRCP-SC PHY

13.2.8.1 General

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the HRCP-SC PHY layer of a DEV.

13.2.8.2 Maximum frame size

The maximum frame length allowed, *pMaxFrameBodySize*, shall be 2099200 octets. This total includes the MAC frame body, but not the PHY preamble, base header, (PHY header, MAC header and HCS). The maximum frame length also does not include the stuff bits.

13.2.8.3 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 2 097 152 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2 097 152 octets minus the security overhead as defined in 7.3.4.2, 7.2.8.1.2, or 7.2.8.2.2.

13.2.8.4 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed with the HRCP-SC PHY shall be 2048 octets as defined in 6.4.13.

13.2.9 MIMO, channel bonding, and channel aggregation

13.2.9.1 Introduction to MIMO in SC PHY

This subclause describes the MIMO transmission PHY specification for high transmission rates, above 100 Gbit/s. SISO is the mode that is described in 13.2.2 to 13.2.8, in which spatial division multiplexing is not used. For such high rate, spatial division multiplexing is required besides higher order constellation and frequency channel aggregation. For MIMO, the number of branches (the number of spatial streams) are: 2, 4, 9, and 16.

— ×2 mode: 2×2 MIMO
— ×4 mode: 4×4 MIMO
— ×9 mode: 9×9 MIMO
— ×16 mode:16×16 MIMO

Table 13-21—MCS using MIMO

Modulation	Code rate	PHY transmission rate, Gbit/s									
		SISO		2×2		4×4		9×9		16×16	
		Without pilot word	With pilot word[a]	Without pilot word	With pilot word[a]	Without pilot word	With pilot word[a]	Without pilot word	With pilot word[a]	Without pilot word	With pilot word[a]
π/2 QPSK	14/15	3.3	2.9	6.6	5.7	13.1	11.5	29.6	25.9	52.6	46.0
16-QAM	11/15	5.2	4.5	10.3	9.0	20.7	18.1	46.5	40.7	82.6	72.3
16-QAM	14/15	6.6	5.7	13.1	11.5	26.3	23.0	59.1	51.7	105.1	92.0
64-QAM	11/15	7.7	6.8	15.5	13.6	31.0	27.1	69.7	61.0	123.9	108.4
64-QAM	14/15	9.9	8.6	19.7	17.2	39.4	34.5	88.7	77.6	157.7	138.0

[a]Pilot word length/sub-block length = 8/64.

13.2.9.2 Channel aggregation and channel bonding

There are a total of 11 possible channel aggregation combinations, as defined in 6.4.13. The bonded channels are defined in Table 13-1 with CHNL_ID from 7 to 16.

13.2.9.3 Link setup procedure for MIMO mode

In this subclause the link setup procedure for MIMO transmission mode is described. Figure 13-11 shows the procedure The PRC intends to use M_1 spatial streams (equal to the number of MIMO branches). The PRC has M_{array} antenna elements in its antenna array. The PRC selects and switches on well-placed M antenna elements before starting MIMO mode. At first, a Beacon frame is sent in SISO mode from the PRC to the DEV, which intends to use M_2 spatial streams. This SISO transmission is done using antenna element #1 at the PRC and antenna element #1 at the DEV. The PRC sends a Beacon frame, which comprises the following:

— MIMO capability (available combination of values of M_1)
— Value of N_{ar} (number of Array Training commands. $N_{ar} = 0 \sim 511$, $N_{ar} < M_{array}$)
— Value of T_{ar} (period of Array Training commands, 10 µs, 20 µs, 40 µs, or 80 µs)
— Channel Aggregation capability
— Channel bonding capability

When the DEV comes into close proximity region with the PRC, the Beacon frame is received by the DEV. The DEV decides the number of branches, M, which is used in the MIMO mode that follows the current SISO mode. The number of branches M is decided, for example, by calculating min(max(M_1), max(M_2)). Channel bonding or aggregation is decided as well. Their decision method depends on the implementations.

After M and channel aggregation or channel bonding are decided, the DEV starts to send an Association Request command to the PRC using its antenna element that was used in the Beacon reception. The Association Request command contains the following:

— Value of M (number of MIMO streams), as MIMO capability field
— Channel aggregation or bonding to be used, as capability field

When the Association Request command is received by the PRC, the PRC knows the number of MIMO branches M, which will be used in the MIMO mode, channel aggregation/bonding. At this time the PRC stops sending Beacon frames and sends an Association Response command. After sending the Association Response command, the PRC switches to the Array Training mode and becomes ready to listen for Array Training commands sent from the DEV. After the DEV receives the Association Response command, the DEV switches to the Array Training mode. At this point, the point-to-point system has completed the link establishment.

If the value of N_{ar} is more than zero, the next step for the MIMO transmission is to start the Array Training mode to select a set of antenna elements in the antenna array of the PRC, whose concept is described in 13.2.9.4.

If $N_{ar} = 0$, the Array Training mode shall be skipped because this means the PRC does not require the Array Training mode. At this time the MIMO PHY frame exchange is started just after the Ack for the Association Response command is received by the PRC.

In the Array Training mode, the DEV starts sending Array Training commands after it recognizes it is not moving around on the array surface of the PRC. The method for the recognition is up to the implementation. For example the NFC communication or optical camera imaging can be used, or by using a timer assuming the user stabilizes the positions of DEV within a certain time (e.g., 2 s).

All Array Training commands shall be transmitted with a No-ACK policy. The number of Array Training commands sent is equal to N_{ar}. These are transmitted from antenna element #1 to allow the PRC to select antenna elements for following MIMO transmission.

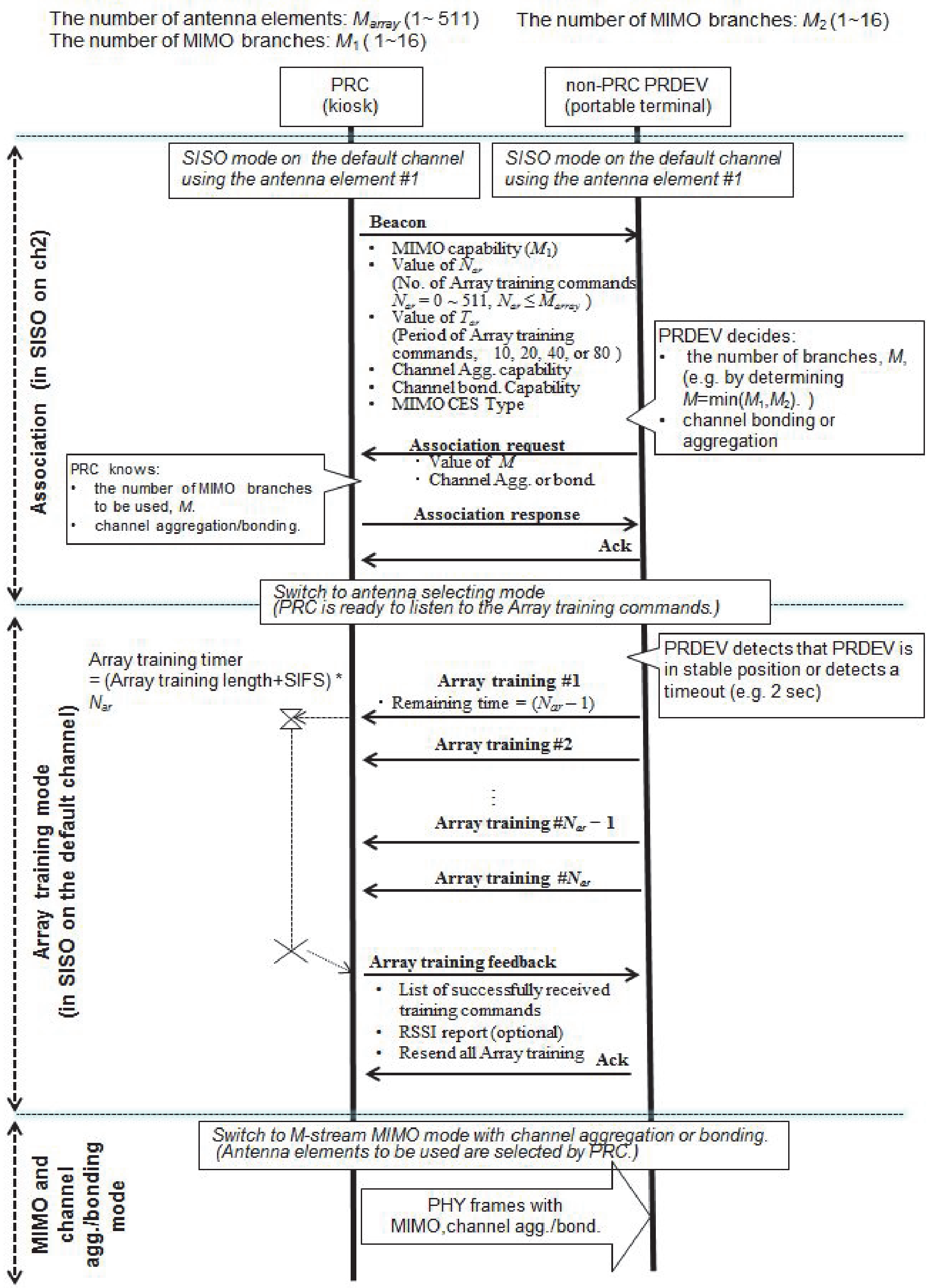

Figure 13-11—Setup sequence for MIMO transmission

Though the PRC's antenna switching procedure in the Array Training mode is up to the implementation, certain steps are recommended to increase the likelihood of successful reception of the first Array Training command. For example, PRC receives Array Training command #1 using the same antenna element that successfully sent the Association Response command. If the first Array Training command is not received, the DEV is disconnected.

When the first Array Training command is received the Array Training timer is started.

Whether the PRC can receive Array Training commands #2 through #N_{ar} or not, the Array Training feedback command shall be sent when the Array Training timer expires.

When the Array Training timer expires, the PRC selects, if necessary, M antenna elements, out of M_{array}, elements that are going to be used in the following MIMO mode.

The Array Training feedback command is sent with a Stk-Ack policy.

The Array Training feedback command comprises the following information:

— List of successfully received training commands field
— RSSI report (optional) field
— Resend all Array Training commands field

If the Array Training feedback command is not received by DEV, the PRC shall retransmit the Array Training feedback command.

If the Resend all Array Training commands field is set to one, the DEV shall resend all Array Training commands after sending the Ack for the Array Training feedback command to the PRC. This mechanism enables the PRC to do the Array Training until acceptable.

After that, both DEVs switch into MIMO mode and start MIMO frame exchange with channel aggregation or channel bonding. The MIMO mode cannot be turned into SISO mode until the communication session ends.

The multiple of Array Training commands transmission is necessary for the antenna selecting procedure, which is described in the following subclauses. The number of Array Training commands sent from DEV (N_{ar}) and their time period (T_{ar}) are notified by the Beacon frame.

When $M < M_{array}$, the PRC selects M antenna elements. In this case, the procedure for selecting antenna elements is as follows: select using reception levels and Array Training commands that are sent N_{ar} times, hence N_{ar} combinations of antennas are switched on to receive these commands. When $M = M_{array}$, the PRC does not have to select the antenna elements.

13.2.9.4 Selecting antenna element

When MIMO transmission is used in the close proximity wireless communication, the use of line-of-sight (LOS) MIMO that requires no multipath propagation effect will be assumed. In LOS-MIMO transmission, the displacement of antenna arrays between transmitter and receiver will cause significant degradation in the channel capacity. Hence the use of a large-scale (e.g., up to 511 elements) array in the PRC (for example the kiosk that is allowed to put an antenna array with a large footprint) and selecting well-placed elements that are faced well to the array of the portable terminal will overcome such issues.

The concept is shown in Figure 13-12. The procedure of selecting antenna elements is done in the Array Training mode shown in Figure 13-11. The method used to select an antenna is an implementation matter. An example is shown here. For example. the kiosk has the large array whose number of elements is $M_{array} = 256$ and the kiosk selects M element for MIMO transmission. While the portable terminal sends N_{ar} Array Training commands from its antenna element #1, the kiosk measures the reception level of these frames. After the measurements, the kiosk selects the M elements with the largest reception level. They will be used for following PHY frame transmission in MIMO mode.

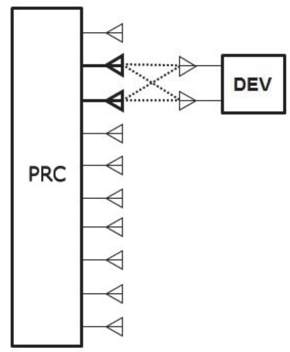

Figure 13-12—Antenna element selection

13.2.9.5 MIMO PHY Preamble

13.2.9.5.1 Overview

The Preamble is comprised of the SYNC, SFD, and CES as shown in Figure 13-13. Note that the preamble defined in 13.2.4.2 is used when channel aggregation, bonding, or both schemes are applied without MIMO.

After SFD transmission each antenna sends CES.

MIMO CES should be one shown in 13.2.9.5.4 or one shown in 13.2.9.5.5. The type of MIMO CES is advertised by the Beacon frame, as shown in 6.4.44.

Tx#*i* (*i* = 1 ~ 16)	SYNC# *i*	SFD# *i*	CES# *i*	Header # *i*	Payload #*i*

Figure 13-13—PHY frame structure in MIMO mode

13.2.9.5.2 SYNC

The SYNC is the same as SISO mode except the cyclic shift delay is applied in each spatial stream. The cyclic shift is in order to prevent the reception level depression due to canceling effect, or unintended beam forming effect, caused by the element spacing. The value of cyclic shift for i^{th} transmitter T_{CSsync_i} is shown in Equation (13-10):

$$T_{CSsync\_i} = 60 * (i - 1) \text{ [ns]} \quad (13\text{-}10)$$

13.2.9.5.3 SFD

The SFD is the same as SISO mode except the cyclic shift delay is applied in each spatial stream. The value of cyclic shift for i^{th} transmitter T_{CSsfd_i} is shown in Equation (13-11):

$$T_{CSsfd\_i} = 4 * (i - 1) \text{ [ns]} \quad (13\text{-}11)$$

13.2.9.5.4 CES for frequency domain channel estimation

The CES for frequency domain channel estimation is generated using Golay complementary sequences of length 256, $\mathbf{a}_{256}$, which is defined in 13.2.4.2.4.

First, the sequence is π/2 BPSK-modulated signal in the frequency domain. Then the signal is converted into time-domain waveform. For each spatial stream, cyclic shift delays are applied in the time domain.

The value of cyclic shift delay for i^{th} transmitter T_{CSces_i} is

— T_{CSces_i} = 128* (i – 1) [chips] when M = 2
— T_{CSces_i} = 64* (i – 1) [chips] when M = 4
— T_{CSces_i} = 28* (i – 1) [chips] when M = 9
— T_{CSces_i} = 16* (i – 1) [chips] when M = 16.

Finally repetition is applied to each stream five times.

13.2.9.5.5 CES for time-domain channel estimation

CES for i^{th} stream, which transmitted from i^{th} transmitter is [$\mathbf{e}_{\_i\_256}\,\mathbf{c}_{1024\_i}\,\mathbf{d}_{\_i\_256}$], where

$\mathbf{c}_{1024\_i}$: shifted $\mathbf{c}_{1024}$
$\mathbf{c}_{1024}$: [$\mathbf{a}_{512}\mathbf{b}_{512}$]
$\mathbf{d}_{\_i\_256}$: First 256 digits of $\mathbf{c}_{1024\_i}$
$\mathbf{e}_{\_i\_256}$: Last 256 digits of $\mathbf{c}_{1024\_i}$

Here $\mathbf{a}_{512}$ and $\mathbf{b}_{512}$ are defined in 13.2.4.2.4. The sequence $\mathbf{c}_{1024\_i}$ is cyclic-shifted $\mathbf{c}_{1024}$. The value of the cyclic shift delay for i^{th} transmitter T_{CSces_i} is shown in 13.2.9.5.4.

13.2.9.6 Data processing for *M*-streams transmission in MIMO mode or channel aggregation

At the transmitter, a bitstream received from the MAC is divided into $M \times N$ streams where the number of spatial streams for MIMO is M and the number of aggregated frequency channels is N. At the receiver, these divided bitstreams must be correctly combined into a single stream. Each substream should have a number tag for correct combination at the receiver.

13.2.9.7 HRCP-SC-MIMO PHY Header

Information for MIMO and channel aggregation bitstream processing, described in 13.2.9.6, shall be included in the header as shown in Figure 13-14. The HRCP-SC-MIMO PHY header shall be processed in the same manner as described in 13.2.4.3.

Bits: b0–b2	**b3**	**b4–b7**	**b8–b9**	**b10–b31**	**b32–b33**	**b34–b37**
MCS	Pilot word	Scrambler Seed ID	PPRE	Frame Length	Reserved	Spatial (MIMO) Stream Number 0 ~ 15

Figure 13-14—PHY header format for MIMO PHY

The Pilot Word field shall be set to the same value in every stream. The MCSs shall be set so that the same rate of FEC is used throughout all channels.

13.2.9.8 HRCP-SC PHY MIMO Payload field

The HRCP-SC MIMO PHY Payload field is the last component of the frame and is constructed as follows:

a) Append stuff bits to the MAC frame body as follows.

The number of blocks in the encoded MAC frame body, $N_{block\text{-}encMFB}$, and the length of stuff bits, L_{STUFF}, are given by Equation (13-12) and Equation (13-13).

$$N_{block-encMFB} = \mathrm{CEIL}\left(L_{ebits} / \sum_{i=1}^{M} L^{i}_{CPBS}\right) \tag{13-12}$$

$$L_{STUFF} = N_{block-encMFB} \times \sum_{i=1}^{M} L^{i}_{CPBS} - L_{ebits} \tag{13-13}$$

where L^{i}_{CPBS} is the number of coded bits per sub-block as given in Table 13-11 for each MCS, M is the number of spatial streams, and L_{ebits} is the length of the coded MAC frame body calculated with the rate of the FEC rate used.

b) Divide the stuffed MAC frame body into M spatial streams.
Procedure: Divide the resulting MAC frame in a round robin fashion assigning bits equal to the number of bits per symbol to each spatial stream.

c) In each spatial stream, Scramble the MAC frame body according to 12.2.3.10.

d) In each spatial stream, Encode the scrambled MAC frame body as specified in 13.2.3.6.

e) In each spatial stream, map the resulting MAC frame body onto the appropriate constellation as described in 13.2.3.5.

f) In each spatial stream, build blocks from the resulting MAC frame body according to 13.2.4.5.1.

g) In each spatial stream, insert PPRE periodically as described in 13.2.4.5.2.

13.2.9.9 Scrambler

The MIMO PHY payload shall use the scrambling process described in 12.2.3.10.

13.2.9.10 Transmitter specifications

The transmitter specifications are the same as the SISO mode as described in 13.2.5.

13.2.9.11 Receiver specifications

13.2.9.11.1 Error rate criterion

The error rate criterion shall be a FER of less than 8% with a frame payload length of 2^{14} octets. The error rate should be determined at the PHY SAP interface after any error correction methods (excluding retransmission) required in the proposed DEV has been applied. The measurement shall be performed in multipath channel.

13.2.9.11.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 13.2.9.11.1 is met. The error ratio shall be determined after any error correction has been applied. A compliant DEV that implements the SC PHY shall achieve at least the reference sensitivity listed in Table 13-22.

Table 13-22—Reference sensitivity levels for each mode

MIMO number of branches	MCS	Receiver sensitivity (dBm)
2	1 (QPSK, 14/15)	–54
2	2 (16QAM, 11/15)	–55
2	3 (16QAM, 14/15)	–50
2	4 (64QAM, 11/15)	–49
2	5 (64QAM, 14/15)	–43
4	1	–54
4	2	–52
4	3	–49
MIMO number of branches	**MCS**	**Receiver sensitivity (dBm)**
4	4	–47
4	5	–43
9	1	–52
9	2	–40
9	3	–47
9	4	–45
9	5	–40
16	1	–50
16	2	–49
16	3	–44
16	4	–42
16	5	–36

13.3 HRCP-OOK PHY

13.3.1 General

The HRCP-OOK PHY is designed for cost effective DEVs that require low power, low complexity, and simple design. The HRCP-OOK PHY supports a single modulation scheme, OOK, and a single FEC scheme, RS. The HRCP-OOK PHY supports channel bonding using up to four channels for high throughput. Channel aggregation and MIMO are not used in HRCP-OOK PHY.

13.3.2 Channel bonding

13.3.2.1 General

HRCP-OOK PHY may use channel bonding using up to four channels. The channels that are bonded together are adjacent or contiguous to one another and they are used simultaneously to achieve high throughput. The chip rate R_c is increased as listed in Table 13-25 due to the expanded bandwidth by bonding contiguous channels.

13.3.2.2 Channelization for HRCP-OOK PHY

Figure 13-15 depicts the channels used by HRCP-OOK PHY.

HRCP-OOK PHY uses the channels defined as CHNL_ID 2, 8, 12, and 15 in Table 13-1. An implementation that implements the HRCP-OOK PHY shall support at least channel 2 (CHNL_ID 2), which is the default channel. When channel bonding is used, contiguous two or three or four channels are used. The phyCurrentChannel is the CHNL_ID of the current channel.

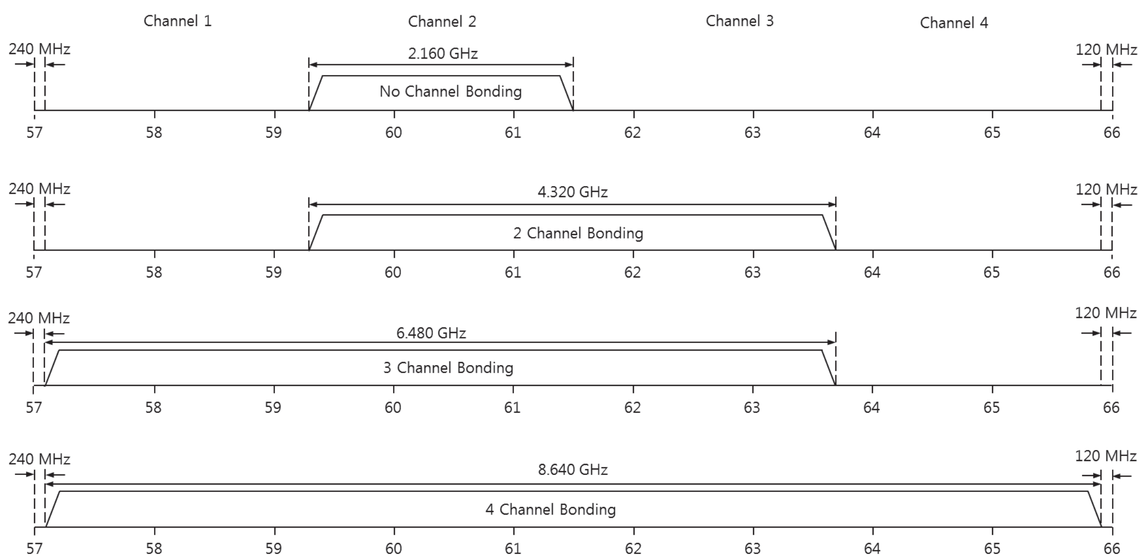

Figure 13-15—Channels used by HRCP-OOK PHY

When the channel bonding is not used, only channel 2 (CHNL_ID 2) shall be used. When two channels are bonded together, channel 2 and 3 (CHNL_ID 8) shall be used. When three channels are bonded together, channel 1, 2, and 3 (CHNL_ID 12) shall be used. When four channels are bonded together, channel 1, 2, 3, and 4 (CHNL_ID 15) shall be used.

13.3.2.3 Transmit PSD mask for HRCP-OOK

The transmit spectral mask specified in Figure 13-1 shall be used for single channel transmission. The transmit spectral mask specified in Figure 13-2, Table 13-2, and Table 13-3 shall be used for transmission using channel bonding.

13.3.3 Modulation and coding

13.3.3.1 MCS dependent parameters

The chip rate of HRCP-OOK PHY is given in Table 13-25. The entire HRCP-OOK frame shall be modulated with OOK as specified in 13.3.3.5.2. The FEC for HRCP-OOK PHY shall be RS coding as specified in 13.3.3.6. The MCS-dependent parameters shall be set according to Table 13-23. An implementation that implements the HRCP-OOK PHY shall support at least Mode 1 with pilot symbol length $L_P = 0$ in the Table 13-23.

Table 13-23—MCS-dependent parameters

Mode	MCS identifier	Data rate (Mb/s) with pilot symbol[a] (L_P = 4 or 8)	Data rate (Mb/s) without pilot symbol (L_P = 0)	Modulation	Spreading factor, L_{SF}	FEC type
Mode 1 (No channel bonding)	1	1630	1643 (Mandatory)	OOK	1	RS (240,224)
Mode 2 (2 channel bonding)	0	1630	1643		2	
	1	3260	3285		1	
Mode 3 (3 channel bonding)	0	2445	2464		2	
	1	4890	4928		1	
Mode 4 (4 channel bonding)	0	3260	3285		2	
	1	6519	6571		1	

[a]Pilot symbols specified in 13.3.4.4.6 are included in the calculation of data rates.

RS(240,224) shall be used for encoding the frame payloads of HRCP-OOK.

When $L_{SF} = 1$ is used, the block length L_{block} of HRCP-OOK PHY payload shall be 512 chips and the pilot symbol length L_P shall be 0 or 4 chips. The pilot symbol length $L_P = 0$ is mandatory and $L_P = 4$ is optional.

When $L_{SF} = 2$ is used, the block length L_{block} of HRCP-OOK PHY payload shall be 1024 chips and the pilot symbol length L_P shall be 0 or 8 chips. The pilot symbol length $L_P = 0$ is mandatory and $L_P = 8$ is optional.

13.3.3.2 Header rate-dependent parameters

The header rate-dependent parameters shall be set according to Table 13-24.

RS(n+16, n), which is a shortened version of RS(240,224) where n is the number of octets in the frame header, shall be used for encoding the frame header of HRCP-OOK. The block length L_{block} of the frame header shall be 512 chips regardless of the spreading factor L_{SF}. Pilot symbols are not used in HRCP-OOK PHY frame header.

The MAC Subheaders in each aggregated subframes shall use the same MCS for the MAC frame body, thus the MCS remains the same within aggregated frames. The Security header shall also use the same MCS for the MAC frame body.

Table 13-24—Header rate-dependent parameters

Channel bonding	Header rate [a] (Mb/s)	Modu-lation	Spreading factor, L_{SF}	Pilot symbol length, L_P	Coded bits per block, N_{CBPB}	Number of occupied blocks, L_{block_hdr}	Number of stuff bits, L_{STUFF}
No channel bonding	55	OOK	16	0	32	8	0
2 channel bonding	110						
3 channel bonding	165						
4 channel bonding	220						

[a]RS (32, 16) is used in the calculation.

13.3.3.3 Timing-related parameters

Table 13-25 lists the general timing parameters associated with the HRCP-OOK PHY.

Table 13-25—Timing-related parameters

Parameter	Description	Value				Unit	Formula
R_c	Chip rate	1760 (1 channel)	3520 (2 channel)	5280 (3 channel)	7040 (4 channel)	Mchip/s	
T_c	Chip duration	0.568	0.284	0.189	0.142	ns	1/ R_c
L_{block}	Block length	512 (Frame header, Payload with L_{SF} =1)				chips	
		1024 (Payload with L_{SF} =2)					
L_P	Pilot symbol length	0 (Frame header, Payload)	4 (Payload with L_{SF} =1)		8 (Payload with L_{SF} =2)	chips	
T_P	Pilot symbol duration	0	2.273, 1.136, 0.758, 0.568 (dependent on L_P and T_c)		2.273, 1.515, 1.136 (dependent on L_P and T_c)	ns	
L_{DC}	Length of data chips per block	512 (Frame header, Payload with L_{SF} =1)	508		1016	chips	
		1024 (Payload with L_{SF} =2)					
T_{block}	Block duration (L_{block} = 512 or 1024)	290.9	145.5	97.0	72.7	ns	L_{block} X T_c
		—	290.9	193.9	145.5		

13.3.3.4 Frame-related parameters

The frame parameters associated with the HRCP-OOK PHY are listed in Table 13-26 where CEIL is the ceiling function, which returns the smallest integer value greater than or equal to its argument. The maximum frame duration occurs when the number of octets in the PHY Payload field is 2099200.

Table 13-26—Frame-related parameters

Parameter	Description	Value
N_{SYNC}	Number of code repetitions in the SYNC sequence	16
T_{SYNC}	Duration of the SYNC sequence	1.16 μs
N_{SFD}	Number of code repetitions in the SFD	4
T_{SFD}	Duration of the SFD	0.29 μs
N_{CES}	Number of code repetitions in the CES	8
T_{CES}	Duration of the CES	0.58 μs
N_{pre}	Number of code repetitions in the PHY preamble	28
T_{pre}	Duration of the PHY preamble	2.036 μs
L_{hdr}	Length of the base headers in octets	32
Parameter	**Description**	**Value**
N_{block_hdr}	Number of blocks in the base frame header	8
T_{hdr}	Duration of the base frame header	N_{block_hdr} X T_{block}
$L_{payload}$	Length of frame payloads in octets	variable
L_{FCS}	Length of FCS in octets	4
L_{MFB}	Length of the MAC frame body in octets	$L_{payload} + L_{FCS}$
N_{CBPB}	Number of coded bits per block in the MAC frame body	$(L_{block} – L_P)/ L_{SF}$ (508 with Pilot symbol, 512 without Pilot symbol)
N_{block_MFB}	Number of blocks for the MAC frame body	CEIL[(L_{MFB} X 8)/(R_{FEC} X N_{CBPB})] (R_{FEC} : FEC rate)
T_{MFB}	Duration of the MAC frame body	N_{block_MFB} X T_{block}
$T_{datafield}$	Duration of the PHY Payload field	N_{block_MFB} X T_{block}
T_{frame}	Duration of the frame	$T_{pre} + T_{hdr} + T_{datafield}$

13.3.3.5 Modulation

13.3.3.5.1 General

After channel encoding and spreading, the resulting bits shall be modulated using OOK as specified in 13.3.3.5.2. The actual transmitted RF signal is described in 13.3.3.5.3.

13.3.3.5.2 OOK

HRCP-OOK frames shall be modulated using OOK. The OOK modulation shall use variable amplitudes to represent the data. As shown in Figure 13-16, OOK shall be represented by two points in the constellation map. The simplest form of OOK represents a binary “1” with the presence of the signal, and a binary “0” with the absence of it. The normalization factor, K_{MOD} shall be $\sqrt{2}$.

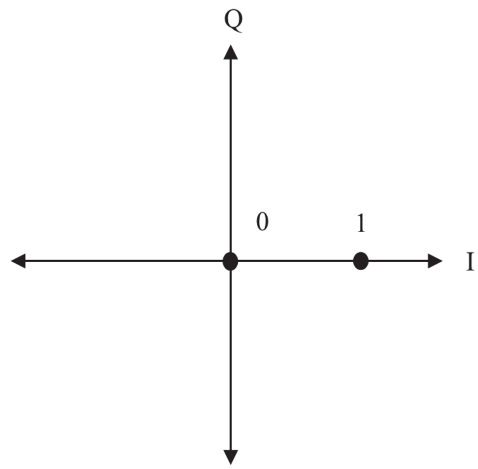

Figure 13-16—Constellation diagram for OOK

13.3.3.5.3 Description of signals

The actual transmitted RF signal can be written as shown in Equation (13-14):

$$S_{RF}(t) = \sum_{k=0}^{N_{chip}-1} a_k s_B(t - kT_c)\cos(2\pi f_c t) \qquad (13\text{-}14)$$

where

$S_{RF}(t)$ is the transmitted RF signal
T_c is the chip duration
N_{chip} is the number of transmitted chips in the transmitted OOK PHY frame
f_c is the center frequency
a_k is a binary value in the transmitted frame
$S_B(t)$ is the baseband pulse shape

13.3.3.6 Forward Error Correction

Only Reed-Solomon (RS) block codes specified in this subclause shall be used for HRCP-OOK PHY.

The RS(240,224), which is the mother code, shall be used for encoding the frame payloads of HRCP-OOK. RS(n+16, n), a shortened version of RS(240,224) where n is the number of octets in the frame header, shall be used for encoding the frame header of HRCP-OOK.

The systematic RS code shall use the generator polynomial shown in Equation (13-15):

$$g(x) = \prod_{k=1}^{16} (x + a^k) \tag{13-15}$$

where

α = 0x02 is a root of the binary primitive polynomial $p(x)=1+x^2+x^3+x^4+x^8$

As notation, the element $M = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$, has the binary representation $b_7b_6b_5b_4b_3b_2b_1b_0$, where b_7 is the MSB and b_0 is the LSB.

The mapping of the information octets $\boldsymbol{m} = (m_{223}, m_{222}, ..., m_0)$ to codeword octets $\boldsymbol{c} = (m_{223}, m_{222}, ..., m_0, r_{15}, r_{14}, ..., r_0)$ is achieved by computing the remainder polynomial $r(x)$ as shown in Equation (13-16):

$$r(x) = \sum_{k=0}^{15} r_k x^k = x^{16} m(x) \mathrm{mod} g(x) \tag{13-16}$$

where

$m(x)$ is the information polynomial:

$$m(x) = \sum_{k=0}^{223} m_k x^k$$

and

r_k (k = 0, …, 15) and
m_k (k = 0, …, 223) are elements of GF(2^8).

The message order is as follows: m_{223} is the first octet of the message and m_0 is the last octet of the message.

For a shortened RS(L_{inf}+ 16, L_{inf}), 224– L_{inf} zero elements are appended to the incoming L_{inf} octet message as follows:

$m_k = 0$, $k = L_{inf}$, …, 223

These inserted zero elements are not transmitted and recovered at the receiver. A shift-register implementation of the RS encoder RS(L_{inf} +16, L_{inf}) is shown in Figure 13-17, with additions and multiplications over GF(2^8). The inserted zero elements are encoded first. After m_0 has been inserted into the shift register, the switch shall be moved from the message polynomial input connection to the shift register output connection (right-to-left).

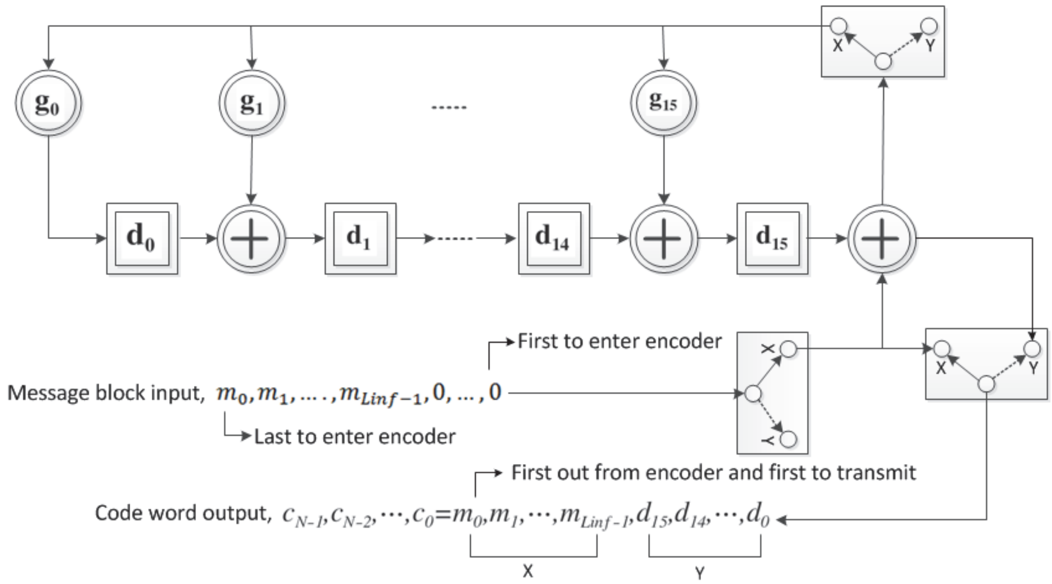

Figure 13-17—Reed-Solomon encoder GF(2^8)

13.3.3.7 Code spreading

13.3.3.7.1 Overview

To increase robustness in frame header and MAC frame body, code spreading is used. The following two categories of spreading are used for HRCP-OOK PHY:

a) PRBS codes by LFSR specified in 13.3.3.7.2 shall be applied for code spreading for HRCP-OOK frame header.

b) Simple bit repetition in which each bit is repeated *n* times, where *n* is the spreading factor, shall be applied for code spreading for HRCP-OOK payloads.

13.3.3.7.2 PRBS generation with LFSR

For HRCP-OOK frame header spreading, spreading factor of length 16 shall be used, and the data bits shall be spread with a PRBS generated using an LFSR, as shown in Figure 13-18. Since the output of the spreader is a factor of L_{SF} larger than the input, the input shall hold while the feedback and output clock.

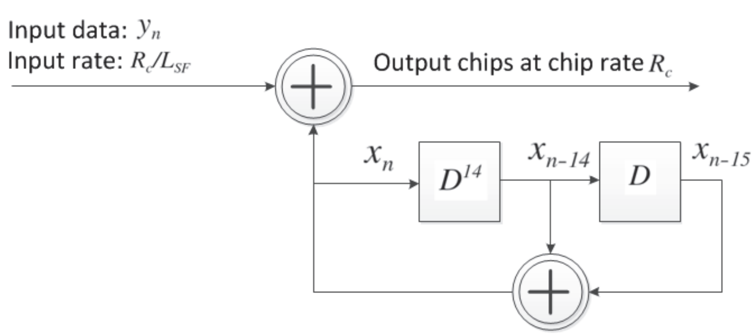

Figure 13-18—PRBS generation by LFSR

The 15-bit seed value of the LFSR shall be: $[x_{-1}, x_{-2}, \ldots, x_{-15}]$ = [0101 0000 0011 111].

13.3.3.8 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as illustrated in Figure 13-18 with $L_{SF} = 1$.

The scrambler shall be used for the MAC header, HCS, and MAC frame body. The PHY preamble, PHY header, and RS bits shall not be scrambled. The polynomial for the PRBS generator used by the scrambler shall be as shown in Equation (13-17):

$$g(D) = 1 + D^{14} + D^{15} \tag{13-17}$$

where

D is a single bit delay element

The polynomial forms not only a maximal length sequence, but also is a primitive polynomial. By the given generator polynomial, the corresponding PRBS, is generated as shown in Equation (13-18):

$${}_{n} = x_{n-14} \oplus x_{n-1}, n = 0, 1, 2, ... \tag{13-18}$$

The initialization sequence is defined by Equation (13-19):

$$x_{init} = [x_{-1}x_{-2}x_{-3}x_{-4}x_{-5}x_{-6}x_{-7}x_{-8}x_{-9}x_{-10}x_{-11}x_{-12}x_{-13}x_{-14}x_{-15}] \tag{13-19}$$

The scrambled data bits, s_n, are obtained by Equation (13-20):

$${}_{n} = b_n \oplus x \tag{13-20}$$

where

b_n represents the unscrambled data bits.

The side-stream descrambler at the receiver shall be initialized with the same initialization vector, x_{init}, used in the transmitter scrambler. The initialization vector is determined from the Scrambler Seed ID field contained in the PHY header of the received frame.

The 15-bit seed value chosen shall be computed from the Scrambler Seed ID field by Equation (13-21):

$$[x_{-1}x_{-2}\ldots x_{-15}] = [11010000101\ S1\ S2\ S3\ S4] \tag{13-21}$$

The seed identifier value is set to 0000 when the PHY is initialized and is incremented in a 4-bit rollover counter for each frame that is sent by the PHY. The value of the seed identifier that is used for the frame is sent in the PHY header.

For a Scrambler Seed ID field set to all zero, the first 16 bits should be as shown in Equation (13-22):

$$[x_0x_1\ldots x_{15}] = [0001111000111010] \tag{13-22}$$

The 15-bit seed value is configured as follows. At the beginning of each PHY frame, the register is cleared, the seed value is loaded, and the first scrambler bit is calculated. The first bit of the data of the MAC header is modulo-2 added with the first scrambler bit, followed by the rest of the bits in the MAC header and MAC frame body. The pilot symbol shall be excluded from the scrambling process.

13.3.4 HRCP-OOK PHY frame format

13.3.4.1 Overview

The HRCP-OOK PHY frame shall be formatted as illustrated in Figure 13-19.

PHY Preamble	Frame Header (Base header)	PHY Payload field

Figure 13-19—HRCP-OOK PHY frame format

The Frame Header field for the PHY frame shall be formatted as illustrated in Figure 13-20.

Frame header (Base header)			
PHY Header	MAC Header	HCS	RS parity bits

Figure 13-20—Frame Header format

The PHY preamble is defined in 13.3.4.2. The MAC header is defined in 6.2. The PHY header is defined in 13.3.4.3.2, and the HCS is defined in 13.3.4.3.3. The PHY Payload field consisting of the MAC frame body and stuff bits is described in 13.3.4.4.

13.3.4.2 PHY Preamble

13.3.4.2.1 General

A PHY preamble shall be added prior to the frame header to aid receiver algorithms related to AGC setting, timing acquisition, frequency offset estimation, frame synchronization, and channel estimation.

The PHY preamble shall be transmitted at the chip rate defined in Table 13-25.

A single mandatory preamble is defined for HRCP-OOK PHY based on the Golay sequence of length 128, denoted $\mathbf{a}_{128}$ and $\mathbf{b}_{128,}$ as shown in Table 13-27. Note that in each hexadecimal-equivalent 4-binary-digit group, the leftmost bit shall be the MSB, and the rightmost bit, the LSB. For example, 3 is denoted as 0011. The order of the octets and bits over the air is the same as defined in 6.1.

Table 13-27—Golay sequence with length 128

Sequence name	Sequence value
$\mathbf{a}_{128}$	0x0536635005C963AFFAC99CAF05C963AF
$\mathbf{b}_{128}$	0x0A396C5F0AC66CA0F5C693A00AC66CA0

Figure 13-21 shows the structure of the HRCP-OOK PHY preamble. The preambles shall be modulated in OOK waveform.

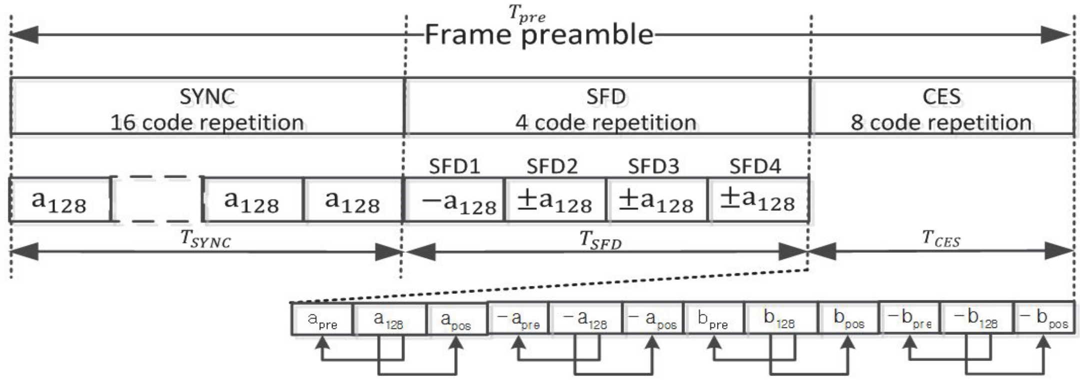

Figure 13-21—HRCP-OOK preamble structure

13.3.4.2.2 SYNC

The SYNC field is used for frame detection and uses a repetition of codes for a higher robustness. The SYNC field shall consist of 16 code repetitions of Golay sequence $\mathbf{a}_{128}$ as given in Table 13-27.

13.3.4.2.3 SFD

The SFD field is used to establish frame timing and to indicate MCS related parameters. The SFD field shall consist of four code repetitions of Golay sequence $\mathbf{a}_{128}$ or $\mathbf{b}_{128}$ as given in Table 13-27. The usage of the four SFD codes shall be as follows: 1 for delimiter and CES selection, 3 for indicating OOK MCS related parameters including number of bonded channels and the spreading factor.

SFD1 is defined as the delimiter and SFD1 also indicates that whether CES is adopted after the SFD4.

- The value of SFD1 is $-\mathbf{a}_{128,}$: SFD 1 indicates delimiter and also indicates CES shall not be adopted.
- The value of SFD1 is $-\mathbf{b}_{128,}$: SFD 1 indicates delimiter and also indicates CES shall be adopted.

SFD2, SFD3, and SFD4 indicate OOK MCS related parameters as shown in Table 13-28.

Table 13-28—SFD for OOK MCS selection

SFD pattern (SFD2, SFD3, SFD4)	Meaning
$+\mathbf{a}_{128}+\mathbf{a}_{128}+\mathbf{a}_{128}$	No channel bonding, $L_{SF} = 1$
$+\mathbf{a}_{128}+\mathbf{a}_{128}-\mathbf{a}_{128}$	Two channel bonding, $L_{SF} = 2$
$+\mathbf{a}_{128}-\mathbf{a}_{128}+\mathbf{a}_{128}$	Two channel bonding, $L_{SF} = 1$
$+\mathbf{a}_{128}-\mathbf{a}_{128}-\mathbf{a}_{128}$	Three channel bonding, $L_{SF} = 2$
$-\mathbf{a}_{128}+\mathbf{a}_{128}+\mathbf{a}_{128}$	Three channel bonding, $L_{SF} = 1$
$-\mathbf{a}_{128}+\mathbf{a}_{128}-\mathbf{a}_{128}$	Four channel bonding, $L_{SF} = 2$

Table 13-28—SFD for OOK MCS selection (*continued*)

SFD pattern (SFD2, SFD3, SFD4)	Meaning
$-\mathbf{a}_{128}-\mathbf{a}_{128}+\mathbf{a}_{128}$	Four channel bonding, L_{SF} = 1
$-\mathbf{a}_{128}-\mathbf{a}_{128}-\mathbf{a}_{128}$ $\sim -\mathbf{b}_{128}-\mathbf{b}_{128}-\mathbf{b}_{128}$	Reserved

In OOK waveform, a negative sequence shall be derived by bit inverting as follows: $-x$ = Bit_Inverting(x), where x is a sequence in the form of binary bit 0 and 1. Bit_Inverting(x) is an operation to invert all the binary bits 0 of a sequence x to 1 and invert all the binary bits 1 of a sequence x to 0.

13.3.4.2.4 CES

If the value of SFD1 is $-\mathbf{b}_{128,}$ and CES is adopted, the CES field shall be constructed from four Golay complementary sequences $\mathbf{a}_{128}$, $-\mathbf{a}_{128}$, $\mathbf{b}_{128}$ and $-\mathbf{b}_{128}$ as shown in Figure 13-22. Each sequence shall be preceded by a cyclic prefix (i.e., a copy of the last 64 bits of the sequence) and followed by a cyclic postfix (i.e., a copy of the first 64 bits of the sequence). The pair of Golay complementary sequences $\mathbf{a}_{128}$ and $\mathbf{b}_{128}$ is given in Table 13-27, where both sequences in the form of binary bit 0 and 1. Another pair of Golay complementary sequences $-\mathbf{a}_{128}$ and $-\mathbf{b}_{128}$ shall be derived from the previous pair of $\mathbf{a}_{128}$ and $\mathbf{b}_{128}$ by bit inverting.

13.3.4.2.5 Preamble Repetition

If channel bonding is used and the number of bonded channels is n, then each subfield of the preamble is repeated n times for higher robustness as depicted in Figure 13-22. By using repetition, duration of preamble, SYNC, SFD, and CES (T_{pre}, T_{SYNC}, T_{SFD}, T_{CES}) remains unchanged when the channel bonding is used.

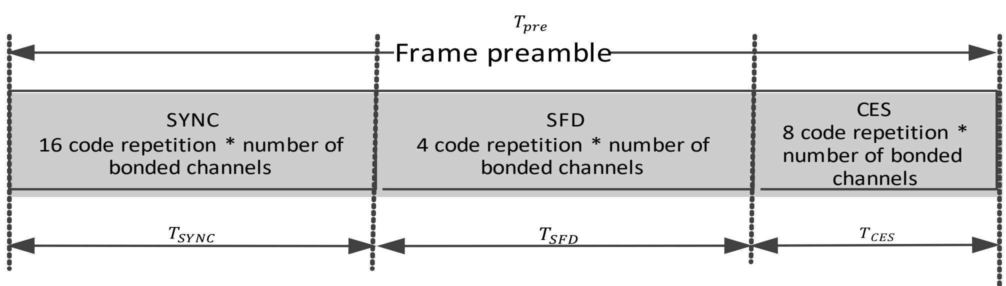

Figure 13-22—Preamble repetition

Figure 13-23 shows an example of preamble repetition when two channel bonding is used.

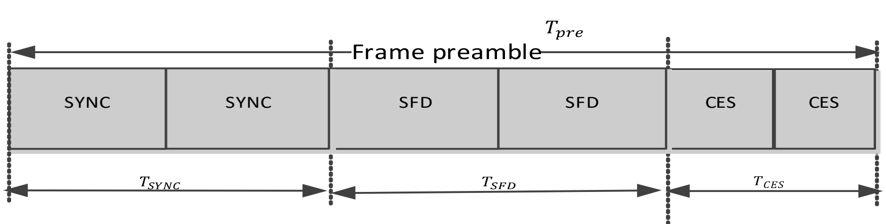

Figure 13-23—Example of Preamble repetition

13.3.4.3 Frame header

13.3.4.3.1 General

A frame header shall be added after the PHY preamble. The frame header conveys information in the PHY and MAC headers necessary for successfully decoding the frame.

The construction of the frame header is shown in Figure 13-24. The detailed process of the construction is as follows:

a) Form the base frame header as follows:
   1) Construct the PHY header based on information provided by the MAC.
   2) Compute the 16 bit HCS using ITU-T CRC-16 over the combined PHY and MAC headers.
   3) Append the HCS to the MAC header.
   4) Scramble the combined MAC header and HCS, as described in 13.3.3.8.
   5) Compute the 128 bit RS parity bits by encoding the concatenation of the PHY header, scrambled MAC header and scrambled HCS into a shortened RS block code, as described in 13.3.3.6. RS(n+16, n) where n is the number of octets in the frame header shall be used.
   6) Form the frame header by concatenating the PHY header, scrambled MAC header, scrambled HCS, and RS parity bits.

   The resulting frame header shall be modulated as shown in Figure 13-24.

b) Spread the frame header with a PRBS generated using an LFSR as described in 13.3.3.7.2, and the spreading factor shall be 16.
c) Map the frame header onto OOK, as described in 13.3.3.5.2.

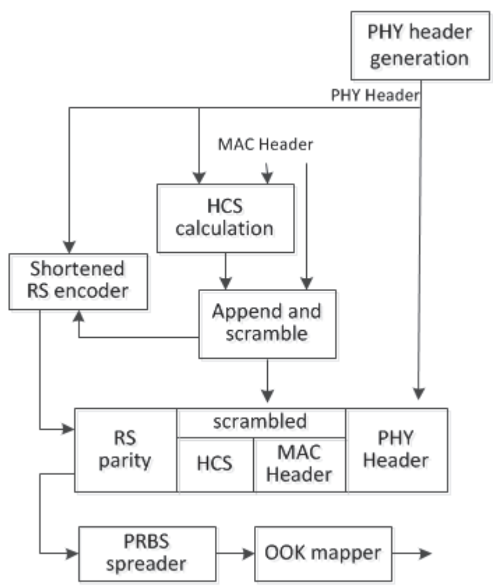

Figure 13-24—Frame header construction process

The block length L_{block} of the frame header shall be 512 chips regardless of the spreading factor L_{SF}. Pilot symbols are not used in the frame header.

The LFSR for the spreader is reset between the header and payload.

13.3.4.3.2 HRCP-OOK PHY header

The HRCP-OOK PHY header shall be formatted as illustrated in Figure 13-25.

bits: b0–b3	b4	b5–b26	b27	b28–b31
Scrambler Seed ID	Aggregation	Frame Length	Pilot Symbol	Reserved

Figure 13-25—PHY header format for HRCP-OOK PHY

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 13.3.3.8.

The Aggregation field shall be set to one if aggregation is used, and it shall be set to zero otherwise.

The Frame Length field shall be an unsigned integer equal to the number of octets in the MAC frame body including frame payload(s), MAC subheader(s) and padding octets in the aggregated frames, and FCS(s), but not including the frame header and the preamble. The frame length includes the length of the security fields such as SECID, SFC, and Integrity Code, if they are present.

The Pilot Symbol field shall be set to one if pilot symbols are used in the frame, and it shall be set to zero otherwise.

13.3.4.3.3 Base header HCS

The combination of the PHY header and MAC header shall be protected with an ITU-T CRC-16 base HCS. The ITU-T CRC-16 is described in 11.2.9.

13.3.4.3.4 Base header FEC

The concatenation of the PHY header, scrambled MAC header, and scrambled HCS shall use shortened systematic RS(n+16,n), a shortened version of RS(240,224), for the FEC, where n is the number of octets in the combined PHY header, MAC header, and HCS. The 128 RS parity bits are appended after the scrambled HCS as shown in Figure 13-24.

13.3.4.4 HRCP-OOK PHY Payload field

13.3.4.4.1 General

The HRCP-OOK PHY Payload field is constructed as shown in Figure 13-26.

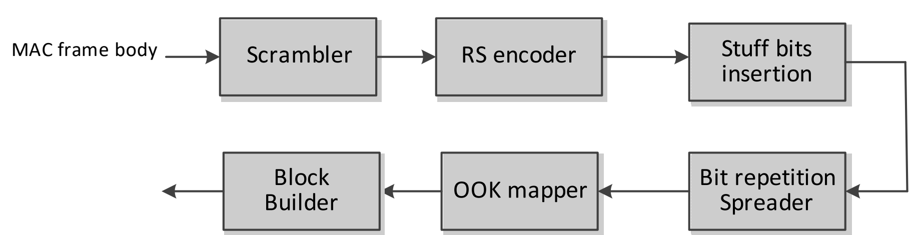

Figure 13-26—HRCP-OOK PHY Payload construction process

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 13.3.3.8.
b) Encode the scrambled MAC frame body as specified in 13.3.3.6. RS(240,224) shall be used for encoding the frame body. The scrambled MAC frame body shall be divided into 224-octet message

blocks and 128 bit RS parity bits shall be appended to each message block. If the size of the last message block is less than 224 octets, RS(n+16, n) where n is the number of octets in the last message block shall be used as described in 13.3.3.6 only for the last message block.

c) Stuff bits are added to the end of the encoded MAC frame body if the number of the encoded data bits is not an integer multiple of N_{CBPB}, which is the length of the data portion in the single block with L_{SF} = 1. N_{CBPB} = 508 if pilot symbols are used and N_{CBPB} = 512 otherwise. In this case, add stuff bits to the end of the encoded MAC frame body until the number of the encoded data bits including the added stuff bits reaches an integer multiple of N_{CBPB}. The stuff bits shall be set to zero and then scrambled using the continuation of the scrambler sequence that scrambled the MAC frame body in 13.3.3.8.

d) If L_{SF} = 2, spread the encoded and scrambled MAC frame body using bit repetition with L_{SF} = 2 in which each bit is repeated twice.

e) Map the resulting MAC frame body onto OOK, as described in 13.3.3.5.2.

f) Build blocks from the resulting MAC frame body as specified in 13.3.4.4.6.

13.3.4.4.2 HRCP-OOK PHY Payload scrambling

The HRCP-OOK PHY payload shall use the scrambling process defined in 13.3.3.8.

13.3.4.4.3 Modulation

Modulation for the MAC frame body is defined in 13.3.3.5.2.

13.3.4.4.4 FEC

FEC for the MAC frame body is defined in 13.3.3.6.

13.3.4.4.5 Code spreading

Simple bit repetition with spreading factor 2 in which each bit is repeated twice shall be applied for code spreading for the MAC frame body.

13.3.4.4.6 Blocks and Pilot symbol

Pilot symbols may be used in HRCP-OOK PHY for timing tracking, compensation for clock drift, and compensation for frequency offset error. Furthermore, pilot symbols act as a known cyclic prefix and enable frequency domain equalization if desired. In frequency domain equalization, the data is handled in the unit of blocks.

In HRCP-OOK data payload, if pilot symbols are used, the transmit symbols shall be divided into block of length $N = 508 \times L_{SF}$, where L_{SF} is the spreading factor. This transmit symbol block shall be preceded by pilot symbol as described in Figure 13-27.

If pilot symbols are not used, the transmit symbols shall be divided into block of length $N = 512 \times L_{SF}$.

The pilot symbols consist of a sequence of length $N_p = 4 \times L_{SF}$. The pilot symbols for L_{SF}=1 and L_{SF}=2 shall be chosen according to Table 13-29.

.

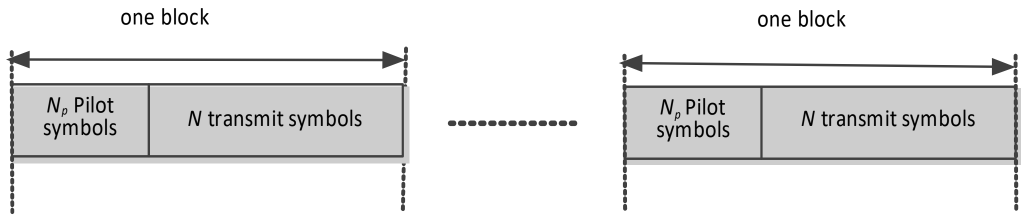

Figure 13-27—HRCP-OOK frame format with pilot symbols

Table 13-29—OOK pilot symbols

Spreading Factor, L_{SF}	Pilot symbols
1	1010
2	11001100

The pilot symbol shall be modulated with OOK.

13.3.5 Transmitter specifications

13.3.5.1 Error Vector Magnitude

Eye opening for OOK is described in G.7.

13.3.5.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ maximum.

13.3.5.3 Symbol rate

The OOK PHY shall be capable of transmitting at the chip rate, as defined in Table 13-25, to within $\pm 30 \times 10^{-6}$.

The MAC parameter, *pClockAccuracy*, shall be $\pm 30 \times 10^{-6}$.

13.3.6 Receiver specifications

13.3.6.1 Error rate criterion

The error rate criterion shall be a FER of less than 8% with a frame payload length of 2^{14} octets. The error rate should be determined at the PHY SAP interface.

13.3.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 13.3.6.1 is met. A compliant DEV that implements the HRCP-OOK PHY shall achieve at least the reference sensitivity listed in Table 13-30.

Table 13-30—Reference sensitivity levels for MCS

Mode	MCS identifier	Receiver sensitivity
Mode 1 (No channel bonding)	1 (L_{SF} = 1)	–59 dBm
Mode 2 (2 channel bonding)	0 (L_{SF} = 2)	–59 dBm
	1 (L_{SF} = 1)	–54 dBm
Mode 3 (3 channel bonding)	0 (L_{SF} = 2)	–56 dBm
	1 (L_{SF} = 1)	–47 dBm
Mode 4 (4 channel bonding)	0 (L_{SF} = 2)	–54 dBm
	1 (L_{SF} = 1)	–45 dBm

13.3.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 13.3.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

13.3.7 PHY layer timing

13.3.7.1 General

The values for the PHY layer timing parameters are defined in Table 13-31.

Table 13-31—PHY layer timing parameters

PHY parameter	Value	Subclause
pPHYSIFSTime	0.2 µs, 2.0 µs, 2.5 µs (default)	13.3.7.4
pPHYChannelSwitchTime	100 µs	13.3.7.6

13.3.7.2 IFS

A conforming implementation shall support the IFS parameters, as described in 7.6.2, given in Table 13-32.

Table 13-32—IFS parameters

MAC parameter		Corresponding PHY parameter	Definition
SIFS		*pPHYSIFSTime*	13.3.7.4
RIFS	PRC	2 × *pPHYSIFSTime* + 4.36 µs	7.6.2
	DEV	4 × *pPHYSIFSTime* + 13.09 µs	

NOTE—The RIFS is derived from the SIFS and the length of time it takes to transmit the Stk-ACK frame(s) at the lowest OOK PHY rate. The RIFS value of the PRC is equal to 2 × *pPHYSIFSTime* plus 1 × Stk-ACK transmission time. The RIFS value of the DEV is equal to 2 × *pPHYSIFSTime* plus 2 × Stk-ACK transmission time plus the RIFS value of the PRC.

13.3.7.3 Receive-to-transmit turnaround time

The receive to transmit turnaround time shall be *pPHYSIFSTime*. The receive to transmit turnaround time shall be measured at the air interface from the trailing edge of the last symbol received until the first symbol of the PHY preamble is present at the air interface.

13.3.7.4 Transmit-to-receive turnaround time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*.

13.3.7.5 Time between transmission

The minimum time between the end of the last transmitted frame and the beginning of the retransmitted frame shall be less than RIFS time specified in Table 13-32. A PRC shall use the shorter RIFS than that of a DEV as defined in Table 13-32.

13.3.7.6 Channel switch

The *channel switch time* is defined as the time from the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pPHYChannelSwitchTime*.

13.3.8 PHY management for HRCP-OOK PHY

13.3.8.1 General

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the HRCP-OOK PHY layer of a DEV.

13.3.8.2 Maximum frame size

The maximum frame length allowed, *pMaxFrameBodySize*, shall be 2 099 200 octets. This total includes the MAC frame body including frame payload(s), MAC subheader(s) and padding octets in the aggregated frames, and FCS(s), but not including the frame header and the preamble. The maximum frame length also does not include the stuff bits. The maximum frame length includes the length of the security fields such as SECID, SFC, and Integrity Code, if they are present.

13.3.8.3 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 2 097 152 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2 097 152 octets minus the security overhead, as defined in 6.3.1.4, 6.3.4.2, 6.3.6.2, and 6.3.8.2.

13.3.8.4 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed with the HRCP-OOK PHY shall be 512 octets.

14. PHY specification for THz

14.1 General requirements

14.1.1 Overview

A compliant THz PHY shall implement at least one of the following PHY modes:

— THz single carrier mode PHY (THz-SC PHY), as defined in 14.2
— THz on-off keying mode PHY (THz-OOK PHY), as defined in 14.3

Unless otherwise stated, in all figures in this clause, the ordering of the octets and bits as they are presented for the THz PHY is the same as defined in 6.1.

14.1.2 RF power measurements

Unless otherwise stated, all RF power measurements for the purpose of this standard are corrected to compensate for the antenna gain in the implementation. The gain of the antenna is the maximum estimated gain given in the data sheet of the antenna, as provided by the manufacturer.

14.1.3 RF channelization

The THz PHY uses the channels defined in Table 14-1.

Table 14-1—THz PHY channelization

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
1	2.16	252.72	253.8	254.88
2	2.16	254.88	255.96	257.04
3	2.16	257.04	258.12	259.2
4	2.16	259.2	260.28	261.36
5	2.16	261.36	262.44	263.52
6	2.16	263.52	264.6	265.68
7	2.16	265.68	266.76	267.84
8	2.16	267.84	268.92	270.0
9	2.16	270.0	271.08	272.16
10	2.16	272.16	273.24	274.32
11	2.16	274.32	275.4	276.48
12	2.16	276.48	277.56	278.64
13	2.16	278.64	279.72	280.8
14	2.16	280.8	281.88	282.96
15	2.16	282.96	284.04	285.12

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
16	2.16	285.12	286.2	287.28
17	2.16	287.28	288.36	289.44
18	2.16	289.44	290.52	291.6
19	2.16	291.6	292.68	293.76
20	2.16	293.76	294.84	295.92
21	2.16	295.92	297.0	298.08
22	2.16	298.08	299.16	300.24
23	2.16	300.24	301.32	302.4
24	2.16	302.4	303.48	304.56
25	2.16	304.56	305.64	306.72
26	2.16	306.72	307.8	308.88
27	2.16	308.88	309.96	311.04
28	2.16	311.04	312.12	313.2
29	2.16	313.2	314.28	315.36
30	2.16	315.36	316.44	317.52
31	2.16	317.52	318.6	319.68
32	2.16	319.68	320.76	321.84
33	4.32	252.72	254.88	257.04
34	4.32	257.04	259.2	261.36
35	4.32	261.36	263.52	265.68
36	4.32	265.68	267.84	270.0
37	4.32	270.0	272.16	274.32
38	4.32	274.32	276.48	278.64
39	4.32	278.64	280.8	282.96
40	4.32	282.96	285.12	287.28
41	4.32	287.28	289.44	291.6
42	4.32	291.6	293.76	295.92
43	4.32	295.92	298.08	300.24
44	4.32	300.24	302.4	304.56
45	4.32	304.56	306.72	308.88
46	4.32	308.88	311.04	313.2
47	4.32	313.2	315.36	317.52
48	4.32	317.52	319.68	321.84

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
49	8.64	252.72	257.04	261.36
50	8.64	261.36	265.68	270.0
51	8.64	270.0	274.32	278.64
52	8.64	278.64	282.96	287.28
53	8.64	287.28	291.6	295.92
54	8.64	295.92	300.24	304.56
55	8.64	304.56	308.88	313.2
56	8.64	313.2	317.52	321.84
57	12.96	252.72	259.2	265.68
58	12.96	265.68	272.16	278.64
59	12.96	278.64	285.12	291.6
60	12.96	291.6	298.08	304.56
61	12.96	304.56	311.04	317.52
62	17.28	252.72	261.36	270.0
63	17.28	270.0	278.64	287.28
64	17.28	287.28	295.92	304.56
65	17.28	304.56	313.2	321.84
66	25.92	252.72	265.68	278.64
67	25.92	278.64	291.6	304.56
68	51.84	252.72	278.64	304.56
69	69.12	252.72	287.28	321.84
70	2.16	321.84	322.92	324
71	2.16	324	325.08	326.16
72	2.16	326.16	327.24	328.32
73	2.16	328.32	329.4	330.48
74	2.16	330.48	331.56	332.64
75	2.16	332.64	333.72	334.8
76	2.16	334.8	335.88	336.96
77	2.16	336.96	338.04	339.12
78	2.16	339.12	340.2	341.28
79	2.16	341.28	342.36	343.44
80	2.16	343.44	344.52	345.6
81	2.16	345.6	346.68	347.76

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
82	2.16	347.76	348.84	349.92
83	2.16	349.92	351	352.08
84	2.16	352.08	353.16	354.24
85	2.16	354.24	355.32	356.4
86	2.16	356.4	357.48	358.56
87	2.16	358.56	359.64	360.72
88	2.16	360.72	361.8	362.88
89	2.16	362.88	363.96	365.04
90	2.16	365.04	366.12	367.2
91	2.16	367.2	368.28	369.36
92	2.16	369.36	370.44	371.52
93	2.16	371.52	372.6	373.68
94	2.16	373.68	374.76	375.84
95	2.16	375.84	376.92	378
96	2.16	378	379.08	380.16
97	2.16	380.16	381.24	382.32
98	2.16	382.32	383.4	384.48
99	2.16	384.48	385.56	386.64
100	2.16	386.64	387.72	388.8
101	2.16	388.8	389.88	390.96
102	2.16	390.96	392.04	393.12
103	2.16	393.12	394.2	395.28
104	2.16	395.28	396.36	397.44
105	2.16	397.44	398.52	399.6
106	2.16	399.6	400.68	401.76
107	2.16	401.76	402.84	403.92
108	2.16	403.92	405	406.08
109	2.16	406.08	407.16	408.24
110	2.16	408.24	409.32	410.4
111	2.16	410.4	411.48	412.56
112	2.16	412.56	413.64	414.72
113	2.16	414.72	415.8	416.88
114	2.16	416.88	417.96	419.04

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
115	2.16	419.04	420.12	421.2
116	2.16	421.2	422.28	423.36
117	2.16	423.36	424.44	425.52
118	2.16	425.52	426.6	427.68
119	2.16	427.68	428.76	429.84
120	2.16	429.84	430.92	432
121	2.16	432	433.08	434.16
122	2.16	434.16	435.24	426.32
123	2.16	426.32	437.4	438.48
124	2.16	438.48	439.56	440.64
125	2.16	440.64	441.72	442.8
126	2.16	442.8	443.88	444.96
127	2.16	444.96	446.04	447.12
128	2.16	447.12	448.2	449.28
129	4.32	321.84	324	326.16
130	4.32	326.16	328.32	330.48
131	4.32	330.48	332.64	334.8
132	4.32	334.8	336.96	339.12
133	4.32	339.12	341.28	343.44
134	4.32	343.44	345.6	347.76
135	4.32	347.76	349.92	352.08
136	4.32	352.08	354.24	356.4
137	4.32	356.4	358.56	360.72
138	4.32	360.72	362.88	365.04
139	4.32	365.04	367.2	369.36
140	4.32	369.36	371.52	373.68
141	4.32	373.68	375.84	378
142	4.32	378	380.16	382.32
143	4.32	382.32	384.48	386.64
144	4.32	386.64	388.8	390.96
145	4.32	390.96	393.12	395.28
146	4.32	395.28	397.44	399.6
147	4.32	399.6	401.76	403.92

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
148	4.32	403.92	406.08	408.24
149	4.32	408.24	410.4	412.56
150	4.32	412.56	414.72	416.88
151	4.32	416.88	419.04	421.2
152	4.32	421.2	423.36	425.52
153	4.32	425.52	427.68	429.84
154	4.32	429.84	432	434.16
155	4.32	434.16	436.32	438.48
156	4.32	438.48	440.64	442.8
157	4.32	442.8	444.96	447.12
158	8.64	326.16	330.48	334.8
159	8.64	334.8	339.12	343.44
160	8.64	343.44	347.76	352.08
161	8.64	352.08	356.4	360.72
162	8.64	360.72	365.04	369.36
163	8.64	369.36	373.68	378
164	8.64	378	382.32	386.64
165	8.64	386.64	390.96	395.28
166	8.64	395.28	399.6	403.92
167	8.64	403.92	408.24	412.56
168	8.64	412.56	416.88	421.2
169	8.64	421.2	425.52	429.84
170	8.64	429.84	434.16	438.48
171	8.64	438.48	442.8	447.12
172	12.96	321.84	328.32	334.8
173	12.96	334.8	341.28	347.76
174	12.96	347.76	354.24	360.72
175	12.96	360.72	367.2	373.68
176	12.96	373.68	380.16	386.64
177	12.96	386.64	393.12	399.6
178	12.96	399.6	406.08	412.56
179	12.96	412.56	419.04	425.52
180	12.96	425.52	432	438.48

Table 14-1—THz PHY channelization (*continued*)

CHNL_ID	Bandwidth (GHz)	Start frequency[a] (GHz)	Center frequency (GHz)	Stop frequency[a] (GHz)
181	12.96	438.48	444.96	451.44
182	17.28	321.84	330.48	339.12
183	17.28	339.12	347.76	356.4
184	17.28	356.4	365.04	373.68
185	17.28	373.68	382.32	390.96
186	17.28	390.96	399.6	408.24
187	17.28	408.24	416.88	425.52
188	17.28	425.52	434.16	442.8
189	25.92	304.56	317.52	330.48
190	25.92	330.48	343.44	356.4
191	25.92	356.4	369.36	382.32
192	25.92	382.32	395.28	408.24
193	25.92	408.24	421.2	434.16
194	34.56	252.72	270	287.28
195	34.56	287.28	304.56	321.84
196	34.56	321.84	339.12	356.4
197	34.56	356.4	373.68	390.96
198	34.56	390.96	408.24	425.52
199	51.84	304.56	330.48	356.4
200	51.84	356.4	382.32	408.24
201	69.12	356.4	390.96	425.52

[a]The start and stop frequencies are nominal values. The frequency spectrum of the transmitted signal shall conform to the transmit PSD mask for the PHY mode.

The bandwidths of all channels are integer multiples of 2.16 GHz. The center frequencies for channels having a CHNL_ID in the range of 33 to 69 and 129 to 201 are integer multiples of 2.16 GHz. Of the channels having CHNL_ID equal to 41, 137, and 149, one shall be defined as the default channel, as described in 6.4.16.

14.1.4 Transmit PSD mask

The transmitted spectrum for the THz-SC PHY shall adhere to the transmit PSD mask shown in Figure 14-1. The transmitted spectrum for the THz-OOK PHY shall adhere to the transmit PSD mask shown in Figure 14-2. The additional single line spectrum of 40 dB above the 0 dB line in Figure 14-2 is within the frequency band of (–6 MHz, +6 MHz) from the carrier frequency. For all transmit mask measurements, the resolution bandwidth is set to 3 MHz and the video bandwidth is set to 300 kHz.

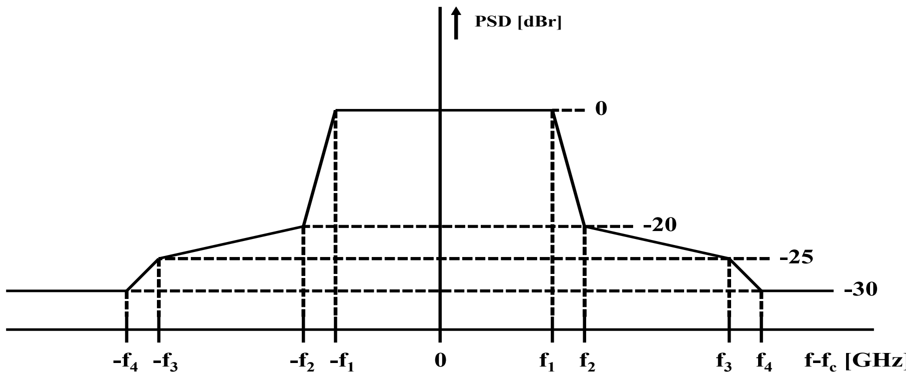

Figure 14-1—Transmit spectral mask for THz-SC PHY

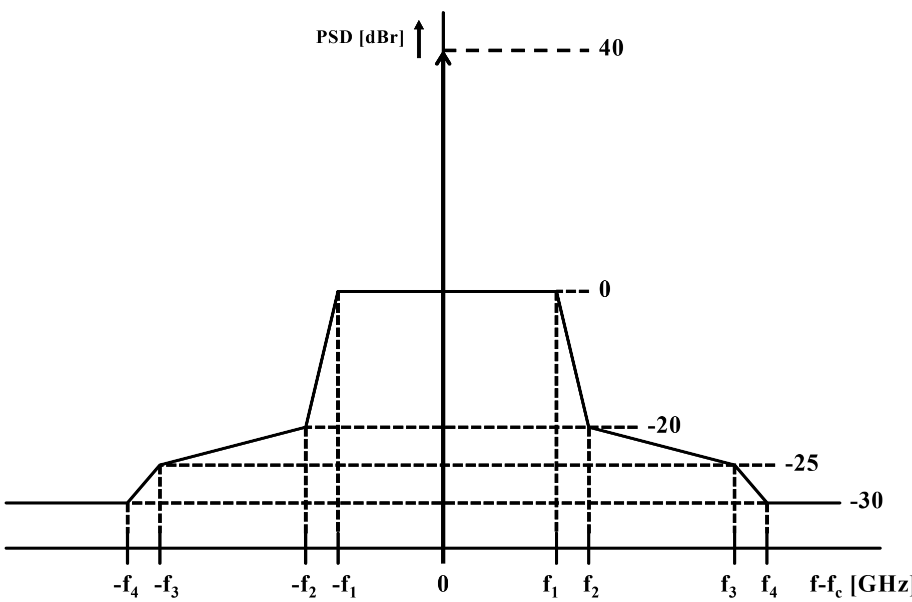

Figure 14-2—Transmit spectral mask for THz-OOK PHY

The parameters of the PSDs indicated in Figure 14-1 and Figure 14-2 are defined in Table 14-2.

Table 14-2—Transmit spectrum mask parameters

Channel bandwidth (GHz)	f_1 (GHz)	f_2 (GHz)	f_3 (GHz)	f_4 (GHz)
2.160	0.94	1.10	1.60	2.20
4.320	2.02	2.18	2.68	3.28
8.640	4.18	4.34	4.84	5.44
12.960	6.34	6.50	7.00	7.60
17.280	8.50	8.66	9.16	9.76
25.920	12.82	12.98	13.48	14.08
34.560	17.14	17.30	17.80	18.40
51.840	25.78	25.94	26.44	27.04
69.120	34.42	34.58	35.08	35.68

14.1.5 Error vector magnitude calculation

The error vector magnitude (EVM) for the THz PHY shall be measured and calculated using the method defined in 12.1.8.1.

14.1.6 THz PHY management

14.1.6.1 General

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the THz PHY layer of a PRDEV.

14.1.6.2 THz PHY PIB

The PHY dependent PIB values for the THz PHY are given in Table 14-3.

14.1.6.3 Maximum frame size

The maximum frame length allowed, *pMaxFrameBodySize*, shall be 2 099 200 octets. This total includes the MAC frame body, but not the PHY preamble or base header (PHY header, MAC header, and HCS). The maximum frame length also does not include the stuff bits. See Table 13-9 for details regarding the relationship between the payload size and the size of the MAC frame body.

14.1.6.4 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 2 097 152 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2 097 152 octets minus the security overhead, as defined in 6.3.1.4, 6.3.4.2, 6.3.6.2, and 6.3.8.2.

Table 14-3—PHY PIB characteristics group parameters

Managed Object	Octets	Definition	Access
phyType	1	0x03 = THz PHY	Read/Write
phyMode	1	bit 1 = THz-SC PHY bit 2 = THz-OOK PHY bit 3–8 = Reserved A bit is set to one if the associated PHY is supported and is set to zero otherwise.	Read/Write
phyRegDomainsSupported	Variable	One octet for each regulatory domain supported, as defined for *phyCurrentRegDomain*.	Read/Write
phyCurrentRegDomain	1	0x00 = European Telecommunications Standards Institute (ETSI) 0x01 = Federal Communications Commission (FCC) 0x02 = Industry Canada (IC) 0x03 = Association of Radio Industries and Businesses (ARIB)	Read/Write
phyDataRateVector	Variable	Two octets for each supported MCS. The MSB indicates the THz PHY mode: MSB 0 = 0 for THz-SC PHY and MSB 0 = 1 for THz OOK PHY. MSB 1–4 indicate the bandwidth identifier described in Table 14-12. For the THz-SC PHY mode, the four LSBs indicate the MCS supported for that mode using the encoding described in Table 14-11. For the THz-OOK PHY mode, the two LSBs indicate the MCS supported for that mode using the encoding described in Table 14-19.	Read/Write
phyChannelBandwidthSupported	2	Bits b33–b42 in the PRC Capability field, as defined 6.4.16.	Read/Write
phySpectrumPartSupported	12	Bits b43–b133 in the PRC Capability field, as defined 6.4.16.	Read/Write
phyCurrentChannel	1	Indicates the channel that is currently being used, as defined in 14.1.3.	Read/Write
phyFrameLengthMax	2	The value of *pMaxFrameBodySize*.	Read/Write

14.1.6.5 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed shall be 2048 octets.

14.2 THz-SC PHY

14.2.1 General

The THz-SC PHY is designed for extremely high PHY-SAP payload bit rates of 100 Gb/s using multiple bandwidths. Higher data rates are achievable, depending on the combination of modulation, bandwidth, and coding.

14.2.2 Channelization of THz-SC PHY

The RF channels are defined in Table 14-1. A compliant implementation shall support at least one of the channels with a default CHNL_ID, as defined in 14.1.3.

The *phyCurrentChannel* is the CHNL_ID of the current channel. For the purpose of the Remote Scan Request and Remote Scan Response commands, as described in 6.5.8.4 and 6.5.8.5, respectively, the Channel Index field is the CHNL_ID in Table 14-1 in 14.1.3.

14.2.3 Modulation and coding

14.2.3.1 Modulation

After channel encoding and spreading, the bits shall be inserted into the constellation mapper.

The constellations of π/2 BPSK, π/2 QPSK, and π/2 8-PSK used for the THz-SC PHY are the same as illustrated in Figure 12-10 (a), (b), and (c), respectively, in 12.2.3.5. The constellations of 16-QAM and 64-QAM used for the THz-SC PHY are the same as illustrated in Figure 12-29 in 12.3.3.6.

The constellation diagrams of π/2 8-APSK, 16-APSK, and 32-APSK are shown in Figure 14-3, Figure 14-4, and Figure 14-5, respectively.

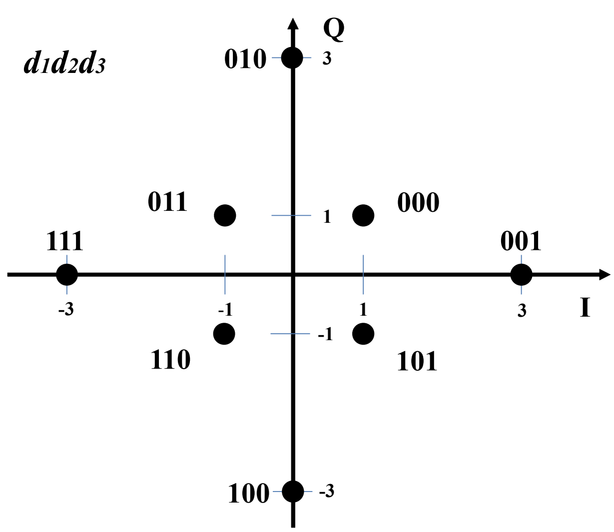

Figure 14-3—π/2 8-APSK

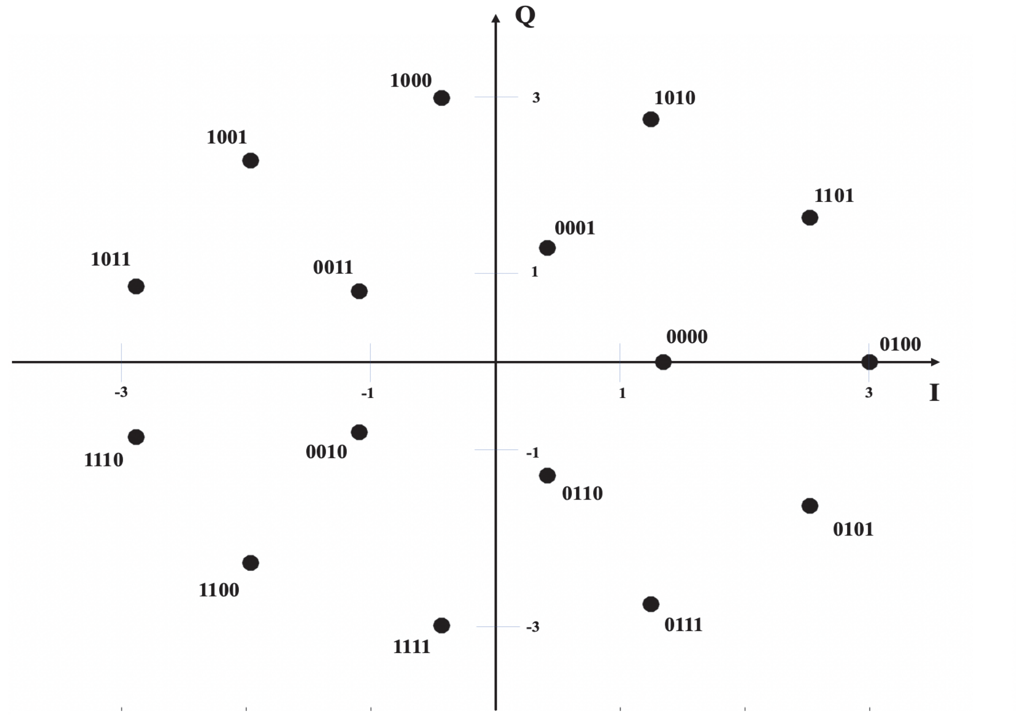

Figure 14-4—16-APSK

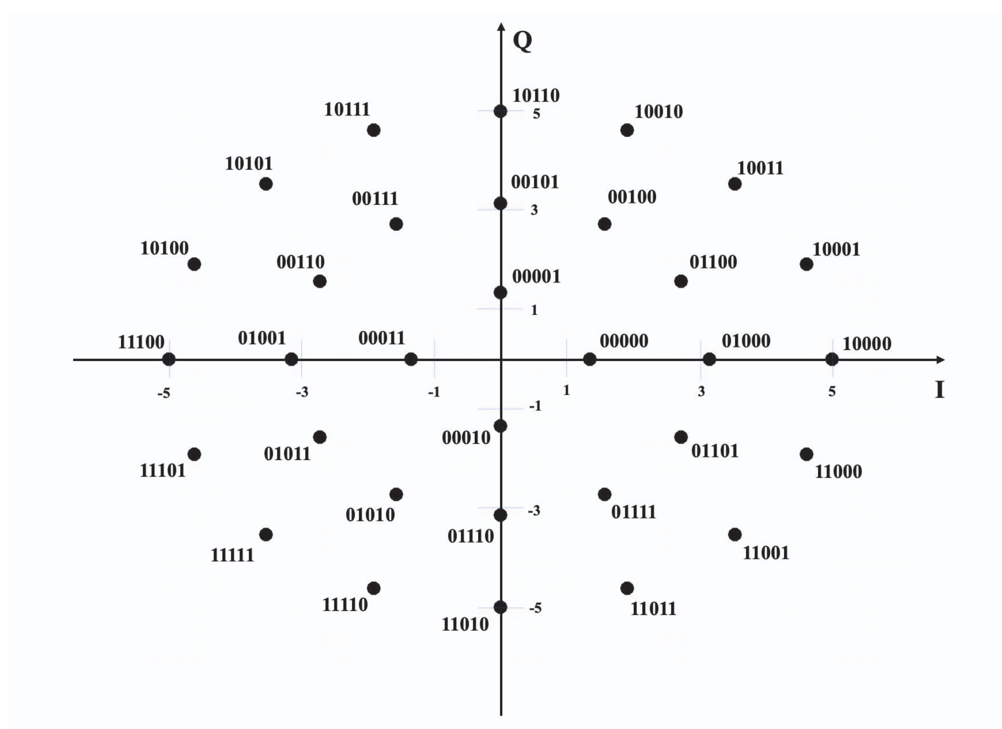

Figure 14-5—32-APSK

The π/2 8-APSK shall encode 3 bits per symbol, with input bit d_1 being the earliest in the stream. The π/2-rotation is performed in the same manner as in 12.2.3.5.2. The coordinates of the outer ring of the 16-APSK are given by Equation (14-1):

$$\left(3\cos\left(\frac{2\pi n}{11}\right), 3\sin\left(\frac{2\pi n}{11}\right)\right) \text{ where } n \in [0, 10] \tag{14-1}$$

The coordinates of the inner ring of the 16-APSK are given by Equation (14-2):

$$\left(1.35\cos\left(\frac{2\pi n}{5}\right), 1.35\sin\left(\frac{2\pi n}{5}\right)\right) \text{ where } n \in [0, 4] \tag{14-2}$$

For the 32-APSK, the coordinates of the outermost ring are given by Equation (14-3):

$$\left(5\cos\left(\frac{2\pi n}{16}\right), 5\sin\left(\frac{2\pi n}{16}\right)\right) \text{ where } n \in [0, 15] \tag{14-3}$$

The coordinates of the middle ring of the 32-APSK are given by Equation (14-4):

$$\left(3.15\cos\left(\frac{2\pi n}{12}\right), 3.15\sin\left(\frac{2\pi n}{12}\right)\right) \text{ where } n \in [0, 11] \tag{14-4}$$

The coordinates of the innermost ring of the 32-APSK are given by Equation (14-5):

$$\left(1.35\cos\left(\frac{2\pi n}{4}\right), 1.35\sin\left(\frac{2\pi n}{4}\right)\right) \text{ where } n \in [0, 3] \tag{14-5}$$

The normalization factors are shown in Table 14-4. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization can be used as long as the DEV conforms to the modulation accuracy requirements described in 14.2.5.1.

Table 14-4—Normalization factors

Modulation	Normalization factor
π/2 QPSK	1
π/2 8-PSK	1
π/2 8-APSK	$(\sqrt{2})/(\sqrt{11})$
16-APSK	$(\sqrt{2})/(\sqrt{13})$
32-APSK	$(\sqrt{2})/(\sqrt{30})$
16-QAM	$1/(\sqrt{10})$
64-QAM	$1/(\sqrt{42})$

All modulation schemes are used for payload, and π/2 BPSK is also used for preamble and header sequences. The modulations of π/2 BPSK and π/2 QPSK are mandatory for THz-SC PHY; other modulations are optional.

14.2.3.2 Forward error correction (FEC)

The FEC schemes are specified in this subclause. Supporting the following two LDPC codes is mandatory for the THz-SC PHY: a rate-14/15 LDPC(1440,1344) code, defined in 12.2.3.6.4, and a rate-11/15 LDPC(1440,1056) code, defined in 13.2.3.6.

14.2.3.3 MCS dependent parameters

The MCS dependent parameters shall be set according to Table 14-5 and Table 14-6. The chip rate for all THz-SC PHY MCSs is given in Table 14-8. The data rates in the table are approximate values.

A block length for THz-SC PHY shall be 64 chips. A block is formed according to 13.2.4.5.1. The PW length, where a PW is defined in 13.2.4.5.1, for the THz-SC PHY shall be 0 chips or 8 chips. The PW length of 8 chips is mandatory and that of 0 chips is optional.

Table 14-5—MCS dependent parameters for the THz-SC PHY for bandwidths 2.16 GHz –17.28 GHz

MCS identifier	Modulation	FEC rate	Bandwidth 2.16 GHz		Bandwidth 4.32 GHz		Bandwidth 8.64 GHz		Bandwidth 12.96 GHz		Bandwidth 17.28 GHz	
			Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)	
			without PW	with PW	without PW	with PW	without PW	with PW	without PW	with PW	without PW	with PW
0	BPSK	11/15	1.29	1.13	2.58	2.26	5.16	4.52	7.74	6.78	10.33	9.04
1	BPSK	14/15	1.64	1.44	3.29	2.87	6.57	5.75	9.86	8.62	13.14	11.50
2	QPSK	11/15	2.58	2.26	5.16	4.52	10.33	9.03	15.49	13.55	20.65	18.07
3	QPSK	14/15	3.29	2.87	6.57	5.75	13.14	11.50	19.71	17.25	26.28	23.00
4	8-PSK	11/15	3.87	3.39	7.74	6.78	15.49	13.55	23.23	20.33	30.98	27.11
5	8-PSK	14/15	4.93	4.31	9.86	8.62	19.71	17.25	29.57	25.87	39.42	34.50
6	8-APSK	11/15	3.87	3.39	7.74	6.78	15.49	13.55	23.23	20.33	30.98	27.11
7	8-APSK	14/15	4.93	4.31	9.86	8.62	19.71	17.25	29.57	25.87	39.42	34.50
8	16-QAM	11/15	5.16	4.52	10.33	9.03	20.65	18.07	30.98	27.10	41.30	36.14
9	16-QAM	14/15	6.57	5.75	13.14	11.50	26.28	23.00	39.42	34.50	52.57	45.99
10	64-QAM	11/15	7.74	6.78	15.49	13.55	30.98	27.10	46.46	40.66	61.95	54.21
11	64-QAM	14/15	9.86	8.62	19.71	17.25	39.42	34.50	59.14	51.74	78.85	68.99
12	16-APSK	11/15	5.16	4.52	10.33	9.03	20.65	18.07	30.98	27.10	41.30	36.14
13	16-APSK	14/15	6.57	5.75	13.14	11.50	26.28	23.00	39.42	34.50	52.57	45.99

Table 14-5—MCS dependent parameters for the THz-SC PHY for bandwidths 2.16 GHz (*continued*)–17.28 GHz (*continued*)

MCS identifier	Modulation	FEC rate	Bandwidth 2.16 GHz		Bandwidth 4.32 GHz		Bandwidth 8.64 GHz		Bandwidth 12.96 GHz		Bandwidth 17.28 GHz	
			Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)	
			without PW	with PW	without PW	with PW	without PW	with PW	without PW	with PW	without PW	with PW
14	32-APSK	11/15	6.45	5.65	12.91	11.29	25.81	22.59	38.73	33.88	51.63	45.18
15	32-APSK	14/15	8.21	7.19	16.43	14.38	32.85	28.75	49.28	43.13	65.71	57.49

Table 14-6—MCS dependent parameters for the THz-SC PHY for bandwidths 25.92 GHz –69.12 GHz

MCS - identifier	Modulation	FEC rate	Bandwidth 25.92 GHz		Bandwidth 34.56 GHz		Bandwidth 51.84 GHz		Bandwidth 69.12 GHz	
			Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)		Data rate (Gb/s)	
			without PW	with PW	without PW	with PW	without PW	with PW	without PW	with PW
0	BPSK	11/15	15.49	13.55	20.66	18.08	30.98	27.11	41.30	36.14
1	BPSK	14/15	19.71	17.25	26.28	23.00	39.42	34.50	52.56	45.99
2	QPSK	11/15	30.98	27.10	41.30	36.14	61.95	54.21	82.60	72.28
3	QPSK	14/15	39.42	34.50	52.56	46.00	78.85	68.99	105.13	91.99
4	8-PSK	11/15	46.47	40.66	61.96	54.22	92.93	81.32	123.91	108.42
5	8-PSK	14/15	59.13	51.74	78.84	69.00	118.27	103.49	157.69	137.98
6	8-APSK	11/15	46.47	40.66	61.96	54.22	92.93	81.32	123.91	108.42
7	8-APSK	14/15	59.13	51.74	78.84	69.00	118.27	103.49	157.69	137.98
8	16-QAM	11/15	61.95	54.21	82.60	72.28	123.90	108.42	165.21	144.55
9	16-QAM	14/15	78.85	68.99	105.14	91.98	157.70	137.98	210.26	183.98
10	64-QAM	11/15	92.93	81.31	123.90	108.42	185.86	162.62	247.81	216.83
11	64-QAM	14/15	118.27	103.49	157.70	137.98	236.54	206.98	315.39	275.97
12	16-APSK	11/15	61.95	54.21	82.60	72.28	123.90	108.42	165.21	144.55
13	16-APSK	14/15	78.85	68.99	105.14	91.98	157.70	137.98	210.26	183.98

Table 14-6—MCS dependent parameters for the THz-SC PHY for bandwidths 25.92 GHz (*continued*)–69.12 GHz (*continued*)

MCS - identi-fier	Modula-tion	FEC rate	Bandwidth 25.92 GHz Data rate (Gb/s)		Bandwidth 34.56 GHz Data rate (Gb/s)		Bandwidth 51.84 GHz Data rate (Gb/s)		Bandwidth 69.12 GHz Data rate (Gb/s)	
			with out PW	with PW	with out PW	with PW	with out PW	with PW	with out PW	with PW
14	32-APSK	11/15	77.44	67.76	103.25	90.35	154.88	135.53	206.51	180.69
15	32-APSK	14/15	98.56	86.24	131.43	114.98	197.13	172.48	262.83	229.98

14.2.3.4 Header dependent parameters

The header dependent parameters shall be set according to Table 14-7. The header rate is proportional to the bandwidth used. The headers use an extended Hamming (EH) code, as defined in 13.2.4.3.4.

Table 14-7—Header rate dependent parameters for the THz-SC PHY

Bandwidth (GHz)	Header rate (Mb/s)	Modulation scheme	Spreading factor, L_{SF}	FEC	PW length (chips), L_{PW}	Code bits per block, L_{CBPS}	Number of occupied blocks, N_{block_hdr}	Number of stuff bits, L_{STUFF}
2.160	162	π/2 BPSK	4	EH	8	14	19	40
4.320	324	π/2 BPSK	4	EH	8	14	19	40
8.640	648	π/2 BPSK	4	EH	8	14	19	40
12.960	972	π/2 BPSK	4	EH	8	14	19	40
17.280	1296	π/2 BPSK	4	EH	8	14	19	40
25.920	1944	π/2 BPSK	4	EH	8	14	19	40
34.56	2592	π/2 BPSK	4	EH	8	14	19	40
51.840	3888	π/2 BPSK	4	EH	8	14	19	40
69.120	5184	π/2 BPSK	4	EH	8	14	19	40

14.2.3.5 Timing-related parameters

Table 14-8 lists the general timing parameters associated with the THz-SC PHY.

14.2.3.6 Frame-related parameters

The frame parameters associated with the THz-SC PHY are the same as for the HRCP-SC PHY, given in Table 13-9, with the exception of the N_{CBPC} parameter. The values of the N_{CBPC} parameter for the various modulation schemes are shown in Table 14-9.

Table 14-8—Timing-related parameters

Parameter	Description	Value		Unit	Formula
R_C	Chip rate	1760 to 42 240		Mchip/s	B×1760/(2.16 GHz)
T_C	Chip duration	0.024 to 0.568		ns	$1/R_C$
L_{block}	Block length	64		chips	—
L_{PW}	PW length	0	8	chips	—
T_{PW}	PW duration	0	0.192 to 4.544	ns	$L_{PW} \times T_C$
L_{DC}	Data chips per block	64	56	chips	—
T_{block}	Block duration	1.536 to 36.352		ns	$L_{block} \times T_C$
R_{block}	Block rate	27.509 to 651.042		MHz	$1/ T_{block}$

Table 14-9— N_{CBPC} parameters

Modulation	N_{CBPC}
BPSK	1
QPSK	2
8-PSK	3
8-APSK	3
16-ASPK	4
32-APSK	5
16-QAM	4
64-QAM	6

14.2.3.7 Stuff bits

Stuff bits shall be added to the end of the encoded MAC frame body if the number of the encoded data bits is not an integer multiple of the length of the data portion in the block. The calculation of stuff bits follows the definition in Table 13.2.3.7, where N_{CBPB} is defined in Table 14-10 for each MCS of the THz-SC PHY.

Table 14-10—MCS dependent coded bits per block for the THz-SC PHY

MCS identifier	N_{CBPB} (PW length = 0)	N_{CBPB} (PW length = 8)
0,1	64	56
2,3	128	112
4,5,6,7	192	168
8,9,12,13	256	224
14,15	320	280
10,11	384	336

14.2.3.8 Code spreading

Code spreading of the frame headers shall be done as described in 13.2.3.8.

14.2.3.9 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as defined in 13.2.3.9.

14.2.4 THz-SC PHY frame format

14.2.4.1 General

The THz-SC PHY frame shall be formatted as illustrated in Figure 12-18.

The Frame Header field for the THz-SC PHY frame shall be formatted as illustrated in Figure 13-4. It shall be constructed according to 13.2.4.3.

The PHY preamble is described in 14.2.4.2. The MAC header is defined in 6.2. The PHY header is defined in 14.2.4.3.2, and the HCS is defined in 13.2.4.3.3. The header FEC is defined in 13.2.4.3.4. The PHY Payload field consisting of the MAC frame body, the PPRE, and stuff bits is described in 14.2.4.4. The PPRE is described in 14.2.4.5.2. The stuff bits are described in 13.2.3.7.

14.2.4.2 PHY preamble

A PHY preamble shall be added prior to the frame header to aid receiver algorithms related to auto-gain control (AGC) setting, frame detection, timing acquisition, frequency offset estimation, frame synchronization, and channel estimation.

The PHY preamble, i.e., PHY long preamble and PHY short preamble, shall be transmitted at the chip rate described in Table 14-8.

A PHY long preamble shall be used during PSP and a PHY short preamble shall be used during PAP.

The structure of the PHY long and PHY short preambles are shown in Figure 13-5

For the PHY preamble, T_{SFD} is 0.07 s and T_{CES} is 0.80 s. For PHY long preamble, T_{SYNC} is 2.01 s and T_{PRE} is 2.91 s. For the PHY short preamble, T_{SYNC} is 1.02 s and T_{PRE} is 1.89 s. The fields for frame synchronization (SYNC), SFD, and CES shall be set as described in 13.2.4.2.2, 12.2.4.2.3, and 13.2.4.2.4.

14.2.4.3 Frame header

14.2.4.3.1 General

A frame header shall be added after the PHY preamble, as described in 13.2.4.3.

14.2.4.3.2 THz-SC PHY frame header

The THz-SC PHY header shall be formatted as illustrated in Figure 14-6.

Bits: b0–b3	b4–b7	b8–b11	b12–b13	b14	b15–b36
MCS	Bandwidth	Scrambler Seed ID	PPRE	PW	Frame Length

Figure 14-6—PHY header format for THz-SC PHY

The MCS field shall be set according to the values in Table 14-11.

Table 14-11—MCS field definition for the THz-SC PHY

MCS field value	MCS identifier
0b0000	0
0b0001	1
0b0010	2
0b0011	3
0b0100	4
0b0101	5
0b0110	6
0b0111	7
0b1000	8
0b1001	9
0b1010	10
0b1011	11
0b1100	12

Table 14-11—MCS field definition for the THz-SC PHY (*continued*)

MCS field value	MCS identifier
0b1101	13
0b1110	14
0b1111	15

The Bandwidth field shall be set according to the values in Table 14-12.

Table 14-12—Bandwidth field definition

Bandwidth field value	Bandwidth (GHz)
0b0000	2.16
0b0001	4.32
0b0010	8.64
0b0011	12.96
0b0100	14.28
0b0101	25.92
0b0110	51.84
0b0111	69.12
0b1000	34.56

The PW field shall be set to one if the PW is used in the current frame and shall be set to zero otherwise.

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 12.2.3.10.

The PPRE field shall be set according to the values in Table 13-15.

The Frame Length field shall be an unsigned integer equal to the number of octets in the MAC frame body of a regular frame, excluding the FCS.

14.2.4.3.3 Header HCS

The combination of the PHY header and MAC header shall be protected with the CRC-16 HCS, as described in 13.2.4.3.3.

14.2.4.3.4 Header FEC

The combination of the PHY header, scrambled MAC header, and HCS shall be encoded, as described in 13.2.4.3.4.

14.2.4.4 THz-SC PHY Payload field

14.2.4.4.1 General

The THz-SC PHY Payload field is the last component of the frame and is constructed as shown in Figure 12-23.

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 12.2.3.10
b) Encode the scrambled MAC frame body, as specified in 14.2.3.2.
c) Add stuff bits to the encoded and scrambled MAC frame body according to 14.2.3.7.
d) Map the resulting MAC frame body onto the appropriate constellation, as described in 14.2.3.1.
e) Build blocks from the resulting MAC frame body according to 13.2.4.5.1.
f) Insert PPRE periodically, as described in 14.2.4.5.2.

14.2.4.4.2 THz-SC PHY payload scrambling

The THz-SC PHY payload shall use the scrambling process defined in 12.2.3.10.

14.2.4.4.3 Modulation

Modulation for the MAC frame body is defined in 14.2.3.1.

14.2.4.4.4 FEC

FEC for the MAC frame body is defined in 14.2.3.2.

14.2.4.5 PW and PPRE

14.2.4.5.1 Block and PW

The block and PW is defined as in 13.2.4.5.1.

14.2.4.5.2 PPRE

The PPRE is defined as in 13.2.4.5.2.

14.2.5 Transmitter specifications

14.2.5.1 EVM requirement

The EVM of a compliant transmitter shall be measured and calculated, as defined in 12.1.8, and shall not exceed the values given in Table 14-13 for the indicated mode.

Table 14-13—Max EVM

MCS identifier	Modulation	FEC rate	Max. EVM (dB)
0	BPSK	11/15	–6
1	BPSK	14/15	–7
2	QPSK	11/15	–9
3	QPSK	14/15	–10
4	8-PSK	11/15	–13
5	8-PSK	14/15	–15
6	8-APSK	11/15	–12
7	8-APSK	14/15	–15
8	16-QAM	11/15	–14
9	16-QAM	14/15	–17
10	64-QAM	11/15	–21
11	64-QAM	14/15	–23
12	16-APSK	11/15	–17
13	16-APSK	14/15	–22
14	32-APSK	11/15	–19
15	32-APSK	14/15	–23

14.2.5.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ at maximum.

14.2.5.3 Symbol rate

The THz-SC PHY shall be capable of transmitting at the chip rate, as defined in Table 14-8, to within $\pm 30 \times 10^{-6}$.

The MAC parameter, *pPHYClockAccuracy*, shall be $\pm 30 \times 10^{-6}$.

14.2.6 Receiver specifications

14.2.6.1 Error rate criterion

The error rate criterion shall be a FER of less than 1.3×10^{-7} with a frame payload length of 2^{14} octets, which corresponds to a BER of 10^{-12}. The error rate shall be determined at the PHY SAP interface after any error correction methods (excluding retransmission) have been applied. The measurement shall be performed in AWGN channel.

14.2.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 14.2.6.1 is met. The error ratio shall be determined after any error correction has been applied. A compliant DEV that implements the THz-SC PHY shall achieve at least the reference sensitivity listed in Table 14-14.

Table 14-14—Reference sensitivity levels for MCS for the THz-SC PHY

MCS identifier	Modulation	FEC rate	Receiver sensitivity (dBm) depending on the bandwidth								
			2.16 GHz	4.32 GHz	8.64 GHz	12.96 GHz	17.28 GHz	25.92 GHz	34.56 GHz	51.84 GHz	69.12 GHz
0	BPSK	11/15	–67	–63	–60	–57	–55	–54	–52	–51	–49
1	BPSK	14/15	–64	–62	–59	–56	–53	–52	–50	–49	–47
2	QPSK	11/15	–64	–60	–57	–55	–51	–50	–48	–47	–45
3	QPSK	14/15	–61	–59	–56	–53	–50	–49	–47	–46	–44
4	8-PSK	11/15	–59	–56	–53	–50	–47	–46	–44	–43	–41
5	8-PSK	14/15	–56	–54	–51	–48	–45	–44	–42	–41	–39
6	8-APSK	11/15	–59	–57	–54	–51	–48	–47	–45	–44	–42
7	8-APSK	14/15	–56	–54	–51	–48	–45	–44	–42	–41	–39
8	16-QAM	11/15	–57	–55	–52	–49	–46	–45	–43	–42	–40
9	16-QAM	14/15	–54	–52	–49	–46	–43	–42	–40	–39	–37
10	64-QAM	11/15	–52	–48	–45	–42	–39	–38	–36	–35	–33
11	64-QAM	14/15	–48	–46	–43	–40	–37	–36	–34	–33	–31
12	16-APSK	11/15	–70	–52	–49	–46	–43	–42	–40	–39	–37
13	16-APSK	14/15	–70	–47	–44	–41	–38	–37	–35	–34	–32
14	32-APSK	11/15	–70	–50	–47	–44	–41	–40	–38	–37	–35
15	32-APSK	14/15	–70	–46	–43	–40	–37	–36	–34	–33	–31

14.2.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 14.2.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

14.2.7 PHY layer timing

14.2.7.1 General

The PHY layer timing parameters are defined in 13.2.7.

14.2.7.2 Interframe space

A conformant implementation shall support the IFS parameters, as described in 13.2.7.2.

14.2.7.3 Receive-to-transmit turnaround time

The receive to transmit turnaround time shall be set as described in 13.2.7.3.

14.2.7.4 Transmit-to-receive turnaround-time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*, as described in Table 13-20.

14.2.7.5 Time between successive transmissions

The minimum time between the end of the last transmitted frame and the beginning of the retransmitted frame shall be less than a RIFS time. The RIFS times for both a PRC and a DEV are given in Table 13-20.

14.2.7.6 Channel switch

The channel switch time is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pPHYChannelSwitchTime*, as defined in Table 13-19.

14.3 THz-OOK PHY

14.3.1 General

The THz-OOK PHY is designed for cost-effective devices that require low power, low complexity, and simple design. For applications using this PHY, transmission ranges of a few tens of centimeters are targeted. The THz-OOK PHY is designed for PHY-SAP payload-bit rates between 1.3 Gb/s, using a single channel with a bandwidth of 2.16 GHz, and the maximum 52.6 Gb/s, using a bandwidth of 69.12 GHz.

14.3.2 Channelization for THz-OOK PHY

The possible channels are the same as defined in 14.1.3. The transmit spectral masks for the THz-OOK PHY are the same as defined in 14.1.4.

14.3.3 Modulation and coding

14.3.3.1 Overview

The entire THz-OOK frame shall be modulated with OOK, as specified in 14.3.3.2. The MCS dependent parameters shall be set according to Table 14-15. The chip rate of the THz-OOK PHY is given in Table 14-8. The FEC for the THz-OOK PHY shall be as specified in 14.3.3.3.

Table 14-15—MCS dependent parameters for the THz-OOK PHY

MCS identifier	Bandwidth (GHz)	FEC rate	Data rate (Gb/s) with PW	Data rate (Gb/s) without PW
0	2.16	224/240	1.64	1.44
0	4.32	224/240	3.29	2.87
0	8.64	224/240	6.57	5.75
0	12.96	224/240	9.86	8.62
0	17.28	224/240	13.14	11.50
0	25.92	224/240	19.71	17.25
0	34.56	224/240	26.28	23.00
0	51.84	224/240	39.42	34.50
0	69.12	224/240	59.14	51.74
1	2.16	11/15	1.29	1.29
1	4.32	11/15	2.58	2.26
1	8.64	11/15	5.16	4.52
1	12.96	11/15	7.74	6.78
1	17.28	11/15	10.33	9.04
1	25.92	11/15	15.49	13.55
1	34.56	11/15	20.66	18.08
1	51.84	11/15	30.98	27.11
1	69.12	11/15	41.30	36.14
2	2.16	14/15	1.64	1.44
2	4.32	14/15	3.29	2.87
2	8.64	14/15	6.57	5.75
2	12.96	14/15	9.86	8.62
2	17.28	14/15	13.14	11.50
2	25.92	14/15	19.71	17.25
2	34.56	14/15	26.28	23.00
2	51.84	14/15	39.42	34.50
2	69.12	14/15	52.56	45.99

14.3.3.2 Modulation

THz-OOK frames shall be modulated using OOK, as described in 13.3.3.5.

14.3.3.3 Forward error correction

The FEC scheme is specified by a (240,224)-Reed Solomon code and two LDPC codes with code rates of 14/15 and 11/15. The Reed Solomon Code, defined in 13.3.3.6, is mandatory for the THz OOK-PHY. The LDPC (1440,1344), defined in 12.2.3.6.4, and the LDPC (1440,1056), defined in 13.2.3.6, are both optional for the THz-OOK PHY.

14.3.3.4 MCS dependent parameters

A block length for the THz-OOK PHY shall be 64 chips. The PW length, where a PW is defined in 13.2.4.5.1, for the THz-OOK PHY shall be 0 chips or 8 chips. The PW length of 8 chips is mandatory and that of 0 chips is optional.

14.3.3.5 Header dependent parameters

The header dependent parameters shall be set according to the values defined in Table 14-16. The header rate is proportional to the bandwidth used. The headers use an EH code, as defined in 13.2.4.3.4.

Table 14-16—Header rate dependent parameters for the THz-OOK PHY

Bandwidth (GHz)	Header rate (Mb/s)	Modulation scheme	Spreading factor, L_{SF}	FEC	PW length (chips), L_{PW}	Code bits per block, L_{CBPS}	Number of occupied blocks, N_{block_hdr}	Number of stuff bits, L_{STUFF}
2.160	162	OOK	4	EH	8	14	19	40
4.320	324	OOK	4	EH	8	14	19	40
8.640	648	OOK	4	EH	8	14	19	40
12.960	972	OOK	4	EH	8	14	19	40
17.280	1296	OOK	4	EH	8	14	19	40
25.920	1944	OOK	4	EH	8	14	19	40
34.56	2592	OOK	4	EH	8	14	19	40
51.840	3888	OOK	4	EH	8	14	19	40
69.120	5184	OOK	4	EH	8	14	19	40

14.3.3.6 Timing-related parameters

The general timing parameters for the THz-OOK PHY shall be set as defined for the THz-SC PHY according to Table 14-8.

14.3.3.7 Frame-related parameters

The frame parameters associated with the THz-OOK PHY are the same as for the THz-SC PHY defined in 14.2.3.6, with the exception that the parameter N_{CBPC} takes the value one.

14.3.3.8 Stuff bits

Stuff bits shall be added to the end of the encoded MAC frame body if the number of the encoded data bits is not an integer multiple of the length of the data portion in the block. The calculation of stuff bits follows the definition in 13.2.3.7, where N_{CBPB} is defined in Table 14-17 for each MCS of the THz-OOK PHY.

Table 14-17—MCS dependent coded bits per block for the THz-OOK PHY

MCS identifier	N_{CBPB} (PW length = 0)	N_{CBPB} (PW length = 8)
0,1,2	64	56

14.3.3.9 Code spreading

Code spreading shall be applied to THz-OOK frame headers according to 13.2.3.8.

14.3.3.10 Scrambling

Scrambling of THz-OOK fields shall be performed as defined in 13.2.3.9.

14.3.4 THz-OOK PHY frame format

14.3.4.1 General

The THz-OOK PHY frame shall be formatted as illustrated in Figure 12-18.

The Frame Header field for the THz-OOK PHY frame shall be formatted as illustrated in Figure 13-4. It shall be constructed according to 13.2.4.3.

The PHY preamble is described in 14.3.4.2. The MAC header is defined in 6.2. The THz-OOK PHY frame header is defined in 14.3.4.2, and the HCS is defined in 13.3.4.3.3. The header FEC is defined in 13.2.4.3.4. The PHY Payload field, consisting of the MAC frame body and stuff bits, is described in 13.3.4.4. The stuff bits are described in 14.3.3.8.

14.3.4.2 THz-OOK PHY frame header

The THz-OOK PHY header shall be formatted as illustrated in Figure 14-7.

Bits: b0–b1	b2–b5	b6–b9	b10–b11	b12	b13–b34
MCS	Bandwidth	Scrambler Seed ID	PPRE	PW	Frame Length

Figure 14-7—PHY header format for THz-OOK PHY

The MCS field shall be set according to the values in Table 14-18.

The Bandwidth field shall be set according to Table 14-12.

Table 14-18—MCS field definition for the THz-OOK PHY

MCS field value	MCS identifier
0b00	0
0b01	1
0b10	2
0b11	Reserved

The PW field shall be set to one if the PW is used in the current frame and shall be set to zero otherwise.

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 12.2.3.10.

The PPRE field shall be set according to the values in Table 13-15.

The Frame Length field shall be an unsigned integer equal to the number of octets in the MAC frame body of a regular frame, excluding the FCS.

14.3.4.3 THz-OOK PHY Payload field

14.3.4.3.1 General

The THz-OOK PHY Payload field is the last component of the frame and is constructed as shown in Figure 13-26.

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 14.3.4.3.2.
b) Encode the scrambled MAC frame body, as specified in 14.3.4.3.4.
c) Add stuff bits to the encoded and scrambled MAC frame body according to 14.3.3.8.
d) Map the resulting MAC frame body onto the appropriate constellation, as described in 14.3.3.2.
e) Build blocks from the resulting MAC frame body according to 13.2.4.5.1.

14.3.4.3.2 THz-OOK PHY Payload scrambling

The THz-OOK PHY payload shall use the scrambling process defined in 12.2.3.10.

14.3.4.3.3 Modulation

Modulation for the MAC frame body is defined in 14.3.3.2.

14.3.4.3.4 FEC

FEC for the MAC frame body is defined in 14.3.3.3.

14.3.4.4 Blocks and PW

The block and PW is defined in 13.2.4.5.1, and the PW shall be modulated with OOK.

14.3.4.5 PPRE

PPRE insertion is defined in 13.2.4.5.2, and the PPRE shall be modulated with OOK.

14.3.5 Transmitter specifications

14.3.5.1 EVM requirement

Eye opening for OOK is described in G.7.

14.3.5.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ at maximum.

14.3.5.3 Symbol rate

The THz-SC PHY shall be capable of transmitting at the chip rate, as defined in Table 14-8, to within $\pm 30 \times 10^{-6}$.

The MAC parameter, *pPHYClockAccuracy*, shall be $\pm 30 \times 10^{-6}$.

14.3.6 Receiver specifications

14.3.6.1 Error rate criterion

The error rate criterion shall be an FER of less than 1.3×10^{-7} with a frame payload length of 214 octets, which corresponds to a BER of 10^{-12}. The error rate shall be determined at the PHY SAP interface after any error correction methods (excluding retransmission) have been applied. The measurement shall be performed in an AWGN channel.

14.3.6.2 Receiver sensitivity

The receiver sensitivity is the minimum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 14.3.6.1 is met. The error ratio shall be determined after any error correction has been applied. A compliant DEV that implements the THz-OOK PHY shall achieve at least the reference sensitivity listed in Table 14-19.

Table 14-19—Reference sensitivity levels for MCS for the THz-OOK PHY

MCS identifier	Modulation	FEC rate	Receiver sensitivity (dBm) depending on the bandwidth								
			2.16 GHz	4.32 GHz	8.64 GHz	12.96 GHz	17.28 GHz	25.92 GHz	34.56 GHz	51.84 GHz	69.12 GHz
0	OOK	224/240	–62	–59	–56	–54	–53	–51	–50	–48	–47
1	OOK	14/15	–65	–62	–59	–57	–56	–54	–53	–51	–50
2	OOK	11/15	–62	–59	–56	–54	–53	–51	–50	–48	–47

14.3.6.3 Receiver maximum input level

The receiver maximum input level is the maximum power level of the incoming signal, in dBm, present at the input of the receiver for which the error rate criterion in 14.3.6.1 is met. A compliant receiver shall have a receiver maximum input level of at least –10 dBm for each of the modulation formats that the DEV supports.

14.3.7 PHY layer timing

14.3.7.1 General

The PHY layer timing parameters are defined in 13.2.7.

14.3.7.2 Interframe space

A conformant implementation shall support the IFS parameters, as described in 13.2.7.2.

14.3.7.3 Receive-to-transmit turnaround time

The receive to transmit turnaround time shall be set as described in 13.2.7.3.

14.3.7.4 Transmit-to-receive turnaround-time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*, as described in Table 13-20.

14.3.7.5 Time between successive transmissions

The minimum time between the end of the last transmitted frame and the beginning of the retransmitted frame shall be less than a RIFS time. The RIFS times for both a PRC and a DEV are given in Table 13-20.

14.3.7.6 Channel switch

The *channel switch time* is defined as the time from when the last valid bit is received at the antenna on one channel until the DEV is ready to transmit or receive on a new channel. The channel switch time shall be less than *pPHYChannelSwitchTime*, as defined in Table 13-19.

15. Beam forming

15.1 Introduction

Beam forming is only applicable to the mmWave PHY, as defined in Clause 12.

This clause specifies an optional beam-forming protocol. A multitude of antenna configurations such as single antenna element, sectored antennas, switched antennas, and one-dimensional (1-D) and two-dimensional (2-D) beam-forming antenna arrays are supported.

Two types of beam-forming protocols are specified: an on-demand beam forming and a proactive beam forming.

On-demand beam forming may be used between two DEVs or between the PNC and a DEV and shall take place in the CTA allocated to the DEV for the purpose of beam forming. The CTA allocated for beam forming shall use one of the beam-forming stream indices, as defined in 6.2.8. If a CTA is allocated with sector-level training stream index, then the beam forming starts from sector-level training. However, if the CTA is allocated with the beam-level training stream index, then the beam forming starts directly from beam-level training. However, a DEV shall not request CTA with the beam-level training stream index without first finishing sector-level training.

Proactive beam forming may be used when the PNC is the source of data to one or multiple DEVs. It allows multiple DEVs to train their own receiver antennas for optimal reception from the PNC with lower overhead. During proactive beam forming, the sector-level training from PNC to DEV shall take place in the beacon. The sector-level training from DEV to PNC and the beam-level training of both directions, shall take place in the CTAP, as described in 15.7.

Proactive and on-demand beam forming are achieved using a two-level training mechanism, namely a sector (coarse) level training and a beam (fine) level training, followed by an optional HRS tracking phase, as detailed in 15.5.2.

Two bean-forming criterion are specified: a beam BST criteria suitable for all antenna configurations, and PET criteria for 1-D linear antenna arrays and 2-D planar antenna arrays. All DEVs that support this bean-forming method shall support the BST criterion. The PET criterion may be used only if the two DEVs support it. BST is based on selecting the best beam from a given set of beams whereas PET is based on finding the optimal beam former and combiner vectors (i.e., antenna weights) that do not necessarily fall into the given set of beams.

Support for beam forming is optional. However, when the beam forming in this clause is implemented, the sector-level training shall be supported for switched/sectored antennas and the two-level training, as defined in 15.5.2, shall be supported for all other antenna configurations. The tracking phase is optional. SC and HSI DEVs that support beam forming need to support CMS as well, as described in 15.5.2.2.

The IFSs used in the bean-forming protocol defined in this clause are as follows:

— Bean-forming beam-level IFS (BBIFS) = 128 SC PHY chips (~73 ns)
— Bean-forming sector-level IFS (BSIFS) = 0.5 µs

15.2 Beam-forming terminology

15.2.1 General

The concept of patterns with increasing resolution level, namely, quasi-omni patterns, sectors, fine beams, and HRS beams is illustrated in Figure 15-1. In addition, the clustering concept is introduced and the convention used in beams numbering and cluster encoding are clarified.

When describing beam forming between two DEVs, the following notation will be used:

a) When two DEVs are communicating, they will be referred to as DEV1 and DEV2. DEV1 may be the PNC. The DEV number, *d*, will be one for DEV1 (or the PNC) and 2 for DEV2.
b) The total number of transmit and receive antenna elements for DEV number *d* are denoted as $M^{(d,t)}$ and $M^{(d,r)}$, respectively. The corresponding transmit and receive antenna elements are denoted as $A_n^{(d,t)}$ where $n = 0{:}M^{(d,t)} - 1$ for the transmit antennas and $A_n^{(d,r)}$ where $n = 0{:}M^{(d,r)} - 1$ for the receive antennas.
c) The total number of quasi-omni transmit and receive patterns of interest for DEV number *d*, is denoted as $I^{(d,t)}$ and $I^{(d,r)}$, respectively. The corresponding quasi-omni transmit and receive patterns are denoted as $Q_n^{(d,t)}$ where $n = 0{:}I^{(d,t)} - 1$ for the transmit patterns and $Q_n^{(d,r)}$ where $n = 0{:}I^{(d,r)} - 1$ for the receive patterns.
d) The best pair of quasi-omni transmit and receive patterns for DEV *d* when communicating with the other DEV are identified by indices $i^{(d,t)}$ and $i^{(d,r)}$, respectively. The corresponding quasi-omni transmit and receive patterns are denoted as $Q_{i^{(d,t)}}^{(d,t)}$ and $Q_{i^{(d,r)}}^{(d,r)}$, respectively.
e) The total number of transmit and receive sectors of interest for DEV number *d* are denoted as $J^{(d,t)}$ and $J^{(d,r)}$, respectively. The corresponding transmit and receive sectors are denoted as $S_n^{(d,t)}$ where $n = 0{:}J^{(d,t)} - 1$ for the transmit sectors and $S_n^{(d,r)}$ where $n = 0{:}J^{(d,r)} - 1$ for the receive sectors.
f) The best pair of transmit and receive sectors for DEV d when communicating with the other DEV are identified by indices $j^{(d,t)}$ and $j^{(d,r)}$, respectively. The corresponding transmit and receive sectors are denoted as $S_{j^{(d,t)}}^{(d,t)}$ and $S_{j^{(d,r)}}^{(d,r)}$, respectively.
g) The total number of transmit and receive fine beams of interest for DEV number d are denoted as $K^{(d,t)}$ and $K^{(d,r)}$, respectively. The corresponding transmit and receive fine beams are denoted as $B_n^{(d,t)}$ where $n = 0{:}K^{(d,t)} - 1$ for the transmit fine beams and $B_n^{(d,r)}$ where $n = 0{:}K^{(d,r)} - 1$ for the receive fine beams.
h) The best pair of transmit and receive fine beams for DEV *d* when communicating with the other DEV are identified by indices $k^{(d,t)}$ and $k^{(d,r)}$, respectively. The corresponding transmit and receive fine beams are denoted as $B_{k^{(d,t)}}^{(d,t)}$ and $B_{k^{(d,r)}}^{(d,r)}$, respectively.
i) The total number of transmit and receive HRS beams of interest for DEV number *d* are denoted as $L^{(d,t)}$ and $L^{(d,r)}$, respectively. The corresponding transmit and receive HRS beams are denoted as $H_n^{(d,t)}$ where $n = 0{:}L^{(d,t)} - 1$ for the transmit HRS beams and $H_n^{(d,r)}$ where $n = 0{:}L^{(d,r)} - 1$ for the receive HRS beams. The transmit (receive) HRS beams are grouped into two clusters. The total number of HRS beams in these two clusters are denoted as $L_1^{(d,t)}$ and $L_2^{(d,t)}$ ($L_1^{(d,r)}$ and $L_2^{(d,r)}$), respectively. HRS beams are just an additional set of fine beams.
j) The best pair of transmit and receive HRS beams for DEV *d* when communicating with the other DEV are identified by indices $l^{(d,t)}$ and $l^{(d,r)}$, respectively. The corresponding transmit and receive HRS beams are denoted as $H_{l^{(d,t)}}^{(d,t)}$ and $H_{l^{(d,r)}}^{(d,r)}$, respectively.
k) If both DEVs are SAS, the superscripts *t* and *r* can be dropped since they are the same.

15.2.2 Quasi-omni patterns

The term *quasi-omni pattern* is the lowest resolution pattern and is used to refer to an antenna pattern that covers a very broad area of the region of space of interest around a DEV (including when the DEV is acting as the PNC). A DEV covers the region of space of interest with a minimal set of, possibly overlapping, quasi-omni patterns. A set size of one indicates that the DEV is able to cover the spatial region of interest with only one quasi-omni pattern, indicating that the DEV is omni-capable.

15.2.3 Sectors

The term *sector* is the second level resolution pattern and is used to refer to an antenna direction or an array pattern that covers a relatively broad area of multiple beams. A sector can cover a set of consecutive or non-consecutive beams and different sectors can overlap.

15.2.4 Beams

For the pattern estimation and tracking option, beams shall be selected from the beam-forming codebooks specified in 15.3. A transmit or receive codebook is identified by the number of transmit or receive antennas, $M^{(t)}$ or $M^{(r)}$, respectively, and the desired number of transmit or receive beams, $K^{(t)}$ or $K^{(r)}$ for fine beams or $L^{(t)}$ or $L^{(r)}$ for HRS beams, respectively. For a 2-D antenna array, separate codebooks are associated with each dimension as well as for transmit and receive.

Beam sets of different number of beams can be generated by setting different phases on the same number of antenna elements. Figure 15-1(c) and Figure 15-1(d) show an example of an 8-element linear antenna array with 8 fine beams and 16 HRS beams, respectively.

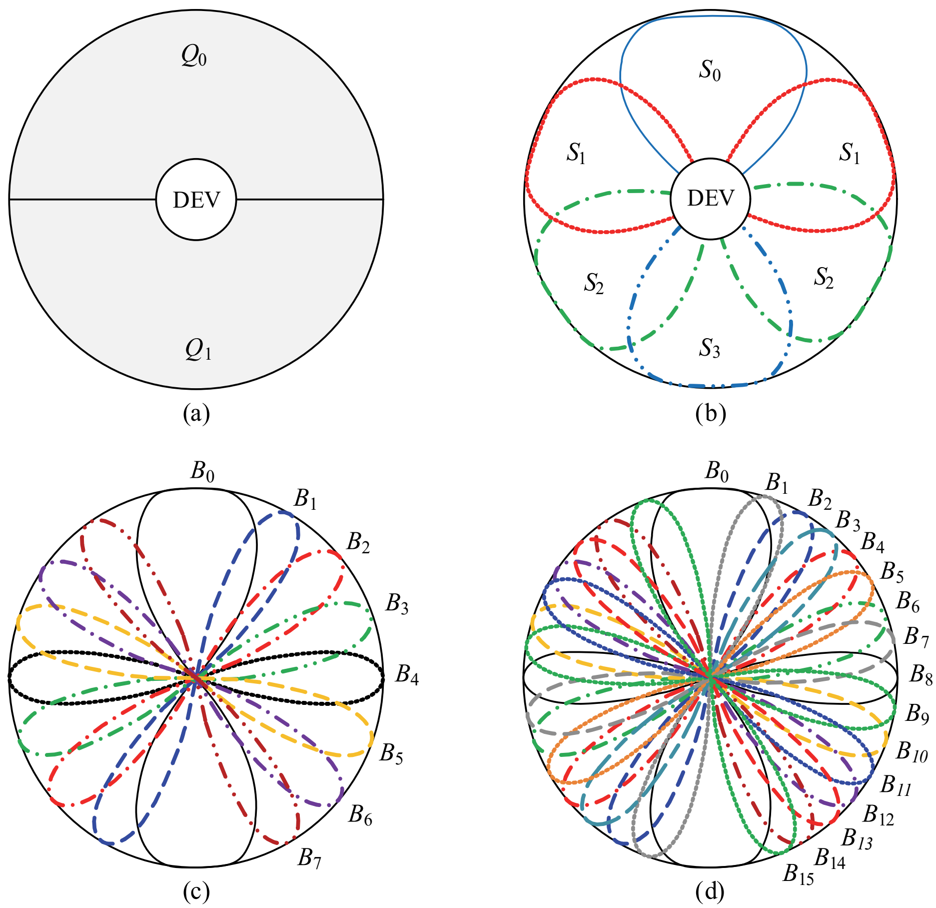

Figure 15-1—Beam patterns: a) quasi-omni patterns, b) sectors, c) fine beams, and d) HRS beams

For a 1-D antenna array with K beams along the *z*-axis, beams shall be identified by indices zero through $K-1$ in the direction of increasing polar angle as shown in Figure 15-1. These beams shall correspond one to one with the beam vectors zero to $K-1$ from the selected beam-forming codebook detailed in 15.3.

For a 2-D antenna array with K_x beams on the x-axis and K_z beams on the z-axis, the K_x beams along the x-axis shall be identified by indices zero through $K_x - 1$ in the direction of increasing polar angle and shall correspond one to one with the beam vectors zero to $K_x - 1$ from the selected x-beam-forming codebook. The K_z beams along the z-axis shall be identified by indices zero through $K_z - 1$ in the direction of increasing polar angle and shall correspond one to one with the beam vectors zero to $K_z - 1$ from the selected z-beams codebook. This is further illustrated in Figure 15-2 for a 2-D antenna array with 8 beams in each direction.

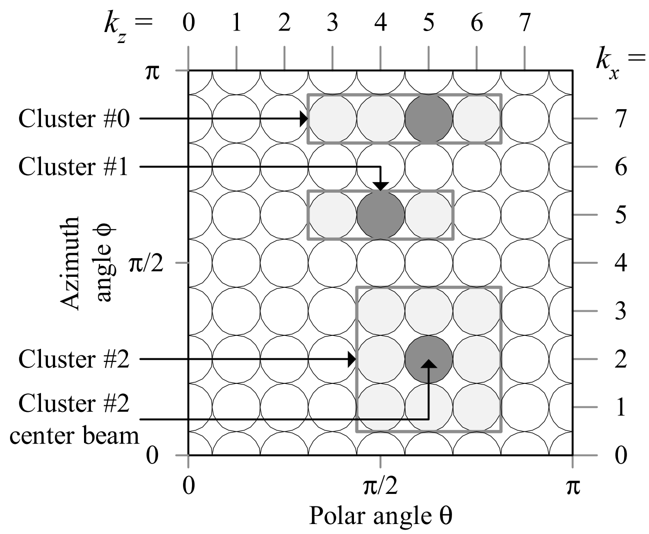

Figure 15-2—Beam numbering and clusters

15.2.5 Clusters

A cluster is a group of beams around a center beam. The clustering concept is introduced to facilitate tracking. The number of clusters per sector(s) is left to the implementer. Figure 15-2 gives examples of clusters of different sizes. The circles in Figure 15-2 represent hypothetical beams.

Cluster encoding shall only be used for DEVs that support the PET option. For DEVs implementing the beam switching and steering option, cluster encoding support is not required.

A cluster shall be encoded by an 8-bit field: $c_7c_6c_5c_4c_3c_2c_1c_0$. The first 3 LSB bits, i.e., $c_2c_1c_0$, encode the beams in the polar angle direction in reference to Figure 15-2, while the second set of 3 bits, i.e., $c_5c_4c_3$, encodes the beams in the azimuth angle direction. The last set of 2 bits, c_7c_6, specifies three different 2-D puncturing patterns, i.e., different cluster geometries.

Bits c_1c_0 shall encode the total number of beams in the polar direction (excluding the center beam). The maximum number of beams (excluding the center beam) shall be 3, which corresponds to the setting c_1c_0=11. Bit c_2 shall be set to zero if the number of beams to the left of the center beam is smaller than or equal to the number of beams to the right of the center beam; otherwise, bit c_2 shall be set to one.

Bits c_4c_3 shall encode the total number of beams in the azimuth direction (excluding the center beam). Bit c_5 shall be set to zero if the number of beams below the center beam is smaller than or equal to the number of beams above the center beam; otherwise, bit c_5 shall be set to one.

Bits c_7c_6 shall encode the 2-D puncturing patterns as follows. When bit c_6 is set to one, this shall indicate that the cluster is punctured; otherwise, all 2-D beams within a cluster shall be used. When $c_7c_6 = 11$, the cluster is fully punctured; i.e., only the beams along the polar angle direction and azimuth angle direction around the center beam are used. When $c_7c_6 = 01$, the beams along the polar angle direction and azimuth angle direction around the center beam as well as the adjacent beams to the center beam are used. Figure 15-3 shows some examples of cluster encoding.

Finally, beams in a cluster are ordered in increasing index k_z and decreasing index k_x in reference to Figure 15-2. When a cluster is transmitted, the first beam, i.e., the beam with lowest k_z index and highest k_x index, shall be transmitted first, and the last beam, i.e., the beam with highest k_z index and lowest k_x index, shall be transmitted last.

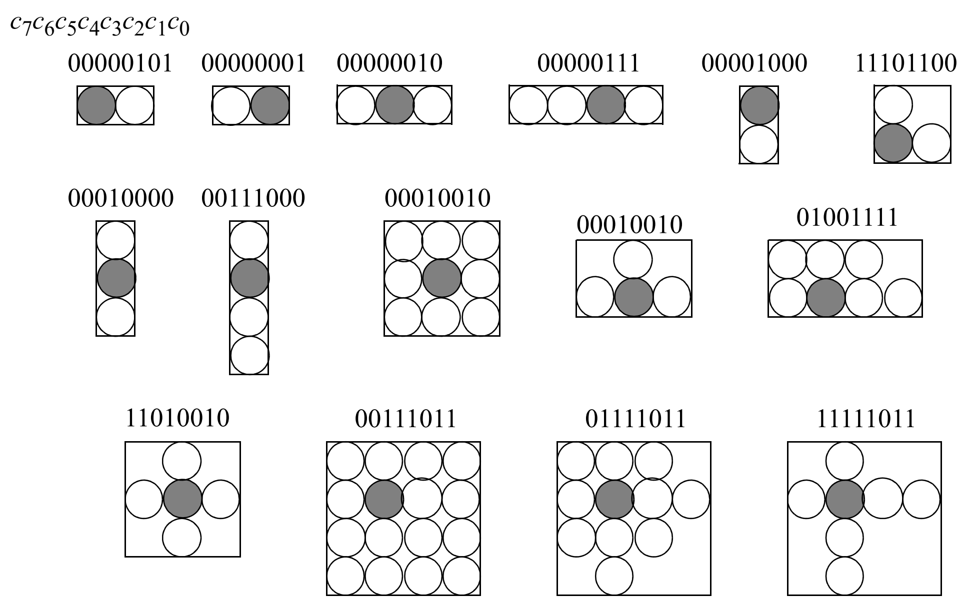

Figure 15-3—Examples of cluster encoding

15.3 Beam-forming codebooks

This subclause specifies beam-forming codebooks for sectored and switched antennas, 1-D and 2-D arrays with uniform spacing of λ/2.

A codebook is a matrix where each column specifies the beam former vector or combiner vector to be used. The term *codeword* is used here as a generic term for beam former vector and combiner vector. Each column specifies a specific pattern or direction. The set of columns span the entire space, which is 360 degrees. Columns shall be numbered in increasing order starting with zero.

The codebook for a sectored antenna array and switched antenna array of M elements is a special case of beam-forming antenna given by the M×M identity matrix shown in Equation (15-6)

$$\mathbf{W}_{M \times M} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{15-6}$$

For the purpose of training, it is sufficient to provide codebooks in one dimension. The codebooks specified here are 1-D.

Each beam-forming codebook is identified by the number of antenna elements, M, and the desired number of beam patterns, K. For the case where $K \geq M$, the codebook beam vectors are given by the column vectors of the following matrix, see Equation (15-7):

$$W(m, k) = j^{\mathrm{fix}\left\{\frac{m \times mod[k + (K/2), K]}{K/4}\right\}} \text{ for } m = 0{:}M-1 \text{ and } k = 0{:}K-1 \tag{15-7}$$

The function fix() returns the biggest integer smaller than or equal to its argument. It is also possible to substitute the function round() for the function fix(), where the function round() returns the closest integer to the input argument.

For the special case where $K = M/2$, the codebook beam vectors are given by the column vectors of the following matrix, see Equation (15-8):

$$W(m, k) = \begin{cases} (-j)^{mod(m, k)} & m = 0{:}N-1 \text{ and } k = 0 \\ (-1)^{\mathrm{fix}\left\{\frac{m \times mod[k + (K/2), K]}{K/4}\right\}} & m = 0{:}N-1 \text{ and } k = 1{:}K-1 \end{cases} \tag{15-8}$$

The function round() can be substituted for the function fix() as before.

The 2-D antenna arrays can be trained by separable codebooks along the polar angle direction and azimuth angle direction. This is illustrated in Figure 15-4 where the 2-D antennas weights w_{m_x, m_z}, $m_x = 0{:}M_x - 1$ and $m_z = 0{:}M_z - 1$ can be computed from the antenna weights along the x-axis w_{m_x}, $m_x = 0{:}M_x - 1$ and the antenna weights along the z-axis w_{m_z}, $m_z = 0{:}M_z - 1$ as follows (see Equation (15-9):

$$w_{m_x, m_z} = w_{m_x} w_{m_z} \text{ for } m_x = 0{:}M_x - 1 \text{ and } m_z = 0{:}M_z - 1 \tag{15-9}$$

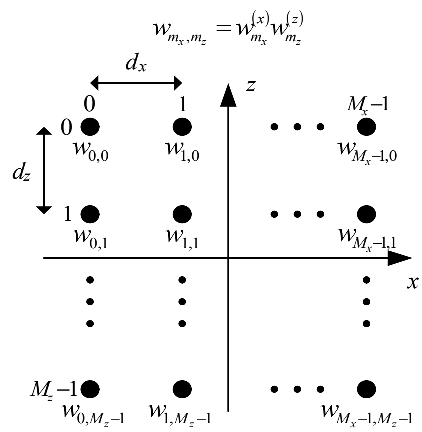

Figure 15-4—Separable 2-D antenna array

Consequently, if the polar codebook has M_x codewords and the azimuth codebook has M_z codewords, the 2-D codebook will have $M_x \times M_z$ codewords.

The beam-forming codebooks for sectored and switched antennas, 1-D and 2-D arrays with uniform spacing of $\lambda/2$:

a) The antenna element, as shown in Figure 15-4, on the m_x^{th} row ($m_x = 0{:}M_x - 1$) and m_z^{th} column ($m_z = 0{:}M_z - 1$) shall be numbered $m_x M_x + m_z M_z$.

b) The Codebook ID field is an 8-bit number where the 7 LSB bits indicate the number of desired beam patterns and the MSB bit indicates whether the round (MSB bit = 0) or fix (MSB bit = 1) function is used. The Codebook ID field is used only if both DEVs support PET.

For all other configurations, the BST criterion shall be used, in which case, the knowledge of codebooks at the receiving side is no longer required.

15.4 Beam-forming reference model

The beam-forming reference model is illustrated in Figure 15-5. In this figure, DEV1 has $M^{(1,\,t)}$ transmit antennas and $M^{(1,\,r)}$ receive antennas while DEV2 has $M^{(2,\,t)}$ transmit antennas and $M^{(2,\,r)}$ receive antennas. Depending on the implementation, the transmit and receive antenna weight vectors **w** and **c** belong to specific alphabets. For example, for switched/sectored antennas where one antenna is active at a time, the weights belong to the alphabet $\{0, 1\}$. For a phased antenna array implementing specific phase shifts, the weights are restricted to those specific phase shifts. For a complex beam-forming antenna array, the weights can be adjusted in both phase and amplitude.

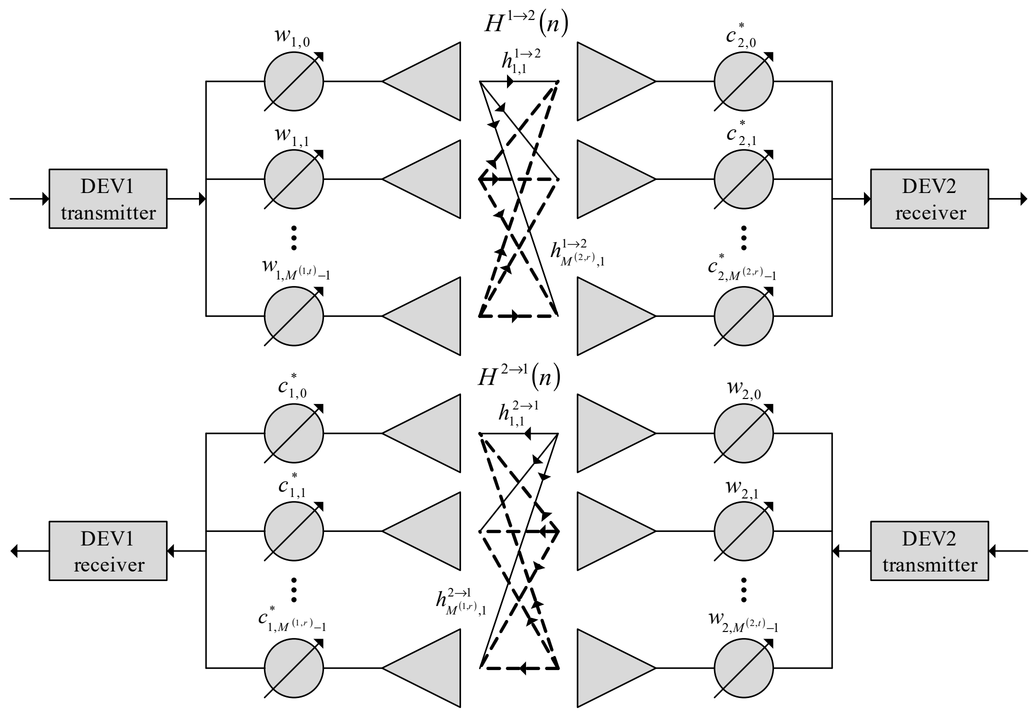

Figure 15-5—Beam-forming reference model

The system model is developed in reference to Figure 15-5. At DEV1's transmitter, the SC PHY, AV PHY, or HSI PHY bit stream is multiplied by the following beam-former vector, see Equation (15-10)

$$\mathbf{w} = \begin{bmatrix} w_{1,\,0} & w_{1,\,1} & \dots & w_{1,\,M^{(1,\,t)}-1} \end{bmatrix}^T \tag{15-10}$$

and then transmitted through a multipath MIMO channel with the following frequency domain channel state information (CSI) matrix, see Equation (15-11):

$$\mathbf{H}^{1 \to 2}(n) \in C^{M^{(1,t)} \times M^{(2,r)}} \tag{15-11}$$

at frequency bin number n, $n = 0, 1, 2, \ldots, N_f$, see Equation (15-12):

$$\mathbf{H}^{1 \to 2}(n) = \begin{bmatrix} h_{1,1}^{1 \to 2}(n) & h_{1,2}^{1 \to 2}(n) & \cdots & h_{1,M^{(2,r)}}^{1 \to 2}(n) \\ h_{2,1}^{1 \to 2}(n) & h_{2,2}^{1 \to 2}(n) & \cdots & h_{2,M^{(2,r)}}^{1 \to 2}(n) \\ \cdots & \cdots & \cdots & \cdots \\ h_{M^{(1,t)},1}^{1 \to 2}(n) & h_{M^{(1,t)},2}^{1 \to 2}(n) & \cdots & h_{M^{(1,t)},M^{(2,r)}}^{1 \to 2}(n) \end{bmatrix} \tag{15-12}$$

where $h_{i,j}^{1 \to 2}(n)$ represents the channel response between DEV1 j^{th} transmit antenna and DEV2's i^{th} receive antenna, where j and i range from 1 to the number of transmit and receive antennas, respectively. The number of frequency bins, N_f, corresponds to the subblock length in SC PHY(i.e., $N_f = 256$) or to the number of used carriers in HSI PHY or AV PHY (i.e., $N_f = 352$).

At DEV2's receiver, the received signals a processed through the following combiner vector, see Equation (15-13):

$$\mathbf{c}_2^T = \begin{bmatrix} c_{2,0} & c_{2,1} & \cdots & c_{2,M^{(2,r)}-1} \end{bmatrix} \tag{15-13}$$

The equivalent channel between DEV1's transmitter and DEV2's receiver is a SISO channel with frequency response at bin n given byEquation (15-14):

$$G^{1 \to 2}(n) = \mathbf{c}_2^H \mathbf{H}^{1 \to 2}(n) \mathbf{w}_1 \text{ for } n = 0, 1, \ldots, N_f - 1 \tag{15-14}$$

In a similar way, the equivalent channel between DEV2's transmitter and DEV1's receiver is a SISO channel with frequency response at bin n given by Equation (15-15):

$$G^{2 \to 1}(n) = \mathbf{c}_1^H \mathbf{H}^{2 \to 1}(n) \mathbf{w}_2 \text{ for } n = 0, 1, \ldots, N_f - 1 \tag{15-15}$$

The objective of the pattern estimation beam forming is to select the beam former vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ and the combiner vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ that optimize a cost function that measures the link quality according to a selected criterion. If for example an effective SNR criterion is selected, then DEV2 has to be able to acquire and track the CSI matrices $\mathbf{H}^{1 \to 2}(n)$ for $n = 0{:}N_f - 1$ in the region of space of interest. Furthermore, if the channel is asymmetric, then DEV1 has to be able to acquire and track the CSI matrices $\mathbf{H}^{2 \to 1}(n)$ for $n = 0{:}N_f - 1$. If a beam-switching option is selected, then a DEV needs only to measure the link quality per beam pair. The exact optimization criterion is left to the implementer.

For the special case of a SAS, the same antenna array is used for transmission and reception and for a symmetric channel, and so the optimal beam former and combiner vectors are related as follows:

$$\mathbf{c}_1 = \mathbf{w}_1{}^* \text{ and } \mathbf{c}_2 = \mathbf{w}_2{}^*$$

where * indicates the complex conjugate.

For this special case, it is sufficient to determine one of the two vectors for each DEV.

The general case where at least one of the DEVs uses a different antenna system for transmission and reception shall be referred to as asymmetric antenna system (AAS).

Measuring the link quality between all beam pairs, or acquisition of the entire set of CSI matrices, is time costly and incurs high overhead. In order to reduce the amount of time and overhead required for training, a two-level beam-forming mechanism shall be used as detailed in 15.5.2.

15.5 Beam-forming protocol

15.5.1 General

The beam-forming protocol consists of a two-level training mechanism and an optional tracking phase. The two-level training mechanism consists of a sector-level and beam-level training and is used to find the best pair of beam patterns between two DEVs with a given beam resolution. Tracking is used to achieve higher resolution and to track the best set of beam patterns between the two DEVs.

15.5.2 Two-level training mechanism

15.5.2.1 General

The sector level is used to limit the region of space that is of interest and to find the best pair of sectors. These sectors are then sliced into beams in preparation for beam-level training. Beam-level training is used to select the optimal transmit beam vector and receiver beam vector as outlined in 15.4. In the simple case of beam switching, this reduces to selecting the best transmit and receive pair of beams. Beam-level training is achieved using a set of beam former and combiner vectors (from selected beam-forming codebooks) covering the sector(s) selected during sector-level training. The beam-former (combiner) codebooks are specified in 15.3.

15.5.2.2 Sector-level training

15.5.2.2.1 General

If the two DEVs are both SAS, then the sector-level training shall be performed, as described in 15.5.2.2.3. If either or both of the DEVs are AAS, then sector-level training shall be performed, as described in 15.5.2.2.2.

15.5.2.2.2 AAS sector-level training

The sector level consists of following stages:

- Sector training
- Sector feedback
- Sector to beam mapping
- Acknowledgment

The first stage, sector training, shall be divided into two parts (sector training from DEV1 to DEV2 and sector training from DEV2 to DEV1) and shall be formatted as illustrated in Figure 15-6.

Sector training DEV1→DEV2 $J^{(1,t)}$ cycles	SIFS	Sector training DEV2→DEV2 $J^{(2,t)}$ cycles	SIFS

Figure 15-6—AAS sector training

The sector training from DEV1 to DEV2 is illustrated in Figure 15-7.

Sector training cycle #0	Sector training cycle #1	…	Sector training cycle #$J^{(1,t)} - 1$

Figure 15-7—AAS sector training DEV1→DEV2

The sector training cycle is illustrated in Figure 15-8.

Sector training #0 $S_{j^{(1,t)}}^{(1,t)} \rightarrow S_0^{(2,r)}$	Sector training #1 $S_{j^{(1,t)}}^{(1,t)} \rightarrow S_1^{(2,r)}$	…	Sector training #$J^{(2,r)} - 1$ $S_{j^{(1,t)}}^{(1,t)} \rightarrow S_{J^{(2,r)}-1}^{(2,r)}$	BSIFS

Figure 15-8—AAS sector training cycle #$J^{(1,t)}$ for sector training DEV1→DEV2

For the SC PHY and HSI PHY, the sector training sequence shall be identical to the CMS preamble, as defined in 12.1.13.5. For the AV PHY, the sector training sequence shall be identical to the HRP preamble.

The sector training from DEV1 to DEV2 consists of $J^{(1,t)}$ cycles. During each cycle, DEV1 shall send $J^{(2,r)}$ repetitions of a sector training sequence in the same direction, i.e., the direction specified by the corresponding sector codeword. Each cycle except the last one shall end with a BSIFS. The $J^{(1,t)}$ cycles shall be sent in $J^{(1,t)}$ different directions, $[S_0^{(1,t)}, S_1^{(1,t)}, \ldots, S_{J^{(1,t)}-1}^{(1,t)}]$ corresponding to the chosen $J^{(1,t)}$ transmit sector codewords.

During a cycle, DEV2 shall attempt to receive each of the $J^{(2,r)}$ sector training sequences using a different listening (receive) direction. The $J^{(2,r)}$ different listening directions, $[S_0^{(2,r)}, S_1^{(2,r)}, \ldots, S_{J^{(2,r)}-1}^{(2,r)}]$, during a cycle shall correspond to DEV2's $J^{(2,r)}$ chosen sector codewords.

At the completion of the full $J^{(1,t)}$ cycles, DEV2 will have had an opportunity to receive a sector training sequence using each combination of DEV1 transmit sector direction (0 to $J^{(1,t)} - 1$) and DEV2 receive sector direction (0 to $J^{(2,r)} - 1$). Based on this information, DEV2 selects the best sector pair, i.e., DEV1's optimal transmit sector, $S_{j^{(1,t)}}^{(1,t)}$, and DEV2's optimal receive sector, $S_{j^{(2,r)}}^{(2,r)}$.

Following the sector training from DEV1 to DEV2, a similar sector training from DEV2 to DEV1 takes place where DEV2 transmits sector training sequences over $J^{(2,t)}$ cycles as shown in Figure 15-9.

Sector training cycle #0	Sector training cycle #1	…	Sector training cycle #$J^{(2,t)} - 1$

Figure 15-9—AAS sector training DEV2→DEV1

The sector training cycles shown in Figure 15-9 is illustrated in Figure 15-10.

Sector training #0 $S_{j^{(2,t)}}^{(2,t)} \rightarrow S_0^{(1,r)}$	Sector training #1 $S_{j^{(2,t)}}^{(2,t)} \rightarrow S_1^{(1,r)}$	…	Sector training #$J^{(1,r)} - 1$ $S_{j^{(2,t)}}^{(2,t)} \rightarrow S_{J^{(1,r)}-1}^{(1,r)}$	BSIFS

Figure 15-10—AAS sector training cycle #$J^{(1,t)}$ for sector training DEV2→DEV1

At the completion of the cycles, DEV1 selects the best sector pair, i.e., DEV2's optimal transmit sector, $S_{j^{(2,t)}}^{(2,t)}$, and DEV1's optimal receive sector, $S_{j^{(1,r)}}^{(1,r)}$.

The second stage of sector-level training is sector feedback and shall be formatted as in Figure 15-11.

DEV1→DEV2 feedback $J^{(1,t)}$ repetitions	SIFS	DEV2→DEV1 feedback $S_{j^{(2,t)}}^{(2,t)} \rightarrow S_{j^{(1,r)}}^{(1,r)}$	SIFS

Figure 15-11—AAS sector feedback

For the SC PHY and HSI PHY, the DEV1→DEV2 feedback is defined as in Figure 15-12. For the AV PHY with LRP omni feedback, only one repetition is required for the DEV1 to DEV2 feedback repetitions block.

Announce command #0 $S_0^{(1,t)} \rightarrow S_{j^{(2,r)}}^{(2,r)}$	BSIFS	Announce command #$j^{(1,t)}$ $S_{j^{(1,t)}}^{(1,t)} \rightarrow S_{j^{(2,r)}}^{(2,r)}$	…	Announce command #$J^{(1,t)} - 1$ $S_{J^{(1,t)}-1}^{(1,t)} \rightarrow S_{j^{(2,r)}}^{(2,r)}$	BSIFS

Figure 15-12—AAS DEV1→DEV2 feedback

The Announce command shall be sent $J^{(1,t)}$ times in the $J^{(1,t)}$ different transmit directions, $[S_0^{(1,t)}, S_1^{(1,t)}, \ldots, S_{J^{(1,t)}-1}^{(1,t)}]$. For the AV PHY with LRP omni feedback, only one repetition is required for the DEV1 to DEV2 feedback repetition block. When more than one Announce command is sent as feedback, all frames shall be sent with ACK policy set to no-ACK, with the exception of the last one, which shall be sent with ACK policy set to Imp-ACK.

This is required since DEV1 does not yet know its optimal transmit sector. DEV2 switches to its optimal receive sector, $J_{j^{(2,r)}}^{(2,r)}$ and attempts to receive at least the transmission sent on DEV1's optimal transmit sector, $J_{j^{(1,t)}}^{(1,t)}$. The Feedback IE informs DEV2 of its optimal transmit sector, $S_{j^{(2,t)}}^{(2,t)}$, second best transmit sector, and the corresponding LQIs.

In return, DEV2 shall transmit its sector feedback in a Feedback IE in an Announce command with Imp-ACK requested. The Announce command shall be sent on DEV2's optimal transmit sector, $S_{j^{(2,t)}}^{(2,t)}$,and DEV1 shall listen on its optimal receive sector, $S_{j^{(1,r)}}^{(1,r)}$. The Feedback IE informs DEV1 of its optimal transmit sector, $S_{j^{(1,t)}}^{(1,t)}$, second best transmit sector, and the corresponding LQIs. For the AV PHY, the Feedback IE sent from both DEV1 and DEV2 shall be transmitted using LRP mode. For SC PHY and HSI PHY, the Feedback IE sent from both DEV1 and DEV2 shall be transmitted using CMS.

Upon completion of the feedback stage, both DEV1 and DEV2 know their optimal transmit and receive sectors. These shall be used for any further frame exchanges in this level.

The mapping stage follows the feedback stage. The mapping stage format is as illustrated in Figure 15-13.

DEV1→DEV2 mapping $S_{j^{(1,t)}}^{(1,t)} \rightarrow S_{j^{(2,r)}}^{(2,r)}$	SIFS	DEV2→DEV1 mapping $S_{j^{(2,t)}}^{(2,t)} \rightarrow S_{j^{(1,r)}}^{(1,r)}$	SIFS

Figure 15-13—AAS sector to beam mapping

DEV1 shall transmit its sector to beam mapping in an Announce command with the ACK Policy set to Imp-ACK with the following IEs:

— The Mapping IE field, as defined in 6.4.35, that for the SC PHY and HSI PHY, specifies the number of DEV1 transmit and receive beams and the preamble type (short or long) to be used in the

beam-level training. For the AV PHY, this IE specifies are the number of DEV1 transmit and receive beams and the HRP preamble. This IE is always present.
- The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that contains number of DEV1 transmit and receive clusters and the number of beams in each cluster, and the cluster encoding (when both DEVs use PET.) The BST Clustering IE shall be exchanged only when both DEVs support tracking.
- The Beam PET IE, as defined in 6.4.38, that contains the number of transmit and receive antennas on the *z*-axis and *x*-axis, the corresponding codebook IDs, and the amplitude and phase resolution capabilities. The Beam PET IE may be exchanged only when both DEVs use PET.

DEV2 shall reply by sending back its own sector to beam mapping in an Announce command with the ACK Policy set to Imm-ACK with the following IEs:

- The Mapping IE, as defined in 6.4.35, that for the SC PHY and HSI PHY, specifies the number of DEV2 transmit and receive beams and the preamble type (short or long) to be used in the beam-level training. For the AV PHY, this IE specifies the number of DEV1 transmit and receive beams and the HRP preamble. This IE is always present.
- The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that specifies the number of DEV2 transmit and receive clusters and the number of beams in each cluster, and the cluster encoding (when both DEVs use PET). The BST Clustering IE shall be exchanged only when both DEVs support tracking.
- The Beam PET IE, as defined in 6.4.38, that specifies the number of transmit and receive antennas on the *z*-axis and *x*-axis, the corresponding codebook IDs, and the amplitude and phase resolution capabilities. The Beam PET IE may be exchanged only when both DEVs use PET.

DEV1 shall reply with an Imm-ACK that completes the sector-level training.

Figure 15-14 illustrates the message flow for a successful AAS sector-level training process.

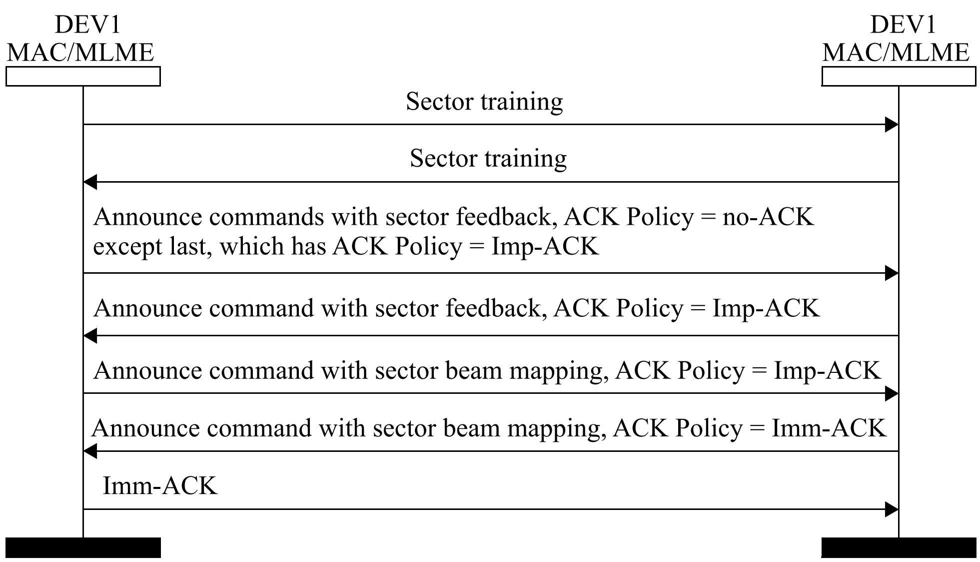

Figure 15-14—AAS sector-level training process

15.5.2.2.3 SAS sector-level training

Sector-level training for a SAS consists of the following stages:
- Sector training
- Sector feedback

— Sector beam-mapping
— Acknowledgment

The sector training stage shall be formatted as in Figure 15-15.

Sector training DEV1→DEV2 $J^{(1)}$ cycles	SIFS

Figure 15-15—SAS sector training

The sector training from DEV1 to DEV2 consists of $J^{(1)}$ sector training cycles and is illustrated in Figure 15-16.

Sector training cycle #0	Sector training cycle #1	…	Sector training cycle # $J^{(1)} - 1$

Figure 15-16—SAS sector training DEV1→DEV2

The sector training cycle is illustrated in Figure 15-17.

Sector training #0 $S^{(1)}_{j^{(1)}} \rightarrow S^{(2)}_0$	Sector training #1 $S^{(1)}_{j^{(1)}} \rightarrow S^{(2)}_1$	…	Sector training # $J^{(2)} - 1$ $S^{(1)}_{j^{(1)}} \rightarrow S^{(2)}_{J^{(2)}-1}$	BSIFS

Figure 15-17—SAS sector training cycle #j$^{(1)}$ for sector training DEV1→DEV2

The sector training sequence shall be identical to the CMS preamble for the SC PHY and HSI PHY. The sector training sequence shall be identical to the HRP preamble for the AV PHY.

The sector training consists of $J^{(1)}$ cycles. During each cycle, DEV1 shall send $J^{(2)}$ repetitions of a sector training sequence in the same direction. Each cycle except the last one shall end with a BSIFS. The $J^{(1)}$ cycles shall be sent in $J^{(1)}$ different directions, $[S^{(1)}_0, S^{(1)}_1, \ldots, S^{(1)}_{J^{(1)}-1}]$ corresponding to the chosen $J^{(1)}$ sector codewords.

During a cycle, DEV2 shall attempt to receive each of the $J^{(2)}$ sector training sequences using a different direction. The $J^{(2)}$ different directions, $[S^{(2)}_0, S^{(2)}_1, \ldots, S^{(2)}_{J^{(2)}-1}]$ during a cycle shall correspond to DEV2's $J^{(2)}$ chosen sector codewords.

At the completion of the full $J^{(1)}$ cycles, DEV2 will have had an opportunity to receive a sector training sequence using each combination of DEV1 transmit sector (0 to $J^{(1)} - 1$) and DEV2 receive sector (0 to $J^{(2)} - 1$). Based on this information, DEV2 selects the best sector pair, i.e., DEV1's optimal transmit and receive sector, $J^{(1)}_{i^{(1)}}$, and DEV2's optimal transmit and receive sector, $J^{(2)}_{i^{(2)}}$.

The second stage of sector-level training is sector feedback and shall be formatted as in Figure 15-18.

DEV2→DEV1 feedback $J^{(1)}$ repetitions	SIFS

Figure 15-18—SAS sector feedback

The DEV2→DEV1 feedback frame is defined as in Figure 15-19.

Announce command #0 $S^{(2)}_{j^{(2)}} \rightarrow S^{(1)}_0$	BSIFS	Announce command # $j^{(2)}$ $S^{(2)}_{j^{(2)}} \rightarrow S^{(1)}_{j^{(1)}}$	…	Announce command # $J^{(2)} - 1$ $S^{(2)}_{j^{(2)}} \rightarrow S^{(1)}_{J^{(1)}-1}$	BSIFS

Figure 15-19—DEV2→DEV1 feedback

For the SC PHY and HSI PHY, DEV2 shall transmit its sector feedback in a Feedback IE, as defined in 6.4.34, by sending an Announce command with ACK policy set to no-ACK with the exception of the last Announce command, which shall have the ACK policy set to Imp-ACK. The Announce command shall be sent in the optimal transmit sector, $S^{(2)}_{j^{(2)}}$, and shall be repeated $J^{(1)}$ times as shown in Figure 15-19. This is required since DEV1 does not yet know which sector to use to receive frames from DEV2, and therefore shall listen on each of the $J^{(1)}$ sectors until it hears the Announce command from DEV2. For the AV PHY, the Feedback IE sent from DEV2 shall be transmitted using LRP mode. For the SC PHY and HSI PHY, the Feedback IE sent from both DEV1 and DEV2 shall be transmitted using CMS. For the AV PHY with LRP omni feedback, only one repetition is required for the DEV2 to DEV1 feedback repetitions block. The Feedback IE informs DEV1 of its optimal sector, $S^{(1)}_{j^{(1)}}$, second best sector, and the corresponding LQIs.

Upon completion of the feedback stage, both DEV1 and DEV2 know their optimal transmit and receive sectors. These shall be used for any further frame exchanges in this level.

The mapping stage follows the feedback stage. The mapping stage format is as illustrated in Figure 15-20.

DEV1→DEV2 mapping $S^{(1)}_{j^{(1)}} \rightarrow S^{(2)}_{j^{(2)}}$	SIFS	DEV2→DEV1 mapping $S^{(2)}_{j^{(2)}} \rightarrow S^{(1)}_{j^{(1)}}$	SIFS

Figure 15-20—SAS sector beam mapping

DEV1 shall transmit its sector to beam mapping in an Announce command with the ACK Policy set to Imp-ACK with the following IEs:

— The Mapping IE, as defined in 6.4.35, that for the SC PHY and HSI PHY, contains the number of DEV1 transmit and receive beams and the preamble type (short or long) to be used in the beam-level training. For the AV PHY, this IE specifies the number of DEV1 beams and HRP preamble. This IE is always present.

— The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that specifies the number of DEV1 transmit and receive clusters and the number of beams in each cluster, and the cluster encoding (when both DEVs use PET.) The BST Clustering IE may be exchanged only when both DEVs support tracking.

— The Beam PET IE, as defined in 6.4.38, that specifies the number of transmit and receive antennas on the *z*-axis and *x*-axis, the corresponding codebook IDs, and the amplitude and phase resolution capabilities. The Beam PET IE shall be exchanged only when both DEVs use PET.

DEV2 shall reply by sending back its own sector to beam mapping in an Announce command with the ACK Policy set to Imm-ACK with the following IEs:

— The Mapping IE, as defined in 6.4.35, that for the SC PHY and HSI PHY, specifies the number of DEV1 transmit and receive beams and the preamble type (short or long) to be used in the beam-level training. For the AV PHY, this IE specifies the number of DEV1 beams and HRP preamble. This IE is always present.

— The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that specifies the number of DEV2 transmit and receive clusters and the number of beams in each cluster, and the cluster encoding (when both DEVs use PET.) The BST Clustering IE shall be exchanged only when both DEVs support tracking.
— The Beam PET IE, as defined in 6.4.38, that specifies the number of transmit and receive antennas on the *z*-axis and *x*-axis, the corresponding codebook IDs, and the amplitude and phase resolution capabilities. The Beam PET IE may be exchanged only when both DEVs use PET.

DEV1 shall reply with an Imm-ACK that completes the sector-level training.

Figure 15-21 illustrates the message flow for a successful SAS sector-level training process.

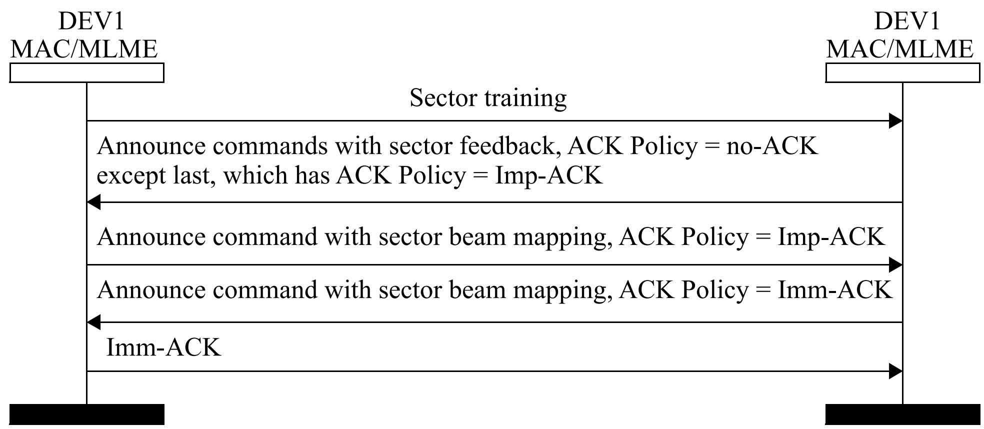

Figure 15-21—SAS sector training process

15.5.2.2.4 Sector-level training failure remedy

Announce commands may be retransmitted according to 7.12. Should the Announce command in either direction fail to be acknowledged after a number of retransmissions that are implementation dependent, or if there is not enough time remaining in the CTA for the entire frame exchange, the DEV that is attempting the retransmissions shall terminate the stream. DEV1 shall then request a CTA again from the PNC and restart the beam-forming process from the beginning.

15.5.2.3 Beam-level training

15.5.2.3.1 General

Once sector-level training is completed, DEV1 and DEV2 shall start beam-level training. The beam-level training explores beams within the best sectors to find the best beam pair (best transmit and receive patterns) for DEV1 and DEV2. The AAS case is described first, followed by the SAS case.

If the two DEVs use PET or if tracking is enabled, then when referring to beams transmission, the cluster mapping shall be transmitted in increasing index number, i.e., cluster number 0 shall be transmitted first. Furthermore, beams within a cluster shall be transmitted first in the order specified in 15.2.5.

The AAS beam-level training is described first, followed by the SAS case.

15.5.2.3.2 AAS beam-level training

The beam-level training consists of the following stages:

— Beam training
— Beam feedback
— Beam to HRS beam mapping
— Acknowledgment

The first stage, beam training, shall be divided into two parts (beam training from DEV1 to DEV2 and beam training from DEV2 to DEV1) and shall be formatted as illustrated in Figure 15-22.

Beam training DEV1→DEV2 $K^{(1,t)}$ cycles	SIFS	Beam training DEV2→DEV2 $K^{(2,t)}$ cycles	SIFS

Figure 15-22—AAS beam training

The beam training from DEV1 to DEV2 is illustrated in Figure 15-23.

Beam training cycle #0	Beam training cycle #1	…	Beam training cycle #$K^{(1,t)}$–1

Figure 15-23—AAS beam training DEV1→DEV2

The beam training cycle is shown in Figure 15-24.

Beam training #0 $B^{(1,t)}_{k^{(1,t)}} \rightarrow B^{(2,r)}_0$	Beam training #1 $B^{(1,t)}_{k^{(1,t)}} \rightarrow B^{(2,r)}_1$	…	Beam training #$K^{(2,r)} - 1$ $B^{(1,t)}_{k^{(1,t)}} \rightarrow B^{(2,r)}_{K^{(2,r)}-1}$	BBIFS

Figure 15-24—AAS beam training cycle #$k^{(1,t)}$ for beam training DEV1→DEV2

For the SC PHY and HSI PHY, the beam training sequence shall be transmitted in the mode (preamble type) agreed upon during the sector mapping stage of the sector-level training. For the AV PHY, the beam training sequence shall be the HRP preamble.

The beam training from DEV1 to DEV2 consists of $K^{(1,t)}$ cycles. During each cycle, DEV1 shall send $K^{(2,r)}$ repetitions of a beam training sequence in the same direction, i.e., the direction specified by the corresponding beam codeword. Each cycle except the last one shall end with a BBIFS. The $K^{(1,t)}$ cycles shall be sent in the $K^{(1,t)}$ different directions, $[B^{(1,t)}_0, B^{(1,t)}_1, \ldots, B^{(1,t)}_{K^{(1,t)}-1}]$ corresponding to the chosen $K^{(1,t)}$ transmit beam codewords.

During a cycle, DEV2 shall attempt to receive each of the $K^{(2,r)}$ beam training sequence repetitions using a different listening (receive) direction. The $K^{(2,r)}$ different listening directions, $[B^{(2,r)}_0, B^{(2,r)}_1, \ldots, B^{(2,r)}_{K^{(2,r)}-1}]$, during a cycle shall correspond to DEV2's $K^{(2,r)}$ chosen beam codewords.

At the completion of the full $K^{(1,t)}$ cycles, DEV2 will have had an opportunity to receive a beam training sequence using each combination of DEV1 transmit beam (0 to $K^{(1,t)} - 1$) and DEV2 receive beam (0 to $K^{(2,r)} - 1$). Based on this information, DEV2 selects the best beam pair, i.e., DEV1's optimal transmit beam, $B^{(1,t)}_{k^{(1,t)}}$, and DEV2's optimal receive beam, $B^{(2,r)}_{k^{(2,r)}}$.

Following the beam training from DEV1 to DEV2, a similar beam training from DEV2 to DEV1 takes place where DEV2 transmits beam training sequences over $K^{(2,t)}$ cycles as shown in Figure 15-25.

Beam training cycle #1	Beam training cycle #0	…	Beam training cycle #$K^{(2,t)} - 1$

Figure 15-25—AAS beam training DEV2→DEV1

The beam training cycle is shown in Figure 15-26.

Beam training #0 $B^{(2,t)}_{k^{(2,t)}} \rightarrow B^{(1,r)}_0$	Beam training #1 $B^{(2,t)}_{k^{(2,t)}} \rightarrow B^{(1,r)}_1$	…	Beam training #$K^{(2,r)} - 1$ $B^{(2,t)}_{k^{(2,t)}} \rightarrow B^{(1,r)}_{K^{(1,r)}-1}$	BBIFS

Figure 15-26—AAS beam training cycle #$K^{(2,t)}$ for beam training DEV2→DEV1

At the completion of the $K^{(2,t)}$ cycles, DEV1 selects the best beam pair, i.e., DEV2's optimal transmit beam, $B^{(2,t)}_{k^{(2,t)}}$, and DEV1's optimal receive beam, $B^{(1,r)}_{k^{(1,r)}}$.

The second stage of beam-level training is beam feedback and shall be formatted as in Figure 15-27. For the AV PHY with LRP omni feedback, only one repetition is required for the DEV1 to DEV2 feedback repetitions block. For SC PHY and HSI PHY, the Feedback IE sent from both DEV1 and DEV2 shall be transmitted using CMS.

DEV1→DEV2 feedback $B^{(1,t)}_{j^{(1,t)}} \rightarrow B^{(2,r)}_{j^{(2,r)}}$	SIFS	DEV2→DEV1 feedback $B^{(2,t)}_{j^{(2,t)}} \rightarrow B^{(1,r)}_{j^{(1,r)}}$	SIFS

Figure 15-27—AAS beam feedback

The DEV1→DEV2 feedback shall be sent using an Announce command on DEV1's optimal transmit sector, $S^{(1,t)}_{j^{(1,t)}}$, and DEV2 shall listen on its optimal receive sector, $S^{(2,t)}_{j^{(2,t)}}$. The Announce command shall contain the following IEs:

— The Feedback IE, as defined in 6.4.34, that contains DEV2's optimal transmit beam, $B^{(2,t)}_{k^{(2,t)}}$, the second best transmit beam, and the corresponding LQIs.
— The PET Phase IE, as defined in 6.4.41, that contains DEV2's phase vector, i.e., the phase for each of the $M^{(2,t)}$ transmit antenna elements. This IE may be exchanged only when both DEVs use PET and DEV2's transmit phase resolution is greater than zero.
— The PET Amplitude IE, as defined in 6.4.40, that contains DEV2's amplitude vector, i.e., the amplitude for each of the $M^{(2,t)}$ transmit antenna elements. This IE may be exchanged only when both DEVs use PET and DEV2's transmit amplitude resolution is greater than zero.

In return, DEV2 shall transmit its beam feedback by sending an Announce command with Imp-ACK requested containing the same IEs. The Announce command shall be sent on DEV2's optimal transmit sector, $S^{(2,t)}_{j^{(2,t)}}$ and DEV1 shall listen on its optimal receive sector, $S^{(1,r)}_{j^{(1,r)}}$. For the AV PHY, the beam feedback sent from both DEV1 and DEV2 shall be transmitted using LRP mode. For SC PHY and HSI PHY, the feedback IE sent from both DEV1 and DEV2 shall be transmitted using CMS. The Announce command shall contain the following IEs:

— Feedback IE
— PET phase IE
— PET amplitude IE

Upon completion of the feedback stage, both DEV1 and DEV2 know their optimal transmit and receive beams (patterns). These shall be used for any further frame exchanges in this level.

If either or both of the DEVs do not support tracking, the beam to HRS beam mapping exchange shall be skipped and the last Announce command from DEV2 to DEV1 in the feedback stage shall be sent with Imm-ACK instead.

If tracking is supported, then following the feedback stage, the beam to HRS beam mapping is performed. The Beam to HRS beam mapping shall be formatted as illustrated in Figure 15-28.

DEV1→DEV2 mapping $B_{k^{(1,t)}}^{(1,t)} \to B_{k^{(2,r)}}^{(2,r)}$	SIFS	DEV2→DEV1 mapping $B_{k^{(2,t)}}^{(2,t)} \to B_{k^{(1,r)}}^{(1,r)}$	SIFS

Figure 15-28—AAS beam to HRS beam mapping

DEV1 shall transmit its beam to HRS beam mapping in an Announce command with the ACK Policy set to Imp-ACK with the following IEs:

— The Mapping IE, as defined in 6.4.35, that for the SC PHY and HSI PHY, specifies the number of DEV1 transmit and receive HRS beams and the synchronization mode to be used in the HRS beam tracking phase. For the AV PHY, this IE specifies the number of DEV1 transmit and receive HRS beams and the HRP preamble. This IE is present if both DEVs support tracking.
— The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that specifies the number of DEV1 transmit and receive clusters and the number of HRS beams in each cluster, and the cluster encoding (when both DEVs use PET). The BST Clustering IE shall be exchanged only when both DEVs support tracking.
— The HRS Beam PET IE, as defined in 6.4.39, that specifies the *z*-axis and *x*-axis HRS codebook IDs. The HRS Beam PET IE shall be exchanged only when both DEVs use PET.

DEV2 shall reply by sending back its own beam to HRS beam mapping in an Announce command with the ACK Policy set to Imm-ACK containing the following IEs:

— Mapping IE
— PET Clustering IE
— HRS Beam PET IE

DEV1 shall reply with an Imm-ACK that completes the beam-level training.

Figure 15-29 illustrates the message flow for a successful AAS beam-level training process.

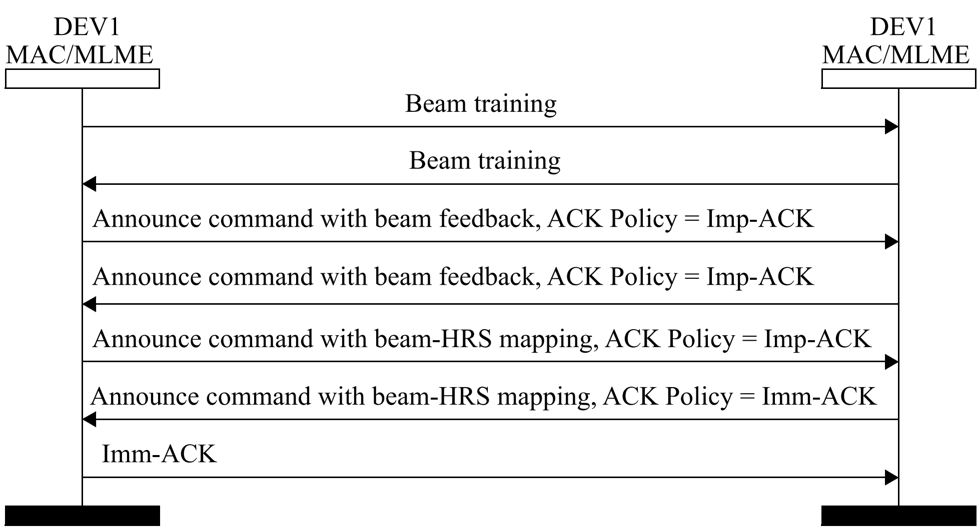

Figure 15-29—AAS beam-level training process

15.5.2.3.3 SAS beam-level training

The beam level consists of the following stages:

— Beam training
— Beam feedback
— Beam to HRS beam mapping
— Acknowledgment

The first stage, beam training, shall be formatted as illustrated in Figure 15-30.

Beam training DEV1→DEV2 $K^{(1)}$ cycles	SIFS

Figure 15-30—SAS beam training

The beam training from DEV1 to DEV2is illustrated in Figure 15-31.

Beam training cycle #0	Beam training cycle #1	…	Beam training cycle #$K^{(1)} - 1$

Figure 15-31—SAS beam training DEV1→DEV2

The beam training cycle is illustrate in Figure 15-32.

Beam training #0 $B^{(1)}_{k^{(1)}} \rightarrow B^{(2)}_0$	Beam training #1 $B^{(1)}_{k^{(1)}} \rightarrow B^{(2)}_1$	…	Beam training #$K^{(2)} - 1$ $B^{(1)}_{k^{(1)}} \rightarrow B^{(2)}_{K^{(2)}-1}$	BBIFS

Figure 15-32—SAS beam training cycle #$K^{(1)}$ for beam training DEV1→DEV2

For the SC PHY and HSI PHY, the beam training sequence shall be transmitted in the mode agreed upon during the sector mapping stage of the sector-level training. For the AV PHY, the beam training sequence shall be the HRP preamble.

The beam training from DEV1 to DEV2 consists of $K^{(1)}$ cycles. During each cycle, DEV1 shall send $K^{(2)}$ repetitions of a beam training sequence in the same direction, i.e., the direction specified by the corresponding beam codeword. Each cycle except the last one shall end with a BBIFS. The $K^{(1)}$ cycles shall be sent in the $K^{(1)}$ different directions, $[B^{(1)}_0, B^{(1)}_1, \ldots, B^{(1)}_{K^{(1)}-1}]$ corresponding to the chosen $K^{(1)}$ transmit beam codewords.

During a cycle, DEV2 shall attempt to receive each of the $K^{(2)}$ beam training sequence repetitions using a different listening (receive) direction. The $K^{(2)}$ different listening directions, $[B^{(2)}_0, B^{(2)}_1, \ldots, B^{(2)}_{K^{(2)}-1}]$, during a cycle shall correspond to DEV2's $K^{(2)}$ chosen beam codewords.

At the completion of the full $K^{(1)}$ cycles, DEV2 will have had an opportunity to receive a beam training sequence using each combination of DEV1 transmit beam (0 to $K^{(1)} - 1$) and DEV2 receive beam (0 to $K^{(2)} - 1$). Based on this information, DEV2 selects the best beam pair, i.e., DEV1's optimal transmit and receive beam, $B^{(1)}_{k^{(1)}}$, and DEV2's optimal receive beam, $B^{(2)}_{k^{(2)}}$.

Following the beam training, beam feedback begins. The beam feedback is formatted as illustrated in Figure 15-33.

DEV2→DEV1 feedback $S^{(2)}_{j^{(2)}} \rightarrow S^{(1)}_{j^{(1)}}$	SIFS

Figure 15-33—SAS Beam feedback

DEV2 shall transmit its beam feedback in an Announce command with Imp-ACK requested. The Announce command shall be sent on DEV2's optimal transmit sector, $S^{(2)}_{j^{(2)}}$, and DEV1 shall listen on its optimal receive sector, $S^{(1)}_{j^{(1)}}$. The Announce command shall contain the following IEs:

— The Feedback IE, as defined in 6.4.34, that contains the DEV1's optimal transmit and receive beam, $B^{(1)}_{k^{(1)}}$, second best transmit and receive beam, and the corresponding LQIs.
— The PET Phase IE, as defined in 6.4.41, that contains the DEV1's phase vector, i.e., the phase for each of the $M^{(1)}$ antenna elements. This IE shall be exchanged only when both DEVs use PET and DEV1's phase resolution is greater than zero.
— The PET Amplitude IE, as defined in 6.4.40, that contains the DEV1's amplitude vector, i.e., the amplitude for each of the $M^{(1)}$ antenna elements. This IE shall be exchanged only when both DEVs use PET and DEV1's amplitude resolution is greater than zero.

Upon completion of the feedback stage, both DEV1 and DEV2 know their optimal transmit and receive beams (patterns). These shall be used for any further frame exchanges in this level.

If either or both of the DEVs do not support tracking, the beam to HRS beam mapping exchange shall be skipped and the last Announce command from DEV2 to DEV1 in the feedback stage shall be sent with Imm-ACK instead.

If tracking is supported, then following the feedback stage, the beam to HRS beam mapping stage is utilized. The Beam to HRS beam mapping is formatted as illustrated in Figure 15-34.

DEV1→DEV2 mapping $B^{(1)}_{k^{(1)}} \rightarrow B^{(2)}_{k^{(2)}}$	SIFS	DEV2→DEV1 mapping $B^{(2)}_{k^{(2)}} \rightarrow B^{(1)}_{k^{(1)}}$	SIFS

Figure 15-34—SAS beam to HRS beam mapping

DEV1 shall transmit its beam to HRS beam mapping in an Announce command with Imp-ACK requested with the following IEs:

— The Mapping IE, as defined in 6.4.35, that, for the SC PHY and HSI PHY, specifies the number of DEV1 transmit and receive HRS beams and the synchronization mode to be used in the HRS beam tracking phase. For the AV PHY, this IE specifies the DEV1 HRS beams and the HRP preamble. This IE is present if both DEVs support tracking.
— The BST Clustering IE, as defined in 6.4.36, or PET Clustering IE, as defined in 6.4.37, that contains the number of DEV1 transmit and receive clusters and the number of HRS beams in each cluster, and the cluster encoding (when both DEVs use PET). The BST Clustering IE shall be exchanged only when both DEVs support tracking.
— The HRS Beam PET IE, as defined in 6.4.39, that contains the polar angle and azimuth angle HRS codebook IDs. The HRS Beam PET IE shall be exchanged only when both DEVs use PET.

DEV2 shall reply by sending back its own beam to HRS beam mapping in an Announce command with Imm-ACK requested with the following IEs:

— Mapping IE,
— BST Clustering IE
— HRS Beam PET IE

DEV1 shall reply with an Imm-ACK that completes the beam-level training.

Figure 15-35 illustrates the message flow for a successful SAS beam-level training process.

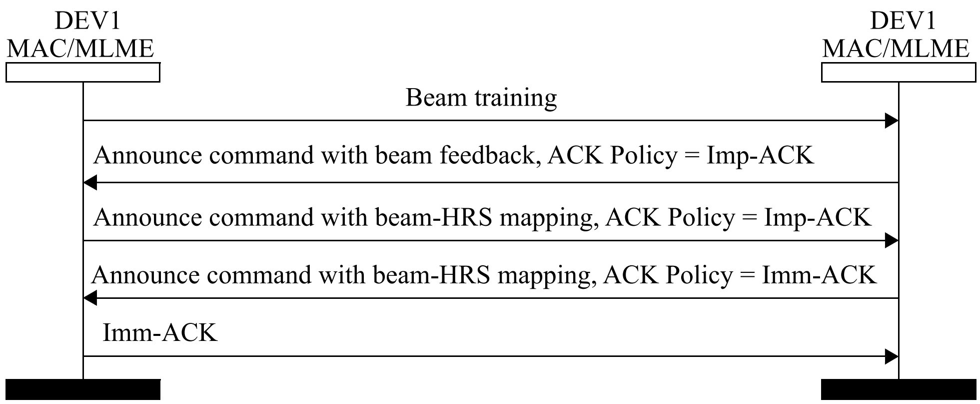

Figure 15-35—SAS beam-level training process

15.5.3 Beam tracking

To improve connectivity, a beam tracking phase is provided. When tracking is enabled, beams within a sector are further grouped into HRS clusters and tracked, described as follows. Transmission of clusters and beams within a cluster shall follow the order outlined in 15.5.2.3. During tracking, each DEV tracks the best and second best clusters.

The tracking phase is used to achieve higher beam resolution and to track the changes in the beam former and combiner vectors due to channel characteristics variability over time. In the more general case, the beam former and combiner directions (patterns) need to be adjusted dynamically to achieve optimal link quality. Tracking is enabled by clustering, as described in 15.2.5. The best beam and its adjacent beams that are to be tracked shall be grouped as the best cluster. Accordingly the second best beam and its adjacent beams shall be grouped into the second best cluster.

Tracking the best cluster and second best cluster is performed quasi-periodically as defined by the tracking frequency in the Mapping IE. The tracking frequency of the best cluster of HRS beams is higher than the tracking frequency of the second best cluster of HRS beams.

Tracking shall take place in the CTA allocated to data transfer from DEV1 to DEV2. Each data frame sent from DEV1 to DEV2 with the Beam Tracking field in the PHY header set to one shall be followed by a BBIFS followed by one or more high resolution beam training (HT) sequences sent in the HRS beams (directions) identified during the beam-level tracking. In the following, such a frame is referred to as a *tracking frame*. Not every frame sent by DEV1 is a tracking frame and the frequency at which tracking frames are sent is left to the implementer.

The odd tracking frame is used to estimate the best cluster and the even tracking frame is used to estimate the second best cluster. The high-resolution beam training sequence frame shall be formatted as illustrated in Figure 15-36.

HRS training cycle #0	HRS training cycle #1	…	HRS training cycle # $L^{(1,t)}_{1+mod(m,2)}-1$	SIFS

Figure 15-36—High resolution beam training (HT)

The HRS training cycles shall be formatted as illustrated in Figure 15-37 and Figure 15-38, respectively.

HT #0 $H^{(1,t)}_{1,l^{(1,t)}} \rightarrow H^{(2,r)}_{1,0}$	HT #1 $H^{(1,t)}_{1,l^{(1,t)}} \rightarrow H^{(2,r)}_{1,1}$	…	HT #$L_1^{(2,r)}-1$ $H^{(1,t)}_{1,l^{(1,t)}} \rightarrow H^{(2,r)}_{1,L_2^{(2,r)}-1}$	BBIFS

Figure 15-37—HRS training cycle #$l^{(1,t)}$ when mod(m, 2) = 0

HT #0 $H^{(1,t)}_{2,l^{(1,t)}} \rightarrow H^{(2,r)}_{2,0}$	HT #1 $H^{(1,t)}_{2,l^{(1,t)}} \rightarrow H^{(2,r)}_{2,1}$	…	HT #$L_1^{(2,r)}-1$ $H^{(1,t)}_{2,l^{(1,t)}} \rightarrow H^{(2,r)}_{2,L_2^{(2,r)}-1}$	BBIFS

Figure 15-38—HRS training cycle #$l^{(1,t)}$ when mod(m, 2) = 1

For DEVs operating in the low-latency mode, as detailed in 7.11.2, only a single high-resolution beam training sequence shall be transmitted in each of the tracking frames as shown in Figure 15-39. The order in which the HRS beams are interrogated remain the same as above. It is worth noting that the low-latency tracking from DEV1 to DEV2 is distributed over $L(1,t) \times L(1,r) + L(2,t) \times L(2,r)$ tracking frames.

Frame	BBIFS	HRS beam HT

Figure 15-39—Frame with beam tracking field set to one for the low-latency case

The tracking frame that interrogates the last beam in either cluster index may be sent with ACK policy set to Imp-ACK and the Last HRS Beam field in the PHY header set to one. The frequency with which this is done is implementation dependent. If DEV2 does not have results indicating that a change in transmit beam is desirable, DEV2 shall respond with an Imm-ACK and tracking continues. If, however, DEV2 has results indicating that DEV1 has a better choice of transmit beam, DEV2 shall respond with a feedback of the results in an empty data frame with ACK policy set to Imm-ACK. Upon receiving any results, DEV1 shall acknowledge the results with an Imm-ACK still using the old transmit beam. It shall then continue the data exchange using the new transmit beam.

If both DEV1 and DEV2 support best and second best beam, switching between best and second best beam will be then triggered by the tracking results at DEV2 if the DEV2 finds the LQI of receive beam is lower than the required threshold and so cannot continue its data exchange anymore. The threshold shall be decided by implementers. If DEV2 does not have results indicating that a change in receive beam is desirable, DEV2 shall respond with an Imm-ACK and tracking continues.

If, however, DEV2 has results indicating that the receive beam cannot be used anymore, DEV2 shall switch to the second best beam immediately and then respond with an empty data frame with ACK policy set to Imm-ACK. Upon receiving any results, DEV1 shall acknowledge the results with an Imm-ACK still using the old transmit beam. It shall then continue the data exchange using the new receive beam.

If both the best and second best beam cannot be used for data exchange, the beam-forming procedure will be restarted after a waiting timer is expired (the timer value shall be defined in MLE).

The switching beams sequence is shown in Figure 15-40.

DEV2→DEV1 Empty frame with Imm-ACK & feedback results	DEV1→DEV2 Imm-ACK (old beam set)	SIFS	DEV2 and DEV1 continue data exchange using new beam set	SIFS

Figure 15-40—Switching beam sequence

If the switching beams takes place, than it should be understood to the two DEVs that the old cluster is automatically replaced by the new cluster of the same size and identified by the new beam center.

The tracking and switching process from DEV2 to DEV1 is implemented in a similar way and can be run independently of tracking and switching from DEV1 to DEV2.

15.6 On-demand beam forming

On-demand beam forming takes place in the CTA allocated to the DEV. DEV1 shall reserve a CTA for the special purpose of beam-forming acquisition. The sector-level training, as described in 15.5.2.2, shall occur first, followed by the beam-level training, as described in 15.5.2.3.

15.7 Proactive beam forming

In the proactive beam forming, the sector training shall be performed according to sector-level training in 15.5.2.2 and shall take place in the sector training section of the beacon part of the superframe. The beacon structure is shown in Figure 15-41.

Quasi-omni beacon section	PNC quasi-omni tracking section	Sector-level training section

Figure 15-41—Beacon structure for proactive beam forming

The message exchange following the sector training, as specified in 15.5.2.2, and the beam-level tracking, as specified in 15.5.2.3, shall take place in the beam-forming CTA allocated to the PNC and DEV.

15.8 TSD

TSD may be used in case the PNC is capable of multiple directional antennas that are switched one to another and the DEV is capable of the antenna with omni. The TSD capability is indicated in the TSD supported field in Capability IE, as described in 6.4.12.

If the PNC sends quasi-omni beacons, then the association procedure of TSD shall follow the directional association, as described in 7.8.6.4. During data transmission from the PNC to the DEV in a CTA, TSD may be used. When TSD is used, the DEV shall check the LQI from the data frame and shall compare the LQI with TXDiversityTheshold, as described in 5.3.19. If the LQI is lower than TXDiversityTheshold, the DEV shall send the Announce command to the PNC with the TSD IE, as defined in 6.4.30, which shall set the Mode field to one and set the Transmit Direction Index field to the next transmit antenna of the PNC. If the PNC successfully receives the Announce command, the PNC shall respond by sending an Announce command to the DEV with TSD IE, which shall set the Mode field to three and the Transmit Direction Index field to the corresponding transmit direction of the PNC. The PNC may transmit data frames after switching the transmit direction to the new transmit direction.

The PNC may announce the TSD switching request time to the DEV by sending an Announce command with TSD IE with the Mode field to zero and the TSD Switching Request Time field set to the new value. The DEV shall feedback to the PNC periodically by using the Announce command according to TSD switching request time whether the current transmit direction of the PNC should be changed or not, where the Announce command includes TSD IE with the Mode field to one and the Transmit Direction Index field set to the corresponding transmit direction of the PNC.

If the DEV does not know the PNC transmit direction index, the DEV may send the Announce command with TSD IE to the PNC with the mode field set to two. The PNC responds with an Announce command with TSD IE with the mode field set to three and set the Transmit Direction Index field set to the current antenna index of the PNC.

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this recommended practice. Reference to these resources is made for informational use only.

[B1] ETS 300-328:1996, Radio Equipment and System (RES); Wideband transmission systems; Technical characteristics and test conditions for data transmission equipment operating in the 2,4 GHz ISM band and using spread spectrum modulation techniques.[19]

[B2] IEEE Std 802.11™-2012, IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.[20, 21]

[B3] IEEE Std 802.11b™-1999 (Amendment to ANSI/IEEE Std 802.11, 1999 Edition), IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz Band.

[B4] IEEE Std 802.15.1™-2005, IEEE Standard for Information technology—Local and metropolitan area networks—Specific requirements. Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPANs).

[B5] IEEE Std 802.15.2™-2003, IEEE Recommended Practice for Information technology—Local and metropolitan area networks—Specific Requirements—Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands.

[B6] IEEE Std 1363™-2000, IEEE Standard Specifications for Public-Key Cryptography.

[B7] IETF RFC 1157:1990, A Simple Network Management Protocol (SNMP).[22]

[B8] Milewski, A., "Periodic Sequences with Optimal Properties for Channel Estimation and Fast Start-Up Equalization," *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 426–431, Sept. 1983.

[B9] Peterson, William Wesley, and E. J. Weldon, *Error Correcting Codes*, Second Edition, 1972.

[B10] Ungerboeck, G., "Channel Coding with Multilevel/Phase and Signals," *IEEE Transactions on Information Theory*, vol. 28, Jan. 1982.

[19]ETS publications are available from the European Telecommunications Standards Institute (http://www.etsi.org/).
[20]IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).
[21]The IEEE standards or products referred to in Annex A are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.
[22]IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

Annex B

(normative)

Frame convergence sublayer

B.1 Generic convergence sublayer

The frame convergence sublayer (FCSL) may contain one or more convergence sublayers [e.g., EtherType Protocol Discrimination (EPD),[23] IEEE 1394™, Universal Serial Bus (USB)] as illustrated in Figure B.1. The FCSL provides the following functions:

a) Receiving protocol data units (PDUs) from upper protocol layers via the appropriate FCSL service access point (FCSL-SAP)
b) Classifying PDUs received from upper protocol layers according to a classification rule set
c) Delivering each classified PDU to the medium access control SAP (MAC-SAP)
d) Receiving PDUs from peer FCSLs via the MAC-SAP
e) Delivering received PDUs to the upper protocol layers via the appropriate FCSL-SAP

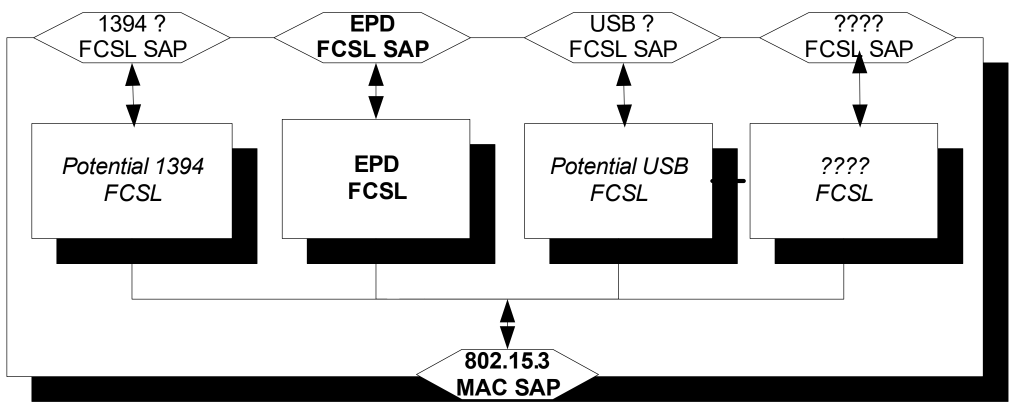

Figure B.1—IEEE 802.15.3 frame convergence sublayer model

An implementer is allowed to send IntServ packages and define IntServ policy functions in the FCSL, but these capabilities are outside of the scope of this standard. The FCSL also allows other quality of service (QoS) services to be supported, e.g., IEEE Std 1394 and others, but these are also outside of the scope of this standard.

The relationship between an FCSL and the rest of the IEEE 802.15.3 protocol entities is illustrated in Figure 5-1.

Because IEEE Std 802.15.3 supports more than one FCSL, a mechanism is required to allow the MAC to deliver each received data frame to the correct FCSL. For instance, an IEEE 1394 FCSL would not correctly process a frame from an EPD FCSL. The Multi-protocol Data frame and Data Header ID field are used to determine the correct higher layer to pass the frame to for processing.

[23]The EPD function is described in IEEE Std 802.

B.2 EPD FCSL PDU classification

The EPD FCSL PDU classification process maps each FCSL PDU to a specific stream index. Each stream index has associated with it a set of QoS characteristics. Consequently, after classification, each FCSL PDU will be delivered using the QoS parameters specified for the stream index. This includes asynchronous data that is mapped to the asynchronous stream index.

The classification process uses one or more classification parameter sets to analyze each frame entering the FCSL. In the case of an EPD classifier, the classification set includes the stream index and protocol specific parameters such as destination MAC address, source MAC address, and optionally a UserPriority parameter. If an FCSL PDU, received from an upper layer protocol, matches the specified protocol specific parameters, it is then sent to the MAC-SAP for delivery using the stream indicated by the stream index. If the FCSL PDU does not match the specified protocol parameters, either the frame may be delivered using a default stream index (i.e., asynchronous stream index) or the frame may be discarded. The policy for deciding the method used to handle a frame in this instance is outside of the scope of this standard. Figure B.2 provides a graphical representation of the entities involved.

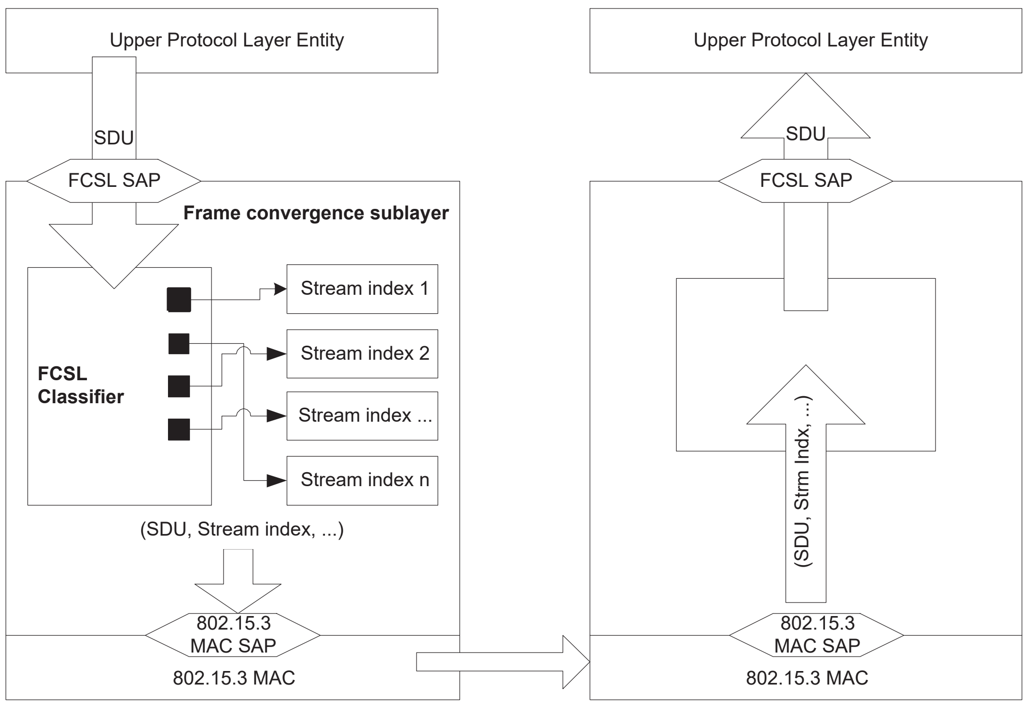

Figure B.2—Classifications and stream index mapping

In the case where more than one classification parameters set is available, the classification process first shall use the classification parameters set containing the highest valued UserPriority parameter. If no match is found with the first classification parameters set, the next highest UserPriority parameters set will be applied. This process will repeat itself until either the incoming frame is properly matched and assigned to a specific stream index for subsequent delivery, or there are no more classification parameters sets available and the incoming frame is either discarded or delivered with a default delivery QoS (i.e., Best Effort).

B.3 EPD FCSL

B.3.1 Overview

The EPD FCSL shall:

— Receive upper layer PDUs via the EPD FCSL SAP.
— Classify each received PDU according to the following attributes:
  1) Destination address
  2) Source address
  3) UserPriority, an IEEE 802.1Q™ hierarchical QoS scheme
— Map each received PDU to a specific StreamIndex according to the rules of the EPD FCSL classifier.
— Map each received PDU source and destination address to a corresponding IEEE 802.15.3 SrcID and DestID.
— Deliver each valid frame service data unit (SDU) to the MAC-SAP.
— Receive frame SDUs from the MAC-SAP.
— Deliver received frame SDUs to the upper layer EPD via the EPD FCSL SAP.

B.3.2 EPD FCSL QoS support

The EPD FCSL shall support one or both of these QoS schemes:

a) Best Effort: This is the default QoS that shall be supported. All PDUs are handled the same, i.e., no QoS guarantees are provided regarding delivery of the received PDU.
b) Hierarchical IEEE 802.1Q QoS UserPriority scheme.

The IEEE 802.1Q UserPriority scheme, which describes up to eight (0 to 7) different QoS levels, may be included in the rule set for the EPD FCSL classifier. Each of the eight different QoS priorities is described in Table B.1.

Table B.1—Traffic types

UserPriority	Traffic type	Used for	Comments
0 (default)	Best effort (BE)	Asynchronous data	Default piconet traffic
1	Background (BK)	Asynchronous data	Bulk transfers
2	Excellent effort (EE)	Isochronous data	For valued customers
3	Critical applications (CA)		Guaranteed minimum bandwidth
4	Video (VI)	Isochronous data	< 100 ms delay and jitter
5	Voice (VO)	Isochronous data	< 10 ms delay and jitter
6	Internetwork control (IC)		Large networks comprising separate administrative domains
7	Network control (NC)		Maintenance of network infrastructure

B.3.3 Data entity inter-relationships

Figure B.3 illustrates the relationship among the SDU, classification parameters set, connection, and PDU entities. These entities and the underlying protocol mechanisms that establish these entities and their relationship with each other are key to enabling support for QoS delivery of PDUs from one MAC entity to another.

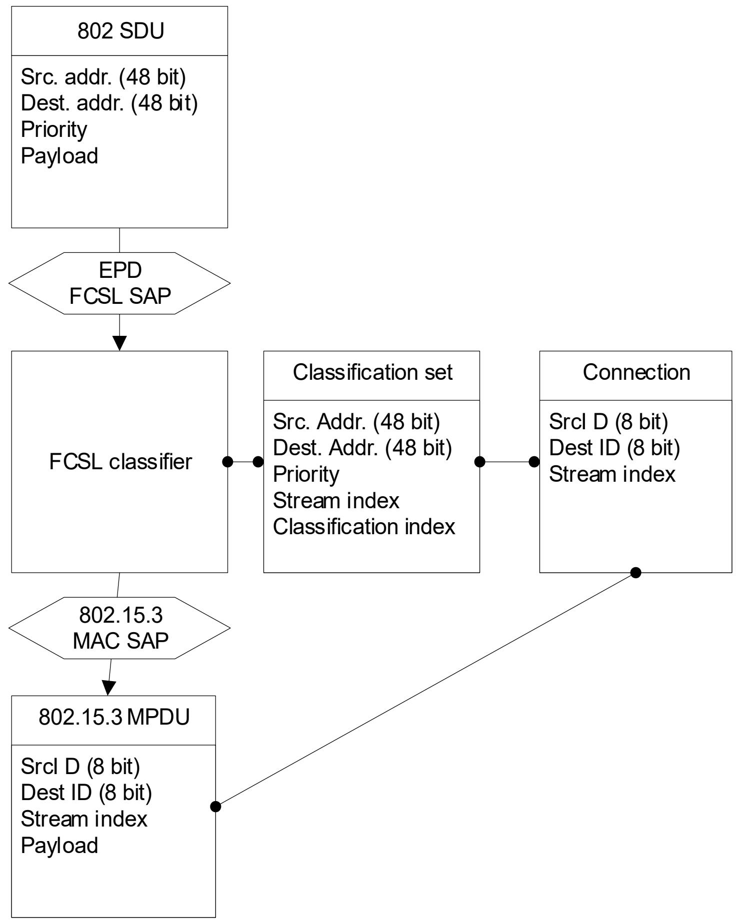

Figure B.3—Data entity inter-relationship model

B.4 EPD FCSL SAP

B.4.1 Overview

The IEEE 802.15.3 MAC supports the following service primitives:

— MA-UNITDATA.request
— MA-UNITDATA.indication
— MA-UNITDATA-STATUS.indication

The definitions of the primitives and the specific parameter value restrictions imposed by IEEE 802.15.3 are given in the following subclauses.

B.4.2 MA-UNITDATA.request

This primitive requests a transfer of an MAC service data unit (MSDU) from a local logical link control (LLC) sublayer entity to a single peer LLC sublayer entity, or multiple peer LLC sublayer entities in the case of broadcast or multicast addresses. The parameters of the primitive are as follows:

```
MA-UNITDATA.request              (
                                 SourceAddress,
                                 DestinationAddress,
                                 RoutingInformation,
                                 Data,
                                 UserPriority,
                                 ServiceClass
                                 )
```

The SourceAddress parameter specifies a MAC address from which the MSDU is being transferred.

The DestinationAddress parameter specifies a MAC address to which the MSDU is to be transferred.

The RoutingInformation parameter specifies the route desired for the data transfer (a null value indicates that source routing is not to be used). For IEEE Std 802.15.3, the RoutingInformation parameter shall be null.

The Data parameter specifies the MSDU to be transmitted by the MAC sublayer entity. For IEEE Std 802.15.3, the length of the MSDU shall be less than or equal to *pMaxTransferUnitSize*.

The UserPriority parameter specifies a prioritized QoS request for the MSDU transfer. IEEE Std 802.15.3 allows 8 integer values between 0 and 7 inclusive for directly indicating prioritized QoS.

The ServiceClass parameter specifies the service class desired for the data unit transfer. For IEEE Std 802.15.3, the ServiceClass parameter shall be null.

On receipt of this primitive the FCSL entity determines whether it is able to fulfill the request according to the requested parameters.

If the FCSL entity cannot fulfill the request according to the requested parameters, it discards the request and indicates the action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive that describes the reason for its inability to fulfill the request.

If the FCSL entity is able to fulfill the request according to the requested parameters, it appends all MAC specified fields that are unique to IEEE Std 802.15.3 to the data parameter, passes the properly formatted frame to the lower layers for transfer to peer FCSL entity or entities, and indicates the action to the LLC sublayer entity using an MA-UNITDATA-STATUS.indication primitive with TransmissionStatus set to “successful,” as described in B.4.4.

B.4.3 MA-UNITDATA.indication

This primitive indicates the transfer of an MSDU from the FCSL entity to the LLC sublayer entity. In the absence of error, the contents of the data parameter are logically complete and unchanged relative to the data parameter in the associated MA-UNITDATA.request primitive. The semantics of the primitive are as follows:

```
MA-UNITDATA.indication              (
                                    SourceAddress,
                                    DestinationAddress,
                                    RoutingInformation,
                                    Data,
                                    ReceptionStatus,
                                    ServiceClass
                                    )
```

The SourceAddress parameter is an individual address as specified by the mapping of the SrcID field of the incoming frame to the corresponding device (DEV) address.

The DestinationAddress parameter is a MAC address as specified by the mapping of the DestID field of the incoming frame to the corresponding DEV address.

The RoutingInformation parameter specifies the route that was used for the data transfer. For IEEE 802.15.3, this field shall be set to null.

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

The ReceptionStatus parameter indicates the success or failure of the received frame for those frames that IEEE 802.15.3 reports via an MA-UNITDATA.indication. This FCSL only reports success as all failures of reception are discarded without generating MA-UNITDATA.indication.

The ServiceClass parameter specifies the receive service class that was used for the data unit transfer. For IEEE 802.15.3, the ServiceClass parameter shall be null.

The MA-UNITDATA.indication primitive is passed from the FCSL entity to the LLC sublayer entity to indicate the arrival of a frame at the local FCSL entity. Frames are reported only if they are validly formatted, received without error, received with valid security properties according to the security policy at the local FCSL entity, and their destination address designates the local FCSL entity.

B.4.4 MA-UNITDATA-STATUS.indication

This primitive has local significance and provides the LLC sublayer with status information for the immediately preceding MA-UNITDATA.request primitive. The semantics of the primitive are as follows:

```
MA-UNITDATA-STATUS.indication       (
                                    SourceAddress,
                                    DestinationAddress,
                                    TransmissionStatus,
                                    ProvidedUserPriority,
                                    ProvidedServiceClass
                                    )
```

The SourceAddress parameter is a MAC address, as specified in the associated MA-UNITDATA.request primitive.

The DestinationAddress is a MAC address, as specified in the associated MA-UNITDATA.request primitive.

The TransmissionStatus parameter is used to pass status information back to the local requesting LLC sublayer entity. IEEE Std 802.15.3 specifies the following values for TransmissionStatus when delivery of the MSDU is attempted:

0—Successful
1—Excessive data length
2—Non-null source routing
3—Unsupported UserPriority (for UserPriority values other than an integer value between 0 and 7 inclusive)
4—Undeliverable (no piconet available)
5—Undeliverable (the local MAC sublayer entity does not have the required credentials or other security data to transmit the frame)
6—Undeliverable (channel conditions are too severe)

The ProvidedUserPriority parameter specifies the user priority that was used for the associated data unit transfer, as defined in B.4.2.

The ProvidedServiceClass shall be null for IEEE Std 802.15.3.

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

B.5 Stream SAP

B.5.1 Overview

The Stream SAP is used to transfer time sensitive data streams. Examples of the data intended for this interface include audio and video, particularly uncompressed audio and video. The primitives defined for this SAP are listed in Table B.2.

Table B.2—Stream SAP primitives

Name	Request	Confirm	Indication	Response
STREAM_INITIATE	B.5.2.2	B.5.2.3	B.5.2.4	—
STREAM_MODIFY	B.5.2.5	B.5.2.6	—	—
STREAM_END	B.5.2.7	B.5.2.8	B.5.2.9	—
STREAM_DATA	B.5.3.2	B.5.3.3	B.5.3.4	—

B.5.2 Stream creation, modification, and deletion

B.5.2.1 Overview

These primitives are used to start, change, or stop a stream connection. The parameters for the primitives are defined in Table B.3.

Table B.3—STREAM_INITIATE, STREAM_MODIFY and STREAM_END primitive parameters

Name	Type	Range	Description
RequestID	Integer	0–255	A unique value created by the originating DEV to match the request primitive to the response primitive.
TargetAddress	MAC address	Any valid MAC address, as defined in 6.1	The address of the target of the primitive.
SourceAddress	MAC Address	Any valid MAC address, as defined in 6.1	The address of the Source of the stream.
StreamIndex	Integer	Any valid stream index, as defined in 6.2.5	The index of the stream that was created or the index of the stream to be modified or ended.
MinThroughput	Integer	$(1 – 2^{64}) – 1$	The minimum required throughput in bits per second at the MAC SAP.
DesiredThroughput	Integer	$(1 – 2^{64}) – 1$	The desired throughput in bits per second at the MAC SAP.
MaxTransmitDelay	Duration	$(1 – 2^{64}) – 1$	The maximum delay in microseconds.
TypicalFrameSize	Integer	0–*pMaxFrameBodySize*	The typical size, in octets, of an MSDU that would be presented to the MAC SAP.
SecMode	Boolean	TRUE, FALSE	Indicates if security is applied to the stream.
ReliabilityExponent	Integer	0–31	The negative power of 10 that is the that is the maximum frame error rate (FER) including retries and frames lost to MaxTransmitDelay. For example, a value of 4 corresponds to an FER < $10^{–4}$. If the value of the parameter is zero, then the parameter is ignored.
TimeOut	Integer	0–65535	The time in milliseconds for the primitive to complete.
AvailableThroughput	Integer	$(1 – 2^{64}) –1$	The estimate of the throughput in bits per second available in the allocated stream.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the request.
ReasonCode	Enumeration	REQUEST_TIMEOUT, NOT_ASSOCIATED, TARGET_UNAVAILABLE, TERMINATED_BY_PNC, TERMINATED_BY_DEST, INVALID_STREAM_INDEX, TRANSMIT_DELAY_ UNSUPPORTED, RESOURCES_ UNAVAILABLE, OTHER	The reason for a ResultCode of FAILURE.

B.5.2.2 STREAM_INITIATE.request

This primitive is used to set up a data stream. The semantics of this primitive are as follows:

```
STREAM_INITIATE.request             (
                                    RequestID,
                                    TargetAddress,
                                    MinThroughput,
                                    DesiredThroughput,
                                    MaxTransmitDelay,
                                    TypicalFrameSize,
                                    SecMode,
                                    ReliabilityExponent,
                                    TimeOut
                                    )
```

The primitive parameters are defined in Table B.3.

B.5.2.3 STREAM_INITIATE.confirm

This primitive is used to report the result of a request to set up a data stream. The semantics of this primitive are as follows:

```
STREAM_INITIATE.confirm             (
                                    RequestID,
                                    StreamIndex
                                    AvailableDataRate,
                                    ResultCode,
                                    ReasonCode
                                    )
```

The primitive parameters are defined in Table B.3.

B.5.2.4 STREAM_INITIATE.indication

This primitive is used to report that the creation of a stream by another DEV. The semantics of this primitive are as follows:

```
STREAM_INITIATE.indication          (
                                    SourceAddress,
                                    StreamIndex
                                    )
```

The primitive parameters are defined in Table B.3.

B.5.2.5 STREAM_MODIFY.request

This primitive is used to modify the parameters of an existing data stream. The semantics of this primitive are as follows:

```
STREAM_MODIFY.request              (
                                   RequestID,
                                   StreamIndex,
                                   MinThroughput,
                                   DesiredThroughput,
                                   MaxTransmitDelay,
                                   TypicalFrameSize,
                                   SecMode,
                                   ReliabilityExponent,
                                   TimeOut
                                   )
```

The primitive parameters are defined in Table B.3.

B.5.2.6 STREAM_MODIFY.confirm

This primitive is used to report the result of a request to modify the parameters of an existing data stream. The semantics of this primitive are as follows:

```
STREAM_MODIFY.confirm              (
                                   RequestID,
                                   StreamIndex
                                   AvailableDataRate,
                                   ResultCode,
                                   ReasonCode
                                   )
```

The primitive parameters are defined in Table B.3.

B.5.2.7 STREAM_END.request

This primitive is used to set up a data stream. The semantics of this primitive are as follows:

```
STREAM_END.request                 (
                                   StreamIndex,
                                   TimeOut
                                   )
```

The primitive parameters are defined in Table B.3.

B.5.2.8 STREAM_END.confirm

This primitive is used to report the result of a request to set up a data stream. The semantics of this primitive are as follows:

```
STREAM_END.confirm                    (
                                      StreamIndex
                                      ResultCode,
                                      ReasonCode
                                      )
```

The primitive parameters are defined in Table B.3.

B.5.2.9 STREAM_END.indication

This primitive is used to report that the creation of a stream by another DEV. The semantics of this primitive are as follows:

```
STREAM_END.indication                 (
                                      StreamIndex,
                                      ReasonCode
                                      )
```

The primitive parameters are defined in Table B.3.

B.5.3 Stream data interface

B.5.3.1 Overview

These primitives are used to send stream data and report the reception of stream data. The parameters for the primitives are defined in Table B.4.

Table B.4—STREAM_DATA primitive parameters

Name	Type	Range	Description
RequestID	Integer	0–255	A unique value created by the originating DEV to match the request primitive to the response primitive.
SourceAddress	MAC Address	Any valid MAC address, as defined in 6.1	The address of the source of the data.
DestinationAddress	MAC Address	Any valid MAC address, as defined in 6.1	The address of the destination of the data.
StreamIndex	Integer	Any valid stream index, as defined in 6.2.5	The index of the stream that was created or the index of the stream to be modified or ended.
TransmitTimeout	Integer	$(1 - 2^{32}) - 1$	The maximum allowed delay in microseconds from when the data is presented to the SAP until the frame has finished transmission and the acknowledgment, if required, is received.

Table B.4—STREAM_DATA primitive parameters (*continued*)

Name	Type	Range	Description
ConfirmRequested	Boolean	TRUE, FALSE	Indicates when a confirm primitive is required for the request.
Length	Integer	$0 – (1 – 2^{32}) – 1$	The length of the Data in octets.
Data	Octet string		The information to be sent in the stream connection.
DataType	Enumeration	VIDEO, AUDIO, DATA, UNCOMPRESSED_VIDEO, UNCOMPRESSED_AUDIO	Indicates the type of data that is sent in the stream.
UepAllowed	Boolean	TRUE, FALSE	Indicates if UEP is allowed for the transmission of the data.
ErrorsFreeData	Boolean	TRUE, FALSE	Indicates if the data that was received contains errors or is free from errors.
InterlacedFieldIndi-cation	Enumeration	TOP_FIELD, BOTTOM_FIELD, NOT_INTERLACED	Indicates if the pixels belong to the top or bottom field of an interlaced image or that the image is not interlaced.
VideoFrameNumber	Integer	As defined in 6.2.12.2.5	A sequential numbering of the video frames that are being transferred.
HPosition	Integer	As defined in 6.2.12.2.5	The horizontal position of the first pixel in the MSDU.
VPosition	Integer	As defined in 6.2.12.2.5	The vertical position of the first pixel in the MSDU.
ResultCode	Enumeration	SUCCESS, FAILURE	Indicates the result of the request.
ReasonCode	Enumeration	TRANSMIT_TIMEOUT, NOT_ASSOCIATED, TARGET_UNAVAILABLE, INVALID_STREAM_INDEX, OTHER	The reason for a ResultCode of FAILURE.

B.5.3.2 STREAM_DATA.request

This primitive is used to send stream data to another DEV. The semantics of this primitive are as follows:

```
STREAM_DATA.request                 (
                                    RequestID,
                                    StreamIndex,
                                    TransmitTimeout,
                                    ConfirmRequested
                                    Length,
                                    Data,
                                    DataType,
                                    UepAllowed,
                                    InterlacedFieldIndication,
                                    VideoFrameNumber,
                                    HPosition,
                                    VPosition
                                    )
```

The primitive parameters are defined in Table B.4.

B.5.3.3 STREAM_DATA.confirm

This primitive is used to send stream data to another DEV. The semantics of this primitive are as follows:

```
STREAM_DATA.request                  (
                                     RequestID,
                                     StreamIndex,
                                     TransmitDelay,
                                     ResultCode,
                                     ReasonCode
                                     )
```

The primitive parameters are defined in Table B.4.

B.5.3.4 STREAM_DATA.indication

This primitive is used to send stream data to another DEV. The semantics of this primitive are as follows:

```
STREAM_DATA.request                  (
                                     SourceAddress,
                                     DestinationAddress,
                                     StreamIndex,
                                     Length,
                                     Data,
                                     DataType,
                                     ErrorFreeData,
                                     InterlacedFieldIndication,
                                     VideoFrameNumber,
                                     HPosition,
                                     VPosition
                                     )
```

The primitive parameters are defined in Table B.4.

Annex C

(informative)

Security considerations

C.1 Background assumptions

C.1.1 General

All security solutions rely on assumptions about devices (DEVs) and the capabilities of potential attackers to thwart possible threats. The goals of mode 1 security are that only authorized DEVs will be able to join a secure piconet or a secure pairnet and that communication is restricted to authorized DEVs.

C.1.2 Physical assumptions

The following assumptions are made about the physical environment for the piconet or pairnet. The physical constraints help to determine the security architecture.

— **Open communications medium:** Since the data being transmitted will be able to be received by any other entity that is sufficiently close and has a sufficiently good receiver, it is assumed that transmissions are be heard by entities that are not part of the piconet or pairnet.
— **Low cost:** Like all other components of a DEV, security is provided with careful attention to cost.
— **Dynamic group membership:** DEVs are expected to be mobile and it is therefore assumed that the DEVs enter or exit the network at any time.
— **No access to external networks:** Security solutions need to be effective without access to external networks.
— **Bandwidth:** Since IEEE 802.15.3 piconets or pairnets provide high data rates, reasonable amounts of bandwidth overhead due to security are acceptable.
— **Computational power:** The DEVs are assumed to have very little computational power with only a small portion of that available for cryptographic computations.
— **Memory:** It is assumed that the low-end DEVs implementing IEEE Std 802.15.3 will have little memory available for security.

C.1.3 Network assumptions

The following assumptions are made about the network structure of the piconet or pairnet. The network constraints help to determine the security architecture.

— **Network size:** Although there is a fixed upper bound of fewer than 255 DEVs in a piconet, the security solution might need to scale to arbitrary sets of DEVs, rather than to a fixed set of limited size. DEVs join and leave the network in an ad hoc fashion and in some cases will not have previously communicated with the other DEV(s).
— **Controller:** One DEV, the piconet coordinator (PNC), or pairnet coordinator (PRC) has the role of managing message control and entry into the piconet or pairnet.
— **Dynamic controller:** The PNC is assumed to have the ability to leave the network or hand over the PNC role to other DEVs.
— **Device relationships:** The wide array of use cases describe multiple models for the pre-existing relationship of DEVs in the piconet or pairnet. It is assumed that DEVs could have pre-existing security relationships or that they have never met and that both types of relationship could exist within a single piconet or pairnet.

C.1.4 Attack model assumptions

In order to make statements about the effectiveness of security measures, it is necessary to describe the capabilities of the attackers and the nature of the attackers.

— **Computational capabilities:** It is assumed that the attacker has state-of-the-art technologies to perform rapid computations.
— **Listening capabilities:** It is assumed that the attacker is within listening range of the DEVs in the piconet or pairnet and understands the communication mechanism.
— **Broadcast capabilities:** It is assumed that the attacker has sophisticated broadcasting equipment that is able to synchronize with the piconet or pairnet and transmit data for the DEVs in the piconet or pairnet at the appropriate time.
— **Security setup:** The security setup for the DEVs occurs either before entry into the piconet or after the piconet has been established. No assumptions are made about the presence of attackers during security setup.

C.1.5 Security key lifecycle issues

C.1.5.1 Key lifecycle

In order to maintain security, care needs to be taken to protect keys from exposure for their entire lifetime. This standard provides the necessary methods for good key lifecycle management. The requirements for key lifecycle management depend on the type of application.

C.1.5.2 Membership lifecycle

The PNC or another DEV is able to require that each DEV with which it has a secure relationship periodically transmit a secure frame using the management key to be certain that the DEV is still in the piconet. If no secure frames are being transmitted by the target DEV, the PNC or PRC or requesting DEV is able send a secure Probe Request command requesting an IE from the target DEV. If the target DEV does not respond with a secure frame within a period of time determined by the PNC or PRC or requesting DEV, the PNC or PRC or requesting DEV will assume that the target DEV is no longer present and disassociate or terminate the secure relationship with the target DEV.

C.1.5.3 Group membership change rekey

Only DEVs that are currently members of the piconet or pairnet are allowed to generate, read, or modify piconet or pairnet data. This implies that when a DEV joins or leaves the piconet or pairnet, the currently active group keys need to be changed. Changes in the group membership key are described in 8.3.3.

C.2 Claimed security services

C.2.1 General

Each of the protocols defined in Clause 8 are designed to offer specific security services. These security services are consistent with the security services required by the IEEE 802.15.3 security model. C.2.2 through C.2.5 describe the security services provided by each protocol and the method implemented to provide the security service.

C.2.2 Beacon protection protocol

Table C.1 specifies the security services provided by the beacon protection protocol specified in 8.4.3 along with a description of the method employed to provide the security service.

Table C.1—Beacon protection security services

Security service	Method provided
Communication of current time token to the DEVs in the piconet or pairnet.	The PNC or PRC increments the time token for each superframe and protects it using the current group key. The integrity protection on the beacon and the storage of the previous time token allows each DEV to determine that the time token is fresh.
Indication of the identity of the PNC or PRC to the DEVs in the piconet. or pairnet	If PNC handover has not occurred, the DEV address of the current PNC appears in the beacon. If PNC handover has occurred, the DEV address of the new PNC appears in the beacon. The DEV address of the current PRC appears in the Beacon frame The integrity protection on the beacon and the freshness from the time token allow each DEV to determine the identity of the current PNC or PRC.

C.2.3 Distribute key protocol

Table C.2 specifies the security services provided by the distribute key protocol specified in 8.4.3 along with a description of the method employed to provide the security service.

Table C.2—Key distribution security services

Security service	Method provided
Privacy protection on distributed key.	The encryption of the key with the shared key encryption key ensures that the key remains private.
Integrity protection on the distributed key.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.
Verification by the key originator that the DEV received the key.	The key originator verifies that the integrity code verifies properly and that the freshness checks succeed.

C.2.4 Key request protocol

Table C.3 specifies the security services provided by the key request protocol specified in 8.4.4 along with a description of the method employed to provide the security service.

C.2.5 Data protection protocol

Table C.4 specifies the security services provided by the data protection protocol specified in 9.2.2, along with a description of the method employed to provide the security service.

Table C.3—Key request security services

Security service	Method provided
Privacy protection on requested key.	The encryption of the key with the shared key encryption key ensures that the key remains private.
Integrity protection on the requested key.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.

Table C.4—Data protection security services

Security service	Method provided
Privacy protection on the data.	The encryption of the data with the shared encryption key ensures that the key remains private.
Integrity protection on the data.	The receiving DEV verifies that the integrity code verifies properly and that the freshness checks succeed.

C.3 Properties of the IEEE 802.15.3 security suite

C.3.1 Key usage

In general, a 128-bit advanced encryption standard (AES) key should not be used more than 2^{64} times to produce an integrity code or to encrypt a frame. If a DEV sends a frame encrypted by a key once every microsecond, it would send approximately 2^{45} frames every year. Thus, to avoid security problems, an implementation should change its management keys at least once every 2^{19} = 524 288 years. More conservative implementations that are concerned with security should change management keys at least once every millennium. Of course, after that many years, computation power will have increased dramatically and 128-bit AES keys likely will no longer be considered secure.

The security operation for pairnets is based on the Galois/counter mode (GCM) mode of the AES encryption algorithm with 128-bit key length, 128-bit integrity code, and 96-bit nonce. To avoid a birthday attack, the number of invocation of the authenticated encryption function using a given key should be limited to 248. In the worst case scenario, secure frames consist of only one AES block per frame may be transmitted at 100 Gbps throughput. In this case, total number of octets that can be encrypted using a single key is 248 * 24 octets = 252 octets. Then, maximum duration using the single key at 100 Gbps throughput is 252 * 8 bit / 1011 = 4.17 days. Since a management key is used only for command frames and command frames are not frequently transmitted, the actual lifetime of a management key is much longer than this duration.

C.3.2 Replay prevention

This standard uses a Time Token field, 6.3.1.1, and Secure Frame Counter (SFC) field, 6.2.10.3, to provide a method to detect and defeat potential replay attacks. For piconets, the SFC allows up to 65 535 frames to be sent in a single superframe or one every microsecond for the largest possible superframe. The Time Token field is 6 octets, and so it will repeat only once every $2^{48}/2^{35} = 2^{13}$ years ~ 8192 years if the PNC uses a 1 ms superframe duration.

For pairnets, the 6 octet SFC field allows up to 2^{48} frames or subframes to be sent in multiple superframes. In the worst case scenario described in C.3.1, the duration for transmitting 2^{48} frames using 6 octet SFC field is 4.17 days. A 6 octet Time Token field is used in pairnets. In the worst case scenario where a PRC keeps transmitting Beacon frames and no DEV is associated, the time token will roll over every 254 years if we assume 28.5 μs beacon interval.

For piconets, because the nonce includes the time token, a replay of one of Distribute Key Request, Distribute Key Response, Request Key, or Request Key Response commands would fail for anything other than the current superframe. A replay of one of these commands would not fail integrity code check if either

- The piconet restarts with a lower time token and so eventually the same time token will be used; or
- The time token rolls over in the current piconet (once every 8192 years for a 1 ms superframe duration) and the same security identifier (SECID) is being used by that DEV (which may be true for the management key in shared key operation).

For piconets, in the case where the command is replayed in the same superframe, the duplicate detection algorithm will discard the second occurrence sent by the attacker. For pairnets, a replay of one of Distribute Key Request, Distribute Key Response, Request Key or Request Key Response commands would fail since the 6 octet SFC is included in the secure frames and it can be used for replay detection. A replay of one of these commands would not fail integrity code check if the SFC rolls over and the same SECID is being used by that DEV.

For piconets, in the case of a piconet starting with a lower time token, the duplicate detection will fail and the integrity code will pass in the case of shared keys if the same management key and SECID are used. If higher layer mutual authentication is used, then the management keys and their SECIDs will change each time the piconet is restarted and the DEVs reauthenticate.

Suppose that the attacker waits until the piconet starts again and the desired time token occurs and replays one of the key distribution commands and it passes integrity code check and duplicate detection.

If the Distribute Key Request command is replayed, the DEV will update its current key to the distributed (old) key. It will then use this data key for communications with the attacking DEV (provided it is using the same DEVID). However, the attacking DEV will not be able to provide correct IDs for any new frames or read any encrypted data because it still does not have either of the two keys (data or management). If it previously had access to the data (i.e., it was part of the piconet as a trusted DEV), it would be able to communicate with the DEV using this key until the DEV found out that the key was "faked." If the key in the replayed Distribute Key Request command has a different SECID than the current one in the piconet, which is likely, the DEV will find out when the next beacon is received. If somehow the current PNC is using the same SECID as the replayed frame, the DEV will get integrity code errors on the beacon and will send a Request Key command to the PNC to get a new key.

If the Distribute Key Response command is replayed, the receiving DEV (the key originator (KO), but not the PNC, because the PNC does not use this command) will think that the remote DEV has had its key updated with the indicated SECID. However, if the KO does not have its Distribute Key state machine operating with a pending key distribution, the KO should drop this indication. If the KO is currently using a different SECID for the relationship, it should probably interpret this as some sort of a failure and send a Distribute Key Request command to the DEV with the current key. If the KO has a pending Distribute Key operation and receives this command with a different SECID, it would interpret it as a failure. Finally, if the KO is distributing a key to that DEV with the same SECID, it will finish the key distribution process. It is possible that the receiving DEV will not have yet received the key, and so it will then receive frames with a new SECID. This will cause that DEV to issue a Key Request command to resync with the KO. This case does not represent a security breach.

If the Request Key command is replayed, the KO will see this as a valid request for a key and sends the encrypted key to the attacking DEV. The attacking DEV, lacking the management key, will be unable to decrypt the resulting Request Key Response command. Thus, it will not be able to get the data key.

Furthermore, the attacker will not be able to repeat the request because the Time Token will have incremented in the next superframe and the replayed command will fail integrity check.

If, for some reason, the DEV accepts the new (old) key, it would begin to use it for data encryption, which will prompt the KO to register a security error and update the data keys.

The attacker, by these methods alone, has not gained access to the shared secret, but he is able to convince the DEV to take some predictable action and to use the same key, possibly with the same nonce (this is extremely unlikely due to all of the items in the nonce that change on a frame-by-frame basis).

Annex D

(informative)

Coexistence, interoperability, and interference

D.1 Interoperability

D.1.1 Overview

IEEE Std 802.15.3 does not require interoperability with any other IEEE 802 wireless standards or other wireless specifications. However, choices were made at the PHY layer to make it easier for implementers to be able to make low-cost, dual-mode radios. The IEEE 802 wireless protocols where dual-mode solutions are more easily created are as follows:

— IEEE 802.11 direct sequence spread spectrum (DSSS) and IEEE Std 802.11b [B3]
— IEEE 802.11 frequency hopping spread spectrum (FHSS)
— IEEE Std 802.15.1 [B4]

For other protocols, no specific facilities were included to enhance interoperability. However, interoperability is not precluded, and an implementer could make dual-mode radios with the following protocols:

— IEEE Std 802.11a™
— IEEE Std 802.11 infrared (IR)
— IEEE Std 802.16™

While it is possible to implement these interoperable radio modules, the details of the techniques used to accomplish this are out of the scope of this standard and so are left to the implementer.

D.1.2 Interoperability with IEEE 802.11 DSSS and IEEE 802.11b

Since IEEE Std 802.11b-1999 [B3] is a superset of IEEE 802.11 DSSS, both will be referred to in this discussion as simply IEEE 802.11b. IEEE 802.11b and IEEE 802.15.3 share the same frequency band, which makes interoperability of radio modules much simpler. Also, the IEEE 802.15.3 physical (PHY) layer uses 11 Mbaud, DQPSK modulation for the base rate, which is the same as the chip rate and modulation for IEEE 802.11b. However, IEEE 802.11b uses a Barker code, complementary code keying (CCK), or packet binary convolutional code (PBCC) as a spreading code, which is not a part of IEEE Std 802.15.3.

The IEEE 802.15.3 PHY was also chosen with the same frequency accuracy, allowing the reuse of reference frequency source and frequency synthesizers. While the IEEE 802.11b and IEEE 802.15.3 frequency plans are slightly different, the synthesizers that would normally be used in either radio would be capable of 1 MHz frequency step size and so would be capable of supporting either frequency plan. The receive/transmit (RX/TX) turnaround time is also the same for both protocols. However, the TX/RX turnaround for IEEE 802.11b is 5 µs vs. 10 µs for IEEE 802.15.3, which could have an impact on the architecture of a dual-mode radio.

To summarize, the similarities between IEEE Std 802.15.3 and IEEE Std 802.11b [B3] are as follows:

— DQPSK modulation

— 11 Mbaud symbol (chip) rate
— Frequency and symbol timing accuracy of $\pm 25 \times 10^{-6}$
— RX/TX turnaround time
— Power ramp up/down

Some of the differences include the following:

— Barker, CCK, or PBCC spreading code
— Power spectral density
— Frequency plan
— Performance criteria (e.g., sensitivity, jamming resistance)
— TX/RX turnaround time
— PHY preamble, header, frame structure
— medium access control (MAC)

D.1.3 IEEE 802.11 FHSS and IEEE 802.15.1

A narrowband, frequency hopping radio, as defined in IEEE 802.11 FHSS and IEEE 802.15.1, has many differences with the IEEE 802.15.3 radio. However, since these protocols share the same frequency, it is possible to architect dual-mode radios that would support a combination of these protocols. The design of dual-mode radios is outside of the scope of this standard.

D.2 Coexistence

D.2.1 General

IEEE Std 802.15.3 provides many facilities that allow it to coexist with other wireless protocols. The following subclauses provide an overview of the methods defined in this standard.

D.2.2 Coexistence with IEEE Std 802.11b

D.2.2.1 Overview

The IEEE 802.15.3 PHY presents the following two challenges in coexisting with IEEE Std 802.11b [B3]:

a) Both use the same frequency range.
b) IEEE Std 802.11b uses carrier sense multiple access with collision avoidance (CSMA/CA) and a polling method with the point coordination function while IEEE Std 802.15.3 uses a hybrid CSMA/CA and time division multiple access (TDMA).

IEEE 802.15.3 piconets use two access methods in the superframe; CSMA/CA during the contention access period (CAP) and TDMA during the channel time allocation period (CTAP). The CAP provides the best method of coexistence with IEEE 802.11b networks, since the CSMA/CA algorithm used in the CAP is similar to the CSMA/CA algorithm used in IEEE 802.11b, i.e., the transmitter uses a listen-before-talk mechanism. In the case of IEEE 802.11, there is more than one clear channel assessment (CCA) method allowed and some of them would not recognize an IEEE 802.15.3 frame. In this case, the IEEE 802.11b transmission might collide with IEEE 802.15.3 frames. However, an IEEE 802.11b station that implemented "energy above threshold" for CCA, i.e., CCA mode 1 or CCA mode 5 (see IEEE Std 802.11b-1999 [B2]), would signal that the medium is busy when a sufficiently strong IEEE 802.15.3 signal is present. The 2.4 GHz PHY of IEEE 802.15.3 requires energy detection as a part of the CCA process. A sufficiently strong IEEE 802.11b signal would result in the IEEE 802.15.3 device (DEV) signaling that the medium is busy, which would improve the coexistence performance.

CTAs provide the best quality of service (QoS) for IEEE 802.15.3 connections, but potentially will also cause the most coexistence problems with IEEE 802.11b products. This is because once a DEV has a CTA, the DEV transmits without using a listen-before-talk mechanism.

To address this issue, IEEE Std 802.15.3 provides the following techniques to handle coexistence with IEEE 802.11b:

— Passive scanning
— Dynamic channel selection
— Ability to request channel quality information
— Link quality and received signal strength indication (RSSI)
— Channel plan that minimizes channel overlap
— Lower transmit power
— Transmit power control
— Neighbor piconet capability

D.2.2.2 Passive scanning

All IEEE 802.15.3 piconet coordinator-capable (PNC-capable) DEVs (i.e., ACs) are required to passively scan, as described in 7.2.2, a potential channel before attempting to start a piconet, as described in 7.2.3. While detecting an IEEE 802.11b wireless local area network (WLAN) is not required, the PNC-capable DEV will, at a minimum, be looking for a channel that is relatively quiet. Passive scanning implies that the PNC-capable DEV, when starting a piconet, or other DEVs that wish to join an existing piconet will not cause interference while searching the channels.

D.2.2.3 Dynamic channel selection

The PNC will periodically request channel status information, as described in 7.13.5, from the DEVs in the piconet via the Channel Status Request command, as described in 6.5.8.2. If the PNC determines, from the number of lost frames, that the channel is having problems, as it would when an IEEE 802.11b network is present, then it would search for a new channel, as described in 7.15.2, that had a lower level of interference. If the PNC finds a channel with less interference then the PNC uses the Piconet Parameter Change IE in the beacon, as described in 6.4.7, to move the piconet to a quieter channel.

Thus, if an IEEE 802.11b network is present, the IEEE 802.15.3 piconet would change channels to avoid interfering with IEEE 802.11b.

D.2.2.4 Ability to request channel quality information

Dynamic channel selection, as described in 7.15.2, requires the ability to obtain an estimate of the interference in a channel. In the case of IEEE 802.15.3, not only does the DEV sense the channel in its area, but it is also capable of asking any other DEV to respond with its own estimate of the channel status, as described in 7.13.5. These commands indicate the frame error rate at a remote DEV. This command is useful for detecting coexistence problems in remote DEVs by the PNC or other DEVs that are unable to detect an interference environment (for example, during a passive scan).

D.2.2.5 Link quality and RSSI

The 2.4 GHz PHY specifies that a DEV returns the RSSI, as described in 11.6.6, and for the higher speed modulations, an estimate of the link quality, as described in 11.6.7. The RSSI provides an estimate of the strength of the received signal, which is useful for transmit power control. The RSSI combined with the link quality indication (LQI) provides a method to differentiate between low signal power and interference causing the loss of frames. For example, if the RSSI is low and frames are being lost, then the cause is low receive power. On the other hand, if the RSSI is relatively high, but the LQI is low, that would indicate the possibility of interference in the channel.

D.2.2.6 Channel plan that minimizes channel overlap

The channel plan for the 2.4 GHz PHY, as described in 11.2.3, balances the requirement of four simultaneous piconets with the desire to coexist with other wireless standards, such as IEEE 802.11b. To do this, two channel plans are available. If there are no IEEE 802.11b networks detected, then the high-density channel plan would be used. On the other hand, if IEEE 802.11b networks are detected, then the PNC would want to choose the IEEE 802.11b coexistence channel plan. The reason for this is that each of the two center channels in the high-density channel plan would overlap 2 IEEE 802.11b channels. The IEEE 802.11b coexistence channel plan roughly aligns the channels so that IEEE 802.15.3 operation in one of these channels would only affect a single IEEE 802.11b channel. By choosing the "quietest" one, the IEEE 802.15.3 piconet would minimize its impact on the IEEE 802.11b networks.

If an IEEE 802.15.3 PNC has selected the center channel of the IEEE 802.11b coexistence channel plan, then other IEEE 802.15.3 piconets that entered or started in the same operational area would also adopt the IEEE 802.11b coexistence channel plan, i.e., the piconets would use one of the two remaining outer channels, either channel 1 or channel 5. This would only occur when the center channel, channel 3, is occupied first. Otherwise, subsequent piconets in the same operational area would be able to use the high-density channel plan.

D.2.2.7 Lower transmit power

The IEEE 802.15.3 standard operates in the U.S. under the 47 CFR 15.249 rules, as described in 11.1, often called the "low power" rules. Under this section of Part 15, the maximum allowed transmit power is approximately 8 dBm EIRP for the IEEE 802.15.3 2.4 GHz PHY. This measurement includes the antenna gain, so a 1 dB increase antenna gain requires a 1 dB decrease in transmit power. IEEE 802.11b WLANs, on the other hand, operate under 47 CFR 15.247, which allow up to 1 W of transmit power with as much as 6 dB of antenna gain at that power level. Most IEEE 802.11b products on the market at this time operate with between 12 dBm and 18 dBm of transmit power, with unspecified antenna gain. Access points often have antennas with moderate gain, 3 dB–6 dB.

This implies that a high-power IEEE 802.15.3 implementation would operate with 7 dB to 13 dB less effective transmit power than a typical IEEE 802.11b implementation.

D.2.2.8 Transmit power control

IEEE Std 802.15.3 provides three methods for controlling transmit power. The first is that the PNC is able to set a maximum power level for the beacon, CAP and directed management channel time allocations (MCTAs). For the 2.4 GHz PHY, the lowest setting possible is 0 dBm. This allows IEEE 802.15.3 piconets to reduce the interference it creates for other networks while maintaining the operation of the piconet.

Individual DEVs in a CTA are able to request a change in the transmit power of the remote DEV for that link, as described in 7.15.3.3. The originating DEV sends a message to the target DEV that tells the target DEV the amount to change its power up or down, depending on the status of the link. Thus, two DEVs that are relatively close to each other are able to both save power and reduce the interference to other networks while maintaining a high quality link.

DEVs are also able to change their transmit power based on their own estimation of the channel. The Probe Request command, as described in 7.13.3, allows DEVs to request information from other DEVs in the piconet to assist in getting this information.

D.2.2.9 Neighbor piconet capability

The neighbor piconet capability, as described in 7.2.9, allows a DEV, which may not be fully IEEE 802.15.3 compliant, to request time to operate a network that is co-located in frequency with the IEEE 802.15.3 network. While current IEEE 802.11b radios do not implement this functionality, it is relatively easy to build IEEE 802.11b radios that could support enough of a subset of IEEE 802.15.3 to request the neighbor piconet capability. One reason that it is possible to do this is that the IEEE 802.15.3 PHY has characteristics that make it easier to build dual-mode radios (see D.1.2 for more information).

Once a dual-mode IEEE 802.11b/IEEE 802.15.3 access point has requested and received a CTA for a neighbor network, it would use the point coordination function (PCF) and network allocation vector (NAV) to set aside time for the operation of the IEEE 802.15.3 piconet, while maintaining clear operation for part of the time for the IEEE 802.11b WLAN.

D.2.3 Coexistence with IEEE 802.15.1 and IEEE 802.11 FHSS

D.2.3.1 General

Narrowband FHSS systems have very different coexistence requirements than IEEE 802.11b, which has a PHY similar to IEEE 802.15.3. For these systems, there are methods in the standard as well as methods that are implementation dependent, which enhance the coexistence performance. Within the standard, the following methods are defined that improve the coexistence with FHSS systems:

a) Lower transmit power, as described in D.2.2.7
b) Transmit power control, as described in D.2.2.8

Techniques to manage coexistence between IEEE 802.11b and narrowband interference such as IEEE 802.15.1 are addressed by IEEE Std 802.15.2-2003 [B5], which is a recommended practice for coexistence. While some clauses of the document are only relevant for IEEE 802.15.1 systems, portions of IEEE Std 802.15.2 [B5] are directly relevant to the IEEE 802.15.3 PHY, and some of the recommended practices for the MAC will also improve performance in a coexistence environment. It is up to the implementer to decide the methods that are included in their implementation.

The terminology in IEEE Std 802.15.2 [B5] describes both collaborative and non-collaborative coexistence mechanisms. Collaborative mechanisms are those that would require direct communication between the IEEE 802.15.3 system and the FHSS system it desires to coexist with. By the definition in IEEE Std 802.15.2 [B5], collaborative systems are collocated within 0.5 m of each other. Non-collaborative systems, which do not have to be collocated, are those that modify their PHY or MAC behavior by inferring the interference environment, not by direct communication between the systems.

D.2.3.2 PHY collaborative coexistence enhancements

In addition, certain PHY mechanisms, as described in IEEE Std 802.15.2 [B5], are available to improve coexistence. In particular, deterministic frequency excision (variable notch filter) is a collaborative mechanism that is able to excise IEEE 802.15.1 signals from the IEEE 802.15.3 signal before detection. In this case, the nulling of the narrowband signal is done at either IF or complex baseband. In the case of IEEE 802.15.1 interference, the jamming tones occur at predetermined channels and times since the IEEE 802.15.3 DEV is assumed to have knowledge of the IEEE 802.15.1 piconet's future behavior. This mechanism and the algorithms needed to implement it are described in the IEEE Std 802.15.2 [B5].

D.2.3.3 MAC coexistence via collaboration

IEEE Std 802.15.2-2003 [B5] also describes the following two collaborative techniques that involve MAC sublayer coordination:

a) The packet traffic arbitrator method uses shared knowledge of the current traffic in the IEEE 802.15.3 and IEEE 802.15.1 piconets as well as future IEEE 802.15.1 frequency utilization, frame type, and frame priority to arbitrate access to the medium. This arbitration occurs on a frame-by-frame basis.
b) The alternating wireless medium access method alternates access to the medium by IEEE 802.15.1 and IEEE 802.15.3 members based on a timing criterion; the IEEE 802.15.3 piconet is restricted to activity during a portion of the beacon interval, while the IEEE 802.15.1 piconet is active during the remainder.

Either of these optional techniques would enhance the coexistence of collocated IEEE 802.15.3 and IEEE 802.15.1 systems. The impact of either method on overall system performance is outside of the scope of this standard.

D.2.3.4 Other techniques

Due to the popularity of IEEE Std 802.11b [B3] and IEEE Std 802.15.1 [B4], many methods have been proposed to improve the coexistence of these two systems. As with the methods published in IEEE Std 802.15.2 [B5], some of these methods are able to dramatically improve the coexistence of the two piconets. Since these methods are outside of the scope of this standard, they are not listed in this document. However, the existence of these techniques shows that similar innovative techniques might be used in the case of IEEE Std 802.15.1 [B4] and IEEE Std 802.15.3.

D.3 Coexistence performance

D.3.1 Allowed operation

The following IEEE wireless protocols are always allowed to operate in an operational area that overlaps with the operational area of an IEEE 802.15.3 piconet due to the fact that they use different frequency bands:

— IEEE 802.11a
— IEEE 802.11 IR
— IEEE 802.16

The following IEEE wireless protocols are allowed to operate in an operational area that overlaps with the operational area of an IEEE 802.15.3 piconet, but could experience reduced throughput:

— IEEE 802.11 DSSS
— IEEE 802.11 FHSS
— IEEE 802.11b
— IEEE 802.15.1

The IEEE 802.15.3 network has mechanisms that allow coexistence with other overlapping IEEE wireless networks, and so overlapping operation with any of the IEEE wireless networks is not prohibited. However, as noted, with some networks, the throughput could be reduced, and so under certain conditions, overlapping operation might be undesirable.

D.3.2 Assumptions for coexistence calculations

For the calculations used to determine the level of coexistence, the following assumptions have been made:

a) The sensitivity for each of the receivers is the reference sensitivity given in each of the standards, i.e.,
   1) –70 dBm for IEEE 802.15.1
   2) –76 dBm for IEEE 802.11b 11 Mb/s CCK
   3) –75 dBm for IEEE 802.15.3 22 Mb/s differential quadrature phase shift keying (DQPSK)
b) The received power at the desired receiver is 10 dB above the receiver sensitivity. The level of 10 dB was selected because a 10 dB margin results in 10% frame error rate (FER) in a Raleigh fading channel. Reliable communications without interference would require at least this margin. The distance between the desired transmitter and receiver is not directly specified by this requirement, instead the transmitter power and the channel model listed as follows would be used to determine the resulting distance.
c) The transmitter power for each of the protocols is
   1) 0 dBm for IEEE 802.15.1
   2) +14 dBm for IEEE 802.11b
   3) +8 dBm for IEEE 802.15.3
d) The channel model is one that was proposed for IEEE Std 802.11 [B2] and used by IEEE Std 802.15.2 [B5], see Equation (D.1) and Equation (D.2):

$$d = 10^{\frac{(Pt - Pr - 40.2)}{20}} \quad \text{for d < 8 m} \tag{D.1}$$

$$d = 8 \times 10^{\frac{(Pt - Pr - 58.5)}{33}} \quad \text{for d > 8 m} \tag{D.2}$$

e) The receiver bandwidths are based on the requirements in the standard:
   1) 1 MHz for IEEE Std 802.15.1 [B4]
   2) 22 MHz for IEEE Std 802.11b [B3]
   3) 15 MHz for IEEE Std 802.15.3
f) The transmitter spectral masks are the maximum allowed in the standards. This is a very pessimistic assumption since the transmitter spectrum will generally be significantly lower than the spectral mask over most of the frequencies. There are usually only narrow peaks that come close to the required limits. The subclauses that define the transmitter spectral mask for the three standards are as follows:
   1) 7.2.3.1 for IEEE Std 802.15.1 [B4]
   2) 18.4.7.3 for IEEE Std 802.11b [B3]
   3) 11.5.3 for IEEE Std 802.15.3
g) The energy from the interfering signal affects the desired signal in a manner equivalent to additive white Gaussian noise (AWGN) in the same bandwidth.
h) The IEEE 802.15.3 piconet operates with the IEEE 802.11b coexistence channel plan, as described in 11.2.3.

The bit error rate (BER) calculations were calculated using the analytical model from IEEE Std 802.15.2-2003 [B5]. The calculation follows the approach outlined in C.3.2 of IIEEE Std 802.15.2-2003 [B5], and the conversion from signal-to-noise ratio (SNR) to BER uses the formulas in C.3.6 of IEEE Std 802.15.2-2003 [B5]. Figure D.1 illustrates the relationship between BER and SNR for IEEE 802.11b, IEEE 802.15.3 base rate, and IEEE 802.15.1.

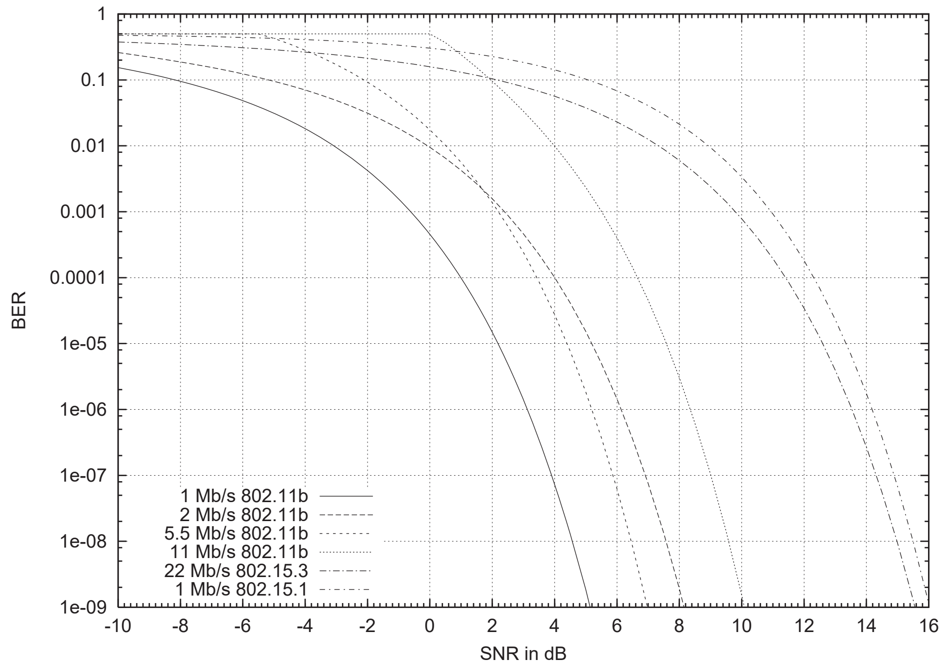

Figure D.1—BER results for IEEE 802.15.3, IEEE 802.11b, and IEEE 802.15.1

For this analysis, the SNR is used instead of the signal to interference ratio. One reason is that IEEE Std 802.11b [B3] and IEEE Std 802.15.3 only specify the sensitivity with respect to noise. IEEE Std 802.15.1 [B4] specifies the receiver performance with respect to an interferer, but only for an IEEE 802.15.1 interferer. The performance of one of these system with respect to a specific interference source depends on the actual implementation of the receiver. The approximation that the interfering signal is equivalent to AWGN is sufficient for the purposes of this analysis.

D.3.3 Performance impact on IEEE 802.15.3 piconets

D.3.3.1 Overview

The following subclauses provide an estimate of the performance impact of other IEEE 802 wireless networks on the operation IEEE 802.15.3 piconets. Any evaluation of the performance impact depends greatly on the assumed channel model, the physical distribution of DEVs in the network and the traffic pattern. Because of this, the IEEE 802.15.3 Working Group adopted a simple model to provide an estimate of the performance impacts. This model only takes into account the PHY parameters of the system to estimate the reduction in the FER for overlapping systems.

There is no performance degradation anticipated from the overlapping operation of IEEE 802.11 IR, IEEE 802.11a, or IEEE 802.16.

The performance degradation of overlapping operation with IEEE 802.11b, IEEE 802.11 FHSS, and IEEE 802.15.1 is addressed in D.3.3.2 and D.3.3.3.

D.3.3.2 IEEE 802.11b overlapping with IEEE 802.15.3

When an IEEE 802.11b network operational area overlaps with the operational area of an IEEE 802.15.3 piconet, the IEEE 802.15.3 piconet will experience reduced throughput. When the PNC notices this throughput reduction by the responses to the Channel Status Request command, as described in 7.13.5, it would then search for a better channel to operate on. The PHY layer rates the channels, best to worst (from lowest to highest interference, as described in 7.2.2), and would choose a new channel to operate on that has the least interference.

If the IEEE 802.15.3 DEV is able to positively detect the presence of IEEE 802.11b networks, it will adopt the IEEE 802.11b coexistence channel plan, as described in 11.2.3, to minimize the overlap of selected channels with IEEE 802.11b channels. It would also then rate as worst any channels that it finds contains IEEE 802.11b networks, as described in 11.2.4. Thus, the PNC would choose a new channel that is not used by the overlapping IEEE 802.11b network.

Note that if the IEEE 802.11b AP supported both the ability to request neighbor piconet status, as described in 7.2.9, and the ability to perform the PCF function, then that AP would be able to negotiate with the IEEE 802.15.3 piconet to share time in the overlapping medium. This capability is not specified in IEEE Std 802.11, and so providing this capability would require an AP that had additional functionality that is outside of the current IEEE Std 802.11. IEEE Std 802.15.2-2003 [B5] has proposed an IE and extra functionality that, if added to IEEE Std 802.11, would make it possible to build a standards-compliant AP that could then support the IEEE 802.15.3 neighbor piconet capability.

Using the modeling assumptions stated in D.3.2, the degradation of the FER of an IEEE 802.15.3 piconet in the presence of an IEEE 802.11b WLAN is illustrated in Figure D.2. In the graph, the adjacent channel is 25 MHz away while the alternate channel is 49 MHz separation.

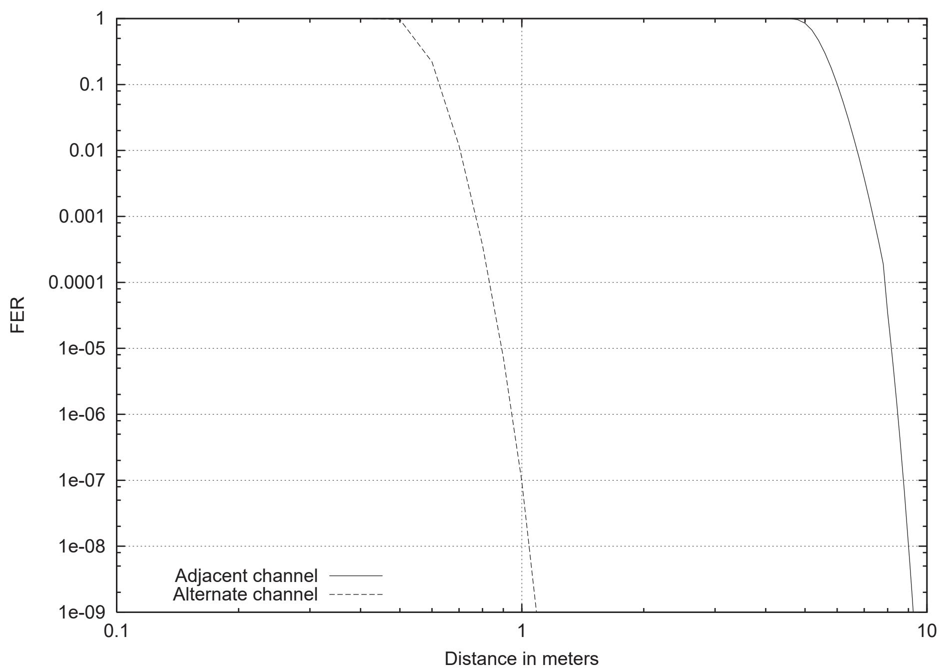

Figure D.2—FER results for IEEE 802.15.3 with IEEE 802.11b as the interferer

The adjacent and alternate channels for IEEE 802.15.3 as the receiver with IEEE 802.11b as the interferer are given in Table D.1.

In the graph of Figure D.2, there is almost no effect when the IEEE 802.11b network is in the alternate channel. When the IEEE 802.11b network is in the adjacent channel, however, the performance is noticeably impacted up to a distance of 6 m. There are two reasons for this. The first reason is that IEEE Std 802.11b [B3] uses higher transmit power than IEEE Std 802.15.3, and so it affects the IEEE 802.15.3 network to a greater degree. The second reason is that the IEEE 802.11b spectral mask is wider than IEEE 802.15.3 so that the impact in the adjacent channel is much greater.

Table D.1—Adjacent and alternate channels for IEEE 802.15.3 as receiver

IEEE 802.15.3 channel	Adjacent IEEE 802.11b channel	Alternate IEEE 802.11b channel
1	6	11
3	1, 11	none
5	6	1

The impact of a IEEE 802.15.3 network on the FER of an IEEE 802.11b network is illustrated in Figure D.3 for the adjacent channel and in Figure D.4 for the alternate channel.

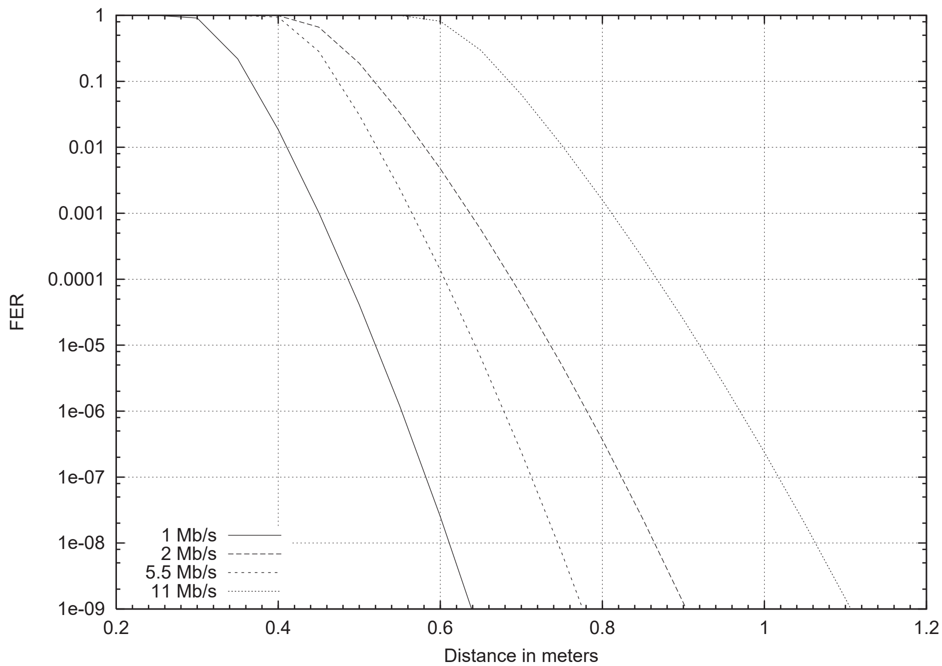

Figure D.3—FER results for IEEE 802.11b with IEEE 802.15.3 as the interferer in the adjacent channel

The adjacent and alternate channel numbers for IEEE 802.11b as the receiver with IEEE 802.15.3 as the interferer are given in Table D.2.

Table D.2—Adjacent and alternate channels for IEEE 802.11b as receiver

IEEE 802.11b channel	Adjacent IEEE 802.15.3 channel	Alternate IEEE 802.15.3 channel
1	3	5
6	1, 5	none
11	3	1

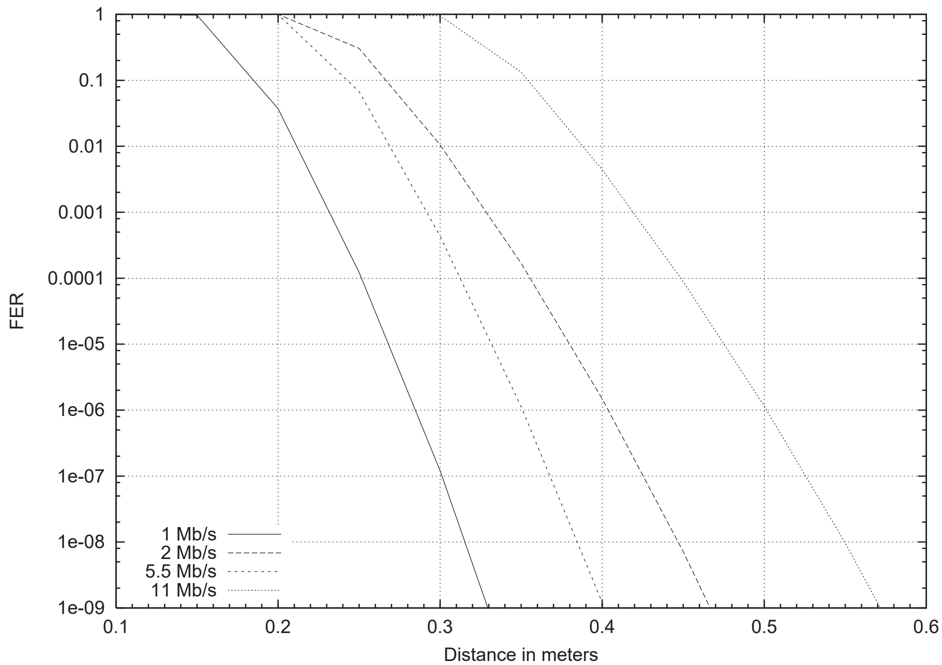

Figure D.4—FER results for IEEE 802.11b with IEEE 802.15.3 as the interferer in the alternate channel

The IEEE 802.15.3 network has very little effect on the IEEE 802.11b network. The IEEE 802.15.3 piconet causes less than 10% FER at a distance of just 0.7 m even when it is in the adjacent channel. Part of the reason for this is that the IEEE 802.15.3 piconet operates at a lower transmitter power. Also, there is not much difference between the FER results for the adjacent and alternate channels. This is due to the much more stringent transmitter mask that is required for IEEE Std 802.15.3 but not for IEEE Std 802.11b [B3].

If the IEEE 802.15.3 piconet occupies the same channel as the IEEE 802.11b network, the performance degradation is much worse since the interference is within the passband of either radio. The results for co-channel interference are shown in Figure D.5.

In this case, both systems suffer significant interference at shorter distance and only achieve reasonable throughputs when the separation exceeds 40 m for IEEE 802.11 and 80 m for IEEE 802.15.3. Similar results are obtained for IEEE 802.11/IEEE 802.11 and IEEE 802.15.3/IEEE 802.15.3 co-channel interference. These results show the importance of dynamic channel selection capability in the IEEE 802.15.3 protocol.

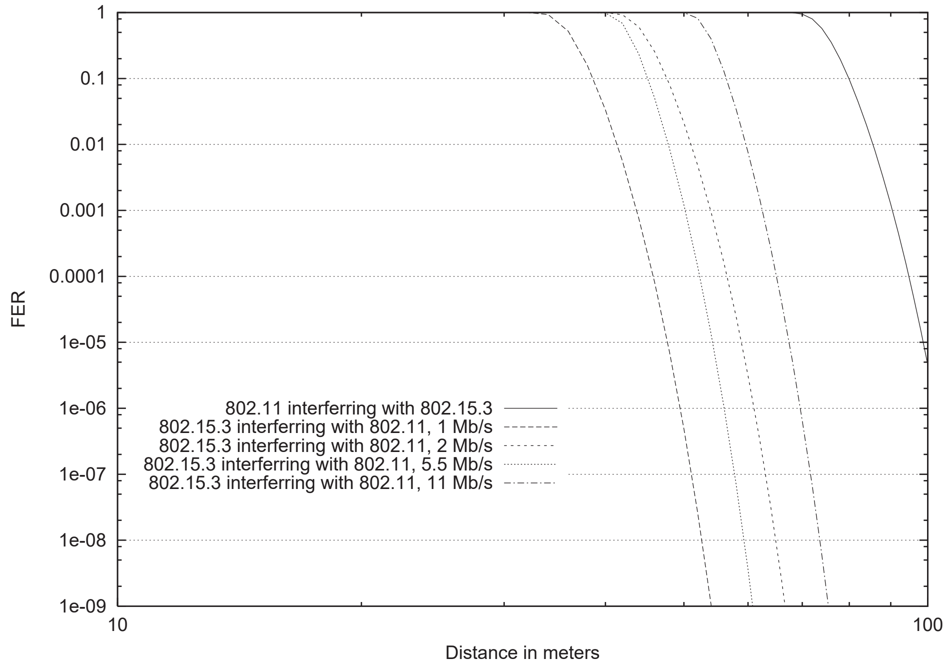

Figure D.5—FER results for IEEE 802.11b and IEEE 802.15.3 co-channel interference

D.3.3.3 IEEE 802.15.1 and IEEE 802.11 FHSS overlapping with IEEE 802.15.3

In the case of an IEEE 802.15.1 piconet sharing the operational space with an IEEE 802.15.3 piconet, the analysis is a little different. Since the IEEE 802.15.1 piconet hops frequencies, the impact on the IEEE 802.15.3 FER will depend on the current hop frequency. Overall, the average FER for either network will be an average of the FER for each of the channels.

IEEE 802.11 FHSS systems will have a similar performance impact as IEEE 802.15.1 piconets on IEEE 802.15.3 piconets, so analysis in this subclause will apply to that situation as well. Thus, in this subclause, only IEEE 802.15.1 piconets are discussed.

The average FER with IEEE 802.15.1 as the receiver and IEEE 802.15.3 as the interferer (“802.15.1 receiving”) and with IEEE 802.15.3 as the receiver and IEEE 802.15.1 as the interferer (“802.15.3 receiving”) is shown in Figure D.6.

The flat part of the FER curve is due to the times when the IEEE 802.15.1 hop frequency is in the IEEE 802.15.3 pass band. Since there is no attenuation of the signal due to channel filtering, these frequencies have an FER of essentially 1. The average FER then works out to be 15/79 = 0.19, as shown in Figure D.6.

The analysis in this subclause assumes that the IEEE 802.15.3 receiver does not use notch filtering or other equalization techniques to minimize the impact of narrowband interferers. If the implementation uses any of the recommended practices listed in D.2.3.2, the performance of the IEEE 802.15.3 piconet in the presence of an IEEE 802.15.1 interferer would see a large improvement.

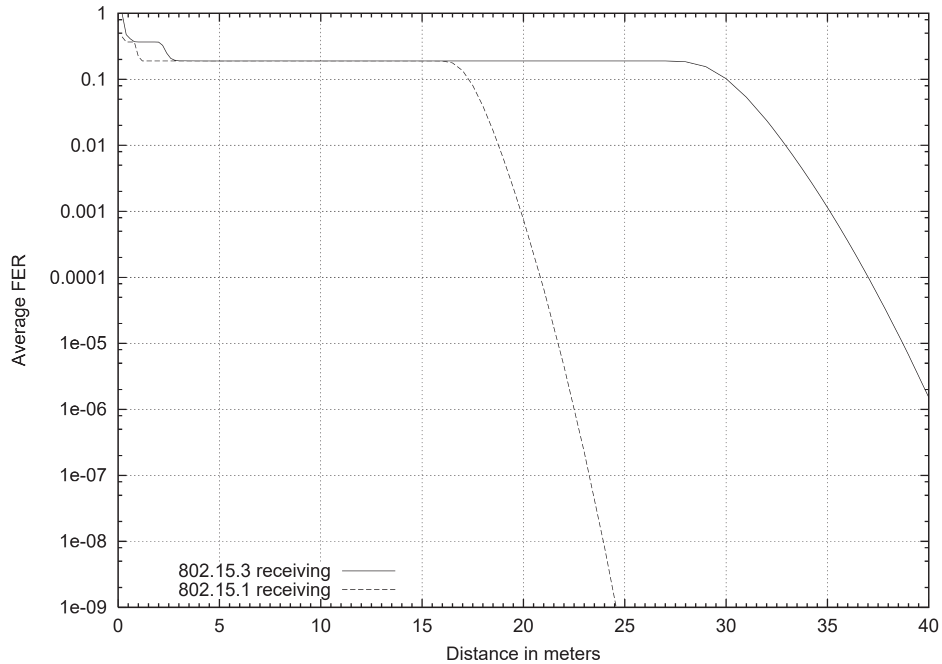

Figure D.6—Average FER results for IEEE 802.15.1 and IEEE 802.15.3

D.4 Notes on the calculations

The calculations for this annex are based on the formulas and descriptions from IEEE Std 802.15.2-2003 [B5].

All source files, including spreadsheets and graphs, are archived online.[24]

[24] For the source files archive, go to http://standards.ieee.org/downloads/802.15/.

Annex E

(informative)

Protocol implementation conformance statement (PICS) proforma[25]

E.1 Introduction

E.1.1 General

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given standard. Such a statement is called a protocol implementation conformance statement (PICS).

E.1.2 Scope

This annex provides the PICS proforma for IEEE Std 802.15.3-2023.

E.1.3 Purpose

The supplier of a protocol implementation claiming to conform to this standard shall complete the following PICS proforma and accompany it with the information necessary to identify fully both the supplier and the implementation.

The PICS of a protocol implementation is a statement of which capabilities and options of the protocol have been implemented. The statement is in the form of answers to a set of questions in the PICS proforma. The questions in a proforma consist of a systematic list of protocol capabilities and options as well as their implementation requirements. The implementation requirement indicates whether implementation of a capability is mandatory, optional, or conditional, depending on options selected. When a protocol implementer answers questions in a PICS proforma, the implementer would indicate whether an item is implemented or not, and provide explanations if an item is not implemented.

E.2 Abbreviations and special symbols

Notations for requirement status:

M	Mandatory
O	Optional
O.n	Optional, but support of at least one of the group of options labeled O.n is required.
N/A	Not applicable
“item”	Conditional, status dependent upon the support marked for the “item.”

For example, “FD1: M” indicates that the status is mandatory if the protocol feature item FD1 is implemented.

[25]Copyright release for PICS proforma: Users of this standard may freely reproduce the PICS proforma in this annex to use it for its intended purpose and may further publish the completed PICS.

E.3 Instructions for completing the PICS proforma

If it is claimed to conform to this standard, the actual PICS proforma to be filled in by a supplier shall be technically equivalent to the text of the PICS proforma in this annex, and shall preserve the numbering and naming and the ordering of the PICS proforma.

A PICS that conforms to this annex shall be a conforming PICS proforma completed in accordance with the instructions for completion given in this annex.

The main part of the PICS is a fixed-format questionnaire, divided into five tables. Answers to the questionnaire are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (such as Yes or No), or by entering a value or a set or range of values.

E.4 Identification of the implementation

*Implementation Under Test (IUT) Identification

IUT Name: ____

IUT Version: ____

____

*System Under Test (SUT) Identification

SUT Name: ____

Hardware Configuration: ________________________________

____

____

Operating System: ______________________________________

*Product Supplier

Name: ____

Address: ____

____

Telephone Number: _____________________________________

Facsimile Number: ______________________________________

Email Address: ____

Additional Information: __________________________________

*Client

Name: ____

Address: ____

____

Telephone Number: ____________________

Facsimile Number: ____________________

Email Address: ____________________

Additional Information: ____________________

*PICS Contact Person

Name: ____________________

Address: ____________________

____

Telephone Number: ____________________

Facsimile Number: ____________________

Email Address: ____________________

Additional Information: ____________________

*PICS/System Conformance Statement

Provide the relationship of the PICS with the System Conformance Statement
for the system:
____

____

____

E.5 Identification of the protocol

This PICS proforma applies to IEEE Std 802.15.3-2023.

E.6 Global statement of conformance

The implementation described in this PICS proforma meets all of the mandatory requirements of the referenced standard.

[ ] Yes

[ ] No

NOTE—Answering “No” indicates non-conformance to the specified protocol standard. Non-supported mandatory capabilities are to be identified in Table E.1 through Table E.12, with an explanation by the implementer on why the implementation is non-conforming.

The supplier will have fully complied with the requirements for a statement of conformance by completing the statement contained in this subclause. However, the supplier may find it helpful to continue to complete the detailed tabulations in E.7.

E.7 Protocol implementation conformance statement (PICS) proforma

E.7.1 Overview

Table E.1 through Table E.13 are composed of the detailed questions to be answered, which make up the PICS proforma. E.7.2 contains the major roles for an IEEE 802.15.3 DEV. E.7.3 contains the major capabilities for the PHY and radio frequencies. E.7.4 contains the major capabilities for the MAC sublayer. E.7.4.5 indicates which level and type of security is supported in the implementation.

E.7.2 Major roles for IEEE 802.15.3 DEVs

Table E.1—Functional DEV types

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
FD1	Is this entity DEV capable?		M			
FD2	Is this DEV PNC capable?	7.2.11	O			
FD3	Supports 2.4 GHz PHY	Clause 11	O			

Table E.2—Functional PRDEV types

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
FHD1	Is this entity PRDEV capable?		M			
FHD2	Is this DEV PRC capable?	4.6	O			
FHD3	Supports HRCP-SC PHY	13.2	O.1			
FHD4	Supports HRCP-OOK PHY	13.3	O.1			
FHD5	Supports THz-SC PHY	14.2	O.1			
FHD6	Supports THz-OOK PHY	14.3	O.1			

E.7.3 PHY functions

Table E.3—PHY functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
PLF1	Conforms to general requirements (e.g., timing, frequency)	11.2	FD3: M			
PLF1.1	Able to detect IEEE 802.11b networks	11.2.4	FD3: O			
PLF2	Supports 22 Mb/s DQPSK modulation	11.3.2, 11.3.3, 11.3.4	FD3: M			
PLF2.1	Supports 33, 22, and 11 Mb/s modulations	11.3.3, 11.3.5	FD3: O			
PLF2.2	Supports 44, 33, 22, and 11 Mb/s modulations	11.3.3, 11.3.5	FD3: O			
PLF2.3	Supports 55, 44, 33, 22, and 11 Mb/s modulations	11.3.3, 11.3.5	FD3: O			
PLF3	Encodes and decodes PHY frame format	11.4	FD3: M			
PLF4	Conforms to transmitter requirements	11.5	FD3: M			
PLF5	Conforms to receiver requirements	11.6	FD3: M			
PLF5.1	Supports link quality assessment	11.6.7	PLF2.1: M, PLF2.2: M, PLF2.3: M			
PLF6	PHY PIB values supported	11.7	FD3: M			
PLF7	CAP mandatory	11.2.10	FD3: M			
PLF8	Provide receive status in Imm-ACK or Dly-ACK frames	11.7	FD3: O			

Table E.4—HRCP SC PHY functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
SC-HPLF1	Conforms to general requirements (e.g., timing, frequency)	13.1	FHD3: M			
SC-HPLF2	Supports a single RF channel	13.2.2	FHD3: M			
SC-HPLF2.1	Supports channel aggregation	13.2.2	FHD3: O			
SC-HPLF3	Supports $\pi/2$ BPSK and $\pi/2$ QPSK modulations	13.2, 13.2.4	FHD3: M			
SC-HPLF3.1	Supports 16QAM modulation	13.2, 13.2.3	FHD3: O			
SC-HPLF3.2	Supports 64QAM modulation	13.2, 13.2.3	FHD3: O			
SC-HPLF3.3	Supports 256QAM modulation	13.2, 13.2.3	FHD3: O			
SC-HPLF4	Supports rate-14/15 and rate-11/15 LDPC codes	13.2, 13.2.3.6	FHD3: M			
SC-HPLF5	Encodes and decodes PHY frame format	13.2.4.2, 13.2.4.3, 13.2.4.4	FHD3: M			
SC-HPLF5.1	Insertion and detection of pilot word	13.2.4.5	FHD3: O			
SC-HPLF5.2	Insertion and detection of pilot preamble (PPRE)	13.2.4.5	FHD3: O			
SC-HPLF6	Conforms to transmitter requirements	13.2.5	FHD3: M			

Table E.4—HRCP SC PHY functions (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
SC-HPLF7	Conforms to receiver requirements	13.2.6	FHD3: M			
SC-HPLF8	Conforms to timing requirements	13.2.7	FHD3: M			
SC-HPLF9	Sends an Association request by using Access slot	4.7.9	FHD3: M			
SC-HPLF10	PHY PIB values supported	13.2.8	FHD3: M			

Table E.5—HRCP OOK PHY functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
OOK-HPLF1	Conforms to general requirements (e.g., timing, frequency)	13.1	FHD4: M			
OOK-HPLF2	Supports single channel transmission	13.3.2	FHD4: M			
OOK-HPLF2.1	Supports two channel bonding	13.3.2	FHD4: O			
OOK-HPLF2.2	Supports three channel bonding	13.3.2	FHD4: O			
OOK-HPLF2.3	Supports four channel bonding	13.3.2	FHD4: O			
OOK-HPLF3	Supports OOK modulations	13.3.3.5	FHD4: M			
OOK-HPLF4	Supports RS(240, 224) and its shortened version	13.3.3.6	FHD4: M			
OOK-HPLF5	Supports PRBS codes by LFSR with spreading factor 16 for frame header	12.3.3.7	FHD4: M			

Table E.5—HRCP OOK PHY functions (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
OOK-HPLF5.1	Supports code spreading using bit repetition with spreading factor 2 for frame payloads	12.3.3.7	FHD4: O			
OOK-HPLF6	Encodes and decodes PHY frame format	13.3.4	FHD4: M			
OOK-HPLF6.1	Insertion and detection of CES	13.3.4.2.3 and 13.3.4.2.4	FHD4: O			
OOK-HPLF6.2	Insertion and detection of pilot symbols	13.3.4.4.6	FHD4: O			
OOK-HPLF7	Conforms to transmitter requirements	13.3.5	FHD4: M			
OOK-HPLF8	Conforms to receiver requirements	13.3.6	FHD4: M			
OOK-HPLF9	Conforms to timing requirements	13.3.7	FHD4: M			
OOK-HPLF10	PHY PIB values supported	13.3.8	FHD4: M			

Table E.6—THz-SC PHY functions

Item number	Item description	References	Status	Support		
				N/A	Yes	No
SC-TPLF1	Conforms to general requirements (e. g., timing, frequency)	14.1	FHD5: M			
SC-TPLF2	**RF channelization**					
SC-TPLF2.1	Supports a bandwidth of 2.16 GHz	14.1.3	FHD5: O			
SC-TPLF2.2	Supports a bandwidth of 4.32 GHz	14.1.3	FHD5: M			
SC-TPLF2.3	Supports a bandwidth of 8.64 GHz	14.1.3	FHD5: O			
SC-TPLF2.4	Supports a bandwidth of 12.96 GHz	14.1.3	FHD5: O			
SC-TPLF2.5	Supports a bandwidth of 17.28 GHz	14.1.3	FHD5: O			
SC-TPLF2.6	Supports a bandwidth of 25.92 GHz	14.1.3	FHD5: O			

Table E.6—THz-SC PHY functions (*continued*)

Item number	Item description	References	Status	Support		
				N/A	Yes	No
SC-TPLF2.7	Supports a bandwidth of 51.84 GHz	14.1.3	FHD5: O			
SC-TPLF2.8	Supports a bandwidth of 69.12 GHz	14.1.3	FHD5: O			
SC-TPLF2.9	Supports a bandwidth of 34.56 GHz	14.1.3	FHD5: O			
SC-TPLF3	**Modulation**					
SC-TPLF3.1	Supports π/2 BPSK and π/2 QPSK modulation	14.2 14.2.3.1	FHD5: M			
SC-TPLF3.2	Supports π/2 8-PSK modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF3.3	Supports π/2 8-APSK modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF3.4	Supports 16-QAM modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF3.5	Supports 64-QAM modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF3.6	Supports 16-APSK modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF3.7	Supports 32-APSK modulation	14.2 14.2.3.1	FHD5: O			
SC-TPLF4	Supports rate 14/15 and rate 11/15 LDPC codes	14.2 14.2.3.2	FHD5: M			
SC-TPLF5	Encodes and decodes PHY frame format	14.2.3.2 14.2.4.4	FHD5: M			
SC-TPLF5.1	Insertion and detection of pilot word (PW)	14.2.4.5	FHD5: O			
SC-TPLF5.2	Insertion and detection of pilot preamble (PPRE)	14.2.4.5	FHD5: O			
SC-TPLF6	Conforms to transmitter requirements	14.2.5	FHD5: M			
SC-TPLF7	Conforms to receiver requirements	14.2.6	FHD5: M			
SC-TPLF8	Conforms to timing requirements	14.2.7	FHD5: M			
SC-TPLF9	Sends an Association Request command by using access slot	4.7.9	FHD5: M			
SC-TPLF10	PHY PIB values supported	14.1.6	FHD5: M			

Table E.7—THz-OOK PHY functions

Item number	Item description	References	Status	Support		
				N/A	Yes	No
OOK-TPLF1	Conforms to general requirements (e. g., timing, frequency)	14.1	FHD6: M			
OOK-TPLF2	**RF channelization**					
OOK-TPLF2.1	Supports a bandwidth of 2.16 GHz	14.1.3	FHD6: O			
OOK-TPLF2.2	Supports a bandwidth of 4.32 GHz	14.1.3	FHD6: M			
OOK-TPLF2.3	Supports a bandwidth of 8.64 GHz	14.1.3	FHD6: O			
OOK-TPLF2.4	Supports a bandwidth of 12.96 GHz	14.1.3	FHD6: O			
OOK-TPLF2.5	Supports a bandwidth of 17.28 GHz	14.1.3	FHD6: O			
OOK-TPLF2.6	Supports a bandwidth of 25.92 GHz	14.1.3	FHD6: O			
OOK-TPLF2.7	Supports a bandwidth of 51.84 GHz	14.1.3	FHD6: O			
OOK-TPLF2.8	Supports a bandwidth of 69.12 GHz	14.1.3	FHD6: O			
OOK-TPLF2.9	Supports a bandwidth of 34.56 GHz	14.1.3	FHD6: O			
OOK-TPLF3	Supports OOK modulation	14.3 14.3.3	FHD6: M			
OOK-TPLF4	Supports RS (240,224) and its shortened version	14.3.3.3	FHD6: M			
OOK-TPLF4.1	Supports rate 14/15 LDPC codes	14.3.3.3	FHD6: O			
OOK-TPLF4.2	Supports rate 11/15 LDPC codes	14.3.3.3	FHD6: O			
OOK-TPLF5	Encodes and decodes PHY frame format	14.3.4.2 14.3.4.3	FHD6: M			
OOK-TPLF5.1	Insertion and detection of PW	14.3.4.4	FHD6: O			
OOK-TPLF5.2	Insertion and detection of PPRE	14.3.4.5	FHD6: O			
OOK-TPLF6	Conforms to transmitter requirements	14.3.5	FHD6: M			
OOK-TPLF7	Conforms to receiver requirements	14.3.6	FHD6: M			
OOK-TPLF8	Conforms to timing requirements	14.3.7	FHD6: M			
OOK-TPLF9	Send an Association Request command by using access slot	4.7.9	FHD6: M			
OOK-TPLF10	PHY PIB values supported	14.1.6	FHD6: M			

E.7.4 Major capabilities for the MAC sublayer

E.7.4.1 MAC frames for piconet

Table E.8—MAC frames for piconet

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF1	**General frame format**					
MF1.1	MAC Header field and MAC Frame Body field formats for piconet	6.2	M		M	
MF1.2	Non–secure MAC Frame Body field format for piconet	6.2	M		M	
MF1.3	Secure MAC Frame Body field format for piconet	6.2	O		S2.1:M	
MF2	**Frame types**					
MF2.1	Non-secure Beacon frame for piconets	6.3.1.1	FD2: M		M	
MF2.2	Secure Beacon frame for piconets	6.3.1.3	FD2: O S2.1 & FD2: M		O S2.1: M	
MF2.3	Imm-ACK frame	6.3.2.1	M		M	
MF2.4	Delayed ACK (Dly-ACK) frame	6.3.2.2	O		O	
MF2.5	Non-secure Command frame	6.3.3.2	M		M	
MF2.6	Secure Command frame	6.3.3.3	O S2.1: M		O S2.1: M	
MF2.7	Non-secure Data frame	6.3.5.1	O		O	
MF2.8	Secure Data frame	6.3.5.2	O		O	
MF2.9	Non-secure Multi-protocol Data frame	6.3.7.1	O		O	
MF2.10	Secure Multi-protocol Data frame	6.3.7.2	O S2.1: M		O S2.1: M	
MF2.11	Synchronization frame for piconets	6.3.9	O		O	
MF3	**IEs**					
MF3.1	CTA IE for piconet	6.4.2	FD2: M		M	
MF3.2	BSID IE for piconet	6.4.3	FD2: M		M	
MF3.3	Parent Piconet IE	6.4.4	FD2: O		O	

Table E.8—MAC frames for piconet (*continued*)

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF3.4	DEV Association IE for piconet	6.4.5	FD2: M		M	
MF3.5	PNC Shutdown IE for piconet	6.4.6	FD2: M		M	
MF3.6	Piconet Parameter Change IE	6.4.7	FD2: M		M	
MF3.7	AS IE for piconet	6.4.8	O		O	
MF3.8	Pending channel time map (PCTM) IE for piconet	6.4.9	FD2: O		O	
MF3.9	PNC Handover IE for piconet	6.4.10	FD2: M		M	
MF3.10	CTA Status IE for piconet	6.4.11	FD2: M		M	
MF3.11	Capability IE for piconet	6.4.12	M		M	
MF3.12	Transmit Power Parameters IE	6.4.19, 7.15.3.3	M		O	
MF3.13	PS Status IE for piconet	6.4.20	FD2: M		O	
MF3.14	CWB IE for piconet	6.4.21	O		O	
MF3.15	Overlapping PNID IE for piconet	6.4.22	M		FD2: M	
MF3.16	Piconet Services IE	6.4.23	O		O	
MF3.17	Vendor Defined IE	6.4.24	O		O	
MF3.18	Group ID IE for piconet	6.4.25	O		O	
MF3.19	Stream Renew IE for piconet	6.4.26	O		O	
MF3.20	Next PNC IE for piconet	6.4.27	O		O	
MF3.21	Piconet Channel Status IE	6.4.28	O		O	
MF3.22	Synchronization IE for piconet	6.4.29	O		O	
MF3.23	TSD IE for piconet	6.4.30	O		O	
MF3.24	UEP Specific IE for piconet	6.4.31	O		O	
MF3.25	IFS IE for piconet	6.4.32	O		O	
MF3.26	CTA Relinquish Duration IE for piconet	6.4.33	O		O	
MF3.27	Feedback IE for piconet	6.4.34	O		O	
MF3.28	Mapping IE for piconet	6.4.35	O		O	
MF3.29	BST Clustering IE for piconet	6.4.36	O		O	

Table E.8—MAC frames for piconet (*continued*)

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF3.30	PET Clustering IE for piconet	6.4.37	O		O	
MF3.31	Beam PET IE for piconet	6.4.38	O		O	
MF3.32	HRS Beam PET IE for piconet	6.4.39	O		O	
MF3.33	PET Amplitude IE for piconet	6.4.40	O		O	
MF3.34	PET Phase IE for piconet	6.4.41	O		O	
MF3.35	Sync Frame Frequency IE for piconet	6.4.42	O		O	
MF4	**Command types**					
MF4.1	Association Request command for piconet	6.5.2.2	M		FD2: M	
MF4.2	Association Response command for piconet	6.5.2.3	FD2: M		M	
MF4.3	Disassociation Request command	6.5.2.4	M		M	
MF4.4	Request Key command	6.5.3.2	S2.1: M		S3.1: M	
MF4.5	Request Key Response command	6.5.3.3	S3.1: M		S2.1: M	
MF4.6	Distribute Key Request command	6.5.3.4	S3.1: M		S2.1: M	
MF4.7	Distribute Key Response command	6.5.3.5	S2.1: M		S3.1: M	
MF4.8	PNC Handover Request command	6.5.4.2	O		O	
MF4.9	PNC Handover Response command	6.5.4.3	O		O	
MF4.10	PNC Handover Information command	6.5.4.4	O		O	
MF4.11	PNC Information Request command for piconet	6.5.5.2	M		FD2: M	
MF4.12	PNC Information command for piconet	6.5.5.3	FD2: M		M	
MF4.13	Security Information Request command	6.5.5.4	O		O	
MF4.14	Security Information command	6.5.5.5	O		O	
MF4.15	Probe Request command	6.5.5.6	M		M	

Table E.8—MAC frames for piconet (*continued*)

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF4.16	Probe Response command	6.5.5.7	M		M	
MF4.17	Piconet Services command	6.5.6.1	FD2: O		O	
MF4.18	Announce command for piconet	6.5.6.2	FD2: O		O	
MF4.19	Channel Time Request command	6.5.7.2	M		FD2: M	
MF4.20	Channel Time Response command	6.5.7.3	FD2: M		M	
MF4.21	Channel Status Request command	6.5.8.2	O		M	
MF4.22	Channel status response	6.5.8.3	M		O	
MF4.23	Remote Scan Request command	6.5.8.4	FD2: O		M	
MF4.24	Remote Scan Response command	6.5.8.5	M		FD2: O	
MF4.25	Transmit Power Change command	6.5.8.6	O		M	
MF4.26	PM Mode Change command	6.5.9.6	O		FD2: M	
MF4.27	SPS Configuration Request command	6.5.9.4	O		FD2: M	
MF4.28	SPS Configuration Response command	6.5.9.5	FD2: M		O	
MF4.29	PS Set Information Request command	6.5.9.2	O		FD2: M	
MF4.30	PS Set Information Response command	6.5.9.3	FD2: M		O	
MF4.31	Security Message command	6.5.10.1	O		O	
MF4.32	Vendor Defined command	6.5.10.2	O		O	
MF4.33	Announce Response command for piconet	6.5.6.3	M		M	
MF4.34	PM Mode Change Response command	6.5.9.7	FD2: M		O	
MF4.35	AS IE Request command for piconets	6.5.10.3	O		O	
MF4.36	AS IE Response command for piconets	6.5.10.4	O		O	

Table E.8—MAC frames for piconet (*continued*)

Item number	Item description	Reference	Transmitter		Receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF4.37	Multicast Configuration Request command	6.5.11.1	O		O	
MF4.38	Multicast Configuration Response command	6.5.11.2	O		O	

E.7.4.2 MAC frames for pairnet

Table E.9—MAC frames for pairnet

Item number	Item description	Reference	Pairnet transmitter		Pairnet receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF1	**General frame format**					
MF1.4	MAC Header field and MAC Frame Body field format for pairnet	6.2	M		M	
MF1.5	Non-secure MAC Frame Body field format for pairnet	6.2	M		M	
MF1.6	Secure MAC Frame Body field format for pairnet	6.2	O S2.2: M		O S2.2: M	
MF2	**Frame types**					
MF2.12	Non-secure Beacon frame for pairnet	6.3.1.2	O FHD2: M		M	
MF2.13	Secure Beacon frame for pairnet	6.3.1.4	O FHD2 & S2.2: M		O S2.2: M	
MF2.14	Pairnet Non-secure command frame	6.3.4.1	M		M	
MF2.15	Pairnet Secure command frame	6.3.4.2	O S2.2: M		O S2.2: M	
MF2.16	Non-secure Pairnet Aggregated Data frame	6.3.6.1	O		M	
MF2.17	Secure Pairnet Aggregated Data frame	6.3.6.2	O		O S2.2: M	
MF2.18	Non-secure Pairnet Aggregated Multi-protocol Data frame	6.3.8.1	O		O	
MF2.19	Secure Pairnet Aggregated Multi-protocol Data frame	6.3.8.2	O		O	

Table E.9—MAC frames for pairnet (*continued*)

Item number	Item description	Reference	Pairnet transmitter		Pairnet receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF3	**IEs**					
MF3.12	Transmit Power Parameters IE	6.4.19, 7.15.3.3	O		O	
MF3.17	Vendor Defined IE	6.4.24	O		O	
MF3.36	PRC Capability IE	6.4.13	O FHD2: M		O	
MF3.37	PRDEV Capability IE	6.4.14	M		M	
MF3.38	Pairnet Operation Parameters IE	6.4.15	O FHD2: M		O	
MF3.39	THz PRC Capability IE	6.4.16	O FHD2: M		O	
MF3.40	THz PRDEV Capability IE	6.4.17	M		M	
MF3.41	THz Pairnet Operation Parameters IE	6.4.18	O FHD2: M		O	
MF3.42	MIMO Information IE	6.4.44	O		O	
MF3.43	Higher Layer Protocol Information IE	6.4.45	O		O	
MF4	**Command types**					
MF4.3	Disassociation Request command	6.5.2.4	M		M	
MF4.4	Request Key command	6.5.3.2	S2.2: M		S3.2: M	
MF4.5	Request Key Response command	6.5.3.3	S3.2: M		S2.2: M	
MF4.6	Distribute Key Request command	6.5.3.4	S3.2: M		S2.2: M	
MF4.7	Distribute Key Response command	6.5.3.5	S2.2: M		S3.2: M	
MF4.13	Security Information Request command	6.5.5.4	O		O	
MF4.14	Security Information command	6.5.5.5	O		O	
MF4.15	Probe Request command	6.5.5.6	O		O	
MF4.16	Probe Response command	6.5.5.7	M		M	

Table E.9—MAC frames for pairnet (*continued*)

Item number	Item description	Reference	Pairnet transmitter		Pairnet receiver	
			Status	Support N/A Yes No	Status	Support N/A Yes No
MF4.25	Transmit Power Change command	6.5.8.6	O		O	
MF4.31	Security Message command	6.5.10.1	O		O	
MF4.32	Vendor Defined command	6.5.10.2	O		O	
MF4.39	Association Request command Payload format for pairnet	6.5.2.2	M		FHD2: M	
MF4.40	Association Response command Payload format for pairnet	6.5.2.3	FHD2: M		M	

E.7.4.3 MAC sublayer functions for piconet

Table E.10—MAC sublayer functions for piconet

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF1	Scanning capable	7.2.2	M			
MLF2	Starting capable	7.2.3	FD2: M			
MLF3	PNC handover capable	7.2.4	O			
MLF4	**Child piconet support**					
MLF4.1	Parent PNC supports request mechanism for creating child piconet	7.2.8	FD2: M			
MLF4.2	PNC-capable DEV support of becoming a child PNC	7.2.8	FD2: O			
MLF5	**Neighbor piconet support**					
MLF5.1	Parent PNC supports request mechanism for creating neighbor piconet	6.5.2.3, 7.2.9	FD2: O			
MLF5.2	PNC-capable DEV support of becoming a neighbor PNC	7.2.9	FD2: O			
MLF6	Stopping piconet operations	7.2.10.2	FD2: M			

Table E.10—MAC sublayer functions for piconet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF7	**Association**					
MLF7.1	Association for piconets	7.4.2	M			
MLF8	Piconet services support	7.4.3	O			
MLF9	Broadcasting of piconet information	7.4.4	FD2: M			
MLF10	**Disassociation**					
MLF10.1	DEV disassociation	7.4.5	M			
MLF11	PNC disassociation	7.4.5	FD2: M			
MLF12	**Contention access methods**					
MLF12.1	CAP channel access during piconet operations	7.6.3	O.1 PLF7: M			
MLF12.2	Open and association MCTA operations	7.6.4.4, 7.6.4.5	O.1			
MLF12.3	Regular MCTA operations	7.6.4.4	M			
MLF13	Asynchronous channel time reservation	7.7.3	FD2: M			
MLF14	**Synchronization**					
MLF14.1	Synchronization for piconet	7.8	M			
MLF14.2	Beacon generation	7.8.3	FD2: M			
MLF14.3	Extended beacon support, reception	7.8.3	M			
MLF14.4	Extended beacon support, generation	7.8.3	FD2: O			
MLF15	Fragmentation and defragmentation	7.10	M			
MLF16	**Acknowledgment and retransmissions**					
MLF16.1	No acknowledgment	7.12.2	M			
MLF16.2	Immediate acknowledgment	7.12.3	M			
MLF16.3	Delayed acknowledgment	7.12.4	O			
MLF16.4	Imp-ACK	7.12.5	O			
MLF16.5	Retransmissions	7.12.8	M			
MLF16.6	Duplicate detection	7.12.9	M			

Table E.10—MAC sublayer functions for piconet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF17	**Peer discovery**					
MLF17.1	PNC information request	7.13.2	M			
MLF17.2	Probe request and response	7.13.3	M			
MLF17.3	Announce	7.13.4	M			
MLF17.4	Channel status request	7.13.5	M			
MLF17.5	Remote Scan	7.13.6	O			
MLF18	**Changing piconet parameters**					
MLF18.1	Moving beacon	7.14.2	FD1: M, FD2: O, MLF4.2: M, MLF5.2: M			
MLF18.2	Changing superframe duration	7.14.3	FD1: M, FD2: O, MLF4.2: M, MLF5.2: M			
MLF18.3	Setting the BSID and PNID	7.2.11, 7.14.4	FD1: M, FD2: M			
MLF18.4	Maintaining synchronization in dependent piconets	7.14.5	MLF4.2: M, MLF5.2: M			
MLF19	Multi-rate support	7.16	O			
MLF20	Dynamic channel selection	7.15.2	O			
MLF21	**Power management**					
MLF21.1	PSPS	7.17.2	FD1: O FD2: M			
MLF21.2	SPS (support for at least one SPS set)	7.17.3	FD1: O FD2: M			
MLF21.3	APS	7.17.4	FD1: O FD2: M			
MLF22	**Transmit power control**					
MLF22.1	Fixed maximum transmit power	7.15.3.2	FD1: M, FD2: O			
MLF22.2	Request transmitter power adjustment	7.15.3.3	O			
MLF22.3	Adjust transmitter power as requested	7.15.3.3	M			

Table E.10—MAC sublayer functions for piconet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF23	**AS IE**					
MLF23.1	Capable of requesting AS IEs	7.19	O			
MLF23.2	Capable of putting AS IEs in a beacon	7.19	FD2: O			
MLF24	**Relinquish CTA**					
MLF24.1	Capable of relinquishing CTA time	7.6.4.9	O			
MLF24.2	Capable of using CTA time released by other DEV	7.6.4.9	O			
MLF25	**Multicast configuration**					
MLF25.1	Request to join or leave a multicast group	7.7.4	FD1: O			
MLF25.2	Maintain list of multicast groups	7.7.4	FD2: O			
MLF26	**Handover extensions**					
MLF26.1	Preliminary handover capable	7.2.5	FD2: O			
MLF26.2	Next PNC Capable	7.2.6	FD2: O			
MLF26.3	Report status with Piconet Channel Status IE	6.4.28	O			
MLF27	**Aggregation**					
MLF27.1	Piconet standard aggregation	7.11.1	O			
MLF27.2	Piconet low-latency aggregation		O			
MLF28	Block ACK	7.12.6	O			
MLF29	Beam forming	Clause 15	O			
MLF30	Channel probing	7.13.8	O			
MLF31	UEP	7.21	O			
MLF32	TSD	15.8	O			

E.7.4.4 MAC sublayer functions for pairnet

Table E.11—MAC sublayer functions for pairnet

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF1	Scanning capable	7.2.2	M			
MLF7	**Association**					
MLF7.2	Association for pairnet	7.5.1	M			
MLF10	**Disassociation**					
MLF10.2	Disassociation for pairnet	7.5.2	M			
MLF12	**Contention access methods**					
MLF12.4	PAP after association	7.6.5	FHD1: M			
MLF14	**Synchronization**					
MLF14.5	Synchronization for pairnet	7.9	M			
MLF14.6	Beacon frame generation	7.9.3	O FHD2: M			
MLF16	**Acknowledgment and retransmissions**					
MLF16.1	No acknowledgment	7.12.2	M			
MLF16.7	Stk-ACK	7.12.7	FHD1: M			
MLF16.8	Recovery Process	7.12.7.2	FHD1: M			
MLF21	**Power management**					
MLF21.4	LLPS	7.18	FHD1:O			
MLF22	**Transmit power control**					
MLF22.2	Request transmitter power adjustment	7.15.3.3	O			
MLF22.3	Adjust transmitter power as requested	7.15.3.3	O			
MLF27	**Aggregation**					
MLF27.3	Pairnet aggregation	7.11.3	FHD1: O			
MLF27.4	Pairnet Deaggregation	7.11.3	FHD1: M			
MLF33	Higher layer protocol setup during association procedure for a pairnet	7.5.3	O			

E.7.4.5 Security support

Table E.12—Security capabilities for piconet

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
S1	Mode 0 support	8.2.2	M			
S2	**Mode 1 support**					
S2.1	Mode 1 support for piconet	8.2.3, 9.2, 9.3	O			
S3	**Supports acting as a key originator**					
S3.1	Supports acting as a key originator for piconet	8.3	FD2 & S2.1: M, S2: O			
S4	Security information request and distribution	8.4.2	O			

Table E.13—Security capabilities for pairnet

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
S1	Mode 0 support	8.2.2	M			
S2	**Mode 1 support**					
S2.2	Mode 1 support for pairnet	8.2.3, Clause 10	O			
S3	**Supports acting as a key originator**					
S3.2	Supports acting as a key originator for pairnet	8.3.3–8.3.10	FHD2 & S2.2: M, S2.2: O			

Annex F

(informative)

Implementation considerations

F.1 Channel time requests

F.1.1 Types of channel time allocations (CTAs)

Various types of channel time allocations (CTAs) are defined in this standard depending on the type of access method, destination, and the ability to be changed by the PNC. The types of CTAs used in the standard are listed in Table F.1.

Table F.1—Types of CTAs in this standard

CTA type	SrcID	DestID	Stream index	Access method(s)
CAP	N/A	N/A	N/A	Uses CSMA/CA, not a real CTA, but it is assigned time in the superframe.
Regular S-CAP	N/A	N/A	N/A	As defined in 7.8.6.
Association S-CAP	N/A	N/A	N/A	As defined in 7.8.6.
Regular CTA	Any valid single DEVID[a]	Any valid single DEVID	A regular stream index	TDMA with transmit control transfer.
Regular MCTA	Any valid single DEVID	Any valid DEVID[b]	MCTA stream index	TDMA with transmit control transfer. This is the same functionality as a regular CTA.
Association CTA	UnassocID	PNCID	Asynchronous stream index	CSMA/CA
Association MCTA	UnassocID	PNCID	MCTA stream index	Slotted aloha
Private CTA	Any valid single DEVID	Any valid single DEVID that is the same as the SrcID	A regular stream index	Not defined by PNC, handled by DEV that has control of the CTA.
Open CTA	BcstID	Any valid DEVID	Asynchronous stream index	CSMA/CA
Open MCTA	BcstID	Any valid DEVID	MCTA stream index	Slotted aloha

[a]A single DEVID is a DEVID that corresponds to a single physical DEV.
[b]Any nonreserved DEVID, as defined in 6.2.8. This includes DEVIDs that refer to multiple DEVs, e.g., GrpIDs, McstID, and BcstID.

In addition, CTAs are either dynamic or pseudo-static, as described in 7.6.4.2. All CTAs, with the exception of regular CTAs and private CTAs, are dynamic CTAs. All private CTAs are pseudo-static CTAs. Regular CTAs can be either dynamic or pseudo-static, depending on the CTA Type field in the Channel Time Request command when the CTA was originally requested.

A device (DEV) is able to request the creation or a change in a CTA using the Channel Time Request command, as described in 6.5.7.2. The piconet coordinator (PNC) interprets the command based on the target ID and stream index. The various interpretations of these requests are listed in Table F.2.

Table F.2—Interpretation of parameters in a Channel Time Request command

DestID	Stream index	CTA type	Description
Any DEVID	Unassigned	Regular CTA	New CTA.
Any DEVID	Assigned stream index	Regular CTA	Modify or terminate existing CTA.
Any DEVID	Asynchronous stream index	Regular CTA	Reserve or terminate asynchronous CTA.
Same as SrcID	Unassigned	Private CTA	New private CTA.
Same as SrcID	Assigned stream index	Private CTA	Modify or terminate existing private CTA.
DEVID different than SrcID	Stream index previously assigned to private CTA	Private CTA	Handover control of the private CTA to the DEV indicated in the DestID field.
UnassocID, Reserved DEVID	Any	N/A	Not allowed in a request, only the PNC assigns association CTAs.
BcstID	MCTA stream index	Open MCTA	Modify request for an open MCTA, PNC takes this as a suggestion.
BcstID	Asynchronous stream index	Open CTA	Modify request for open CTAs, PNC takes this as a suggestion.
PNCID	MCTA stream index	Regular MCTA	Modify request for DEV to PNC CTAs, PNC takes this as a suggestion.

F.1.2 Interpretation of channel time requests

The channel time request is based on the Channel Time Request time unit (TU), as defined in 6.5.7.2, which indicates the smallest unit of time that the DEV needs in the allocation. The Channel Time Request TU is specified because it allows the PNC to allocate time in useful amounts and also to split up an allocation so that the MaxTransmitDelay requirements, as defined in 5.3.14, for other allocations can be met. Consider the following example:

— Channel time request 1: Channel time is required one half of the superframe duration, and MaxTransmitDelay required is twice the superframe duration.
— Channel time request 2: Channel time required is one tenth of the superframe duration, and MaxTransmitDelay required is one quarter of the superframe duration.

To meet the needs of both requests, the PNC will have to split channel time request 1 into multiple allocations so that channel time request 2 will have at least four separate allocations spread throughout the superframe with the required MaxTransmitDelay.

The PNC interprets the channel time request based on the Channel Time Request TU, Minimum Number of TUs, Desired Number of TUs, CTA Rate Factor, and the CTA Rate Type fields. The Channel Time Request TU is used both to change the other numbers into time and to specify the smallest usable time for the request. Examples of how the PNC interprets these Channel Time Request command parameters are shown in Table F.3.

Table F.3—Possible allocations based on Channel Time Request command parameters

Minimum number of TUs	Desired number of TUs	CTA rate factor	CTA rate type	Allocation by the PNC
11	14	4	0 (super-rate)	11 to 14 Channel Time Request TUs per superframe in every superframe, in at least four allocations spread out evenly. One possible allocation is four allocations, two with 4 Channel Time Request TUs and two with 3 Channel Time Request TUs.
11	14	4	1 (sub-rate)	11 to 14 Channel Time Request TUs in a superframe, but the allocation occurs every 4th superframe and does not occur in the three intervening superframes. This is an average allocation of 2.75 to 3.5 Channel Time Request TUs per superframe.
5	5	0	0 (super-rate)	Not allowed; the CTA Rate Factor and CTA Rate Type fields cannot simultaneously be zero.
10	10	1	0 (super-rate)	10 Channel Time Request TUs per superframe in every superframe, possibly in one allocation. However, the PNC is free to split this allocation into multiple CTAs to support MaxTransmitDelay requirements of other channel time requests.

If a DEV sets the Desired Number of TUs field equal to the Minimum Number of TUs field, then it is indicating that its minimum required bit rate and desired bit rate are the same value. The DEV is also allowed to set the Desired Number of TUs field to a value greater than the Minimum Number of TUs field to indicate a desired bit rate greater than the minimum required bit rate. Asynchronous traffic is also able to use the isochronous method for channel time allocation by setting the Minimum Number of TUs to zero and the Desired Number of TUs such that it specifies the entire superframe. In this last case the PNC will understand that the DEV needs as much time as possible, but not at the expense of other time-critical streams.

The PNC will indicate the number of Channel Time Request TUs allocated per superframe with the Available Number of TUs field in the Channel Time Response command. If the PNC is unable to grant the request due to unavailable channel time, the PNC uses the Available Number of TUs field in the Channel Time Response command to indicate the number of Channel Time Request TUs that it would have been able to grant.

F.1.3 Determining CTA Rate Factor from stream requirements

F.1.3.1 General

In order to provide timely delivery of data, applications need periodic communication opportunities in such a way that the time between these opportunities is bounded. In some cases, a single channel time allocation in each superframe is sufficient to provide this level of service. However, some applications have latency requirements that require more than one channel time allocation in each superframe. For example, if an application needs to keep its latency below 10 ms and the superframe duration is 65 ms, then more than one channel time allocation per superframe, a super-rate allocation, is required. The following subclauses discuss a method that determines the correct CTA Rate Factor to request for a channel time allocation based on an upper bound on the maximum amount of time required between transmit opportunities.

The steps involved are as follows:

a) Determine the maximum time allowed between transmit opportunities (MaxTransmitDelay).
b) Determine the channel time required per superframe to support both the minimum bit rate (TimeRequiredPerSuperframe) and the desired bit rate (TimeDesiredPerSuperframe).
c) Calculate AllocationCriteria using:
   AllocationCriteria = 2 × (SuperframeDuration – TimeRequiredPerSuperframe).
d) If the AllocationCriteria is greater than the MaxTransmitDelay, then a super-rate allocation is required [proceed to step e)]. Otherwise, a sub-rate allocation will suffice [skip to step f)].
e) Make the following super-rate calculations:
   1) CTA Rate Factor = (SuperframeDuration – TimeRequiredPerSuperframe) / MaxTransmitDelay.
   2) Round this value up to the next highest integer.
   3) MinTimeToRequest = TimeRequiredPerSuperframe.
   4) DesiredTimeToRequest = TimeDesiredPerSuperframe.
   5) Skip to step g).
f) Make the following sub-rate calculations:
   1) CTA Rate Factor = MaxTransmitDelay / (SuperframeDuration – TimeRequiredPerSuperframe).
   2) Round this value down to the next power of 2 (because sub-rates are required to be powers of 2).
   3) MinTimeToRequest = (CTA Rate Factor × TimeRequestedPerSuperframe).
   4) DesiredTimeToRequest = (CTA Rate Factor × TimeDesiredPerSuperframe).
   5) If the amount of channel time requested is a significant fraction of the superframe, the request could be denied by the PNC, and so a lower CTA Rate Factor may be selected by the DEV. In this case, different MinTimeToRequest and DesiredTimeToRequest values would be calculated based on the CTA Rate Factor that was selected. If the amount of channel time is a still a significant fraction of the superframe even with a sub-rate CTA Rate Factor of 2, the DEV may request a super-rate allocation and use the calculations of step e).
g) Use the Channel Time Request TU to convert MinTimeToRequest and DesiredTimeToRequest into Minimum Number of TUs and Desired Number of TUs for the Channel Time Request command.

F.1.3.2 Example 1

The following is an example assuming a SuperframeDuration of 10 ms:

a) The stream requires a MaxTransmitDelay of less than 2 ms.
b) The stream requires 2 ms of channel time per superframe (TimeRequiredPerSuperframe = 2 ms) and desires 3 ms of channel time per superframe (TimeDesiredPerSuperframe = 3 ms).
c) AllocationCriteria = 2 × (10 ms – 2 ms) = 16 ms.
d) AllocationCriteria is greater than MaxTransmitDelay, so a super-rate allocation is required [proceed to step e)].
e) Make the following super-rate calculations:
   1) CTA Rate Factor = (10 ms – 2 ms)/(2 ms) = 8/2 = 4.
   2) No rounding required.
   3) MinTimeToRequest = TimeRequiredPerSuperframe = 2 ms.
   4) DesiredTimeToRequest = TimeDesiredPerSuperframe = 3 ms.
f) (Skipped)

g) If the Channel Time Request TU is 1 ms, then:
Minimum Number of TUs = MinTimeToRequest / 1 ms = 2,
Desired Number of TUs = DesiredTimeToRequest / 1 ms = 3.

The parameters for a Channel Time Request command for this example are as follows:

— CTA Rate Type = 0 (super-rate)
— CTA Rate Factor = 4
— Channel Time Request TU = 1 ms
— Minimum Number of TUs = 2
— Desired Number of TUs = 3

F.1.3.3 Example 2

The following in another example assuming a SuperframeDuration of 10 ms:

a) The stream requires a MaxTransmitDelay of less than 50 ms.
b) The stream requires 1 ms of channel time per superframe (TimeRequiredPerSuperframe = 1 ms) and desires 1.25 ms of channel time per superframe (TimeDesiredPerSuperframe = 1.25 ms).
c) AllocationCriteria = 2 × (10 ms – 1 ms) = 18 ms.
d) AllocationCriteria is less than MaxTransmitDelay, so a sub-rate allocation will suffice [proceed to step f)].
e) (Skipped)
f) Make the following sub-rate calculations:
   1) CTA Rate Factor = 50 ms / (10 ms – 1 ms) = 51/9 = 5.6.
   2) 5.6 is rounded down to 4 (next lower power of 2).
   3) MinTimeToRequest = (CTA Rate Factor × TimeRequiredPerSuperframe) = 4 ms.
   4) DesiredTimeToRequest = (CTA Rate Factor × TimeDesiredPerSuperframe) = 5 ms.
   5) At this point, if MinTimeToRequest and DesiredTimeToRequest are considered a significant portion of the superframe, then a lower sub-rate factor could be selected and the MinTimeToRequest and DesiredTimeToRequest recalculated (not done in this example).
g) If the Channel Time Request TU is 1 ms, then:
Minimum Number of TUs = MinTimeToRequest / 1 ms = 4,
Desired Number of TUs = DesiredTimeToRequest / 1 ms = 5.

The parameters for a Channel Time Request command for this example are as follows:

— CTA Rate Type = 1 (sub-rate)
— CTA Rate Factor = 4
— Channel Time Request TU = 1 ms
— Minimum Number of TUs = 4
— Desired Number of TUs = 5

F.1.4 PNC interpretation of CTA rate factor

The PNC interprets the CTA Rate Factor field as follows:

a) If the CTA Rate Type = 0 (super-rate), the PNC calculates the MaxTransmitDelay required using:

$$\text{MaxTransmitDelay} = (\text{SuperframeDuration} - \text{TimeRequiredPerSupeframe})/(\text{CTA Rate Factor}).$$

b) If the CTA Rate Type = 1 (sub-rate), the PNC uses the CTA Rate Factor to create a sub-rate allocation. The maximum MaxTransmitDelay that will be provided by the PNC using the provided CTA Rate Factor is calculated using:

$$\text{MaxTransmitDelay} = (\text{CTA Rate Factor} \times \text{SuperframeDuration}) - \text{TimeRequiredPerSuperframe}.$$

The MaxTransmitDelay calculated by the PNC will not always be equal to the MaxTransmitDelay desired by the DEV. However, the MaxTransmitDelay will always be less than or equal the MaxTransmitDelay desired by the DEV.

Regularly occurring opportunities to transmit (MaxTransmitDelay) does not guarantee that a specific latency will be achieved. Due to the nature of wireless communications, there is no guarantee that all transmission attempts will be successful. Therefore, specifying a maximum of 10 ms between transmit opportunities does not guarantee that a latency of 10 ms will necessarily be achieved.

F.1.5 Creating channel time requests from MLME requests

This subclause describes one possible method of converting MLME-CREATE-STREAM.request and MLME-MODIFY-STREAM.request parameters into a Channel Time Request command. However, the implementer is free to choose another method.

As an example, consider the following requirements for a stream:

- A SourceDataRate of 8 Mb/s throughput.
- A DesiredDataRate of 10 Mb/s throughput.
- The MaxTransmitDelay is 5 ms.
- High priority stream.
- MaxRetries is four.
- ReliabilityExponent is 4 (i.e., the frame error rate (FER) needs to be less than 10^{-4}).
- AggregateDataFrameSize of 1000 octets.

The medium access control (MAC) of the source DEV has the current parameters of the piconet as well as an estimate of the channel quality between it and the destination DEV. Assume the following parameters for the channel and piconet:

- The FER is less than 5%.
- Data rates available on the link between the two DEVs are 22, 33, 44.
- Superframe duration = 10 ms.
- ACK policy is Imm-ACK.
- Overhead is approximately 30 µs per frame, an ACK duration is 30 µs, and the SIFS is 10 µs.

Based on the preceding information, the source MAC will perform the following calculations:

- Need 8 Mb/s / 8000 bits/frame = 1000 frames/second.
- Desire 10 Mb/s / 8000 bits/frame = 1200 frames/second.
- NumberofRetries = ceiling($\log_{10}$(FER)/ReliabilityExponent) = ceiling($\log_{10}$(0.05)/4) = 4.
- With up to four retries of a frame, and additional 5.27% of frames is required, which translates to a minimum 1053 frames/second and a desired 1264 frames/second.
- Time per frame is = 8000 bits/44 Mb/s + 90 µs overhead = 271 µs/frame, so Channel Time Request TU = 271 µs.
- In each superframe, the DEV:
  Needs to send minimum of 1053 frames/second × 10 ms/superframe = 11 frames/superframe.
  Desires to send 1263 frames/second × 10 ms superframe = 13 frames/superframe.
- Time per frame is = 8000 bits/44 Mb/s + 90 µs overhead = 272 µs/frame.
- The TimeRequiredPerSuperframe = 11 frames/superframe × 0.272 ms = 3 ms and the TimeDesiredPerSuperframe = 13 frames/superframe × 0.272 ms = 3.6 ms.
- Using F.1.3, calculate AllocationCriteria = 2 × (10 ms – 3 ms) = 14 ms.
- The AllocationCriteria is greater than the MaxTransmitDelay (5 ms), so a super-rate is required.
- Using F.1.3, CTA Rate Factor = (10 ms – 3 ms) / 5 ms = 1.4, round up to 2.

- Using F.1.3, MinTimeToRequest = TimeRequiredPerSuperframe = 3 ms/superframe.
- Using F.1.3, DesiredTimeToRequest = TimeDesiredPerSuperframe = 3.6 ms/superframe.
- Using F.1.3, Minimum Number of TUs to request is 3 ms / 0.272 ms = 11 and the Desired Number of TUs to request is 3.8 ms / 0.272 ms = 13.

The DEV needs to perform two more checks. The first is to check to see if the number of Channel Time Request TUs in a CTA are sufficient to allow timely retries. In order to meet the latency requirement for successfully transmitting a frame, there needs to be time for the retransmissions within a single CTA. Otherwise, the retransmissions will occur too late. The NumberofRetries for this example is 4, thus a minimum of 5 Channel Time Request TUs are required in each CTA (1 for the frame plus 4 for retries). In this case, the minimum allocation will be 5 Channel Time Request TUs in one CTA and 6 in another, which will meet this requirement. If the NumberofRetries had been higher, 6 for example, the DEV needs increase the Minimum Number of TUs to 14 so that there is at least 7 Channel Time Request TUs in each CTA. Alternately, the DEV could request a higher CTA rate factor.

NOTE—Using this method will improve the latency guarantee so that it includes the maximum retries required within one CTA. However, this will also greatly increase the over-allocation of time in the superframe.

The second check is to verify that using the maximum number of retries does not delay frames in the following CTA past the end of that CTA. For this example, the number of frames required per superframe, without retries, is ten. If one frame requires 4 retries, then the request needs to allow for up to 14 Channel Time Request TUs per superframe. Likewise, to support the desired data rate, 12 frames plus 4 retries are required for a total of 16 Channel Time Request TUs per superframe. To meet this, the DEV will need to increase the numbers in the request.

Thus the DEV should use the following parameters in its Channel Time Request command:

- CTA Rate Type = 0 (super-rate)
- CTA Rate Factor = 2
- Channel Time Request TU = 272 µs
- Minimum Number of TUs = 14
- Desired Number of TUs = 16

The PNC, upon receiving the preceding parameters in the DEV's request, would calculate the maximum space allowed between allocated CTAs:

- Minimum amount of time requested = 11 × 0.271 ms = 3 ms.
- Using F.1.3, calculate MaxTransmitDelay = (10 ms – 3 ms) / 2 = 3.5 ms.

Based on this calculation the PNC is able to satisfy the request by providing a minimum 11 Channel Time Request TUs and a maximum 13 Channel Time Request TUs per superframe, allocated as 2 CTAs per superframe, with the maximum space between CTAs of 3.5 ms and a minimum CTA size of 1 Channel Time Request TU (272 µs). It is not possible for the PNC to know that the desired MaxTransmitDelay required was 5 ms. The PNC will indicate the number of Channel Time Request TUs allocated per superframe in the Channel Time Response command. If the request cannot be granted by the PNC because the minimum number of TUs per superframe is not available, the PNC will indicate the number of Channel Time Request TUs that are available in the Channel Time Response command.

As a second example, consider the preceding example modified such that the MaxTransmitDelay requirement was 20 ms instead of 5 ms. Because 20 ms is greater than the calculated AllocationCriteria of 14 ms, a sub-rate allocation would suffice.

- Using F.1.3, sub-rate CTA Rate Factor = 20 ms / (10 ms – 3 ms) = 20 / 7 = 2.86, round down to 2.

- Using F.1.3, MinTimeToRequest = (CTA Rate Factor × TimeRequiredPerSuperframe) = (3 ms × 2) = 6 ms.
- Using F.1.3, DesiredTimeToRequest = (CTA Rate Factor × TimeDesiredPerSuperframe) = (3.6 ms × 2) = 7.2 ms.
- Using F.1.3, the DEV may continue with the sub-rate request with a sub-rate factor of 2, or the DEV may decide to make a super-rate request instead. (For this example, assume that the DEV continues with the sub-rate request.)
- Using F.1.3, Minimum Number of TUs to request is 6 ms / 0.272 ms = 22 and the Desired Number of TUs to request is 7.2 ms / 0.272 ms = 26.

Again, for first reliability, the minimum CTA should be no less than 5 Channel Time Request TUs based on the NumberofRetries. The minimum number of Channel Time Request TUs requested is 22, which meets this requirement. For the second criteria, 20 frames are required every 2 superframes, and 24 frames are desired every 2 superframes. Allowing for an additional 4 retries increases the request to use 24 and 28.

Thus the DEV could use the following parameters in its request:

- CTA Rate Type = 1 (sub-rate)
- CTA Rate Factor = 2
- Channel Time Request TU = 272 µs
- Minimum Number of TUs = 24
- Desired Number of TUs = 28

The PNC is able to satisfy the request by providing a minimum 22 Channel Time Request TUs and a maximum 26 Channel Time Request TUs every other superframe, with a minimum CTA size of 1 Channel Time Request TU (272 µs). The PNC does not need to worry about MaxTransmitDelay requirements because the requesting DEV has already determined that the sub-rate factor selected will meet the MaxTransmitDelay requirements. The MaxTransmitDelay provided by the PNC using a sub-rate factor of 2 can be calculated as follows:

- TimeRequiredPerSuperframe = 22 × 0.271 ms = 6 ms
- Using F.1.3, MaxTransmitDelay = (2 × 10 ms) – 6 ms = 14 ms

F.1.6 Interpreting channel time requests

When the PNC grants channel time in response to a Channel Time Request command, the value of the Available Number of TUs field returned in the Channel Time Response command will be greater than or equal to Minimum Number of TUs from the corresponding request. If the request was a super-rate request, the Available Number of TUs field indicates the number of Channel Time Request TUs allocated per superframe. If the request was a sub-rate request, the Available Number of TUs field indicates the number of Channel Time Request TUs allocated per superframe at the requested sub-rate.

When the PNC denies a Channel Time Request command, the value of the Available Number of TUs returned in the Channel Time Response command will be less than Minimum Number of TUs from the corresponding request. If the request was a super-rate request, the Available Number of TUs field indicates the number of Channel Time Request TUs that the PNC could have allocated per superframe. If the request was a sub-rate request, the Available Number of TUs field indicates the number of Channel Time Request TUs the PNC could have allocated per superframe at the requested sub-rate.

Regardless of the result of the channel time request, the Available Number of TUs field can be converted to an AvailableDataRate and returned in the MLME-CREATE-STREAM.confirm primitive. If the request was granted, the AvailableDataRate is the data rate received. If the request was denied, the AvailableDataRate is the data rate the PNC could have provided.

If a super-rate was requested, the steps involved in determining an AvailableDataRate from Available Number of TUs are as follows:

— Frames Per Superframe = Available Number of TUs
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
— Frames Per Second = Total Frame Per Second – FER
— AvailableDataRate = Frames Per Second × Bits Per Frame

If a sub-rate was requested, the steps involved in determining an AvailableDataRate from Available Number of TUs are as follows:

— Frames Per Superframe = Available Number of TUs / CTA Rate Factor
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
— Frames Per Second = Total Frame Per Second × (1 – FER)
— AvailableDateRate = Frames Per Second × Bits Per Frame

As an example, consider the stream requirements, channel conditions, and calculated channel time request parameters for the first example shown in F.1.5.

If the super-rate request is granted and the Available Number of TUs field of the corresponding Channel Time Response command contains the value 12, the AvailableDataRate received can be calculated as follows:

— Frames Per Superframe = Available Number of TUs = 12
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
  = 12 frames/superframe / 0.010 s/superframe = 1200 frames/s
— Frames Per Second = Total Frame Per Second × (1 – FER) = 1200 × (1 – 5.26%) = 1140 frames/s
— AvailableDataRate = Frames Per Second × Bits Per Frame = 1140 frames/s × 8000 bits/frame = 9.12 Mb/s

If the request is denied and the Available Number of TUs field of the corresponding Channel Time Response command contains the value 8, the same steps can be followed to determine the data rate that the PNC could have provided:

— Frames Per Superframe = Available Number of TUs = 8 frames/superframe
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
  = 8 frames/superframe / 0.010 s/superframe = 800 frames/s
— Frames Per Second = Total Frame Per Second × (1 – FER) = 800 × (1 – 5.26%) = 760 frames/s
— AvailableDataRate = Frames Per Second × Bits Per Frame = 760 frames/s × 8000 bits/frame = 6.08 Mb/s

As a second example, consider the stream requirements, channel conditions, and calculated channel time request parameters for the second example shown in F.1.5.

If the sub-rate request is granted and the Available Number of TUs field of the corresponding Channel Time Response command contains the value 24, the AvailableDataRate can be calculated as follows:

— Frames Per Superframe = Available Number of TUs / CTA Rate Factor
  = 26 / 2 = 13 frames/superframe
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
  = 13 frames/superframe / 0.010 s/superframe = 1300 frames/s
— Frames Per Second = Total Frames Per Second × (1 – FER) = 1300 × (1 – 5.26%) = 1235 frames/s
— AvailableDataRate = Frames Per Second × Bits Per Frame = 1237 frames/s × 8000 bits/frame = 9.88 Mb/s

If the request is denied and the Available Number of TUs field of the corresponding Channel Time Response command contains the value 20, the same steps can be followed to determine the AvailableDataRate that the PNC could have provided at the requested sub-rate:

— Frames Per Superframe = Available Number of TUs / CTA Rate Factor
  = 20 / 2 = 10 frames/superframe
— Total Frames Per Second = Frames Per Superframe / Seconds Per Superframe
  = 10 frames/superframe / 0.010 s/superframe = 1000 frames/s
— Frames Per Second = Total Frame Per Second × (1 – FER) = 1000 × (1 – 5.26%) = 950 frames/s
— AvailableDataRate = Frames Per Second × Bits Per Frame = 950 frames/s × 8000 bits/frame = 7.60 Mb/s

F.2 Sample frames

The subclause presents sample frames that provide examples of the header check sequence (HCS) and frame check sequence (FCS) calculations as well as the scrambler from the 2.4 GHz PHY. Two data frames are presented with the following characteristics:

— Scrambler seed = 0b01
— Data rate = 55 Mb/s
— Payload length = 20 octets
— Protocol version = 0
— Frame type = data frame
— Security off (SEC = 0)
— ACK policy = Imm-ACK
— Retry = 0
— More data = 1
— PNID = 100
— DestID = 5
— SrcID = 3
— MSDU number = 320 (0x140)
— Fragment number = 3
— Last fragment = 4
— Stream index = 13

Two frame payloads are provided. The first has pseudo-random data in the frame payload. The second frame payload has octets that increase in value by one, i.e., octet 0 has value 0, octet 10 has value 10, etc.

In the figures that follow, the bits are listed least significant bit (LSB) on the right, most significant bit (MSB) on the left with four octets per line. The lowest numbered octets are the first line and higher number octets on subsequent lines, i.e., octets 3, 2, 1 and 0 are on line 1 (octet 3 on the left, octet 0 on the right) while octets 7, 6, 5 and 4 would be on line 2 (octet 7 on the left, octet 4 on the right).

The PHY header, MAC header and HCS are common to both frames and are shown in Figure F.1.

The Frame Payload field and FCS for pseudo-random data are shown in Figure F.2.

The MAC Frame Body field length is 160 + 32 = 192 bits. Because the frame is encoded with 5 bits per symbol, three stuff bits, 11.4.6, are added to make it an integer number of symbols, bringing the length to 195 bits. After the stuff bits are added, the scrambler is applied, 11.4.4, to the MAC header, HCS and MAC Frame Body field, a total of 291 bits, resulting in the bit stream, shown in Figure F.3. This is the bit stream will be modulated and sent with the PHY preamble and header over the air.

```
# name: PHY header
# length: 16
 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1
# name: MAC header
# length: 80
 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0
 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1
 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0
# name: HCS
# length: 16
 0 0 1 0 1 1 1 0 1 0 1 1 0 1 0 0
```

First bit sent over the air (bit 0 of octet 0)

First bit of fifth octet of MAC header (bit 0 of octet 4 of MAC header, octet 6 transmitted in the medium)

Last bit of HCS sent over the air. (bit 15 of octet 1 of the HCS, octet 13 transmitted in the medium)

Figure F.1—PHY header, MAC header, and HCS for sample frame

```
# name: Frame payload
# length: 160
 0 0 0 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 0 0 1 1 0 1
 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 0 0 1
 0 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 0
 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0
# name: FCS
# length: 32
 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 1 0 0 0 1 1 1 0 0 0 1 0
```

Figure F.2—Frame payload and FCS for sample frame with pseudo-random data

```
# name: Scrambled MAC header, HCS, frame payload and FCS
# length: 291
 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1
 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1
 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0
 1 0 0 1 1 1 1 1 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 1
 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1
 1 0 1 1 0 0 1 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0
 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0
 0 1 1 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0
 0 0 0
```

Figure F.3—MAC Header field, HCS, and MAC Frame Body field for sample frame with pseudo-random data after scrambler has been applied

The Frame Payload field and FCS for incremented data are shown in Figure F.4.

```
# name: Frame payload
# length: 160
 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0
 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0
 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 0
 0 0 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0
# name: FCS
# length: 32
 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 0
```

Figure F.4—Frame payload and FCS for sample frame with incremented data

The scrambled MAC Header field, HCS, and MAC Frame Body field for the incremented data frame are illustrated in Figure F.5.

```
# name: Scrambled MAC header, HCS, frame payload and FCS
# length: 291
 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1
 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1
 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0
 1 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0
 0 0 1 0 1 1 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 0
 1 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0
 1 0 1 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0
 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 0 0
 0 0 0 0 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 0 0
 0 0 0
```

Figure F.5—MAC Header field, HCS, and MAC Frame Body field for sample frame with incremented data after scrambler has been applied

F.3 Use of multiple millimeter wave (mmWave) PHYs in a single piconet

The MAC protocol allows the use in a CTA of any PHY mode or rate that is supported by both the source and the destination. Because of this, a DEV that joins a piconet that implements one PHY mode is able to use a different PHY mode when communicating with a DEV that also supports this mode.

As an example, a DEV that supports common mode signaling (CMS) and a high speed interface (HSI) modulation and coding scheme (MCS), as defined in 12.3.3.1, is able to join an SC piconet and request a CTA to another DEV that also supports CMS and HSI MCS. The two DEVs are allowed to carry on communications in that CTA using HSI MCSs, even though the piconet is operating in SC mode.

As another example, if a DEV supports HSI, but not CMS, it would need to find an HSI piconet, one of independent, parent or child, to join before it can transfer data. A DEV that supports both CMS and the HSI base rate could act as a bridge to an HSI-only DEV by creating an HSI child piconet in the SC parent piconet. The HSI only DEV could then join this child piconet and transfer data with the dual mode DEV. The dual mode DEV would then be able to either bridge the communications using the IEEE 802.1 sublayer or route the data using layers above layer 2.

As a further example, an HSI PNC-capable DEV may implement the CMS mode instead of HSI mode 0. In that case, the HSI PNC-capable DEV would operate the piconet in SC PHY, sending the beacon using CMS. DEVs in the piconet that support HSI PHY are then able to use the HSI MCSs for communication in CTAs.

Annex G

(normative)

Optional OOK/DAMI modes

G.1 Introduction

Piconet-capable (PNC-capable) on-off keying (OOK)/dual alternative mark inversion (DAMI) devices (DEVs) shall use child piconet, as described in G.2, for non-PNC-capable OOK/DAMI DEV communication.

G.2 Child piconet operation

When a PNC-capable OOK/DAMI DEV detects an single carrier (SC) piconet, it may join the SC piconet and request to create child piconet using the procedure described in 7.2.8, for non-PNC-capable OOK/DAMI DEV communication.

Alternatively, if a PNC-capable OOK/DAMI DEV does not detect any piconet, it may first start an SC piconet by using common mode signaling (CMS). To support communication with other OOK/DAMI DEVs, the PNC-capable OOK/DAMI DEV shall reserve private channel time allocation(s) (CTA), as described in 7.6.4.2, and operate a piconet within this reserved private CTA(s) by transmitting OOK/DAMI beacons.

Within the OOK/DAMI piconets in both scenarios described previously, OOK/DAMI DEVs shall follow the optional OOK/DAMI mode descriptions in 12.2.9 and the following PHY requirements.

G.3 DAMI

In DAMI modulation, the coded binary serial input data, $b[k]$, where $k = 0, 1, 2, \ldots$, shall be first pre-coded to form an intermediate data, $\bar{b}[k]$, defined as shown in Equation (G.1):

$$\bar{b}[k] = \bar{b}[k-2] \oplus b[k] \tag{G.1}$$

where the two initial values $\bar{b}[-2]=\bar{b}[-1]=0$ shall be used for pre-coding. The output $d[k]$ is formed by Equation (G.2):

$$\bar{d}[k] = K_{mod}(I[k] + jQ[k]) \tag{G.2}$$

where $I[k]$ and $Q[k]$ are given by Table G.1. The resulting constellation is illustrated in Figure 12-25(b). The normalization factor is sqrt(2).

Table G.1—DAMI encoding table

Pre-coded input bits $\bar{b}[k–2]\bar{b}[k]$	$I[k]$	$Q[k]$
00	0	0
01	1	
10	–1	
11	0	

The transmitted radio frequency (RF) signal for a DAMI system is a single-sideband (SSB) modulated signal accompanied by two low-power pilot tones. The SSB signal can be written as shown in Equation (G.3):

$$s_{SSB}(t) = s(t)\cos(2\pi f_c t) + \bar{s}(t)\sin(2\pi f_c t) \tag{G.3}$$

where

f_c is the center frequency
$s(t)$ is the baseband signal
$\bar{s}(t)$ is the Hilbert transform of $s(t)$

The baseband signal $s(t)$ can be represented by Equation (G.4):

$$s(t) = \sum_{k=0}^{N_p-1} d[k]g(t-kT_{sym}) \tag{G.4}$$

where

N_p is the number of symbols in the frame
T_{sym} is the symbol length
$g(t)$ is the baseband pulse shape

It is noted that one symbol corresponds to one bit for a DAMI system, meaning that the symbol length is the same as the bit length. The two pilot tones shall have frequencies f_c and f_c–1/(2T_{sym}), respectively. Both of them shall be in phase with the SSB signal. Their amplitudes shall be chosen such that the integrated power of each pilot is 25 dB (with ± 1 dB tolerance) below the integrated power of the SSB signal.

G.4 PHY preamble

In the OOK piconet, the preambles shall be modulated in OOK waveform with spreading factor of 2 in bit repetition. A single mandatory preamble is defined based on the Golay sequence of length 128, denoted $\mathbf{a}_{128}$, as shown in Table 12-6. Figure G.1 shows the structure of the PHY preamble.

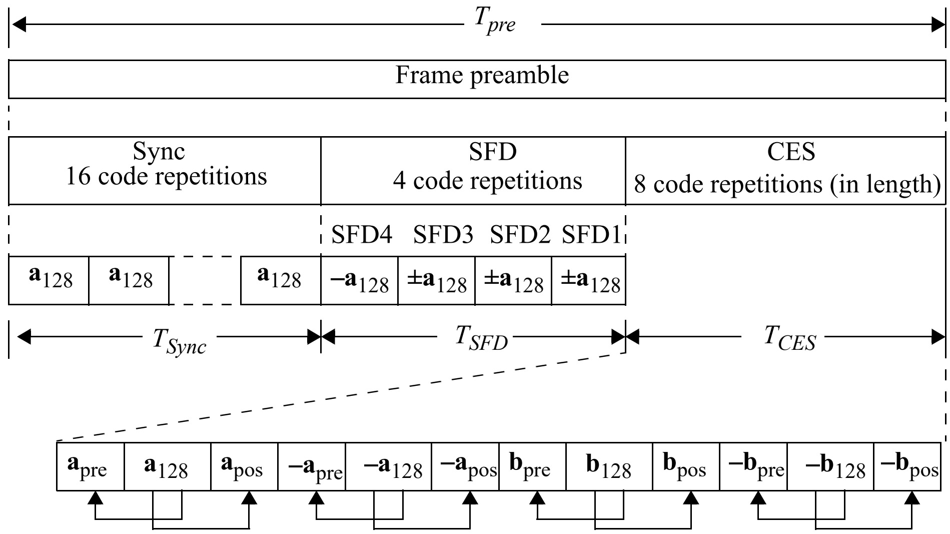

Figure G.1—OOK preamble structure

The frame synchronization (SNYC) field consists of 16 code repetitions of Golay sequence $\mathbf{a}_{128}$ as given in Table 12-6. The start-of-frame delimiter (SFD) consists of 4 code repetitions of Golay sequence $\mathbf{a}_{128}$ as given in Table 12-6. The usage of the four SFD codes shall be as follows: 1 for delimiter, 1 for channel estimation sequence (CES) selection, and 2 for OOK modulation and coding scheme (MCS) selections. SFD1 is defined as the delimiter. SFD2, the code for CES selection shall specify whether CES is adopted after the SFD4. If the value of SFD2 is positive, then CES shall be adopted, and if SFD2 is negative, then CES shall not be adopted. SFD3 and SFD4 are OOK MCS selection codes as shown in Table G.2.

Table G.2—SFD for OOK MCS selection

SFD pattern, SFD3 SFD4	OOK MCS	Spreading factor
+ +	OOK1	2
+ –	OOK2	1
– +	Reserved	Reserved
– –		

In OOK waveform, a negative sequence shall be derived by bit inverting as follows: $-x$ = Bit_Inverting(x), where x is a sequence in the form of binary bit 0 and 1. Bit_Inverting(x) is an operation to invert all the binary bits 0 of a sequence x to 1 and invert all the binary bits 1 of a sequence x to 0.

If SFD2 is positive and CES is adopted, the CES field shall be constructed from four Golay complementary sequences $\mathbf{a}_{128}$, $-\mathbf{a}_{128}$, $\mathbf{b}_{128}$ and $-\mathbf{b}_{128}$ as shown in Figure G.1. Each sequence shall be preceded by a cyclic prefix (i.e., a copy of the last 64 bits of the sequence) and followed by a cyclic postfix (i.e., a copy of the first 64 bits of the sequence). The pair of Golay complementary sequences $\mathbf{a}_{128}$ and $\mathbf{b}_{128}$ is given in Table 12-6, where both sequences in the form of binary bit 0 and 1. Another pair of Golay complementary sequences, $\mathbf{a}_{128}$ and $-\mathbf{b}_{128}$, shall be derived from the previous pair of $\mathbf{a}_{128}$ and $\mathbf{b}_{128}$ by bit inverting.

G.5 PHY frame format

In the OOK piconet, the frame header and the frame payload shall be modulated in OOK waveform with spreading factor of 2 in bit repetition.

The frame header that is defined in 12.2.4.3 shall be used in the OOK piconet. Then the frame header shall be modulated as shown in Figure G.2.

— Spread with a pseudo-random binary sequence (PRBS) generated using an linear feedback shift register (LFSR), as defined in 12.2.3.8.3, and the spreading factor is 16. The 15-bit seed value of the LFSR shall be equal to the value defined in 12.2.3.8.3.
— Spread with bit repetition and the spreading factor is 2.
— Map the frame header onto OOK waveform.

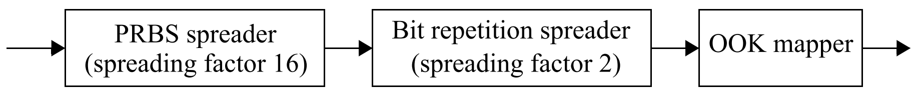

Figure G.2—OOK frame header spreading process

In OOK data payload, the transmit symbols shall be divided into block of length $N = 508 \times \mathrm{SF}$, where SF is the spreading factor in Table 12-11. This transmit symbol block shall be appended with pilot symbol, as described in Figure G.3. The pilot symbols consist of a sequence of length $N_p = 4 \times \mathrm{SF}$. The pilot symbols for OOK1 and OOK2 modes shall be chosen according to Table G.3.

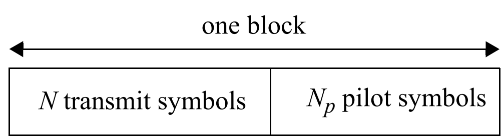

Figure G.3—OOK frame format with pilot symbols

Table G.3—OOK pilot symbols

Mode	Pilot symbols
OOK1	11001100
OOK2	1010

G.6 Interframe space

In the OOK/DAMI piconet, compliant implementation shall use the PHY layer timing parameter values, as defined in Table G.4.

Table G.4—PHY layer timing parameters

PHY parameter	Value	Subclause
pMifsTime	0.6 µs, 1.0 µs, 2.0 µs (default)	12.2.7.5
pSifsTime	1.0 µs, 2.0 µs, 2.5 µs, 6.0 µs (default)	12.2.7.4
pCcaDetectTime	5 µs	12.2.6.4
pChannelSwitchTime	100 µs	12.2.7.6

G.7 Eye opening for OOK

OOK shall have eye amplitude opening of 70% or more. The eye amplitude opening is defined as 2B/(A + B) × 100%, where A is the maximum amplitude and B is the minimum amplitude in eye diagram, as illustrated in Figure G.4. The test equipment shall have an amplitude measurement accuracy of 1/100 of the average amplitude of the recovered baseband signals. Timing measurement accuracy to detect the signal shall be 1/10 of the clock rate recovered from data.

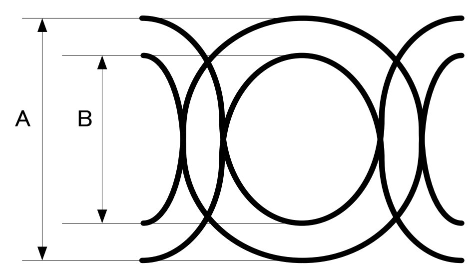

Figure G.4—OOK eye opening measurement

G.8 Error vector magnitude (EVM) for DAMI

The error vector magnitude (EVM) for DAMI shall be measured following the EVM calculation given in 12.1.8.1 for other SC MCS averaging over 1000 bits. The EVM value for DAMI shall be –14 dB or less.

IEEE SA
STANDARDS
ASSOCIATION

RAISING THE WORLD'S STANDARDS

Connect with us on:

Twitter: twitter.com/ieeesa

Facebook: facebook.com/ieeesa

LinkedIn: linkedin.com/groups/1791118

Beyond Standards blog: beyondstandards.ieee.org

YouTube: youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060

IEEE