IEEE STANDARDS ASSOCIATION IEEE

IEEE Standard for High Data Rate Wireless Multi-Media Networks

Amendment 1: High-Rate Close Proximity Point-to-Point Communications

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.15.3e™-2017
(Amendment to
IEEE Std 802.15.3™-2016)

IEEE Std 802.15.3e™-2017
(Amendment to
IEEE Std 802.15.3™-2016)

IEEE Standard for High Data Rate Wireless Multi-Media Networks

Amendment 1: High-Rate Close Proximity Point-to-Point Communications

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 February 2017
IEEE-SA Standards Board

Abstract: An alternative physical layer (PHY) and a modified medium access control (MAC) layer is defined in this amendment.

Two PHY modes have been defined that enable data rates up to 100 Gb/s using the 60 GHz band. MIMO and aggregation methods have been defined to increase the maximum achievable communication speeds. Stack acknowledgment has been defined to improve the medium access control (MAC) efficiency when used in a point-to-point (P2P) topology between two devices.

Keywords: 60 GHz, close proximity, fast setup, IEEE 802.15.3™, IEEE 802.15.3e™, millimeter wave, point-to-point, P2P, wireless

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

Print: ISBN 978-1-5044-4007-3 STD22571
PDF: ISBN 978-1-5044-4008-0 STDPD22571

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

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

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

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. 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.

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

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

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

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

Translations

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

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

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

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

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

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

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

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org/browse/standards/collection/ieee or contact IEEE at the address listed previously. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

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

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of 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.

Participants

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

Robert F. Heile, *Working Group Chair*
Rick Alfvin, *Working Group Vice Chair*
Patrick Kinney, *Working Group Vice Chair*, *Working Group Secretary*
James P. K. Gilb, *Working Group Technical Editor*
Benjamin A. Rolfe, *Working Group Treasurer*

Andrew Estrada, Task Group 15.3e *Chair*
Thomas Kürner, Task Group 15.3e *Vice-Chair*
Ken Hiraga, Task Group 15.3e *Secretary*
Ko Togashi, Task Group 15.3e *Technical Editor*

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

Major contributions were received from the following individuals:

Keiji Akiyama
Hideki Iwami
Jun Kobayashi
Keitarou Kondou
Jae Seung Lee
Moon-Sik Lee
Itaru Maekawa
Hiroshi Nakano
Makoto Noda
Ichiro Seto
Masashi Shimizu
Kiyoshi Toshimitsu

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

Tomoko Adachi
Iwan Adhicandra
Thomas Alexander
Richard Alfvin
Nobumitsu Amachi
Carol Ansley
Butch Anton
Yusuke Asai
Harry Bims
Nancy Bravin
Vern Brethour
William Byrd
Radhakrishna Canchi
William Carney
Juan Carreon
Yesenia Cevallos
Keith Chow
Charles Cook
Richard Doyle
Sourav Dutta
Richard Edgar
Andrew Estrada
David Evans
Avraham Freedman
Yukihiro Fujimoto
Devon Gayle
James P. K. Gilb
Randall Groves
Michael Gundlach
Robert F. Heile
Marco Hernandez
Ken Hiraga
Werner Hoelzl
Noriyuki Ikeuchi
Yasuhiko Inoue
Sergiu Iordanescu
Atsushi Ito
Raj Jain
Michael Johas Teener
Adri Jovin
Piotr Karocki
Jeritt Kent
Stuart Kerry
Yongbum Kim
Tero Kivinen
Jun Kobayashi
Keitarou Kondou
Yasushi Kudoh
Thomas Kuerner
Thomas Kurihara
Hyeong Ho Lee
Jae Seung Lee
Moon-Sik Lee
Arthur H Light
Michael Lynch
Elvis Maculuba
Itaru Maekawa
Michael McInnis
Michael Montemurro
Hiroshi Nakano
Michael Newman
Charles Ngethe
Paul Nikolich
John Notor
Yoshihiro Ohba
Satoshi Oyama
Arumugam Paventhan
Clinton C Powell
Venkatesha Prasad
Verotiana Rabarijaona
Maximilian Riegel
Robert Robinson
Benjamin Rolfe
Kazuyuki Sakoda
Naotaka Sato
Thomas Starai
Rene Struik
Walter Struppler
Mark Sturza
Bo Sun
Chen Sun
Masayuki Tsujita
Mark-Rene Uchida
Lorenzo Vangelista
Prabodh Varshney
George Vlantis
Hung-Yu Wei
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 14 February 2017, it had the following membership:

Jean-Philippe Faure, *Chair*
Vacant Position, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Gary Hoffman
Michael Janezic
Thomas Koshy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly
Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.15.3e-2017, IEEE Standard for High Data Rate Wireless Multi-Media Networks—Amendment 1: High-Rate Close Proximity Point-to-Point Communications.

IEEE Std 802.15.3e 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

Interest in developing a close proximity version of a 60 GHz band PHY and associated MAC changes began in 2014. Activity was initially conducted as part of Task Group 3d, formed in May 2014, which covered switched, point-to-point connections operating in the frequencies from 60 GHz up to the lower THz bands. The Application Requirements Document (ARD), Technical Requirements Documents (TRD), channel models, and regulatory issues 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 and a decision was made in November 2014 to split the Task Group 3d into two, one optimized for ranges under 10 centimeters (3e) and another covering several meters or more (3d). The first meeting of 3e as a task group 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.

Contents

IEEE Standard for High Data Rate Wireless Multi-Media Networks

Amendment 1: High-Rate Close Proximity Point-to-Point Communications

(This amendment is based on IEEE Std 802.15.3™-2016)

NOTE—The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

The editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

2. Normative references

Insert the following new reference in alphabetical order:

NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/ Counter Mode (GCM) and GMAC, Morris Dworkin, November 2007.[2]

[1]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.
[2]NIST publications are available from the National Institute of Standards and Technology (http://csrc.nist.gov/).

3. Definitions, acronyms, and abbreviations

3.1 Definitions

Insert the following definitions in alphabetical order:

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

pairnet associated phase: Phase during which the pairnet devices are associated.

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.

3.2 Acronyms and abbreviations

Change the following acronym as indicated:

PNID	piconet or pairnet identifier

Insert the following acronyms in alphabetical order:

GCM	Galois/counter mode
HRCP	high rate close proximity
IP	internet protocol
LLPS	low-latency power save
MIMO	multiple-input, multiple-output
OBEX[3]	object exchange
PAP	pairnet associated phase
PPRE	pilot preamble
PRC	pairnet coordinator
PRCID	pairnet coordinator identifier
PRDEV	pairnet DEV
PSP	pairnet setup phase
SISO	single-input, single-output
Stk-ACK	stack acknowledgment

4. General description

4.1 What is a piconet?

After 4.1, insert the following new subclause as 4.1a:

[3]The OBEX word mark and logo are trademarks owned by Infrared Data Association (IrDA®).

4.1a Pairnet

A pairnet consists of at most two DEVs, as shown in Figure 4-0a. Typical communication distance is 10 cm or less for a PRDEV. A PRDEV always connects as a pairnet.

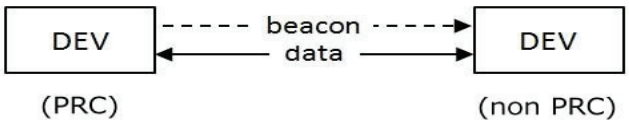


Figure 4-0a—IEEE 802.15.3 pairnet elements

4.2 Components of an IEEE 802.15.3 piconet

Insert the following new paragraph after the third paragraph of 4.2:

For HRCP PHYs, the IEEE 802.15.3 piconet is not used.

After 4.2, insert the following new subclause as 4.2.1:

4.2.1 Pairnet components

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

4.3 Overview of medium access control (MAC) functionality

4.3.1 Coordination

Insert the following new paragraph after the first paragraph of 4.3.1:

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

Change the title of 4.3.1.1, and insert the following new paragraph at the end of 4.3.1.1:

4.3.1.1 Starting a piconet <u>or a pairnet</u>

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 11a.1.4. The PRC need not scan the channels beforehand for availability.

Change the title of 4.3.2, and insert the following new paragraph at the end of 4.3.2:

4.3.2 Ending a piconet or a pairnet

In the case of 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.

4.3.4 Association and disassociation

Insert the following new paragraph after the second paragraph of 4.3.4:

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 7.3a.1. The peer-to-peer data transfer begins after completion of the association process.

Insert the following new paragraph after the third paragraph of 4.3.4:

Disassociation for pairnet takes place as specified in 7.3a.2.

4.3.5 Security overview

Change the text in 4.3.5 as indicated:

Security for the piconet or pairnet is 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 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 9a.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 8.4.2. For piconets, ~~T~~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.

4.3.6 The IEEE 802.15.3 superframe

Insert the following new text and Figure 4-2a at the end of 4.3.6:

The superframe structure for pairnets is shown in Figure 4-2a. 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 P2P connection. A Beacon frame includes information on the number and duration of the access slots which 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 6-50b. 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 an 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 SIFS or 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.

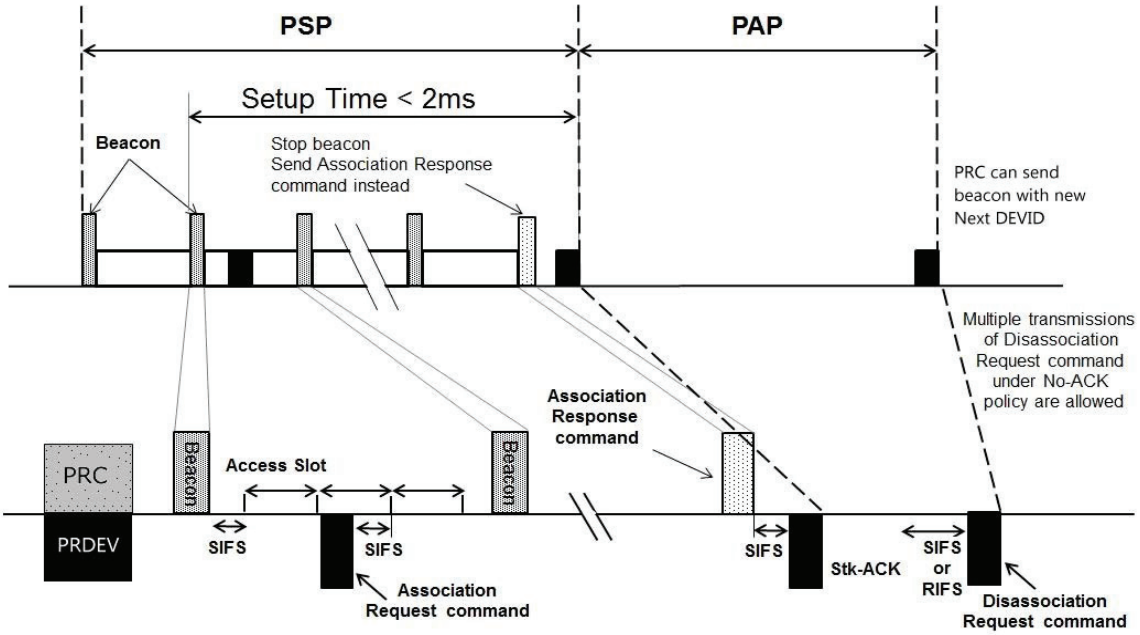


Figure 4-2a—One pairnet session

4.3.7 Channel time management

After 4.3.7, insert the following new subclause title as 4.3.7.1 and move all of 4.3.7 to 4.3.7.1:

4.3.7.1 Channel time management for piconets

After 4.3.7.1, insert the following new subclause as 4.3.7.2:

4.3.7.2 Channel time management for pairnets

There is one method for transmitting data between DEVs that form a pairnet. Data exchanges are achieved during a PAP, which is described in 7.4.4.

4.3.10 Dynamic channel selection

Add the following new sentence to the end of 4.3.10:

This operation is not applicable for pairnets.

4.3.14 Frame aggregation

Change the first paragraph in 4.3.14 as follows:

Frame aggregation, as described in 7.8, is supported for the purpose of high throughput. Two aggregation methods are provided for mmWave DEVs, one is suitable for normal data and A/V 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.

4.3.16 Channel probing

Insert the following new paragraph at the end of 4.3.16:

This operation is not applicable to pairnets.

4.5 Characteristics of the mmWave PHY

After 4.5, insert the following new subclauses as 4.5a, 4.5a.1, and 4.5a.2:

4.5a Characteristics of HRCP PHY

4.5a.1 HRCP PHY characteristics

The HRCP PHY, as described in Clause 11a, is designed for a high data-rate application. The frequency band is 57.0–66.0 GHz, the same as that for the mmWave PHY.

Two PHYs are defined for the HRCP PHY. They are as follows:

a) Single Carrier mode in HRCP (HRCP-SC) PHY, as described in 11a.2
b) OOK mode in HRCP (HRCP-OOK) PHY, as described in 11a.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-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 low-density parity-check (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.

4.5a.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 4-3a 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.

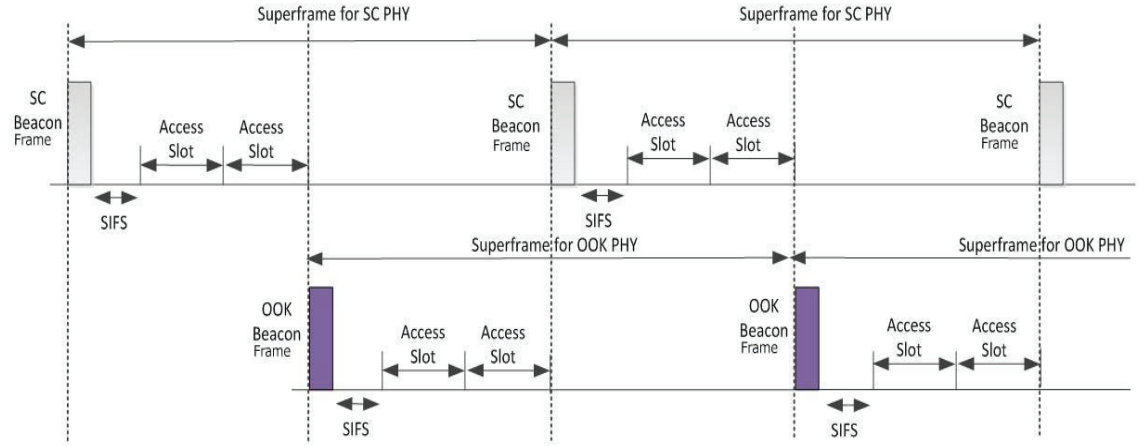


Figure 4-3a—Example of transmitting dual mode beacons

5. Layer management

5.3 MLME SAP interface

Change Table 5-3 and insert the following note (the entire table is not shown) as indicated:

Table 5-3—Summary of MLME primitives

Name	Request	Confirm	Indication	Response
MLME-ASSOCIATE	5.3.5.1	5.3.5.2	5.3.5.3	5.3.5.4
NOTE 1—MLME-ASSOCIATE.Response primitive is valid only for pairnets.				

Insert the following new paragraph at the end of 5.3:

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.

Change the title and first paragraph of 5.3.2 as follows:

5.3.2 Scanning for piconets and pairnets

These primitives support the process of determining the presence or absence of piconets or pairnets, as described in 7.2.1. The parameters used for these primitives are defined in Table 5-5.

Change Table 5-5 (the entire table is not shown) as indicated:

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 7.2.1. 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 5-6a	A set containing zero or more instances of a PairnetDescription	The PairnetDescriptionSet is returned to indicate the results of the scan request.

Add the following new paragraph and Table 5-6a at the end of 5.3.2:

In Table 5-5, a PairnetDescriptionSet is a set of PairnetDescriptions. Each PairnetDescription consists of the elements shown in Table 5-6a.

5.3.2.1 MLME-SCAN.request

Change the first two paragraphs of 5.3.2.1 as follows:

Table 5-6a—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 6.4.11a	As defined in 6.4.11a	Capability of the PRC in the Beacon frame.
HigherLayerProtocolInformation	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.38	Included 3 octets Unique ID and Higher Layer Protocol Information in variable length.
PhyMode	Enumeration	HRCP_SC_PHY, HRCP_OOK_PHY, HRCP_BOTH_PHY	The PHY mode that is being used in the pairnet that was found.

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
                            )
```

5.3.2.2 MLME-SCAN.confirm

Change the first paragraph of 5.3.2.2 as follows:

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:

Insert the following new parameter at the end of the list in 5.3.2.2 (before the closing parenthesis):

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

Change the second paragraph of 5.3.2.2 as follows:

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.

5.3.2.3 MLME-SCAN.indication

Insert the following new parameter at the end of the list in 5.3.2.3 (before the closing parenthesis):

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

Change the second paragraph as indicated:

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.

5.3.3 Starting a piconet or pairnet

Change the first paragraph of 5.3.3 as follows:

These primitives support the process of creating a new piconet or pairnet with the DEV acting as PNC or PRC, as described in 7.2.2. The parameters used for these primitives are defined in Table 5-8.

Change Table 5-8 (the entire table is not shown) as indicated:

Table 5-8—MLME-START primitive parameters

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.2	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 6.2.3	The assigned DEVID for the DEV that is acting as the PNC, as described in 7.2.2. 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 7.2.7 and 7.2.8. 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 7.2.7 and 7.2.8.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, HRCP SC PHY, HRCP OOK PHY, HRCP_BOTH_PHY	The PHY that will be used for the Beacon frames and the CP(s) in the piconet or pairnet that will be started.
PrcCapabilityIe	PRC Capability, as defined in 6.4.11a	As defined in 6.4.11a	Capability of the PRC in the Beacon frame.
HigherLayerProtocol Information	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.38	Includes 3 octets Unique ID and HigherLayerProtocol Information in variable length

5.3.3.1 MLME-START.request

Change the first paragraph of 5.3.3.1 as follows:

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:

Insert the following new parameter at the end of the list in 5.3.3.1 (before the closing parenthesis):

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

Change the title and first paragraph of 5.3.4 as follows:

5.3.4 Stopping a piconet or pairnet

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 7.2.9, or the handover of PNC operations to another DEV in the piconet, as described in 7.2.3 and 7.2.6. The parameters used for these primitives are defined in Table 5-9.

Change Table 5-9 (the entire table is not shown) as indicated:

Table 5-9—MLME-STOP primitive parameters

Name	Type	Valid range	Description
RequestType	Enumeration	SHUTDOWN, HANDOVER	If SHUTDOWN, the current piconet 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 PRDEVs.
HandoverTargetList	List of DEVIDs	0 to maximum number of DEVIDs, as defined in 6.2.3	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_DENIED,~~ HIGHER LAYER REQUESTED, OTHER	Indicates the reason for a ResultCode. NOT_A_PNC and HANDOVER_FAILED are not valid responses for PRDEVs.

5.3.4.1 MLME-STOP.request

Change the first paragraph of 5.3.4.1 as follows:

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

5.3.4.2 MLME-STOP.confirm

Change the first paragraph of 5.3.4.2 as follows:

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

Change the title and first paragraph of 5.3.5 as follows:

5.3.5 Associating with a piconet or pairnet

The following primitives support the process of a DEV associating with a PNC or PRC, as defined in 7.3.1. The parameters used for these primitives are defined in Table 5-10.

Change Table 5-10 as indicated. Only those rows being changed or inserted are shown:

Table 5-10—MLME-ASSOCIATE primitive parameters

Name	Type	Valid range	Description
Bsid	Octet string	As defined in 6.4.2	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 6.2.2.
PncAddress	MAC address	Any valid individual MAC address	The MAC address of the target PNC or PRC for the association.
PiconetServicesInquiry	Boolean	TRUE, FALSE	Requests that the PNC or PRC send the services information about the piconet or pairnet, as described in 7.3.2.
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 6.4.11b	As defined in 6.4.11b	Capability of the PRDEV in the Association Request command.
PairnetOperationParametersIe	As defined in 6.4.11c	As defined in 6.4.11c	Capability of the PRDEV in the Association Response command.
ReasonCode	Enumeration	REQUEST_TIMEOUT, PNC_NOT_FOUND, PNC_DENIED, PNCBUSY, ALREADY_ASSOCIATED, NEIGHBOR_REFUSED, HIGHER_LAYER_DENIED, OTHER	Indicates the reason for a ResultCode of FAILURE. For PRDEVs, NEIGHBOR_REFUSED is not applicable.
HigherLayerProtocolInformation	As defined in the Higher Layer Protocol Information IE	As defined in 6.4.38	Includes 3 octets, Unique ID and Higher Layer Protocol Information in variable length.

Change the first and second paragraphs of 5.3.5.1 as follows:

5.3.5.1 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 5-10.

Change the first and second paragraphs of 5.3.5.2 as follows:

5.3.5.2 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
                            )
```

5.3.5.3 MLME-ASSOCIATE.indication

Change the first paragraph of 5.3.5.3 as follows:

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,
                            )
```

After 5.3.5.3, insert the following new subclause as 5.3.5.4:

5.3.5.4 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 5-10.

Change the title and first paragraph of 5.3.6 as follows:

5.3.6 Disassociation from a piconet or pairnet

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 7.3.4. 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 5-11.

Change Table 5-11 (the entire table is not shown) as indicated:

Table 5-11—MLME-DISASSOCIATE primitive parameters

Name	Type	Valid range	Description
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.

5.3.6.1 MLME-DISASSOCIATE.request

Change the first and second paragraphs of 5.3.6.1 as follows:

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,
                            ReasonCode,
                            )
```

The primitive parameters are~~is~~ defined inTable 5-11.

5.3.6.3 MLME-DISASSOCIATE.indication

Change the first paragraph of 5.3.6.3 as follows:

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:

5.3.7 Security management

Change Table 5-12 and Table 5-13 as follows:

Table 5-12—MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitive parameters

Name	Type	Valid range	Description
SECID	2 octets	As defined in 6.2.7.2	The identifier for the key.
OrigId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a 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.
TrgtId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a 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 9a.3 for pairnet	The key used for protecting frames between this DEV and the TrgtId DEV.

Table 5-12—MLME-MEMBERSHIP-UPDATE and MLME-SECURITY-ERROR primitive parameters (*continued*)

Name	Type	Valid range	Description
SrcID	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a 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.

Table 5-13—MLME-SECURITY-MESSAGE primitive parameters

Name	Type	Valid range	Description
TrgtId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a for pairnet	Specifies the DEVID of the target of the MLME request.
OrigId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a for pairnet	Specifies the DEVID of the originator of the MLME request.
UniqueId	Octet string	Any valid OUI or CID, as defined in 6.4.7	A unique identifier for the entity that defines the format of the security information, as described in 6.4.7.
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.

5.3.7.4 MLME-SECURITY-MESSAGE.request

Change the first paragraph of 5.3.7.4 as follows:

This primitive initiates the sending of a Security Message command, as described in 6.5.9.1, to the target DEV in the piconet or pairnet. The semantics of this primitive are as follows:

5.3.7.6 MLME-SECURITY-MESSAGE.indication

Change the first paragraph of 5.3.7.6 as follows:

This primitive reports the reception of a Security Message command, as described in 6.5.9.1, from a DEV in the piconet or pairnet. The semantics of this primitive are as follows:

5.3.8 PNC handover

Insert the following new paragraph before the first paragraph of 5.3.8:

This function is not applicable for pairnets.

5.3.9 Requesting DEV information from the PNC

Insert the following new paragraph before the first paragraph of 5.3.9:

This function is not applicable for pairnets.

5.3.10 Security information retrieval

Change the first paragraph of 5.3.10 and Table 5-16 as follows:

These primitives are used to request security information about other DEVs in the piconet or pairnet, as described in 8.4.1. The parameters used for the MLME-SECURITY-INFO primitives are defined in Table 5-16.

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

Name	Type	Valid range	Description
QueriedDevId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a 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 6.2.3 for piconet and as defined in 6.2.3a for pairnet	The DEVID of the DEV for which the security information request is intended.
OrigId	Integer	Any valid DEVID, as defined in 6.2.3 for piconet and as defined in 6.2.3a for pairnet	Specifies the DEVID of the DEV that initiated the MLME request.

Table 5-16—MLME-SECURITY-INFO primitive parameters (*continued*)

Name	Type	Valid range	Description
NumSecurityRecords	Integer	0–65535	Number of entries in the SecurityRecordSet.
SecurityRecordSet	A set of Security Record fields, as defined in 6.5.4.4	A set containing 0 or more instances of variable-length Security Record field. The maximum number of instances depends on the size of the records, *pMaxFrameBodySize* and the length of the secure command security fields, as defined in 6.3.3.2 for piconet and as defined in 6.3.3a 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.

5.3.10.1 MLME-SECURITY-INFO.request

Change the first paragraph of 5.3.10.1 and Table 5-16 as follows:

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:

5.3.10.2 MLME-SECURITY-INFO.confirm

Change the first paragraph of 5.3.10.2 as follows:

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:

5.3.10.3 MLME-SECURITY-INFO.indication

Change the first paragraph of 5.3.10.3 as follows:

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:

5.3.13 Stream management

Insert the following new paragraph before the first paragraph of 5.3.13:

This function is not applicable for pairnets.

5.3.15 Power management

Change the first sentence of 5.3.15 as follows:

This mechanism supports the process of establishment and maintenance of power management (PM) modes of a DEV, as described in 7.14, and is applicable for piconets only.

5.3.16 Multicast operations

Change the first sentence of 5.3.16 as follows:

These primitives support multicast operations and thus are not applicable for pairnets that do not use multicast.

5.3.17 Timing synchronization

Insert the following new paragraph after the first paragraph of 5.3.17:

This function is not applicable for pairnets.

5.3.18 Transmit switched diversity (TSD)

Insert the following new paragraph before the first paragraph of 5.3.18:

This subclause only applies to mmWave DEVs.

5.4 MAC management

Change the title of 5.4.1 as follows:

5.4.1 MAC PIB PNC and PRC group

Insert the following new paragraph and Table 5-28a after Table 5-28:

The MAC PIB PRC group, given in Table 5-28a, describes the DEV's PRC capabilities as well as the characteristics of the current pairnet.

Table 5-28a—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

Table 5-28a—MAC PIB PRC group parameters (*continued*)

Managed Object	Octets	Definition	Access
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 6.4.3, that is sent in the Association Response command, as described in 6.5.1.2, when the PRDEV is acting as the PRC.	Read/Write
macAssocHigherLayerIe	Variable	A Higher layer protocol information IE, as defined in 6.4.38, that is sent in the Beacon frame and association related commands, as described in 7.3a.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

5.4.2 MAC PIB characteristic group

Insert the following new paragraph and Table 5-29a after Table 5-29:

The MAC PIB PRDEV characteristics group, given in Table 5-29a, contains information about the capabilities and characteristics of the PRDEV.

Table 5-29a—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

Table 5-29a—MAC PIB PRDEV characteristic group parameters (*continued*)

Managed Object	Octets	Definition	Access
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

Change Table 5-30 as follows:

Table 5-30—Summary of MAC SAP primitives

Name	Request	Confirm	Indication	Response
MAC-ASYNC-DATA	5.5.1	5.5.2	5.5.3	—
MAC-ISOCH-DATA	5.5.4	5.5.5	5.5.6	—
MAC-HRCP-DATA	5.5.7	5.5.8	5.5.9	—
MAC-HRCP-MUL-DATA	5.5.10	5.5.11	5.5.12	—

Change the title and insert the following rows at the end of Table 5-31:

Table 5-31—MAC-ISOCH-DATA, ~~and~~ MAC-ASYNC-DATA, MAC-HRCP-DATA, and MAC-HRCP-MUL-DATA primitive parameters

Name	Type	Valid range	Description
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 11a-6	MCS used in the transmitted PHY frame. Only applicable for HRCP SC PHY.
ChIdentifier	Enumeration	Any valid combinations of channel, as defined in Figure 11a-1	The frequency channel used in the transmitted PHY frame. Only applicable for HRCP SC PHY.

Table 5-31—MAC-ISOCH-DATA, ~~and~~ MAC-ASYNC-DATA, MAC-HRCP-DATA, and MAC-HRCP-MUL-DATA primitive parameters (*continued*)

Name	Type	Valid range	Description
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"
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 802-2014	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.

After 5.5.6, insert the following new subclauses as 5.5.7–5.5.12:

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 HRCP PHY. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-31.

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 HRCP PHY. This primitive is only generated if the ConfirmRequested parameter in the MAC-HRCP-DATA.request with the same RequestID value is ALWAYS or is ON_ERROR and the ResultCode is FAILURE. 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 only applicable for HRCP SC PHY. The semantics of this primitive are as follows as follows:

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

The primitive parameters are defined in Table 5-31.

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    (
                            Length,
                            Data
                            )
```

The primitive parameters are defined in Table 5-31.

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 HRCP PHY. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-31.

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. This primitive is only generated if the ConfirmRequested parameter in the MAC-HRCP-MUL-DATA.request primitive 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-HRCP-MUL-DATA.confirm   (
                            RequestID,
                            TransmitDelay,
                            ResultCode,
                            ReasonCode,
                            MCSIdentifier,
                            ChIdentifier
                            )
```

The primitive parameters are defined in Table 5-31.

5.5.12 MAC-HRCP-MUL-DATA.indication

This primitive is used to indicate the reception of a Multi-protocol MSDU. The semantics of this primitive are as follows:

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

The primitive parameters are defined in Table 5-31.

6 . MAC frame formats

6.2 General frame format

Change the first paragraph and dashed list in 6.2 as follows:

The MAC frame format, illustrated in Figure 6-3 for piconet DEVs and in Figure 6-3a for PRDEVs, comprises a set of fields that occur in a fixed order in all frames. The figures in this subclause 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 6-3 for piconet DEVs and in Figure 6-3a for PRDEVs. For piconet DEVs, t~~T~~he 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:

— 10.2.8.1 for the 2.4 GHz PHY
— 11.2.7.1 for the SC PHY mode
— 11.3.6.3 for the HSI PHY mode
— 11.4.1.3.1 for the AV PHY mode
— 11a.2.7.1 for the HRCP-SC PHY mode
— 11a.3.7.1 for the HRCP-OOK PHY mode

Change the title of Figure 6-3 as follows:

Figure 6-3—MAC Header field and MAC Frame Body field formats for piconet

Insert Figure 6-3a as follows:

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 6-3a—MAC Header field and MAC Frame Body field format for pairnet

Change the title of Figure 6-4 as follows:

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

After Figure 6-4, insert the following new paragraph and Figure 6-4a:

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

Octets: variable
Frame Payload
MAC Frame Body

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

Change the title of Figure 6-5 as follows:

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

After Figure 6-5, insert the following new paragraph and Figure 6-5a:

The secure MAC frame body for PRDEVs shall be formatted as illustrated in Figure 6-5a 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 6-5a—Secure MAC Frame Body field format for pairnet

6.2.1 Frame Control field

Change the title of Figure 6-6 as follows:

Figure 6-6—Frame control field format for piconet

After Figure 6-6, insert the following new paragraph and Figure 6-6a:

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

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

Figure 6-6a—Frame control field format for pairnet

6.2.1.1 Protocol Version field

Change the first paragraph of 6.2.1.1 as follows:

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.

6.2.1.2 Frame Type field

Change the title of Table 6-1 as follows:

Table 6-1—Valid frame type values for piconet

After Table 6-1, add the following new text and Table 6-1a:

Table 6-1a lists the valid frame type values and their description for pairnet.

Stk-ACK frames with no data which 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.

Change the title of 6.2.1.4 as follows:

Table 6-1a—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	6.3.3a
0b100	Data frame	6.3.4a
0b101	Multi-protocol Data frame	6.3.5a
0b110–0b111	Reserved	—

6.2.1.4 ACK Policy field, Imp-ACK Request field, and Blk-ACK field <u>for piconet</u>

Change the title of Table 6-2 as follows:

Table 6-2—Valid ACK policy field type values <u>for piconet</u>

After 6.2.1.4, insert new subclause 6.2.1.4a as follows:

6.2.1.4a ACK Policy field for pairnet

The ACK Policy field 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 7.9.

Table 6-2a—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 7.9.1.
0b01	Reserved	—
0b10	Stk-ACK	The addressed recipient uses a Stk-ACK procedure for subframe exchange described 7.9.2a.
0b11	Reserved	—

After 6.2.1.8, insert new subclause 6.2.1.9 as follows:

6.2.1.9 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.

After 6.2.2, insert new subclause 6.2.2a as follows:

6.2.2a PairnetID

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.

Change the title of subclause 6.2.3 as follows:

6.2.3 SrcID and DestID fields for piconet

After 6.2.3, insert the following new subclause as 6.2.3a:

6.2.3a SrcID and DestID fields for pairnet

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 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 reserved:

- The DEVID value of 0x00 shall be reserved for the PRC
- The DEVID values of 0xED through 0xFE shall be reserved
- The DEVID value of 0xFF shall be reserved for broadcast frames (BcstID)

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

6.2.5 Stream Index field

Insert text immediately after the last paragraph in 6.2.5 as follows:

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

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

6.2.6 MAC header validation

Change the dashed list after the first paragraph as follows:

- 10.2.9 for the 2.4 GHz PHY

— 11.2.3.2.2 for the SC PHY mode
— 11.3.3.4 for the HSI PHY mode
— 11.4.1.4 for the AV PHY mode
— 11a.2.3.2.2 for HRCP-SC PHY mode
— 11a.3.3.2.2 for HRCP-OOK PHY mode

6.2.7 MAC Frame Body field

Change the first paragraph in 6.2.7.1 as follows:

6.2.7.1 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 6.2.7.9 for pairnet.

Change the second paragraph in 6.2.7.2 as follows:

6.2.7.2 Secure session ID (SECID) field

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.

Change the first paragraph of 6.2.7.3 as follows:

6.2.7.3 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. ~~A~~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 8.3.5. For pairnets, a DEV shall not reuse a frame counter with the same key, as described in 8.3.5. ~~T~~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. ~~W~~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.

6.2.7.4 Secure Payload field

Change the first paragraph in 6.2.7.4 as follows:

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 9a. 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 6.2.7.9 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.

6.2.7.5 Integrity Code field

Change the first paragraph of 6.2.7.5 as follows:

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 9a.3 for pairnets.

After 6.2.7.6, insert the following new subclauses as 6.2.7.7, 6.2.7.8, and 6.2.7.9:

6.2.7.7 MAC frame body for pairnet

The MAC frame body for pairnet is described in 6.3.1.1a, 6.3.3a.1, 6.3.4a.1 and 6.3.5a.2.

6.2.7.8 Security header for pairnet

The Security header for pairnet shall be formatted as illustrated in Figure 6-8a.

Octets: 2	6
SECID	SFC

Figure 6-8a—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 6.2.7.9.

The SFC field contains a counter that is used to ensure the uniqueness of the nonce of a secure frame, as defined in 6.2.7.3.

6.2.7.9 Secure MAC frame body for pairnet

The Secure MAC frame body for pairnet is described in 6.3.1.2a, 6.3.3a.2, 6.3.4a.2 and 6.3.5a.2.

After 6.2.9.5, insert the new subclause 6.2.10:

6.2.10 TX and ACK Information field for pairnets

The TX and ACK Information field shall be formatted as illustrated in Figure 6-46a.

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 6-46a—TX and ACK Information field format for pairnets

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 7.9.2a. 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

Change the title of subclause 6.3.1.1 as follows:

6.3.1.1 Non-secure Beacon frame <u>for piconet</u>

After 6.3.1.1, insert the following new subclause as 6.3.1.1a:

6.3.1.1a Non-secure Beacon frame for pairnet

The Beacon frame shall be formatted as illustrated in Figure 6-50a.

Octets:15	variable	…	variable	4
Pairnet Synchronization Parameters	Information element-1		Information element-n	FCS

Figure 6-50a—Non-secure Beacon frame Frame Payload field format for pairnets

The individual information elements (IEs) in the Beacon frame body are listed in Table 6-13a. 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 6-50b.

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 6-50b—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. 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 6-50c.

Bits: b0–b3	b4	b5–b7
Reserved	SEC Mode	Reserved

Figure 6-50c—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 6-5.

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 6-6a.

Table 6-6a—MAC header settings for a Beacon frame for pairnets

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-1a	Decoded
SEC	0	Decoded
ACK policy	No-ACK value in Table 6-2a	May be ignored
DestID	BcstID	Decoded
SrcID	PRCID	Decoded
TX and ACK Information	0x000000	May be ignored
Stream Index	0x00	May be ignored

Change title of subclause 6.3.1.2 as follows:

6.3.1.2 Secure Beacon frame <u>for piconet</u>

After 6.3.1.2, insert the following new subclause 6.3.1.2a:

6.3.1.2a Secure Beacon frame for pairnet

The Secure Beacon frame shall be formatted as illustrated in Figure 6-51a. 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 6-51a—Secure Beacon frame format for pairnets

The SECID field is defined in 6.2.7.2.

The SFC field is used by the DEV for this frame to ensure the uniqueness of the nonce, as defined in 6.2.7.3.

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 6.3.1.1a.

The Integrity Code is defined in 6.2.7.5.

The MAC header settings for a Secure Beacon frame shall be set and interpreted as described in Table 6-7a.

Table 6-7a—MAC header settings for a Secure Beacon frame for pairnets

Header field	Setting on transmission	Interpretation on reception
Frame type	Beacon value in Table 6-1a	Decoded
SEC	1	Decoded
ACK policy	No-ACK value in Table 6-2a	May be ignored
DestID	BcstID	Decoded
SrcID	PRCID	Decoded
TX and ACK Information	0x000000	May be ignored
Stream Index	0x00	May be ignored

Change the title of subclause 6.3.3 as follows:

6.3.3 Command frame <u>for piconet</u>

6.3.3.2 Secure Command frame

Change the title of Table 6-11 as follows:

Table 6-11—MAC header settings of a Secure Command frame <u>for piconet</u>

After 6.3.3.2, insert the following new subclauses as 6.3.3a, 6.3.3a.1, and 6.3.3a.2:

6.3.3a Command frame for pairnet

6.3.3a.1 Non-secure Command frame

The Non-secure command frame format shall be structured as illustrated in Figure 6-56a.

Octets: 4	variable	4
MAC Subheader	Command Block	FCS

Figure 6-56a—Non-secure Command frame format for pairnets

The MAC Subheader field is defined in 6.3.4a.1.

The Command Block field shall be formatted as shown in Figure 6-56b.

Octets: 2	2	variable
Command Type	Length	Payload

Figure 6-56b—Command block format for pairnet

The Command Type field indicates the type of command and is defined in Table 6-22a.

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 header settings for a command frame shall be set and interpreted as described in Table 6-11a.

Table 6-11a—MAC header settings for a non-secure command frame for pairnet

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-1a	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

6.3.3a.2 Secure command frame

The Secure command frame format shall be formatted as illustrated in Figure 6-56c. 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 6-56c—Secure command frame format

The SECID is defined in 6.2.7.2.

The SFC is defined in 6.2.7.3.

The Integrity code field is defined in 6.2.7.5.

The FCS field is defined in 6.2.7.6. 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 6-56b.

The MAC header settings for a secure command frame shall be set and interpreted as described in Table 6-11b.

Table 6-11b—MAC header settings for a Secure Command frame for pairnet

Header field	Setting on transmission	Interpretation on reception
Frame type	Command value in Table 6-1a	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	0x00	May be ignored

Change the title of subclause 6.3.4 as follows:

6.3.4 Data frame <u>for piconet</u>

After 6.3.4.2, insert the following new subclauses as 6.3.4a, 6.3.4a.1, and 6.3.4a.2:

6.3.4a Data frame for pairnet

6.3.4a.1 Non-Secure Pairnet Aggregated Data frame

Figure 6-58a 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 6-58a—Frame Payload field format for Non-Secure Pairnet Aggregated Data frame

The MAC Subheader field shall be formatted as illustrated in Figure 6-58b.

Octets: 3	1
Subframe Information	Subheader HCS

Figure 6-58b—MAC Subheader field format

The Subframe Information field shall be formatted as illustrated in Figure 6-58c.

Bits: b0	b1–b10	b11–b23
Last Fragment	Sequence Number	Payload length

Figure 6-58c—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 6-58d.

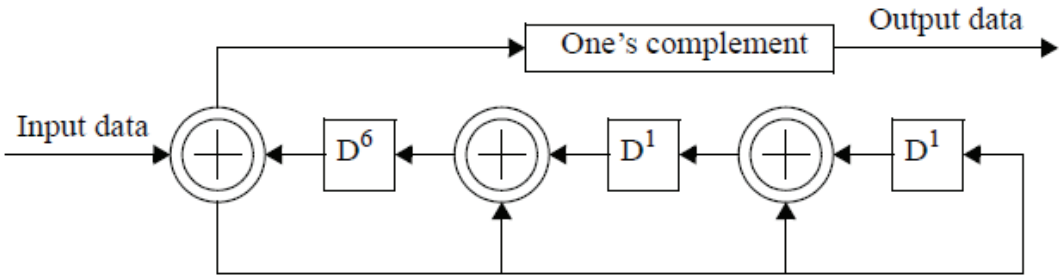


Figure 6-58d—HCS implementation example for Subheader HCS

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 7.16.

The format of the MAC Subframe Body field for aggregation shall be formatted as illustrated in Figure 6-58e.

Octets: variable	4	variable
Payload	FCS	Padding

Figure 6-58e—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 6.2.7.6.

The Padding is defined in 7.8.3.

The MAC header settings for a Non-Secure Pairnet Aggregated Data frame shall be set and interpreted as described in Table 6-11c.

Table 6-11c—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 6-1a	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

6.3.4a.2 Secure Pairnet Aggregated Data frame

Figure 6-58f and Figure 6-58g illustrate the Secure Pairnet Aggregated Data frame.

Octets: 2	6	variable
SECID	SFC	Secure Pairnet Aggregated Data Frame Payload

Figure 6-58f—Frame Payload field format for Secure Aggregated Data Frame

The SECID is defined in 6.2.7.2.

The SFC is defined in 6.2.7.3. 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 6-58g.

Octets: 4	variable	16	4	variable	…	4	variable	16	4	variable
MAC Sub-header 1	Secure Payload 1	Integrity Code	FCS	Pad-ding		MAC Sub-header *n*	Secure Payload *n*	Integrity Code	FCS	Pad-ding
	Secure MAC Subframe Body 1						Secure MAC Subframe Body *n*			

Figure 6-58g—Secure Pairnet Aggregated Data Frame Payload field format

The MAC Subheader field is defined in 6.3.4a.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 9a, that is to be transferred to a DEV.

The Integrity Code field is defined in 6.2.7.5.

The Padding is defined in 7.8.3.

The FCS field is defined in 6.2.7.6. 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 6-11d.

Table 6-11d—MAC header settings for a Secure Pairnet Aggregated Data frame

Header field	Setting on transmission	Interpretation on reception
Frame type	Data value in Table 6-1a	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	0x00	May be ignored

6.3.5a Multi-protocol Data frame for pairnet

6.3.5a.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 6-58e but the Payload field is replaced by the Pairnet Multi-protocol Data Payload format illustrated in Figure 6-61a.

Octets: 1	variable	variable
Data ID	Data Header	Data Payload

Figure 6-61a—Pairnet Multi-protocol Data Payload format

The Frame Type field shall be set to the Multi-protocol Data frame value in Table 6-1a, 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 6-12a.

Table 6-12a—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 6-1a	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 6.3.5.1.

6.3.5a.2 Secure Pairnet Aggregated Multi-protocol Data frame

The Secure Pairnet Aggregated Multi-protocol Data frame uses the same frame format as Figure 6-58g but the Payload 1 through *n* fields are replaced by the Pairnet Multi-protocol Data Payload format illustrated in Figure 6-61a. Figure 6-62b illustrates the Secure Pairnet Aggregated Multi-protocol Data Frame Payload field format.

<table>
<tr><th>Octets: 4</th><th>1</th><th>vari able</th><th>vari able</th><th>16</th><th>4</th><th>varia ble</th><th>…</th><th>4</th><th>1</th><th>vari able</th><th>variable</th><th>16</th><th>4</th><th>vari able</th></tr>
<tr><td rowspan="2">MAC Sub-header 1</td><td>Data ID</td><td>Data Head er</td><td>Data Pay-load 1</td><td>Integ rity Code</td><td>FCS</td><td>Pad-ding</td><td></td><td rowspan="2">MAC Sub-header n</td><td>Dat a ID</td><td>Data Hea der</td><td>Data Payload n</td><td>Integ rity Code</td><td>F C S</td><td>Pad-ding</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 6-62b—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 6-1a 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 6-12b.

Table 6-12b—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 6-1a	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 6.2.7.2.

The SFC field is defined in 6.2.7.3.

The MAC Subheader field is defined in 6.3.4a.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 6.3.5.1.

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 9a.

The Integrity Code field is defined in 6.2.7.5.

The FCS field is defined in 6.2.7.6. 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 7.8.3.

6.4 Information elements (IEs)

Change title of Table 6-13 as indicated:

Table 6-13—IEs <u>for piconet</u>

After Table 6-13, insert the following new text, Table 6-13a, and Table 6-13b:

The IEs for PRDEVs are listed in Table 6-13a.

Table 6-13a—IEs for pairnet

Element ID hex value	Element	Subclause	Present in Beacon frame
0x00	Reserved	—	—
0x01	BSID	6.4.2	In every Beacon frame
0x02–0x09	Reserved	—	—
0x0a	PRC Capability	6.4.11a	In every Beacon frame
0x0b–0x22	Reserved	—	—
0x23	MIMO Information	6.4.37	As needed
0x24	PRDEV Capability	6.4.11b	Non-Beacon frame IE
0x25	Pairnet Operation Parameters	6.4.11c	Non-Beacon frame IE
0x26	Higher Layer Protocol Information	6.4.38	As needed
0x27–0x7F	Reserved	—	—
0x80–0xFF	Vendor Specific IE	6.4.17	As needed

The requirements for supporting an IE are listed in Table 6-13b for HRCP PHY DEVs.

Table 6-13b—Requirements for supporting an IE

Element	HRCP PHY
CTA IE	Not used
BSID IE	Mandatory
Parent Piconet IE	Not used
DEV Association IE	Not used
PNC Shutdown IE	Not used
Piconet Parameter Change IE	Not used
Application Specific IE (ASIE)	Not used
Pending Channel Time Map (PCTM) IE	Not used
PNC Handover IE	Not used
CTA Status IE	Not used
Capability IE	Not used
PRC Capability IE	Mandatory
Transmit Power Parameters IE	Not used
PS status IE	Not used

Table 6-13b—Requirements for supporting an IE (*continued*)

Element	HRCP PHY
Continued Wake Beacon (CWB) IE	Not used
Overlapping PNID IE	Not used
Piconet Services IE	Not used
Group ID IE	Not used
Stream Renew IE	Not used
Next PNC IE	Not used
Piconet Channel Status IE	Not used
Synchronization IE	Not used
TSD IE	Not used
UEP Specific IE	Not used
IFS IE	Not used
CTA Relinquish Duration IE	Not used
Feedback IE	Not used
Mapping IE	Not used
BST Clustering IE	Not used
PET Clustering IE	Not used
Beam PET IE	Not used
HRS Beam PET IE	Not used
PET Amplitude IE	Not used
PET Phase IE	Not used
Sync Frame Frequency IE	Not used
Directional Peer IE	Not used
MIMO Information IE	Optional
PRDEV Capability IE	Mandatory
Pairnet Operation Parameters IE	Mandatory
Higher Layer Protocol Information IE	Optional
Vendor Specific IE	Optional

After 6.4.11, insert the following new subclauses as 6.4.11a, 6.4.11b, and 6.4.11c:

6.4.11a PRC Capability IE

The PRC Capability IE Content field shall be formatted as illustrated in Figure 6-87a. The PRC Capability IE shall be included in each Beacon frame.

Octets: 7
PRC Capability

Figure 6-87a—PRC Capability IE Content field format

The PRC Capability field shall be formatted as illustrated in Figure 6-87b.

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	Reserved	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 6-87b—PRC Capability field format

The SC Capable field shall be set to one if the DEV supports the SC PHY, as defined in 11a.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 11a.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 6.3.5a.1 and 6.3.5a.2, and shall be set to zero otherwise.

The LLPS Control field contains LLPS related parameters, as defined in Figure 6-87c.

Bits: b0	b1	b2	b3	b4	b5	b6	b7
LLPS Allow	LLPS Interval			LLPS Start		LLPS Extend	

Figure 6-87c—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 6-17a.

Table 6-17a—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 6-17b.

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 6-17c.

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 6-17d.

Table 6-17b—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

Table 6-17c—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

Table 6-17d—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 6-87b, 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 6-17e.

Table 6-17e—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

Table 6-17e—Preferred Total Aggregation Size field values (*continued*)

Bits: b0	b1	b2	Preferred Total Aggregation Size
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 6-87d. Each field shall be set to one for supported capability, and otherwise set to zero.

Bits: b0	b1
64 bit unit of padding supported	128 bit unit of padding supported

Figure 6-87d—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 6-87e.

Bits: b0	b1	b2
SC 16-QAM supported	SC 64-QAM supported	SC 256-QAM supported

Figure 6-87e—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 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 6-87f.

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 6-87f—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 6-87g and Table 6-17f. 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 6-17f means the allowable aggregation channel for each pattern.

Bits: b0	b1	b2	b3	b4	b5	b6
pattern0	pattern1	pattern2	pattern3	pattern4	pattern5	pattern6

Figure 6-87g—SC Supported Channel Aggregation pattern field format

Table 6-17f—Channel Aggregation Patterns for SC PHY in Figure 6-87g

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 6-87h.

Bits: b0	b1	b2
OOK 2 channel bonding is supported	OOK 3 channel bonding is supported	OOK 4 channel bonding is supported

Figure 6-87h—OOK Supported Channel Bonding field format

The CHNL_ID used for OOK channel bonding is specified in 11a.3.1.1.

6.4.11b PRDEV Capability IE

The PRDEV Capability IE Content field shall be formatted as illustrated in Figure 6-87i. The PRDEV Capability shall be included in each association request command frame.

Octets: 7
PRDEV Capability

Figure 6-87i—PRDEV Capability IE Content field format

The PRDEV Capability field shall be formatted as illustrated in Figure 6-87b, with the LLPS control bits changed to reserved.

The SC Capable field is defined in 6.4.11a.

The OOK capable field is defined in 6.4.11a.

The Supported SIFS field is defined in 6.4.11a.

The Multi-protocol Support field is defined in 6.4.11a.

The Preferred Payload Size field is defined in 6.4.11a.

The Preferred Total Aggregation Size field is defined in 6.4.11a.

The Supported Unit of Subframe Padding field is defined in 6.4.11a.

The Pilot Symbol Capable field is defined in 6.4.11a.

The SC Supported MCS field is defined in 6.4.11a.

The SC Supported Channel Bonding Capability field is defined in 6.4.11a.

The SC Supported Channel Aggregation Pattern field is defined in 6.4.11a.

The OOK Spreading field is defined in 6.4.11a.

The OOK Supported Channel Bonding field is defined in 6.4.11a.

6.4.11c Pairnet Operation Parameters IE

The Pairnet Operation Parameters IE Content field shall be formatted as illustrated in Figure 6-87j. 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.

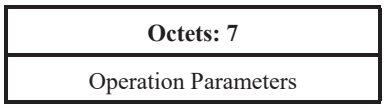


Figure 6-87j—Pairnet Operation Parameters IE Content field format

The Operation Parameters field shall be formatted as illustrated in Figure 6-87k.

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	Reserved	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 6-87k—Operation Parameters field format

The PHY Mode field indicates which PHY mode is used in the session as defined in Table 6-17g. A value of 10b indicates SC PHY is used while a value of 01b indicates OOK PHY is used. Other bit patterns shall not be used.

Table 6-17g—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 6.4.11a. The larger value of SIFS in PRC and DEV capability shall be encoded in this field.

The Multi-protocol Support field is defined in 6.4.11a. 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 6.4.11a. 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 6.4.11a. 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 6.4.11a. 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 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 6-87l 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 6-87l—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 6-87f 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 6-87m 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 6-87m—OOK Channel Bonding field format

After 6.4.36, insert the following new subclauses as 6.4.37 and 6.4.38:

6.4.37 MIMO Information IE

The MIMO Information IE Content field shall be formatted as illustrated in Figure 6-125a.

Octets: 2
MIMO Parameter

Figure 6-125a—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 6-125b.

Bits: b0–b3	b4–b12	b13–b14	b15
SC Supported MIMO	Number of MIMO Array Training	Array Training Interval	MIMO CES Type

Figure 6-125b—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 6-125c.

Bits: b0	b1	b2	b3
2×2 MIMO supported	4×4 MIMO supported	9×9 MIMO supported	16×16 MIMO supported

Figure 6-125c—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 6-21a.

Table 6-21a—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 11a.2.8.5.3, and shall be set to zero when MIMO CES for time domain channel estimation is used as described in 11a.2.8.5.4.

6.4.38 Higher Layer Protocol Information IE

The Higher Layer Protocol Information IE Content field shall be formatted as illustrated in Figure 6-125d.

Octets: 3	variable
Unique ID	Higher Layer Protocol Information

Figure 6-125d—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 7.3a.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

Change the first paragraph of 6.5 as follows:

The MAC commands are listed in Table 6-22 and Table 6-22a. If the column labeled “Associated” in Table 6-22 or Table 6-22a is marked with an “X,” then that MAC command shall only be sent by a DEV that is associated in the piconet or the pairnet. If the column labeled “Secure membership (if required)” in Table 6-22 or Table 6-22a is marked with an “X” and secure membership is required for the piconet or the pairnet, then that 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 6-22 and in Table 6-22a. The “Required” column indicates the type of DEVs that are required to support the command.

Change the title of Table 6-22 as follows:

Table 6-22—Command types for piconet

After Table 6-22, insert the following new text and tables as Table 6-22a and Table 6-22b:

The command types for PRDEVs are given in Table 6-22a.

The requirements for MAC commands based on the PHY type are listed in Table 6-22b.

Table 6-22a—Command types for pairnet

Command type hex value b15–b0	Command name	Subclause	Associated	Secure membership (if required)
0x0000	Association Request command	6.5.1.1	—	
0x0001	Association Response command	6.5.1.2	X	
0x0002	Disassociation Request command	6.5.1.3	X	
0x0003	Request Key command	6.5.2.1	X	X
0x0004	Request Key Response command	6.5.2.2	X	X
0x0005	Distribute Key Request command	6.5.2.3	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	6.5.4.4	X	X
0x000E	Probe Request command	6.5.4.5	X	
0x000F	Probe Response command	6.5.4.6	X	
0x0010–0x0017	Reserved	—	—	
0x0018	Transmit Power Change command	6.5.7.5	X	
0x0019	Array Training command	6.5.9.5	X	
0x001A	Array Training feedback	6.5.9.6	X	
0x001B–0x001D	Reserved	—	—	
0x001E	Security Message command	6.5.9.1	X	
0x001F–0x00FF	Reserved	—	—	
0x0100–0xFFFF	Vendor Defined	6.5.9.2	X	

Table 6-22b—MAC Command usage requirements

Command name	HRCP PHY
Association Request command	Mandatory
Association Response command	Mandatory
Disassociation Request command	Mandatory
Request Key command	Optional
Request Key Response command	Optional
Distribute Key Request command	Optional

Table 6-22b—MAC Command usage requirements (*continued*)

Command name	HRCP PHY
Distribute Key Response command	Optional
PNC Handover Request command	Not used
PNC Handover Response	Not used
PNC Handover Information command	Not used
PNC Information Request command	Not used
PNC Information command	Not used
Security Information Request command	Optional
Security Information command	Optional
Probe Request command	Optional
Probe Response command	Optional
Piconet Services command	Not used
Announce command	Not used
Channel Time Request command	Not used
Channel Time Response command	Not used
Channel Status Request command	Not used
Channel Status Response command	Not used
Remote Scan Request command	Not used
Remote Scan Response command	Not used
Transmit Power Change command	Optional
PS Set Information Request command	Not used
Array Training command	Optional
Array Training Feedback command	Optional
PS Set Information Response command	Not used
SPS Configuration Request command	Not used
SPS Configuration Response command	Not used
PM Mode Change command	Not used
Security Message command	Optional
Announce Response command	Not used
PM Mode Change Response command	Not used
ASIE Request command	Not used
ASIE Response command	Not used
Multicast Configuration Request command	Not used

Table 6-22b—MAC Command usage requirements (*continued*)

Command name	HRCP PHY
Multicast Configuration Response	Not used
Vendor Defined	Optional

6.5.1 Association and disassociation commands

Change the first paragraph of 6.5.1as follows:

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 pairnet.

6.5.1.1 Association Request command

Change the first paragraph of 6.5.1.1 as follows:

The Association Request command Payload field shall be formatted as illustrated in Figure 6-126 for piconet and Figure 6-126a 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, t~~T~~he SrcID shall be set to either the UnassocID, as described in 6.2.3, or the DEV's newly allocated DEVID, as described in 7.3.1. For pairnet operation, the SrcID shall be set to the DEVID obtained from the Next DEVID field in the Beacon frame, as described in 7.3a.1.

Change the title of Figure 6-126 as follows:

Figure 6-126—Association Request command Payload field format for piconets

After Figure 6-126, insert Figure 6-126a:

Octets: 6	As defined in 6.4.11b	2	variable
DEV Address	PRDEV Capability	ATP	IEs

Figure 6-126a—Association Request command Payload field format for pairnets

Change the fourth and fifth paragraphs of 6.5.1.1 as follows:

The Association Timeout Period (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 7.3.4.

NOTE—It is recommended that a PRDEV should use a short ATP length value less than or equal to 500 ms.

6.5.1.2 Association Response command

Change the first paragraph of 6.5.1.2 and the title of Figure 6-128 as follows:

The Association Response command Payload field shall be formatted as illustrated in Figure 6-128 for a piconet and in Figure 6-128a 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 6.2.3, for piconet DEVs or the DEVID that has been announced in the Beacon frame, as described in 6.2.3a, for PRDEVs.

Figure 6-128—Association Response command Payload field format for piconets

After Figure 6-128, insert the following new Figure 6-128a:

Octets: 6	1	2	1	variable/0
DEV address	DEVID	ATP	Reason code	IEs

Figure 6-128a—Association Response command Payload field format for pairnets

After the fourth paragraph of 6.5.1.2, add the following note:

NOTE —It is recommended that a PRDEV should use a short ATP length value less than or equal to 500 ms.

Change the fourth paragraph of 6.5.1.2 as indicated:

The ATP field contains the finalized value for the Association Timeout Period in milliseconds. This value may be different from that requested by the DEV in its Association Request command if the PNC or PRC is not able to support the value requested.

Change the dashed list in 6.5.1.2 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
- ~~9~~10–254 –› Reserved
- 225 –› Other failure

6.5.1.3 Disassociation Request command

Change the dashed list in 6.5.1.3 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
— ~~5~~7–254 –› Reserved
— 225 –› Other failure

6.5.2 Security commands

Change the first paragraph of 6.5.2 as follows:

This set of commands is used to establish the security and privacy functions between a DEV and the PNC in a piconet, ~~and~~ between DEVs in the piconet, and between a DEV and the PRC in a pairnet.

6.5.2.3 Distribute Key Request command

Change the first paragraph of 6.5.2.3 as follows:

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 piconet, t~~T~~his 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 6-131.

6.5.4.4 Security Information command

Change the sixth paragraph of 6.5.4.4 as follows:

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.

6.5.4.5 Probe Request command

Change the first paragraph of 6.5.4.5 as follows:

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 6-23 and Table 6-24. 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 6-143.

Change the fourth paragraph of 6.5.4.5 as follows:

If the IE Request Type field indicates that the IEs Requested field is a bitmap, then the sender shall set a value of one in a bit to request the IE that corresponds to the bit position. Otherwise, the sender shall set the bit to zero. The bit position for an IE is same as the value of the element-ID for that IE. That is, the bit position of *n* in information request field corresponds with the IE whose element ID, Table 6-13 for non-PRDEVs and Table 6-13a for PRDEVs, is *n*.

Change the eighth paragraph of 6.5.4.5 as follows:

Table 6-23 lists the rules that shall apply to requesting IEs from another non-PRDEV based on the identity of the originator of the request.

Change the title of Table 6-23 as follows:

Table 6-23—Rules for requesting IEs in a Probe Request command for non-PRDEVs

After the eighth paragraph of 6.5.4.5, add the following new paragraph and Table 6-23a:

Table 6-23a lists the rules that shall apply to requesting IEs from another PRDEV based on the identity of the originator of the request.

Table 6-23a—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	6.4.11a	Shall not request	May request
MIMO Information	6.4.37	Shall not request	May request
PRDEV Capability	6.4.11b	May request	Shall not request
Pairnet Operation Parameters	6.4.11c	Shall not request	May request

6.5.4.6 Probe Response command

Change the third paragraph of 6.5.4.6 as follows:

Table 6-24 lists the rules that shall apply to a non-PRDEV responding to a request for an IE based on the sender of the request.

Change the title of Table 6-24 as follows:

Table 6-24—Rules for responding to requests in Probe commands for non-PRDEVs

After the third paragraph of 6.5.4.6, add the following new paragraph and Table 6-24a:

Table 6-24a lists the rules that shall apply to PRDEVs responding to a request for an IE based on the sender of the request.

6.5.9 Special commands

6.5.9.1 Security Message command

Change the first paragraph in 6.5.9.1 as follows:

Table 6-24a—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	6.4.11a	Shall ignore	Shall respond
MIMO Information	6.4.37	Shall ignore	May ignore
PRDEV Capability	6.4.11b	Shall respond	Shall ignore
Pairnet Operation Parameters	6.4.11c	May respond	Shall respond

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 6-166.

After 6.5.9.4, add the following new subclauses as 6.5.9.5 and 6.5.9.6:

6.5.9.5 Array Training command

The Array Training command is used to select a set of antenna elements used in MIMO communication after association is established, as described in 11a.2.8.3. 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 6-169a.

3
MIMO Parameter

Figure 6-169a—Array Training command Payload field format

MIMO Parameter field shall be formatted as illustrated in Figure 6-169b.

Bits:b0–b8	b9–b17	b18–23
Number of Array Training from DEV	Number of Array Training Remained	Reserved

Figure 6-169b—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.

6.5.9.6 Array Training Feedback

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 6-169c.

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 6-169c—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 6-169d.

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 6-169d—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 6-169e.

Octets: 1	1	…	1
RSSI of Array Training command #1	RSSI of Array Training command #2	…	RSSI of Array Training command #N_{ar}

Figure 6-169e—RSSI report field

Values in the RSSI of Array Training command field are shown in Table 6-26a. The resolution of this field is 1 dB and therefore has a range of –71 dBm to –10 dBm.

Table 6-26a—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

Table 6-26a—Valid Number of RSSI of Array Training command field value (*continued*)

Value	RSSI of the Array Training command [dBm] or reception status
…	…
0x3E	–10
0x3F—0xFF	Reserved

7. MAC functional description

After 7.2, insert the following new subclause as 7.2a:

7.2a 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 6.2.10 and shall initialize the Last Received Sequence Number fields for both data frames and command frames using the values given in 6.3.4a.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.

After 7.3, insert the following new subclauses as 7.3a–7.3a.3:

7.3a Association and disassociation with a pairnet

7.3a.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 6.5.1.1, 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 which 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 6.5.1.2, 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 7-20a.

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 7-20b, or b) a data frame from the DEV, as shown in Figure 7-20c. 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 7-20d.

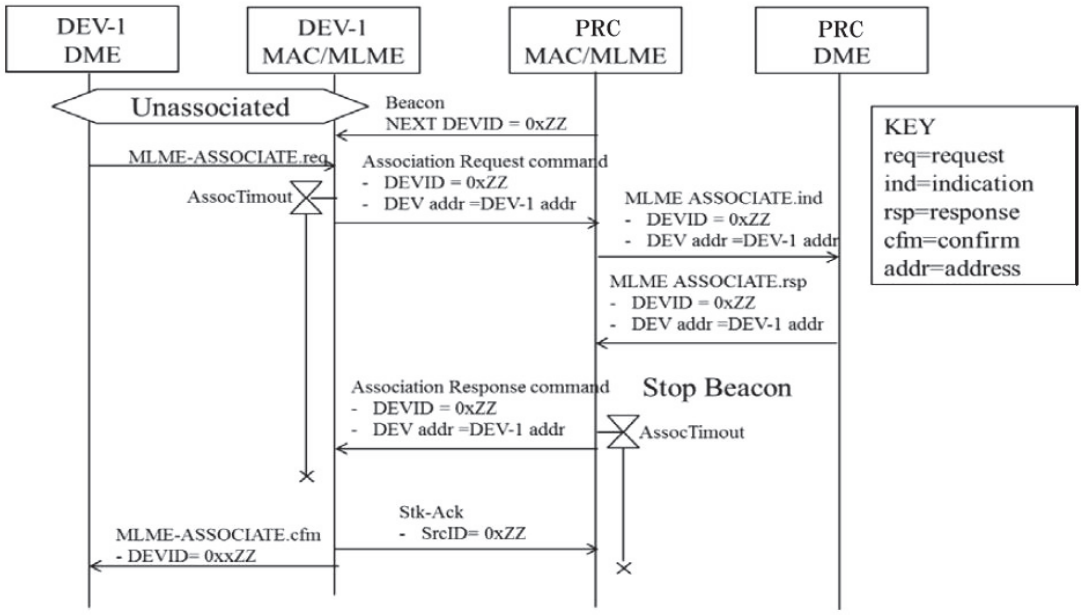


Figure 7-20a—MSC of DEV-1 associating

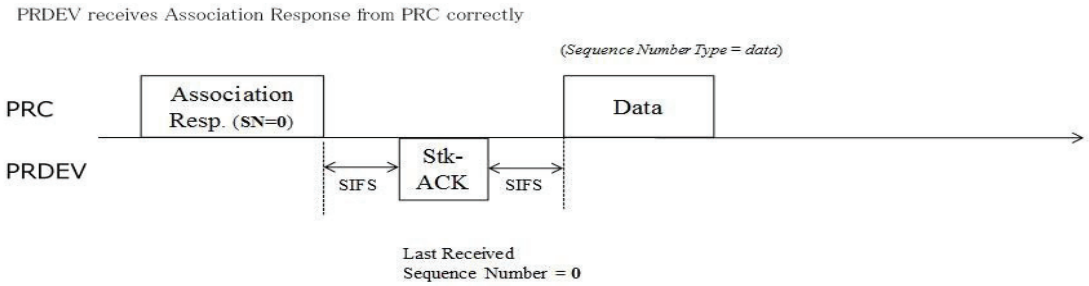


Figure 7-20b—Association correctly completed (Case 1)

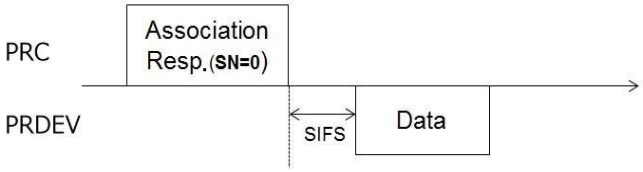


Figure 7-20c—Association correctly completed (Case 2)

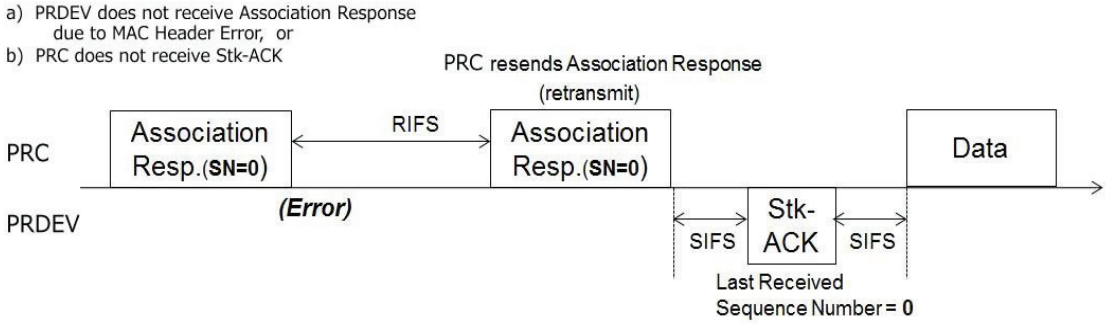


Figure 7-20d—Recovery of Association Response command

An additional setup procedure is required when using the HRCP SC PHY in MIMO mode as specified in 11a.2.8.3.

7.3a.2 Disassociation

When a PRC wants to remove a PRDEV from the pairnet, the PRC shall send a Disassociation Request command, as described in 6.5.1.3, 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.

7.3a.3 Higher layer protocol setup during association procedure for a pairnet

Higher layer protocol setup, such as Internet Protocol (IP) layer setup or object exchange (OBEX) 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 6.4.38, 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.

7.4 Channel access

7.4.1 Interframe Space (IFS)

Change 7.4.1 as follows:

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 10.2.7.1 for the 2.4 GHz PHY
— In 11.2.6 for the SC PHY
— In 11.3.5 for the HSI PHY
— In 11.4.1.2. for the AV PHY
— In 11a.2.6.1 for the HRCP-SC PHY
— In 11a.3.6.1 for the HRCP-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 which 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 CP, however, the retransmissions shall follow the CAP rules described in 7.4.2. The rules for acknowledgment and retransmissions are described in 7.9. The interframe space requirement for the beacon is ensured by the location of the CTAs, which is determined by the PNC, as described in 7.4.3.6.

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 7.8.

After 7.4.3.8, insert the following new subclause as 7.4.4:

7.4.4 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 7.9.2a.2.

During the Synchronous Phase, when the DEV of either side receives the frame and the MAC header has no error, after the end of the PPDU following the SIFS containing the frame, 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.

7.6 Synchronization for piconet

Insert 7.6a at the end of 7.6.

7.6a Synchronization for pairnet

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 7-52a. 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.

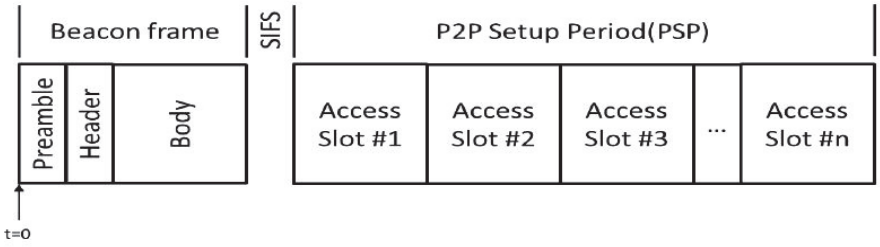


Figure 7-52a—Pairnet timing relative to the beacon

7.6a.1 Time accuracy

A compliant implementation shall maintain the accuracy of the timer to be at least as accurate as *pClockAccuracy.*

7.6a.2 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.

7.6a.3 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 7-3a.

Table 7-3a— IEs included in beacons as needed

IE	Format	Usage
MIMO Information IE	6.4.37	11a.2.8.1
Higher Layer Information IE	6.4.38	7.3a.3

7.8 Aggregation

Delete the first paragraph of 7.8 as shown:

~~Aggregation may be performed for high-speed data/video transmission or low-latency bidirectional data transmission. Accordingly, there are two aggregation methods defined: standard aggregation and low-latency aggregation.~~

After 7.8.2, insert the following new subclause as 7.8.3:

7.8.3 Pairnet aggregation

Pairnet data transmission shall use the pairnet aggregation frame format.

Figure 7-54a 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 which 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 6.3.4a, 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 6.2.1.4a.

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 6.2.10, up to 256 subframes are aggregated into a single frame. Figure 7-54b 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 6.3.4a.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.

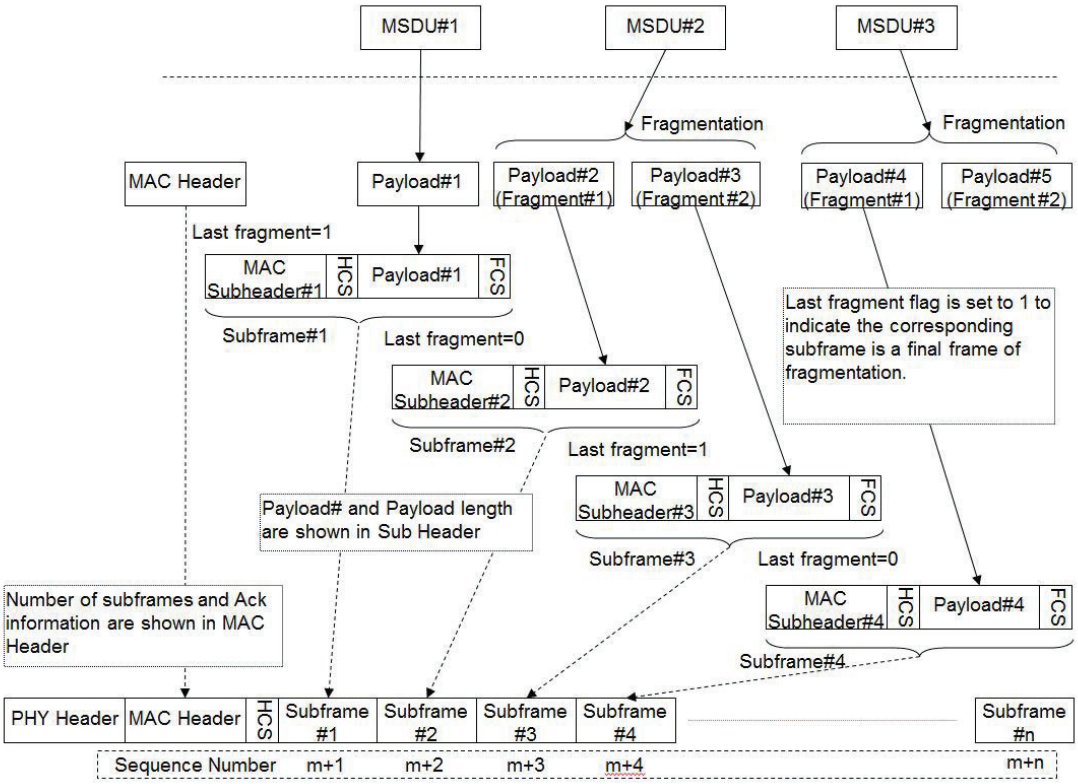


Figure 7-54a—Aggregation at originating DEV

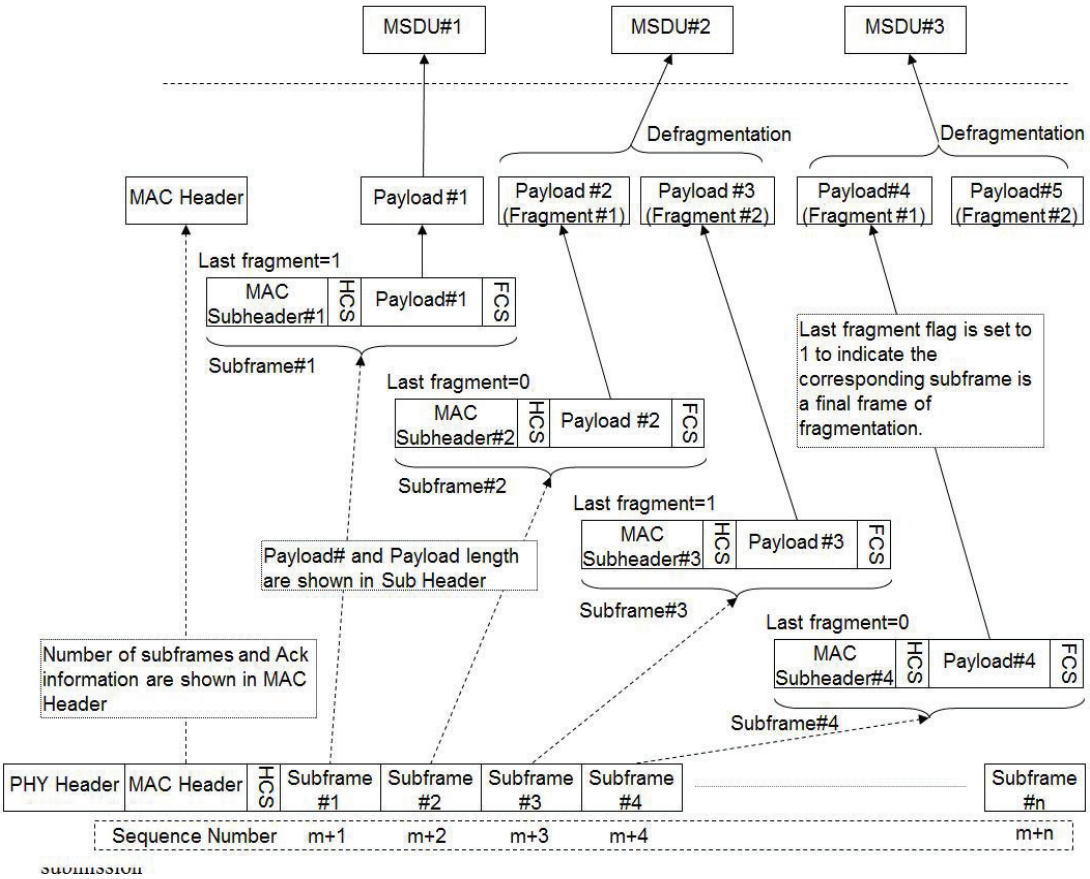


Figure 7-54b—Deaggregation at target DEV

7.9 Acknowledgment and retransmission

Change the dashed list in 7.9 as follows:

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)

After 7.9.2, insert the following new subclauses as 7.9.2a, 7.9.2a.1, and 7.9.2a.2:

7.9.2a Stk-ACK

Stk-ACK is used for acknowledgment for pairnet.

7.9.2a.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 7.4.1. It is also used for re-transmission control for the pairnet aggregation frame that is defined in 7.8. 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 7-54c 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 7-54c, 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 7-54c. 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.

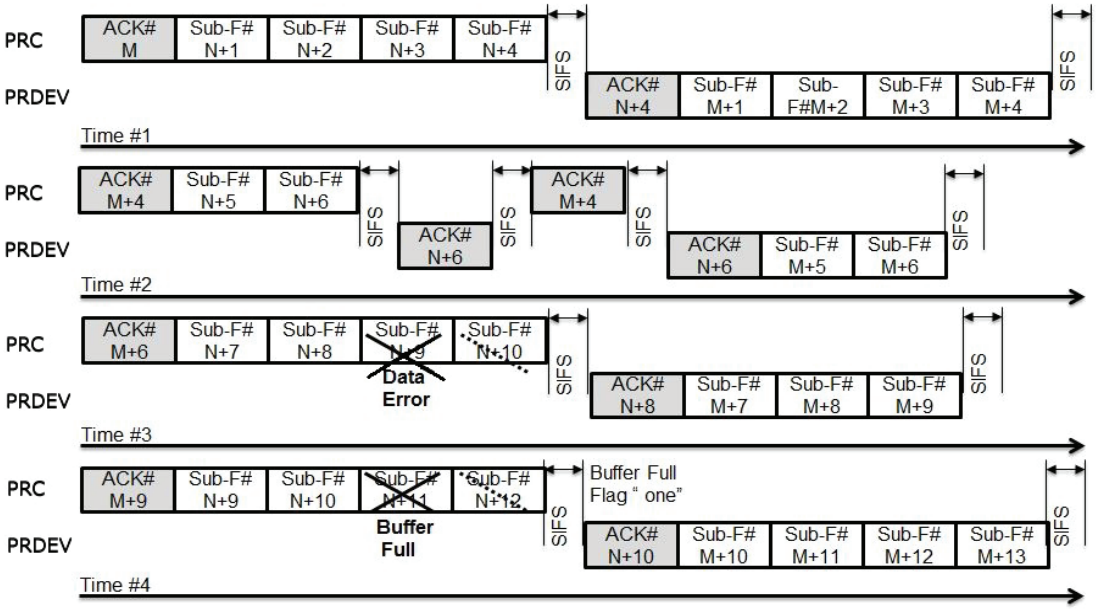


Figure 7-54c—Ping-pong channel access with Stk-ACK (synchronized state)

7.9.2a.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 7-54d illustrates the recovery process. As described in 7.4.4, 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.

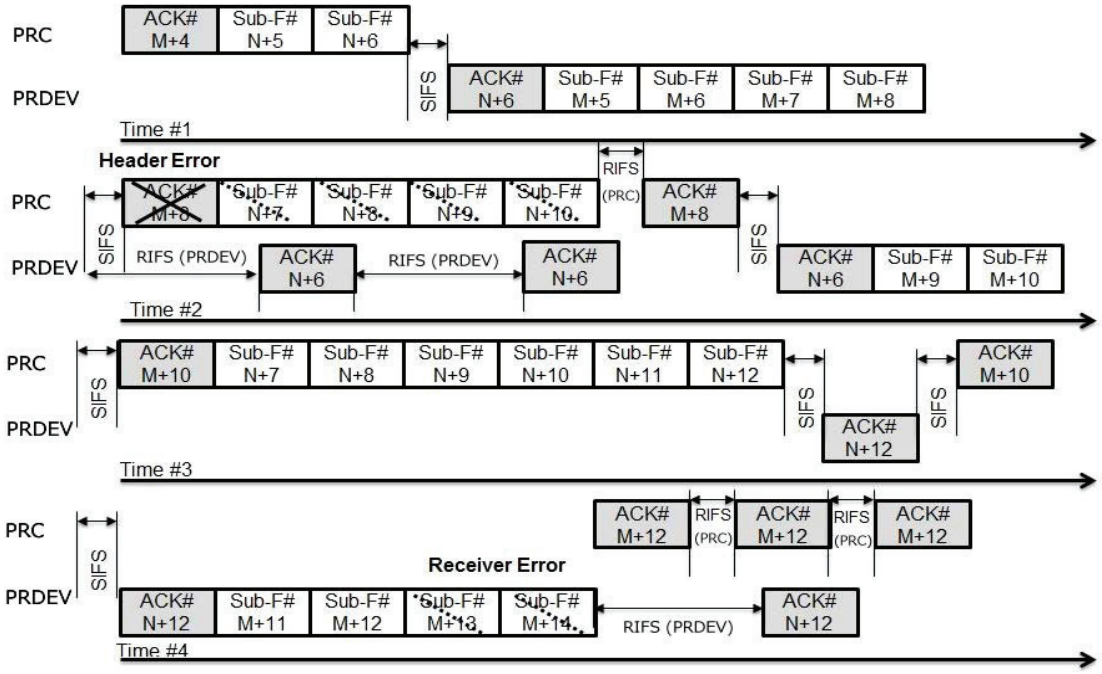


Figure 7-54d—Synchronization recovery process

7.13 Multi-rate support

Change the dashed list after the first paragraph as follows:

- In 10.3 for the 2.4 GHz PHY
- In 11.2.2.1 for the SC PHY mode
- In 11.3.2.1 for the HSI PHY mode
- In 11.4 for the AV PHY mode
- In 11a.2 for the HRCP-SC PHY mode
- In 11a.3 for the HRCP-OOK PHY mode

Change the second paragraph as follows:

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~~three~~ methods:

a) Check the capabilities of the target DEV broadcast by the PNC when the DEV becomes a member of the piconet.
b) Send a Probe Request command, as described in 6.5.4.5, to the target DEV to request its Capability IE, as described in 6.4.11.
c) Request the information from the PNC using the PNC Information Request command, as described in 6.5.4.1.
d) Check the capabilities of the target DEV exchanged during the association process of the pairnet as described in 5.3.5.

Change the title and first paragraph of 7.14 as follows:

7.14 Power management for piconets

There are four power management (PM) modes defined ~~in this standard~~ for piconets: ACTIVE, APS, PSPS, and DSPS modes. The latter three modes are collectively referred to as power save (PS) modes. A DEV that is in ACTIVE, APS, PSPS, or DSPS mode is said to be an ACTIVE DEV, an APS DEV, a PSPS DEV, or a DSPS DEV, respectively. In any given PM mode, a DEV may be in one of two power states, either AWAKE or SLEEP states. AWAKE state is defined as the state of the DEV where it is either transmitting or receiving. SLEEP state is defined as the state in which the DEV is neither transmitting nor receiving. A DEV, regardless of its PM mode, is allowed to enter the SLEEP state during a CTA for which it is neither the source nor the destination. A DEV is also allowed to enter the AWAKE state during any time when it is in a power save mode.

After 7.14, insert the following new subclause as 7.14a:

7.14a 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 7-72a 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 7.9.2a.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 7.9.2a.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.

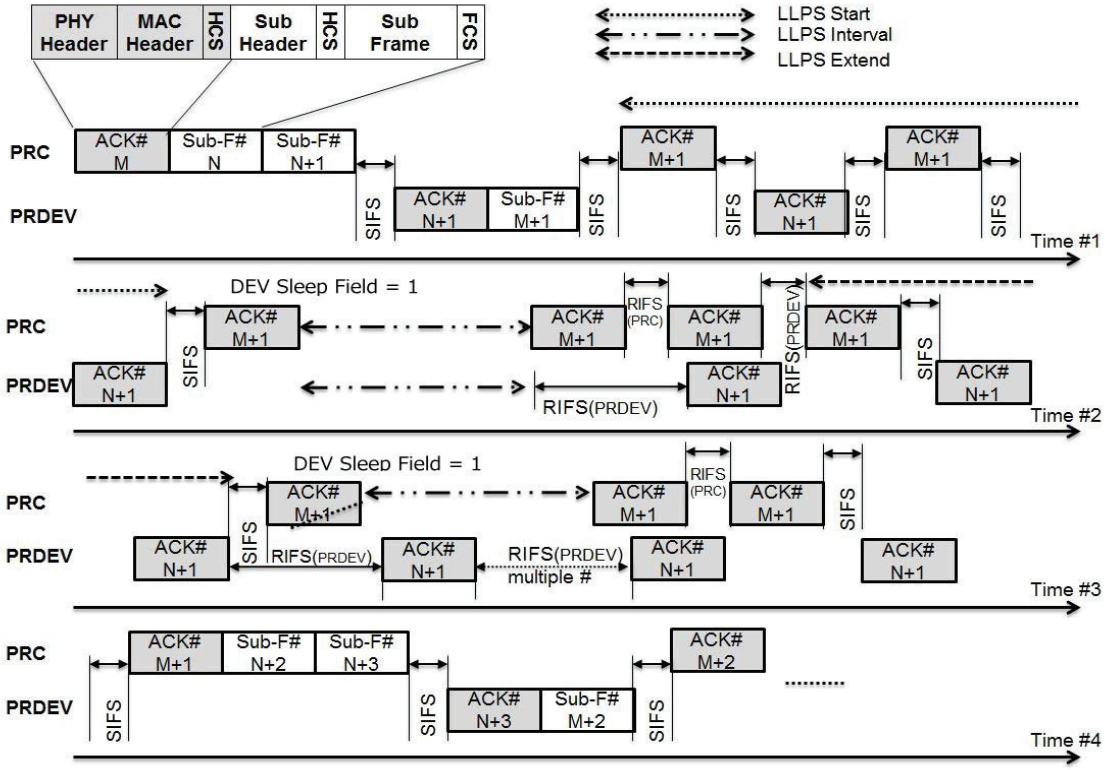


Figure 7-72a—LLPS Time Domain Behavior

7.16 MAC sublayer parameters

After Table 7-10, insert the following new text and tables as Table 7-10a, Table 7-10b, and Table 7-10c:

The parameters that define some of the pairnet MAC characteristics are given in Table 7-10a.

Table 7-10a—MAC sublayer parameters for HRCP SC PHY and HRCP 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 7-10b for the HRCP SC PHY.

Additional characteristics that are PHY dependent are listed in Table 7-10c for the HRCP OOK PHY.

Table 7-10b—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 11a.2.7.1
pMaxBeaconLengthTime	8 μs
pMaxAssocReqCommandTime	8 μs

Table 7-10c—MAC sublayer parameters—HRCP OOK PHY dependent

Parameter	Values or Subclause
pNAccessSlot	mod((Beacon Interval – pMaxBeaconLengthTime-SIFS)/pDAccessSlot)
pDAccessSlot	pMaxAssocReqCommandTime+SIFS
pMaxFrameBodySize	Defined in 11a.3.7.1
pMaxBeaconLengthTime	Unspecified (depends on optional IEs)
pMaxAssocReqCommandTime	Unspecified (depends on optional IEs)

8. Security

8.1 Security mechanisms

Change the first paragraph in 8.1 as follows:

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.

Change the first paragraph in 8.1.2 as follows:

8.1.2 Key transport

All keys that are transmitted from one DEV to another shall be encrypted as specified in the key request, as described in 8.4.3, and distribute key protocols, as described in 8.4.2. 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 integrity

Change the first paragraph in 8.1.4 as follows:

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.

8.1.5 Beacon integrity protection

Change the first paragraph in 8.1.5 as follows:

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 or pairnets, 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.

8.1.6 Command integrity protection

Change the first paragraph in 8.1.6 as follows:

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.

8.1.7 Freshness protection

Change the first paragraph in 8.1.7 as follows:

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 ~~and~~. 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 has not been compromised.

8.2 Security modes

Change the first paragraph in 8.2 as follows:

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 1

Change the first and second paragraphs in 8.2.2 as follows:

Security mode 1 provides a mechanism for a DEV 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 6-22 for piconet and Table 6-22a for pairnet, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

8.3 Security support

Change the first paragraph in 8.3 as follows:

The security policies determine the actions taken to preserve the security of the piconet or pairnet. Subclauses 8.3.1 through 8.3.9 specify the methods that are provided in this standard to support specific security policies.

8.3.1 PNC handover

Insert the following new paragraph at the end of 8.3.1:

This operation is not applicable for pairnets.

Change the title and first and second paragraphs of 8.3.2 as follows:

8.3.2 Changes in the piconet group data key or pairnet 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 6.5.2.3.

For piconet, o~~O~~nce 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 6.5.2.3, from the PNC, the DEV shall use the new key for all outgoing secure frames that require the use of the piconet group data key once it sees the corresponding SECID in the beacon. The DEV may continue to accept frames protected by the old piconet

group data key for up to *mMaxKeyChangeDuration* since the DEV last received a valid beacon protected by the old piconet-wide group data key.

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 PRC or PRC to obtain the new key.

Change the title and first paragraph of 8.3.3 as follows:

8.3.3 Joining a secure piconet or secure pairnet

Change the first paragraph in 8.3.3 as follows:

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.

8.3.4 Membership update

Change the second through fifth paragraphs in 8.3.4 as follows:

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 pairnet. 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 or PRCID 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 8.3.9. 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 8.3.9.

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 8.3.9, 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.

8.3.5 Secure frame generation

Change the first and second paragraphs in 8.3.5 as follows:

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 8-1a provides the listing for PRDEVs. A DEV shall not send a secure frame if the only key selection in Table 8-1 or Table 8-1a 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.

Change the fourth paragraph in 8.3.5 (splitting into three separate paragraphs) as follows:

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 9a-1, used to protect the secure frame. The SECID included in the frame shall be the value corresponding to the keying material being used.

~~T~~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 9a.4.2, to the GCM input specified in 9a.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.

Change the last paragraph of 8.3.5 as follows:

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.

8.3.6 Updating CurrentTimeToken

Change the first paragraph in 8.3.6 as follows:

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.

8.3.7 Secure frame reception

Change the first through fifth paragraphs in 8.3.7 as follows:

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 8-1a provides the listing for pairnets. A DEV may ignore any secure frame if the only key selection in Table 8-1 or Table 8-1a is "none." A DEV shall ignore any non-secure frame or a secure frame with an incorrect SECID when security is required.

An associated DEV 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 6.3.1.2 for piconet and 6.3.1.2a 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 8.3.5. 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 new pairnet group data key, as described in 8.3.2. If ~~both of~~ 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. ~~A~~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 9a.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 6-22 for piconet and Table 6-22a for pairnet, the MLME shall generate an MLME-SECURITY-ERROR.indication with the ReasonCode set to INVALID-SEC-VALUE.

8.3.8 Selecting the SECID for a new key

Change the first and second paragraphs in 8.3.8 as follows:

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 6.2.7.2, that identifies the key.

When a PNC-capable DEV starts a secure piconet, as described in 7.2.2, 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 a 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 8.3.3.

8.3.9 Key selection

Change the first paragraph in 8.3.9 as follows:

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 8-1a for pairnets. All secure commands between the PNC or PRC and other DEVs shall be protected with the PNC management key or PRC management key. All secure data frames with the PNC or PRC 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 8-1a summarizes the keys that shall be used for each type of frame.

Insert the following table in 8.3.9 after Table 8-1 as Table 8-1a:

Table 8-1a—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		

Table 8-1a—Key selection for secure pairnet frames (*continued*)

Frame type or command	None	PRC-DEV mgmt key	Pairnet group data key	Comment
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.
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

Change the first paragraph in 8.4 as follows:

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 frames transmitted in this subclause 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.

8.4.1 Security information request and distribution

Change the third and fourth paragraphs in 8.4.1 as follows:

Figure 8-2 illustrates the message flows for a~~n~~ 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 a~~n~~ Security Information Request from the new PNC to the old PNC. This operation is not applicable for pairnets.

8.4.2 Key distribution protocol

Change the first paragraph of 8.4.2 as follows:

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 the 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, 10.4.2, 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.

8.4.3 Key request protocol

Change the first paragraph of 8.4.3 as follows:

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

Insert the following paragraph at the beginning of Clause 9:

The security operation specified in this clause does not apply to pairnets, and the security operation specified in 9a shall be used for pairnets.

After Clause 9, insert the following new clause as Clause 9a:

9a. Security specifications for pairnets

This clause specifies the security operations that shall be used when security is implemented in the PRDEV.

9a.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.

9a.2 Symmetric cryptography building blocks

The following cryptographic primitives and data elements are defined for use in this standard.

9a.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 most significant bit first and least significant bit last. This ordering shall be irrespective of the transmission order of the bits. See Figure 6-1 for the mapping of bit transmission order to most significant or least significant.

9a.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 Secure Frame Counter (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 Advanced Encryption Standard (AES) algorithm. GCM is defined in NIST Special Publication 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 9a.4. The parameters for these operations shall be as specified in 9a.2.3.

9a.2.3 GCM parameters

The GCM operations shall be parameterized by the following selections: the AES encryption algorithm as specified in 9a.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 9a.2.4.

9a.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 Secure Frame Counter (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 Secure Frame Counter (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 9a-1.

Octets:6	6
DEV address (source DEV)	Secure Frame Counter (SFC)

Figure 9a-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 6.2.7.3. 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.

9a.2.5 AES encryption

The advanced encryption standard (AES) encryption algorithm used for symmetric key security operations shall be performed as specified in NIST FIPS Pub 197. This encryption algorithm is parameterized by the use of 128-bit keys and 128-bit block size. Only AES-128 GCM shall be used for pairnets.

9a.3 Symmetric cryptography implementation

9a.3.1 Symmetric cryptography data formats

Table 9a-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 9a.3.2.

Table 9a-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 9a.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 9a.2.2, that is computed along with the encrypted seed.

Table 9a-1—Symmetric cryptography frame object formats for GCM

Notation	Length	Value	Description
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 9a.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 9a.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.			

9a.3.2 Symmetric cryptographic operations

Figure 9a-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**	**2**	**2**
Frame Header	SECID	SFC	Time Token	Pairnet Synchronization Parameters	IE-1	…	IE-*n*	Auth Data Length	Enc Data Length

Figure 9a-2—GCM input for secure Beacon frames

Figure 9a-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**	**2**	**2**
Frame Header	SECID	SFC	MAC Subheader	Command Type	Length	Auth Data	Enc Data	Auth Data Length	Enc Data Length

Figure 9a-3—GCM input for secure commands

Figure 9a-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.

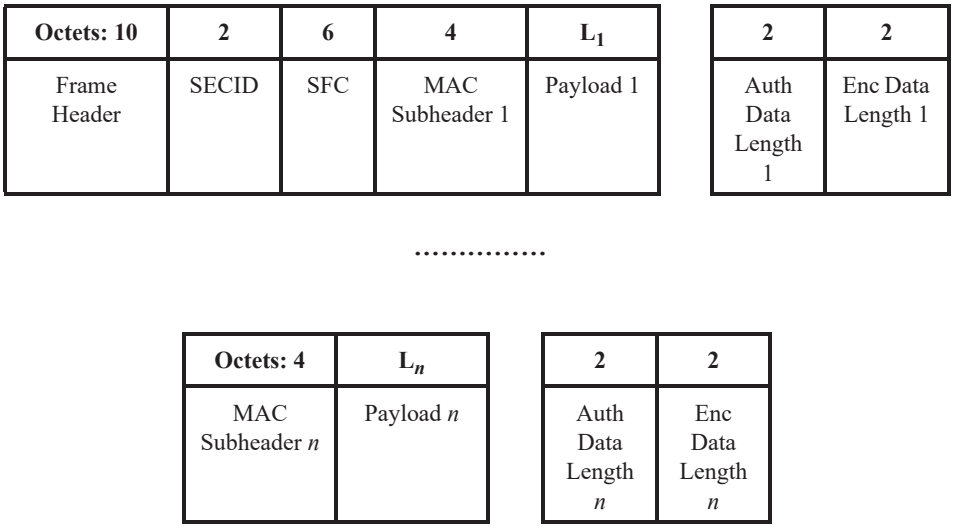
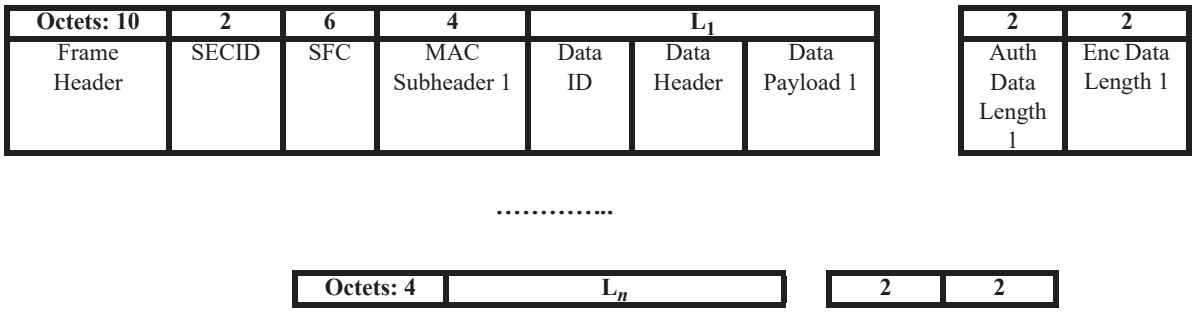


Figure 9a-4—GCM input for secure data frames

Figure 9a-5 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.



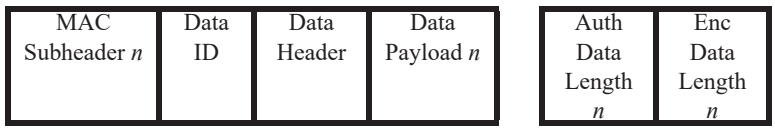


Figure 9a-5—GCM input for Secure Multi-Protocol Data frames

9a.4 GCM mode

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 9a.2.3.

9a.4.1 Inputs for authenticated encryption

To send a message, the sender shall provide the following information (see Table 9a-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 9a.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 9a.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 9a.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 9a-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	Additional authenticated data (AAD)	$l(a)$ octets	String of octets

9a.4.2 Authenticated encryption

The inputs for encryption that are defined in 9a.3.2 shall be divided into 16-octet message blocks as follows in Figure 9a-6. The blocks are ordered in the same order it is 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 the following equation:

$l(p)*8 = (n - 1)*128 + u$, where $1 \leq u \leq 128$

Octets: 16	…	16	1 –16
P_1	….	P_{n-1}	P^*_n

Figure 9a-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, … , 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 9a.3.2 shall be divided into 16-octet message blocks as shown in Figure 9a-7. The blocks are ordered in the same order it is 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 the following equation:

$l(a)*8 = (m - 1)*128 + v$, where $1 \leq v \leq 128$

Octets: 16	…	16	1 – 16
A_1	….	A_{m-1}	A^*_m

Figure 9a-7—Block ordering for AAD

The procedure defined in NIST Special Publication 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 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 $X, Y \in \mathrm{GF}\left(2^{128}\right)$ is denoted as $X \cdot Y$. GCM shall use the following polynomial:

$$f = 1 + \alpha + \alpha^2 + \alpha^7 + \alpha^{128}$$

— 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 following polynomial:

$$x_0 + x_1\,\alpha + x_2\,\alpha^2 + ... + x_{127}\,\alpha^{127}$$

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 9a.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 $\mathrm{incr}(F||I)$ 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 follows:

$$H = E(K, 0^{128})$$

$$Y_0 = IV \,||\, 0^{31}1$$

$$Y_i = \mathrm{incr}(Y_{i-1}) \text{ for } i = 1, \ldots, n$$

$$C_i = P_i \oplus E(K, Y_i) \text{ for } i = 1, \ldots, n$$

$$C_n^* = P_n^* \oplus \mathrm{MSB}_u(E(K, Y_n))$$

$$T = \mathrm{MSB}_t[\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0)]$$

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 follows:

$$X_i = 0 \qquad \text{for } i = 0$$

$$X_i = (X_{i-1} \oplus A_i) \cdot H \qquad \text{for } i = 1, \ldots, m - 1$$

$$X_i = \left(X_{m-1} \oplus \left(A_m^* \,||\, 0^{128-v}\right)\right) \cdot H \qquad \text{for } i = m$$

$$X_i = (X_{i-1} \oplus C_{i-m}) \cdot H \qquad \text{for } i = m+1, \ldots, m+n-1$$

$$X_i = \left(X_{m+n-1} \oplus \left(C_n^* \,||\, 0^{128-u}\right)\right) \cdot H \qquad \text{for } i = m+n$$

$X_i = [X_{m+n} \oplus (\text{len}(A) \| \text{len}(C))] \cdot H$ for $i = m+n+1$

9a.4.3 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.

9a.4.4 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.

9a.4.5 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 Special Publication 800-38D shall be used for authenticated decryption. The procedure described below is provided as an informative overview of the authenticated decryption procedure.

The authenticated decryption shall be processed as follows:

$H = E(K, 0^{128})$

$Y_0 = IV \| 0^{31}1$

$\text{MSB}_t[\text{GHASH}(H, A, C) \oplus E(K,$

$Y_i = \text{incr}(Y_{i-1})$ for $i = 1, \ldots, n$

$= C_i \oplus E(K, Y$ for $i = 1, \ldots, n$

$= C^*_n \oplus \text{MSB}_u[E(K, Y$

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.

9a.4.6 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.

9a.4.7 List of symbols

Table 9a-3 provides a list of the symbols used for the above specification of GCM.

Table 9a-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	—

After Clause 11, insert the following new clause as Clause 11a:

11a. PHY specification for HRCP

11a.1 General requirements

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 11a.2.
b) On-off keying mode PHY (OOK PHY), as defined in 11a.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.

11a.1.1 Regulatory Information

The HRCP PHY operating frequency is the same as that described in 11.1.1.

11a.1.2 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.

11a.1.3 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.

11a.1.4 RF channelization

The HRCP PHY uses the channels defined in Table 11a-1. CHNL_ID from 1 to 4 are the same as those in Table 11-3.

Table 11a-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
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

Table 11a-1—HRCP PHY channelization (*continued*)

CHNL_ID	Start frequency[b]	Center frequency	Stop frequency[a]
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 power spectral density (PSD) mask for the PHY mode as well as any regulatory requirement.
[b]The start and stop frequencies are nominal values. The frequency spectrum of the signal needs to conform to the transmit power spectral density (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 11a.2.8.2.

11a.1.5 Transmit PSD mask

The transmitted spectrum for HRCP PHY shall adhere to the transmit spectrum masks for a single channel shown in Figure 11a-1 and for channel bonding shown in Figure 11a-2.

The transmit spectral mask for HRCP PHY that supports channel bonding shall conform to the values indicated in Table 11a-2 and Table 11a-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 11a-1 and Figure 11a-2 within the frequency band of [–6 MHz,+6 MHz] from the carrier frequency.

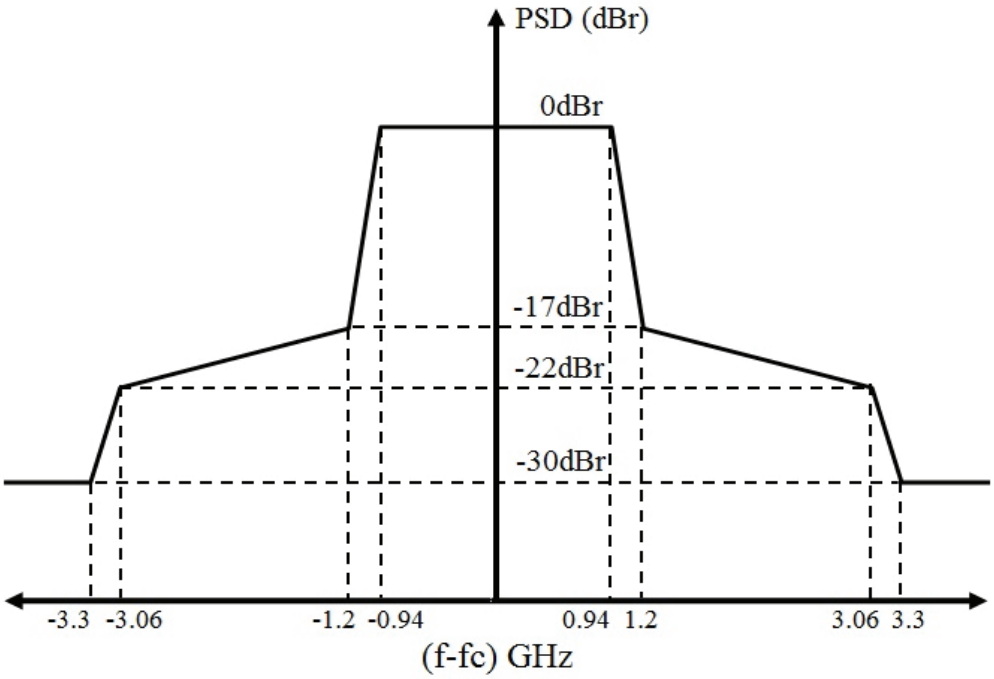


Figure 11a-1—Transmit spectral mask for a single channel

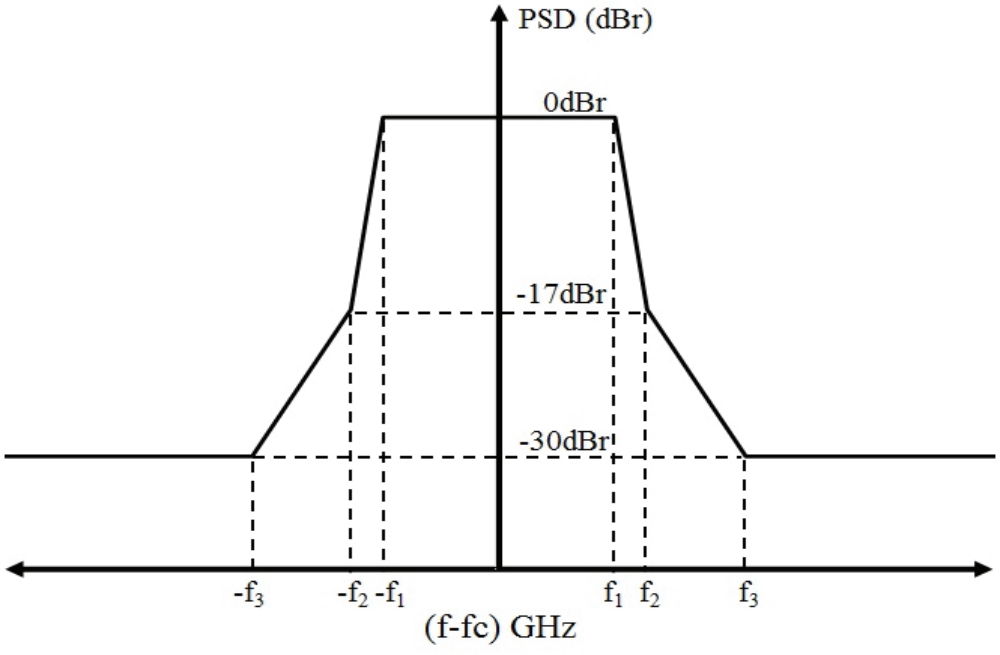


Figure 11a-2—Transmit spectral mask for channel bonding

Table 11a-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 11a-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

11a.1.6 Error vector magnitude calculation

The error vector magnitude (EVM) shall be measured and calculated using the method defined in 11.1.7.1.

11a.1.7 HRCP-PHY management

11a.1.7.1 Supported MCSs

The Supported data rates field in the DEV capabilities field is defined as described in 6.4.11a.

11a.1.7.2 HRCP-PHY PIB

The PHY dependent PIB values for the HRCP PHY are given in Table 11a-4 and Table 11a-5. The PHY PIB characteristics group, Table 11a-4, contains information that is common to most implementations. The PHY PIB implementation group, Table 11a-5, contains information that is more characteristic of a particular PHY implementation than of the PHY as a whole.

11a.2 HRCP-SC PHY

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.

Table 11a-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 11a.1.4. 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 11a.1.4.	Read/ Write
phyFrameLengthMax	2	pMaxFrameBodySize.	Read/ Write

Table 11a-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

Table 11a-5—PHY PIB implementation group parameters (*continued*)

Managed Object	Octets	Definition	Access
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

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.

11a.2.1 Channelization of HRCP-SC PHY

The RF channels are defined in Table 11a-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 6.4.11a. Chip rates used for bonded channels are described in 11a.2.2.3 and data assignments for aggregated channels are described in 11a.2.8.2.

The phyCurrentChannel is the CHNL_ID of the current channel.

11a.2.2 Modulation and coding

11a.2.2.1 MCS dependent parameters

The MCS dependent parameters shall be set according to Table 11a-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 11a-8.

Table 11a-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	014/15	3.2853	2.8747
2	16-QAM	11/15	5.1627	4.5173

Table 11a-6—MCS dependent parameters (*continued*)

MCS identifier	Modulation	FEC rate	Data rate (Gb/s) w/o PW	Data rate (Gb/s) w/ PW
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 11a.2.3.4.1, for HRCP-SC PHY shall be 64 chips. The pilot word (PW) length, where a PW is defined in 11a.2.3.4.1, 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.

11a.2.2.2 Header rate-dependent parameters

The header rate-dependent parameters shall be set according to Table 11a-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 11a.2.3.2.3.

Table 11a-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

11a.2.2.3 Timing-related parameters

Table 11a-8 lists the general timing parameters associated with the SC PHY.

Table 11a-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 × 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	

Table 11a-8—Timing-related parameters (*continued*)

Parameter	Description	Value	Unit	Formula
T_{block}	Block duration	~36.364 / N_B	ns	$L_{block} \times T_c$
R_{block}	Block rate	27.5 × N_B	MHz	1/ T_{block}

11a.2.2.4 Frame-related parameters

The frame parameters associated with the PHY are listed in Table 11a-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 11a-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} × 8 × L_{SF} / (L_{block} – L_{PW})] = 19
T_{hdr}	Duration of the frame header	N_{block_hdr} × T_{block} = ~ 0.691/N_B μs
$L_{payload}$	Length of frame payload in octets	Variable
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})·$N_{subframe}$
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, 64QAMand 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} × N_{block_PPRE} + T_{PW}

Table 11a-9—Frame-related parameters (*continued*)

Parameter	Description	Value
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[$(L_{MFB} \times 8) / (R_{FEC} \times N_{CBPB})$] ($R_{FEC}$: 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	$T_{pre} + T_{hdr} + T_{datafield}$

11a.2.2.5 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 11-10 (a) and (b) with a counter-clock wise phase offset of π/4, respectively, in 11.2.2.5. The constellations of 16-QAM and 64-QAM used for the HRCP-SC PHY are the same as illustrated in Figure 11-29 in 11.3.2.6 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 11a.2.4.1.

The constellation map of 256-QAM used for the HRCP-SC PHY is illustrated in Figure 11a-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.

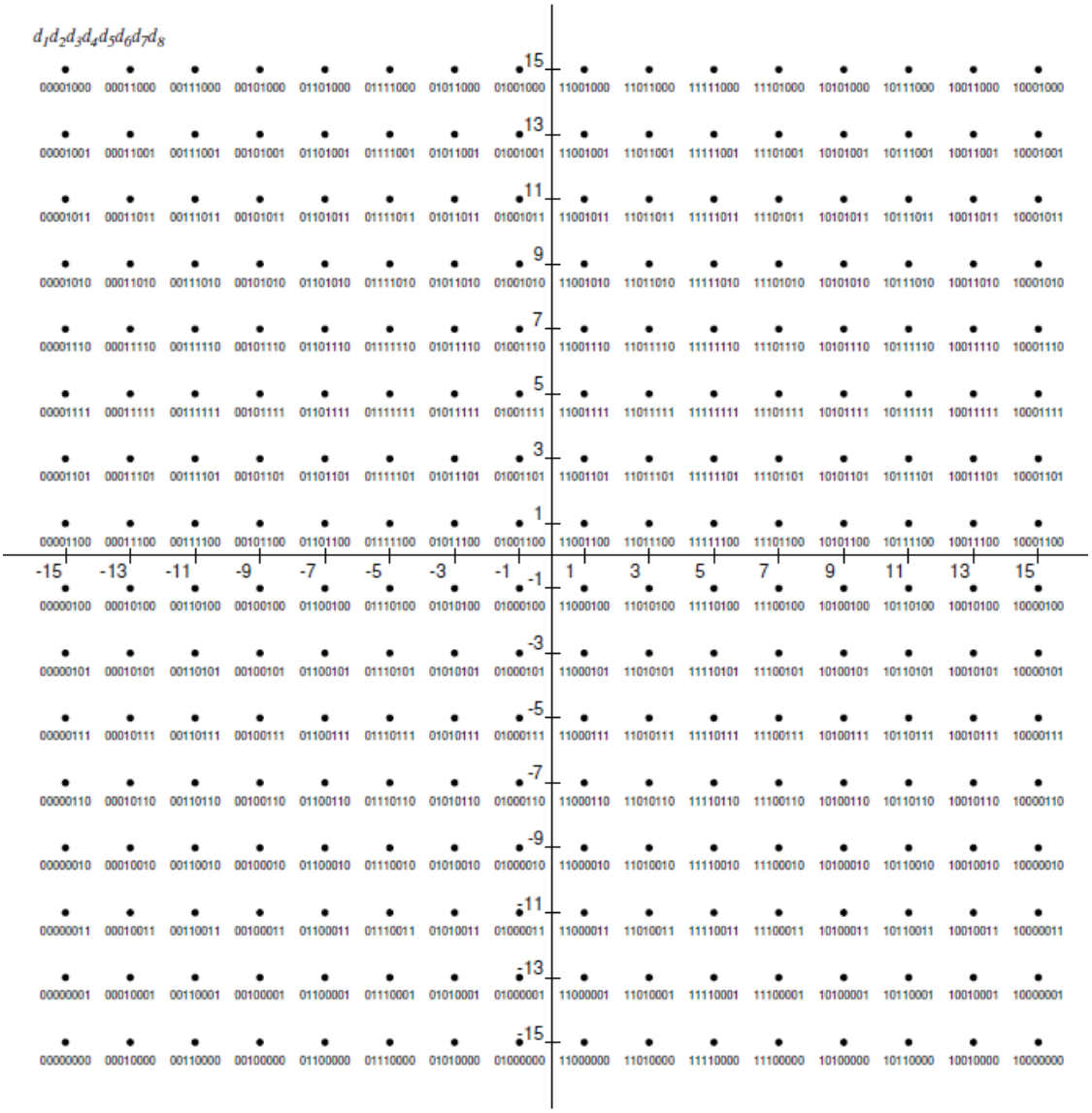


Figure 11a-3—Constellation map of 256 QAM

11a.2.2.6 Forward Error Correction

The forward error correction (FEC) schemes are specified in this subclause. Supporting the following two rate-compatible low-density parity-check (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 11.2.2.6.3.

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}, $0 \leq i < (1440 - k)$ and $0 \leq j < 1440$.

Table 11a-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 11-17.

Table 11a-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 following equation:

$$h_{i,j} = h_{96*\text{floor}(i/96) + \text{mod}(i + \text{floor}(j/15), 96), \text{mod}(j, 15)},$$

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.

11a.2.2.7 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 the following equation:

$$N_{FEC} = \text{CEIL}[(L_{MFB} \times 8)/(1440 \times R_{FEC})]$$

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 in11a.2.2.9. The length of bits in the encoded MAC frame body, L_{ebits} is given by the following equation:

$$L_{ebits} = 8 \times L_{MFB} + N_{FEC} \times (1 – R_{FEC}) \times 1440$$

The number of blocks in the encoded MAC frame body, $N_{subblock\text{-}encMFB}$, and the length of stuff bits, L_{STUFF}, are given by following equations:

$$N_{subblock–encMFB} = \text{CEIL}(L_{ebits}/N_{CBPB})$$

$$L_{STUFF} = N_{subblock–encMFB} \times N_{CBPB} – L_{ebits}$$

where

N_{CBPB} is the number of coded bits per subblock as given in Table 11a-11 for each MCS.

Table 11a-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

Table 11a-11—Rate dependent bits per block (*continued*)

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 11a-7. The bit pattern of the header stuff bits shall be set to 01, where 0 is transmitted first.

11a.2.2.8 Code spreading

Table 11a-12 is a spreading table for a frame header. The most significant bit of the output shall be transmitted first in Table 11a-12.

Table 11a-12—Spreading table

Input	Output
0	1010
1	0101

11a.2.2.9 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as illustrated in Figure 11-14 with $L_{SF} = 1$.

11a.2.3 HRCP-SC PHY frame format

The HRCP-SC PHY frame shall be formatted as illustrated in Figure 11-18.

The Frame Header field for the PHY frame shall be formatted as illustrated in Figure 11a-4.

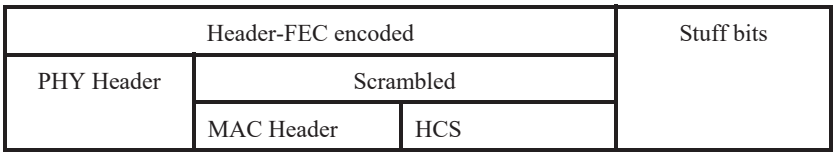


Figure 11a-4—Frame header format

The PHY preamble is described in 11a.2.3.1. The MAC header is defined in 6.2. The PHY header is defined in 11a.2.3.2.1, and the HCS is defined in 11a.2.3.2.2. The header FEC is defined in 11a.2.3.2.3. The PHY Payload field consisting of the MAC frame body, the pilot preamble (PPRE) and stuff bits, is described in 11a.2.3.3. The PPRE is described in 11a.2.3.4.2. The stuff bits are described in 11a.2.2.7.

11a.2.3.1 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 1760 MHz.

A PHY-long preamble shall be used during PSP and a PHY-short preamble shall be used during PAP.

Figure 11a-5 shows the structure of the PHY-long or PHY-short preambles.

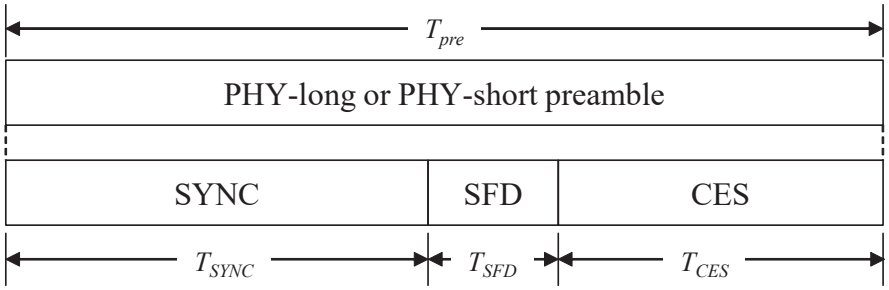


Figure 11a-5—HRCP-SC PHY preamble structure

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.

11a.2.3.1.1 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 11a-13 shows the sequence $\mathbf{a}_{128}$ used for the SYNC and a sequence $\mathbf{b}_{128}$ used for the CES defined in 11a.2.3.1.3. 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 11a-13—Golay sequences with length 128

Sequence name	Sequence value
$\mathbf{a}_{128}$	5A5599963C33FFF00F00CCC36966AAA5
$\mathbf{b}_{128}$	A5AA6669C3CC000F0F00CCC36966AAA5

11a.2.3.1.2 SFD

The SFD field is used to establish frame timing. The SFD field shall consist of the sign inversion sequence of $\mathbf{a}_{128}$.

11a.2.3.1.3 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 as:

$\mathbf{a}_{512} = [\mathbf{b}_{256}\ \mathbf{a}_{256}]$

$\mathbf{b}_{512} = [-\mathbf{b}_{256}\ \mathbf{a}_{256}]$

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 as:

$\mathbf{a}_{256} = [\mathbf{b}_{128}\ \mathbf{a}_{128}]$

$\mathbf{b}_{256} = [-\mathbf{b}_{128}\ \mathbf{a}_{128}]$

where the number on the right $\mathbf{a}_{128}$ are transmitted first.

11a.2.3.2 Frame header

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 11a-6.

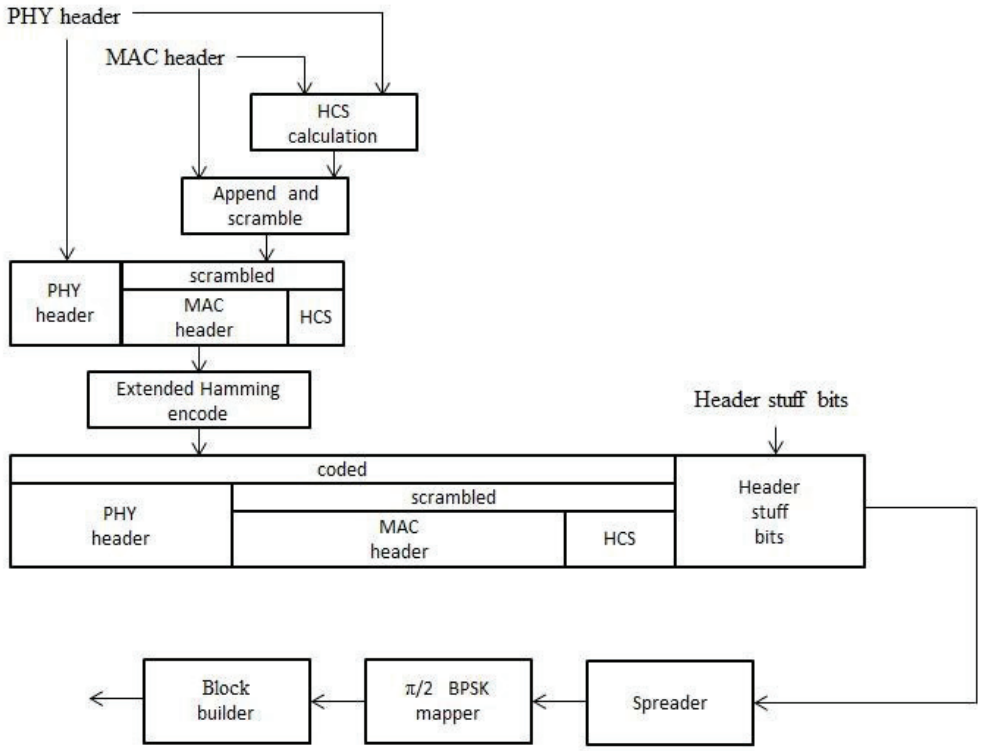


Figure 11a-6—Frame header construction process

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 11.2.2.10
   5) Encode the concatenation of the PHY header, scrambled MAC header and scrambled HCS into a concatenated extended Hamming codes, as described in 11a.2.3.2.3
   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 11a-6.

b) Spread the frame header, as described in 11a.2.2.8.

c) Map the frame header onto π/2 -shift BPSK, as described in 11.2.2.5.1.

d) Build blocks from the frame header, as described in 11a.2.3.4.1.

The LFSR for the spreader is reset between the header and payload.

11a.2.3.2.1 HRCP-SC PHY header

The HRCP-SC PHY header shall be formatted as illustrated in Figure 11a-7.

Bits: b0–b2	b3	b4–b7	b8–b9	b10-b31	b32–b35
MCS	Pilot word	Scrambler Seed ID	PPRE	Frame Length	Reserved

Figure 11a-7—PHY header format for HRCP-SC

The MCS field shall be set according to the values in Table 11a-14.

Table 11a-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 11.2.2.10.

The PPRE field shall be set according to the values in Table 11a-15.

Table 11a-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.

11a.2.3.2.2 Header HCS

The combination of the PHY header and MAC header shall be protected with a CRC-16 header check sequence (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 11a-8.

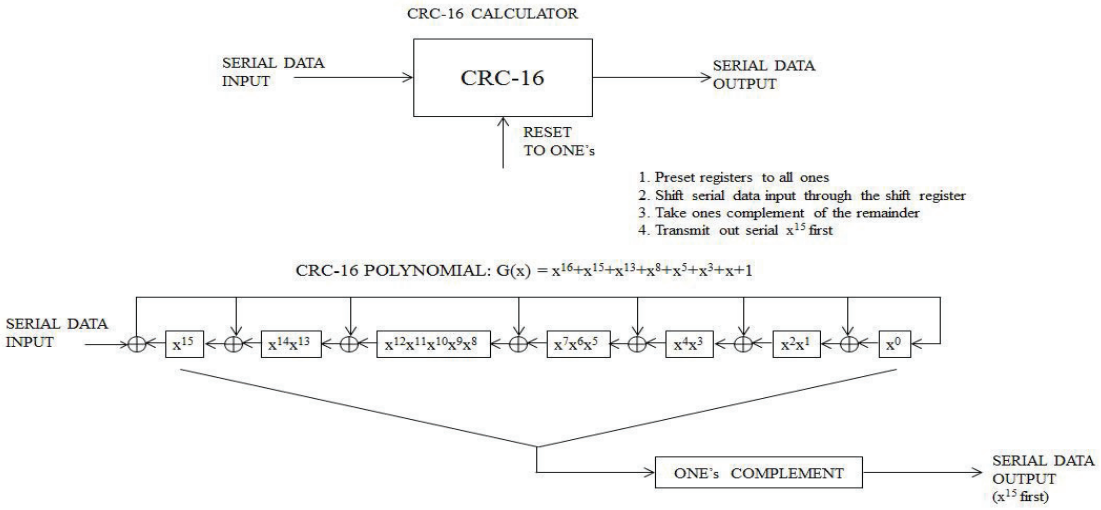


Figure 11a-8—CRC-16 Implementation

11a.2.3.2.3 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 11a-16.

Table 11a-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

Table 11a-16—Parity assignment of the Header FEC (*continued*)

i_0	i_1	i_2	i_3	p_0	p_1	p_2	p_3
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

11a.2.3.3 HRCP-SC PHY Payload field

The HRCP-SC PHY Payload field is the last component of the frame, and is constructed as shown in Figure 11-23. This payload is used in SISO operations.

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 11.2.2.10.
b) Encode the scrambled MAC frame body as specified in 11a.2.2.6.
c) Add stuff bits to the encoded and scrambled MAC frame body according to 11a.2.2.7.
d) Map the resulting MAC frame body onto the appropriate constellation as described in 11a.2.2.5.
e) Build blocks from the resulting MAC frame body according to 11a.2.3.4.1.
f) Insert PPRE periodically as described in 11a.2.3.4.2.

11a.2.3.3.1 HRCP-SC PHY Payload scrambling

The HRCP-SC PHY payload shall use the scrambling process defined in 11.2.2.10.

11a.2.3.3.2 Modulation

Modulation for the MAC frame body is defined in 11a.2.2.5.

11a.2.3.3.3 FEC

FEC for the MAC frame body is defined in 11a.2.2.6.

11a.2.3.4 Pilot word and PPRE

11a.2.3.4.1 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 11a-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.

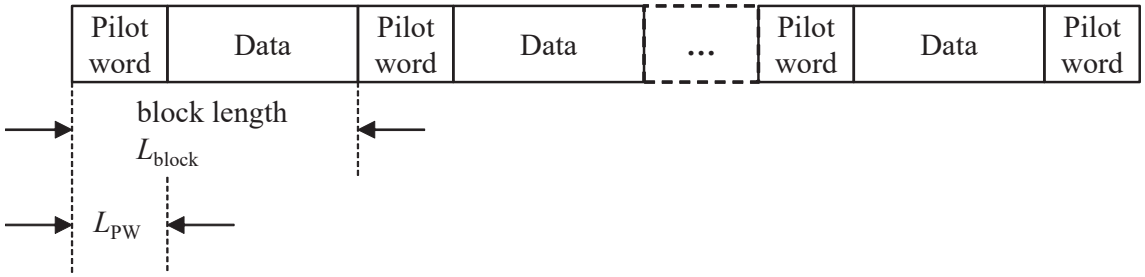


Figure 11a-9—Frame format with pilot words

11a.2.3.4.2 PPRE

PPRE insertion is an optional feature that allows a DEV to periodically re-adjust the receiver algorithms as described in 11a.2.3.1. PPRE is inserted into the scrambled, encoded, spread, and modulated MAC frame body as illustrated in Figure 11a-10 with an interval specified in Table 11a-9. The PPRE field shall be the concatenation of SYNC, SFD and CES field in the preamble defined in 11a.2.3.1. PPRE shall be modulated with π/2 BPSK.

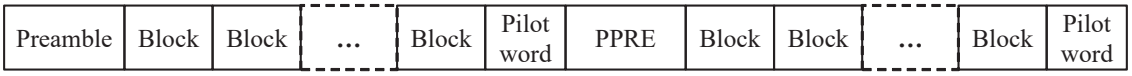


Figure 11a-10—Data Blocks with PPRE

11a.2.4 Transmitter specifications

11a.2.4.1 EVM requirement

The EVM of a compliant transmitter shall be measured and calculated as defined in 11.1.7 and shall not exceed the values given in Table 11a-17 for the indicated mode. Note that this requirement assumes a conducted measurement.

Table 11a-17—Maximum allowed EVM for HRCP-SC PHY with a single channel

MCS identifier	Maximum EVM (dB)
0	–12
1	–15
2	–18

Table 11a-17—Maximum allowed EVM for HRCP-SC PHY with a single channel

MCS identifier	Maximum EVM (dB)
3	–22
4	–25
5	–29
6	–36

11a.2.4.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ maximum.

11a.2.4.3 Symbol rate

The SC PHY shall be capable of transmitting at the chip rate, as defined in Table 11a-8, to within $\pm 30 \times 10^{-6}$

The MAC parameter, pPHYClockAccuracy, shall be $\pm 30 \times 10^{-6}$.

11a.2.4.4 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 11a.1.3.

11a.2.5 Receiver specifications

11a.2.5.1 Error rate criterion

The error rate criterion shall be a frame error rate (FER) of less than 8% with a frame payload length of 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.

11a.2.5.2 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 11a.2.5.1 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 11a-18.

Table 11a-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

11a.2.5.3 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 11a.2.5.1 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.

11a.2.6 PHY layer timing

The values for the PHY layer timing parameters are defined Table 11a-19.

Table 11a-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	11a.2.6.3
pPHYChannelSwitchTime	100 μs	11a.2.6.5

11a.2.6.1 Interframe space

A conforming implementation shall support the IFS parameters, as described in 7.4.1, given in Table 11a-20.

Table 11a-20—IFS parameters

MAC parameter		Corresponding PHY parameter	Definition
SIFS		*pPHYSIFSTime*	11a.2.6.3
RIFS	PRC	2**pPHYSIFSTime* + 3.01 μs	7.4.1
	DEV	4**pPHYSIFSTime* + 9.05 μs	

11a.2.6.2 Receive-to-transmit turnaround time

The receive to transmit turnaround time shall be *pPHYSIFSTime*, including the power-up ramp specified in 11a.2.4.4. 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.

11a.2.6.3 Transmit-to-receive turnaround-time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*, including the power-down ramp specified in 11a.2.4.4.

11a.2.6.4 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 11a-20. A PRC shall use the shorter RIFS than that of a DEV in Table 11a-20.

11a.2.6.5 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*.

11a.2.7 PHY management for HRCP-SC PHY

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the HRCP-SC PHY layer of a DEV.

11a.2.7.1 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.

11a.2.7.2 Maximum transfer unit size

The maximum size data frame passed from the upper layers, *pMaxTransferUnitSize*, shall be 2097152 octets. If security is enabled for the data connection, the upper layers should limit data frames to 2097152 octets minus the security overhead as defined in 7.3.4.2, 7.2.8.1.2, or 7.2.8.2.2.

11a.2.7.3 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed with the HRCP-SC PHY shall be 2048 octets as defined in 6.4.11a.

11a.2.8 MIMO, channel bonding, and channel aggregation

11a.2.8.1 Introduction to MIMO in SC-PHY

This subclause describes the MIMO transmission PHY specification for high transmission rates, above 100 Gbit/s. Single-input and single-output (SISO) is the mode that is described in 11a.2.1–11a.2.7, 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 11a-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.

11a.2.8.2 Channel aggregation and channel bonding

There are a total of 11 possible channel aggregation combinations, as defined in 6.4.11a. The bonded channels are defined in Table 11a-1 with CHNL_ID from 7 to 16.

11a.2.8.3 Link setup procedure for MIMO mode

In this subclause the link setup procedure for MIMO transmission mode is described. Figure 11a-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.

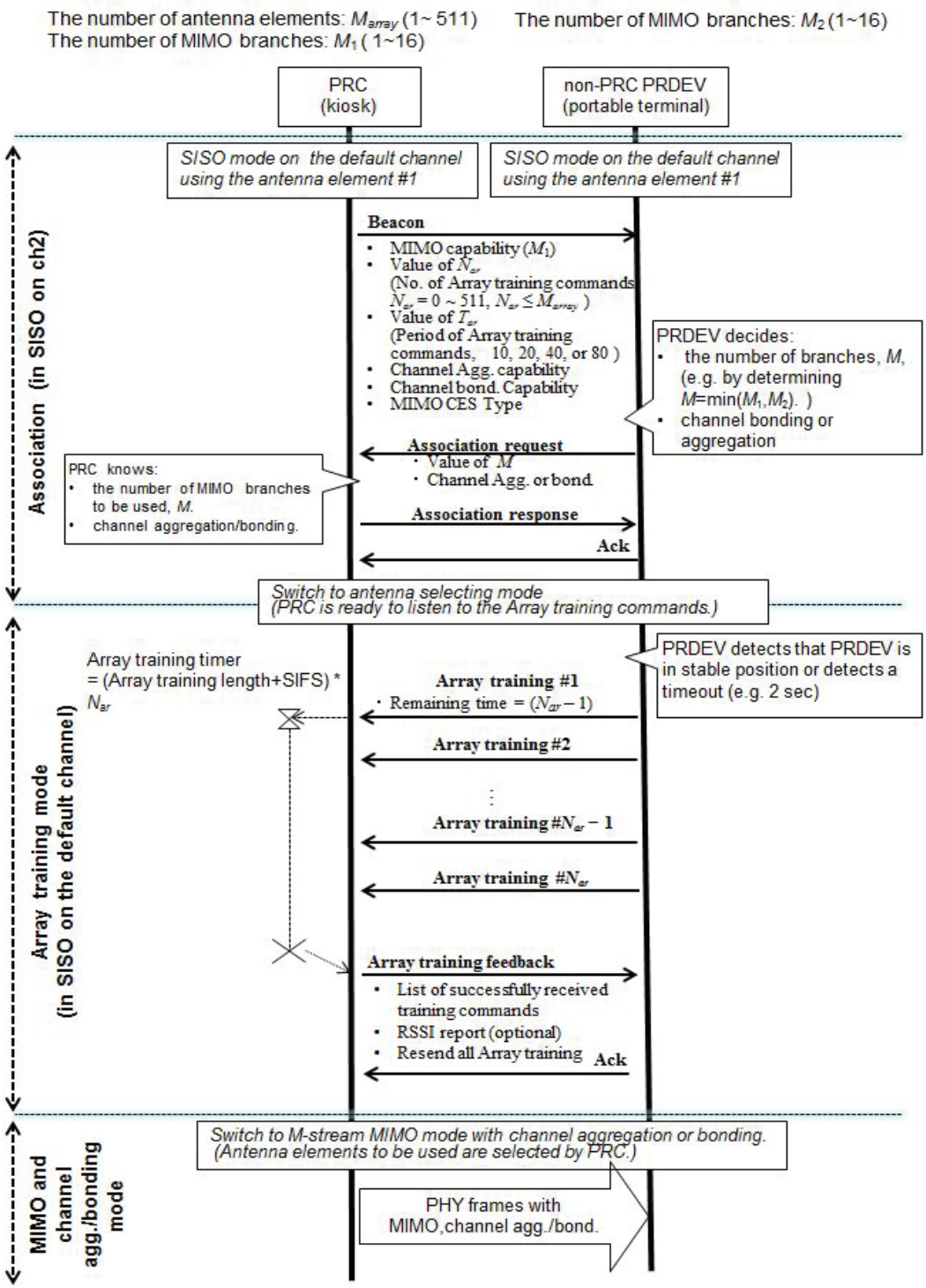


Figure 11a-11—Setup sequence for MIMO transmission

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 P2P 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 11a.2.8.4.

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.

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 information below:

— 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.

11a.2.8.4 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 11a-12. The procedure of selecting antenna elements is done in the Array Training mode shown in Figure 11a-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.

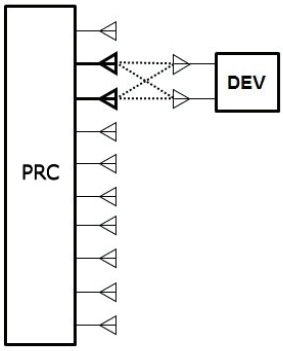


Figure 11a-12—Antenna element selection

11a.2.8.5 MIMO PHY Preamble

The Preamble is comprised of the SYNC, SFD and CES as shown in Figure 11a-13. Note that the preamble defined in 11a.2.3.1 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 11a.2.8.5.3 or one shown in 11a.2.8.5.4. The type of MIMO CES is advertised by the Beacon frame, as shown in 6.4.37.

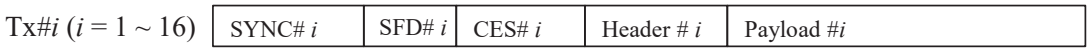


Figure 11a-13—PHY frame structure in MIMO mode

11a.2.8.5.1 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 beamforming effect, caused by the element spacing. The value of cyclic shift for i^{th} transmitter T_{CSsync_i} is

$T_{CSsync\_i} = 60 * (i - 1)$ [ns]

11a.2.8.5.2 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

$T_{CSsfd\_i} = 4 * (i - 1)$ [ns]

11a.2.8.5.3 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 11a.2.3.1.3.

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.

11a.2.8.5.4 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 $\mathbf{b}_{512}$ is defined in 11a.2.3.1.3. 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 11a.2.8.5.3.

11a.2.8.6 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*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.

11a.2.8.7 HRCP-SC-MIMO PHY Header

Information for MIMO and channel aggregation bitstream processing, described in 11a.2.8.6, shall be included in the header as shown in Figure 11a-14. The HRCP-SC-MIMO PHY header shall be processed in the same manner as described in 11a.2.3.2.

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 11a-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.

11a.2.8.8 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 shown below.

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 following equations.

$$N_{block\_encMFB} = \mathrm{CEIL}(L_{ebits} / \sum_{i=1}^{M} L_{CPBS}^{i})$$

$$L_{STUFF} = N_{block\_encMFB} \times \sum_{i=1}^{M} L_{CPBS}^{i} - L_{ebits}$$

where L_{CPBS}^{i} is the number of coded bits per sub-block as given in Table 11a-11 for each MCS and 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 11.2.2.10.

d) In each spatial stream, Encode the scrambled MAC frame body as specified in 11a.2.2.6.

e) In each spatial stream, map the resulting MAC frame body onto the appropriate constellation as described in 11a.2.2.5.

f) In each spatial stream, build blocks from the resulting MAC frame body according to 11a.2.3.4.1.

g) In each spatial stream, insert PPRE periodically as described in 11a.2.3.4.2.

11a.2.8.9 Scrambler

The MIMO PHY payload shall use the scrambling process described in 11.2.2.10.

11a.2.8.10 Transmitter specifications

The transmitter specifications are the same as the SISO mode as described in 11a.2.4.

11a.2.8.11 Receiver specifications

11a.2.8.11.1 Error rate criterion

The error rate criterion shall be a frame error rate (FER) of less than 8% with a frame payload length of 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.

11a.2.8.11.2 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 11a.2.8.11.1 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 11a-22.

Table 11a-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

Table 11a-22—Reference sensitivity levels for each mode (*continued*)

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

11a.3 HRCP-OOK 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. 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.

11a.3.1 Channel bonding

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 11a-25 due to the expanded bandwidth by bonding contiguous channels.

11a.3.1.1 Channelization for HRCP-OOK PHY

Figure 11a-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 11a-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.

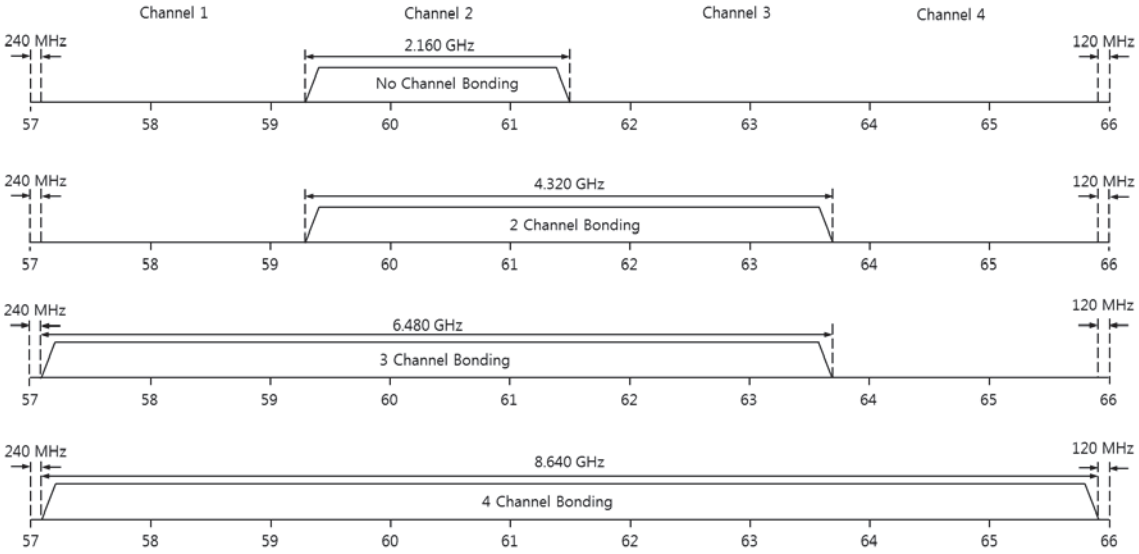


Figure 11a-15—Channels used by HRCP-OOK PHY

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 channel 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.

11a.3.1.2 Transmit PSD mask for HRCP-OOK

The transmit spectral mask specified in Figure 11a-1 shall be used for single channel transmission. The transmit spectral mask specified in Figure 11a-2, Table 11a-2, and Table 11a-3 shall be used for transmission using channel bonding.

11a.3.2 Modulation and coding

11a.3.2.1 MCS dependent parameters

The chip rate of HRCP-OOK PHY is given in Table 11a-25. The entire HRCP-OOK frame shall be modulated with OOK as specified in 11a.3.2.5.1. The FEC for HRCP-OOK PHY shall be RS coding as specified in 11a.3.2.6. The MCS dependent parameters shall be set according to Table 11a-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 11a-23.

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.

Table 11a-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 11a.3.3.3.5 are included in the calculation of data rates.

11a.3.2.2 Header rate-dependent parameters

The header rate-dependent parameters shall be set according to Table 11a-24.

Table 11a-24—Header rate-dependent parameters

Channel Bonding	Header rate [a] (Mb/s)	Modulation	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.

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.

11a.3.2.3 Timing-related parameters

Table 11a-25 lists the general timing parameters associated with the HRCP-OOK PHY.

Table 11a-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		

11a.3.2.4 Frame-related parameters

The frame parameters associated with the HRCP-OOK PHY are listed in Table 11a-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 11a-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

Table 11a-26—Frame-related parameters (*continued*)

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	$T_{pre} + T_{hdr} + T_{datafield}$

11a.3.2.5 Modulation

After channel encoding and spreading, the resulting bits shall be modulated using OOK as specified in 11a.3.2.5.1. The actual transmitted RF signal is described in 11a.3.2.5.2.

11a.3.2.5.1 OOK

HRCP-OOK frames shall be modulated using OOK. The OOK modulation shall use variable amplitudes to represent the data. As shown in Figure 11a-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 $\sqrt{2}$.

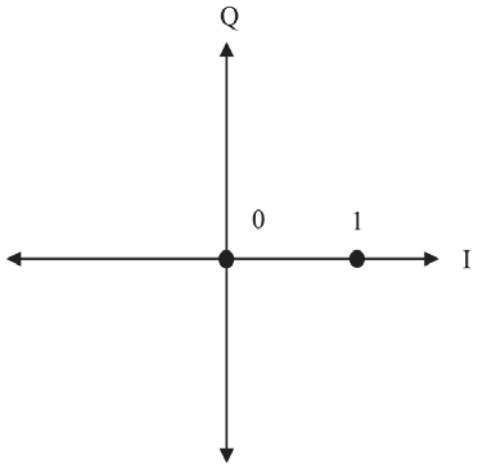


Figure 11a-16—Constellation diagram for OOK

11a.3.2.5.2 Description of signals

The actual transmitted RF signal can be written as follows:

$$S_{RF}(t) = \sum_{k=0}^{N_{chip}-1} a_k s_B (t - kT_c) \cos(2\pi f_c t)$$

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

11a.3.2.6 Forward Error Correction

Only RS block codes as specified in 11a.3.2.6.1 shall be used for HRCP-OOK PHY.

11a.3.2.6.1 Reed-Solomon block codes in GF(2^8)

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 following generator polynomial:

$$g(x) = \prod_{k=1}^{16} (x + a^k)$$

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 $\boldsymbol{m} = (m_{223}, m_{222}, \ldots, m_0)$ to codeword octets $\boldsymbol{c} = (m_{223}, m_{222}, \ldots, m_0, r_{15}, r_{14}, \ldots, r_0)$ is achieved by computing the remainder polynomial $r(x)$:

$$r(x) = \sum_{k=0}^{15} r_k x^k = x^{16} m(x) \bmod g(x)$$

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, $k = L_{inf}, \ldots, 223$

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 11a-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).

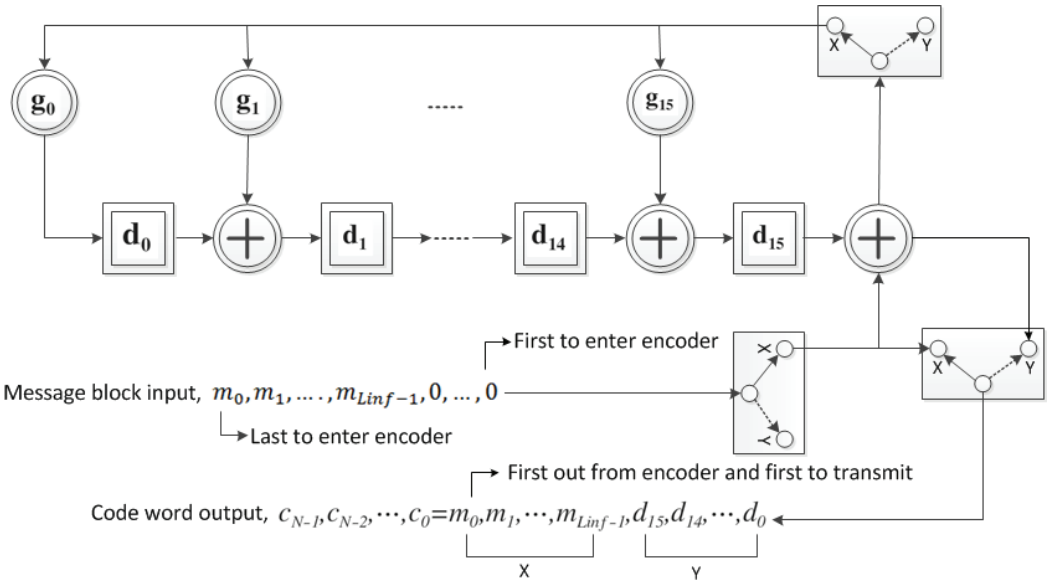


Figure 11a-17—Reed-Solomon encoder GF(2^8)

11a.3.2.7 Code spreading

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) Pseudo random binary sequence (PRBS) codes by linear feedback shift register (LFSR) specified in 11a.3.2.7.1 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.

11a.3.2.7.1 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 11a-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.

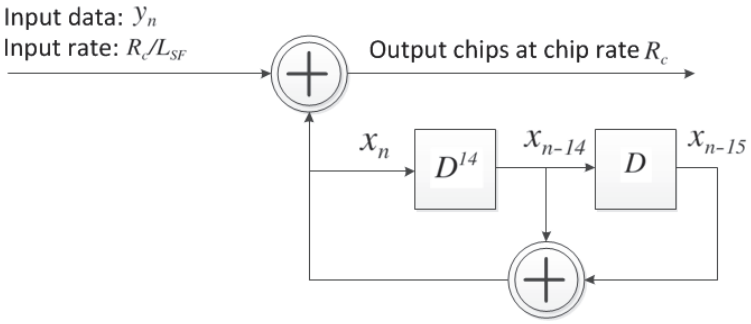


Figure 11a-18—PRBS generation by LFSR

The 15-bit seed value of the LFSR shall be: $[x_{-1}, x_{-2}, \ldots, x_{-15}] = [0101\ 0000\ 0011\ 111]$.

11a.3.2.8 Scrambling

The frames shall be scrambled by modulo-2 addition of the data with the output of a PRBS generator, as illustrated in Figure 11a-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 the following equation:

$$g(D) = 1 + D^{14} + D^{15}$$

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 the following equation:

$$_n = x_{n-14} \oplus x_{n-1},\ n = 0, 1, 2, \ldots$$

The initialization sequence is defined by the following equation:

$$x_{init} = [x_{-1}x_{-2}x_{-3}x_{-4}x_{-5}x_{-6}x_{-7}x_{-8}x_{-9}x_{-10}x_{-11}x_{-12}x_{-13}x_{-14}x_{-15}]$$

The scrambled data bits, s_n, are obtained by the following equation:

$$_n = b_n \oplus x$$

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 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 the following equation:

$$[x_{-1}x_{-2}\ldots x_{-15}] = [11010000101\ S1\ S2\ S3\ S4]$$

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 the following equation:

$$[x_0x_1\ldots x_{15}] = [0001111000111010]$$

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.

11a.3.3 HRCP-OOK PHY frame format

The HRCP-OOK PHY frame shall be formatted as illustrated in Figure 11a-19.

PHY Preamble	Frame Header (Base header)	PHY Payload field

Figure 11a-19—HRCP-OOK PHY frame format

The Frame Header field for the PHY frame shall be formatted as illustrated in Figure 11a-20.

Frame header (Base header)			
PHY Header	MAC Header	HCS	RS parity bits

Figure 11a-20—Frame Header format

The PHY preamble is defined in 11a.3.3.1. The MAC header is defined in 6.2. The PHY header is defined in 11a.3.3.2.1, and the HCS is defined in 11a.3.3.2.2. The PHY Payload field consisting of the MAC frame body and stuff bits, is described in 11a.3.3.3.

11a.3.3.1 PHY Preamble

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 11a-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 11a-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 11a-27—Golay sequence with length 128

Sequence name	Sequence value
$\mathbf{a}_{128}$	0x0536635005C963AFFAC99CAF05C963AF
$\mathbf{b}_{128}$	0x0A396C5F0AC66CA0F5C693A00AC66CA0

Figure 11a-21 shows the structure of the HRCP-OOK PHY preamble. The preambles shall be modulated in OOK waveform.

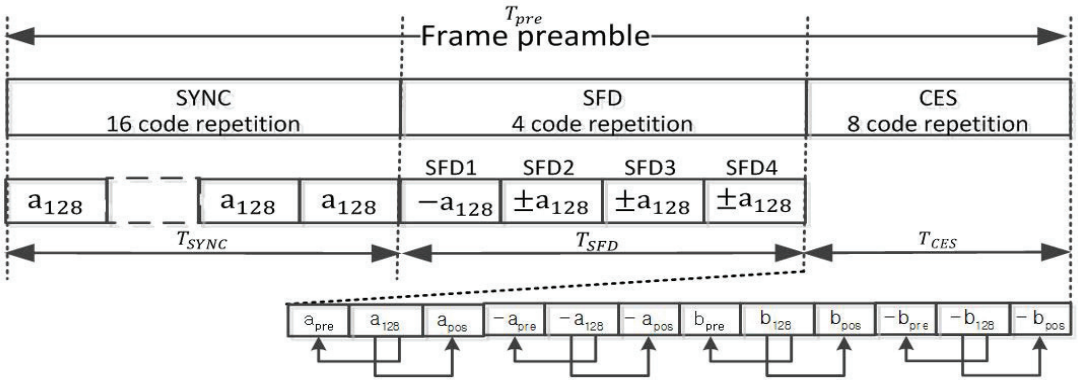


Figure 11a-21—HRCP-OOK preamble structure

11a.3.3.1.1 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 11a-27.

11a.3.3.1.2 SFD

The SFD field is used to establish frame timing and to indicate MCS related parameters. The SFD field shall consist of 4 code repetitions of Golay sequence $\mathbf{a}_{128}$ or $\mathbf{b}_{128}$ as given in Table 11a-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 11a-28.

Table 11a-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 11a-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.

11a.3.3.1.3 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 11a-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 11a-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.

11a.3.3.1.4 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 11a-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.

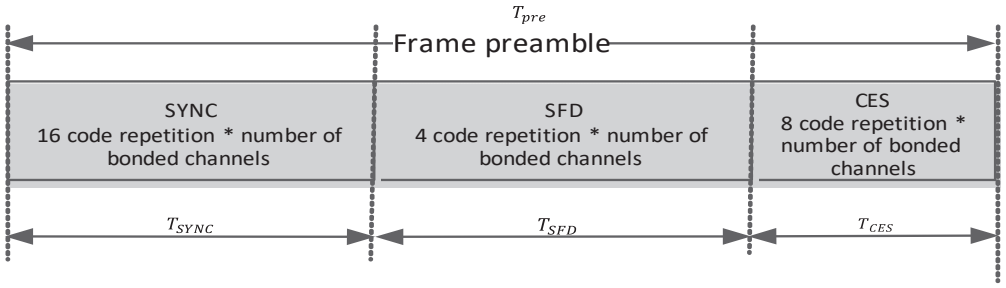


Figure 11a-22—Preamble repetition

Figure 11a-23 shows an example of preamble repetition when two channel bonding is used.

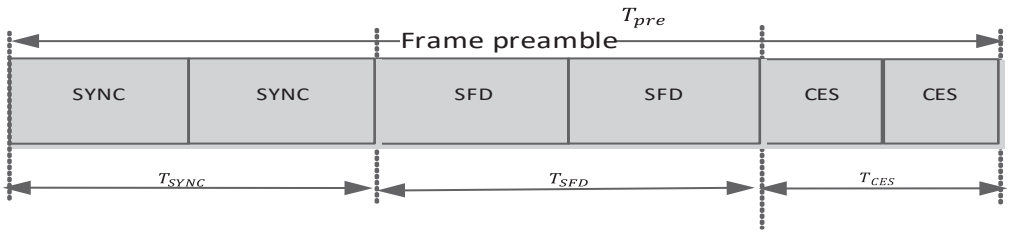


Figure 11a-23—Example of Preamble repetition

11a.3.3.2 Frame Header

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 11a-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 11a.3.2.8.
   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 11a.3.2.6.1. 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 11a-24.

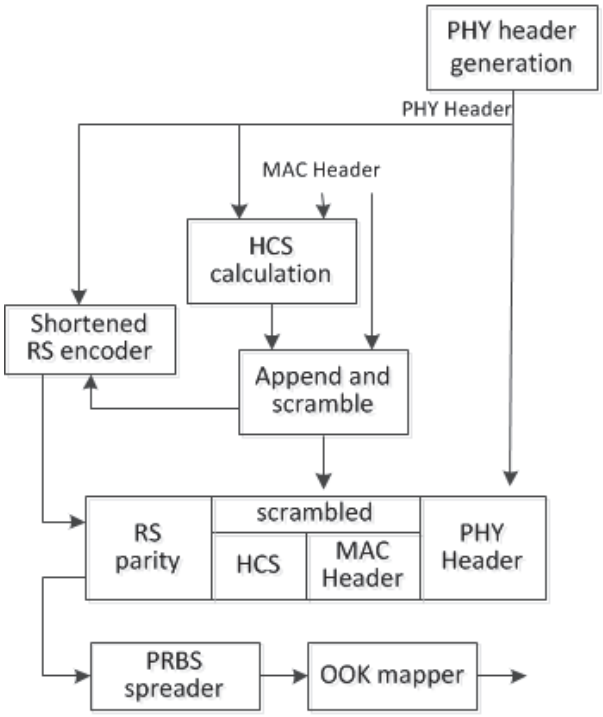


Figure 11a-24—Frame header construction process

b) Spread the frame header with a PRBS generated using an LFSR as described in 11a.3.2.7.1, and the spreading factor shall be 16.

c) Map the frame header onto OOK, as described in 11a.3.2.5.1.

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.

11a.3.3.2.1 HRCP-OOK PHY header

The HRCP-OOK PHY header shall be formatted as illustrated in Figure 11a-25.

bits: b0–b3	b4	b5–b26	b27	b28–b31
Scrambler Seed ID	Aggregation	Frame Length	Pilot Symbol	Reserved

Figure 11a-25—PHY header format for HRCP-OOK PHY

The Scrambler Seed ID field contains the scrambler seed identifier value, as defined in 11a.3.2.8.

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.

11a.3.3.2.2 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 10.2.9 (Header check sequence).

11a.3.3.2.3 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 11a-24.

11a.3.3.3 HRCP-OOK PHY Payload field

The HRCP-OOK PHY Payload field is constructed as shown in Figure 11a-26.

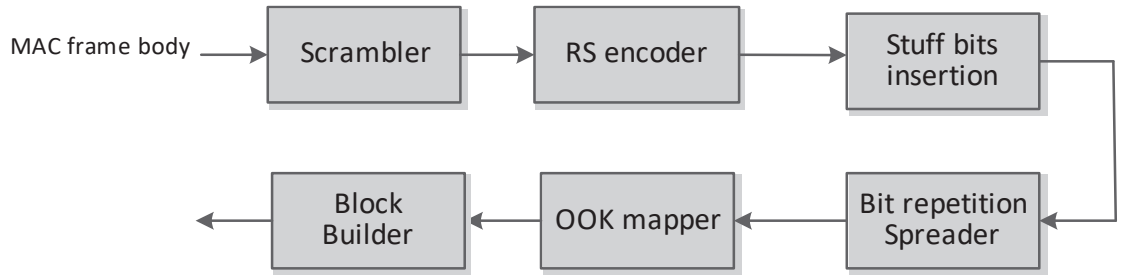


Figure 11a-26—HRCP-OOK PHY Payload construction process

The PHY Payload field shall be constructed as follows:

a) Scramble the MAC frame body according to 11a.3.2.8.

b) Encode the scrambled MAC frame body as specified in 11a.3.2.6. 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 11a.3.2.6.1 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 11a.3.2.8.

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 11a.3.2.5.1.

f) Build blocks from the resulting MAC frame body as specified in 11a.3.3.3.5.

11a.3.3.3.1 HRCP-OOK PHY Payload scrambling

The HRCP-OOK PHY payload shall use the scrambling process defined in 11a.3.2.8.

11a.3.3.3.2 Modulation

Modulation for the MAC frame body is defined in 11a.3.2.5.1.

11a.3.3.3.3 FEC

FEC for the MAC frame body is defined in 11a.3.2.6.

11a.3.3.3.4 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.

11a.3.3.3.5 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 enables 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 11a-27.

If pilot symbols are not used, the transmit symbols shall be divided into block of length $N = 512 \times L_{SF}$.

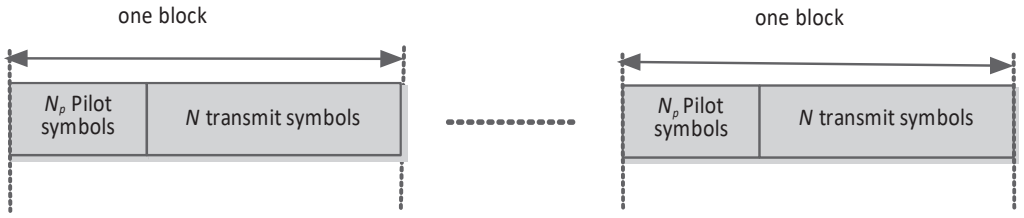


Figure 11a-27—HRCP-OOK frame format with pilot symbols

The pilot symbols consist of a sequence of length $Np = 4 \times L_{SF}$. The pilot symbols for L_{SF}=1 and L_{SF}=2 shall be chosen according to Table 11a-29.

The pilot symbol shall be modulated with OOK.

11a.3.4 Transmitter specifications

11a.3.4.1 Error Vector Magnitude

Eye opening for OOK is described in G.7.

Table 11a-29—OOK pilot symbols

Spreading Factor, L_{SF}	Pilot symbols
1	1010
2	11001100

11a.3.4.2 Transmit center frequency tolerance

The transmitted center frequency tolerance shall be $\pm 30 \times 10^{-6}$ maximum.

11a.3.4.3 Symbol rate

The OOK PHY shall be capable of transmitting at the chip rate, as defined in Table 11a-25, to within $\pm 30 \times 10^{-6}$.

The MAC parameter, *pClockAccuracy*, shall be $\pm 30 \times 10^{-6}$.

11a.3.5 Receiver specifications

11a.3.5.1 Error rate criterion

The error rate criterion shall be a frame error rate (FER) of less than 8% with a frame payload length of 2^{14} octets. The error rate should be determined at the PHY SAP interface.

11a.3.5.2 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 11a.3.5.1 is met. A compliant DEV that implements the HRCP-OOK PHY shall achieve at least the reference sensitivity listed in Table 11a-30.

Table 11a-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

11a.3.5.3 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 11a.3.5.1 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.

11a.3.6 PHY layer timing

The values for the PHY layer timing parameters are defined in Table 11a-31.

Table 11a-31—PHY layer timing parameters

PHY parameter	Value	Subclause
pPHYSIFSTime	0.2 μs, 2.0 μs, 2.5 μs (default)	11a.3.6.3
pPHYChannelSwitchTime	100 μs	11a.3.6.5

11a.3.6.1 IFS

A conforming implementation shall support the IFS parameters, as described in 7.4.1, given in Table 11a-32.

Table 11a-32—IFS parameters

MAC parameter		Corresponding PHY parameter	Definition
SIFS		*pPHYSIFSTime*	11a.3.6.3
RIFS	PRC	2 × *pPHYSIFSTime* + 4.36 μs	7.4.1
	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.

11a.3.6.2 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.

11a.3.6.3 Transmit-to-receive turnaround time

The transmit to receive turnaround time shall be less than *pPHYSIFSTime*.

11a.3.6.4 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 11a-32. A PRC shall use the shorter RIFS than that of a DEV as defined in Table 11a-32.

11a.3.6.5 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*.

11a.3.7 PHY management for HRCP-OOK PHY

The PHY PIB comprises the managed objects, attributes, actions, and notifications required to manage the HRCP-OOK**a** PHY layer of a DEV.

11a.3.7.1 Mximum 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.

11a.3.7.2 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 6.3.1.2a, 6.3.3a.2, 6.3.4a.2, and 6.3.5a.2.

11a.3.7.3 Minimum fragment size

The minimum fragment size, *pMinFragmentSize*, allowed with the HRCP-OOK PHY shall be 512 octets.

Annex C

(informative)

Security considerations

C.1 Background assumptions

Change the first paragraph of C.1 as follows:

All security solutions rely on assumptions about DEVs and the capabilities of potential attackers to thwart possible threats. The goals of mode 1 security are that only authorized DEVs will be able to join a secure piconet or a secure pairnet and that communication is restricted to authorized DEVs.

C.1.1 Physical assumptions

Change the first paragraph and dashed list of C.1.1 as follows:

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 802.15.3 will have little memory available for security.

C.1.2 Network assumptions

Change the first paragraph and dashed list of C.1.2 as follows:

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 PNC, or 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.
— **DEV 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.1.3 Attack model assumptions

Change the first paragraph and dashed list of C.1.3 as follows:

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 pairnet or after the piconet or pairnet has been established. No assumptions are made about the presence of attackers during security setup.

C.1.4 Security key lifecycle issues

C.1.4.2 Membership lifecycle

Change the first paragraph of C.1.4.2 as follows:

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 to 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.1.4.3 Group membership change rekey

Change the first paragraph of C.1.4.3 as follows:

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 8.3.2.

C.2 Claimed security services

C.2.1 Beacon protection protocol

Change Table C-1 as shown:

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.3 Properties of the IEEE 802.15.3 security suite

C.3.1 Key usage

Insert the following new paragraph at the end of C.3.1:

The security operation for pairnets is based on the 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 limitied to 2^{48}. 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 $2^{48} * 2^4$ octets = 2^{52} octets. Then, maximum duration using the single key at 100 Gbps throughput is $2^{52} * 8$ bit / 10^{11} = 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

Change the first four paragraphs (and include the addition of two new paragraphs shown) of C.3.2 as follows:

This standard uses a Time Token, 6.3.1.1, and Secure Frame Counter (SFC), 6.2.7.3, to provide a method to detect and defeat potential replay attacks. For piconets, tThe SFC allows up to 65535 frames to be sent in a single superframe or one every microsecond for the largest possible superframe. The Time Token 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 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 is 4.17 days. A 6 octet time token 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, bBecause 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 SECID is being used by that DEV (which may be true for the management key in shared key operation).

For piconets, iIn 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, iIn 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.

Annex E

(informative)

Protocol implementation conformance statement (PICS) proforma

E.1 Introduction

E.1.1 Scope

Change the first paragraph of E.1.1 as follows:

This annex provides the PICS proforma for IEEE Std 802.15.3-2016 and IEEE Std 802.15.3e-2017.

E.5 Identification of the protocol

Change the first paragraph of E.5 as follows:

This PICS proforma applies to IEEE Std 802.15.3-2016 and IEEE Std 802.15.3e-2017.

Change the title and first paragraph of E.7 as follows:

E.7 PICS proforma—IEEE Std 802.15.3-2016 and IEEE Std 802.15.3e-2017

Table E-1 through ~~Table E-5~~Table E-5a are composed of the detailed questions to be answered, which make up the PICS proforma. Subclause E.7.1 contains the major roles for an IEEE 802.15.3 and IEEE 802.15.3 DEV. Subclause E.7.2 contains the major capabilities for the PHY and radio frequencies. Subclause E.7.3 contains the major capabilities for the MAC sublayer. Subclause E.7.4 indicates which level and type of security is supported in the implementation.

E.7.1 Major roles for IEEE 802.15.3 DEVs

Insert Table E-1a after Table E-1:

E.7.2 PHY functions

Insert Table E-2a and Table E-2b after Table E-2:

Change the title of subclause E.7.3.1:

E.7.3.1 MAC frames for piconet

Table E-1a—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?	4.2.1	O			
FHD3	Supports HRCP-SC PHY	11a.2	O1			
FHD4	Supports HRCP-OOK PHY	11a.3	O1			

Table E-2a—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)	11a.1	FHD3:M			
SC-HPLF2	Supports a single RF channel	11a.2.1	FHD3:M			
SC-HPLF2.1	Supports channel aggregation	11a.2.1	FHD3:O			
SC-HPLF3	Supports $\pi/2$ BPSK and $\pi/2$ QPSK modulations	11a.2, 11a.2.3	FHD3:M			
SC-HPLF3.1	Supports 16QAM modulation	11a.2, 11a.2.2	FHD3:O			
SC-HPLF3.2	Supports 64QAM modulation	11a.2, 11a.2.2	FHD3:O			
SC-HPLF3.3	Supports 256QAM modulation	11a.2, 11a.2.2	FHD3:O			
SC-HPLF4	Supports rate-14/15 and rate-11/15 LDPC codes	11a.2, 11a.2.2.6	FHD3:M			
SC-HPLF5	Encodes and decodes PHY frame format	11a.2.3.1, 11a.2.3.2, 11a.2.3.3	FHD3:M			
SC-HPLF5.1	Insertion and detection of pilot word	11a.2.3.4	FHD3:O			
SC-HPLF5.2	Insertion and detection of pilot preamble (PPRE)	11a.2.3.4	FHD3:O			

Table E-2a—HRCP SC PHY functions (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
SC-HPLF6	Conforms to transmitter requirements	11a.2.4	FHD3:M			
SC-HPLF7	Conforms to receiver requirements	11a.2.5	FHD3:M			
SC-HPLF8	Conforms to timing requirements	11a.2.6	FHD3:M			
SC-HPLF9	Send an Association request by using Access slot	4.3.6	FHD3:M			
SC-HPLF10	PHY PIB values supported	11a.2.7	FHD3:M			

Table E-2b—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)	11a.1	FHD4:M			
OOK-HPLF2	Supports single channel transmission	11a.3.1	FHD4:M			
OOK-HPLF2.1	Supports two channel bonding	11a.3.1	FHD4:O			
OOK-HPLF2.2	Supports three channel bonding	11a.3.1	FHD4:O			
OOK-HPLF2.3	Supports four channel bonding	11a.3.1	FHD4:O			
OOK-HPLF3	Supports OOK modulations	11a.3.2.5	FHD4:M			
OOK-HPLF4	Supports RS(240, 224) and its shortened version	11a.3.2.6	FHD4:M			
OOK-HPLF5	Supports PRBS codes by LFSR with spreading factor 16 for frame header	11a.3.2.7	FHD4:M			

Table E-2b—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	11a.3.2.7	FHD4:O			
OOK-HPLF6	Encodes and decodes PHY frame format	11a.3.3	FHD4:M			
OOK-HPLF6.1	Insertion and detection of CES	11a.3.3.1.2 and 11a.3.3.1.3	FHD4:O			
OOK-HPLF6.2	Insertion and detection of pilot symbols	11a.3.3.3.5	FHD4:O			
OOK-HPLF7	Conforms to transmitter requirements	11a.3.4	FHD4:M			
OOK-HPLF8	Conforms to receiver requirements	11a.3.5	FHD4:M			
OOK-HPLF9	Conforms to timing requirements	11a.3.6	FHD4:M			
OOK-HPLF10	PHY PIB values supported	11a.3.7	FHD4:M			

Change the title of Table E-3:

Table E-3—MAC frames <u>for piconet</u>

Insert new subclause E.7.3.1a after E.7.3.1:

E.7.3.1a MAC frames for pairnet

Table E-3a—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					
MF1.1	MAC Header field and MAC Frame Body field formats for piconet	6.2	X		X	

Table E-3a—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
MF1.2	MAC Header field and MAC Frame Body field format for pairnet	6.2	M		M	
MF1.3	Non–secure MAC Frame Body field format for piconet	6.2	X		X	
MF1.4	Non-secure MAC Frame Body field format for pairnet	6.2	M		M	
MF1.5	Secure MAC Frame Body field format for piconet	6.2	X		X	
MF1.6	Secure MAC Frame Body field format for pairnet	6.2	O (S2a:M)		O (S2a:M)	
MF2	Frame Types					
MF2.1	Non-secure Beacon frame	6.3.1.1	X		X	
MF2.1a	Non-secure Beacon frame for pairnet	6.3.1.1a	O (FHD2:M)		M	
MF2.3	Secure Beacon frame	6.3.1.2	X		X	
MF2.3a	Secure Beacon frame for pairnet	6.3.1.2a	O (FHD2 & S2a:M)		O (S2a:M)	
MF2.4	Imm-ACK frame	6.3.2.1	X		X	
MF2.5	Delayed ACK (Dly-ACK) frame	6.3.2.2	X		X	
MF2.6	Non-secure command frame	6.3.3.1	X		X	
MF2.6a	Pairnet Non-secure command frame	6.3.3a.1	M		M	
MF2.7	Secure command frame	6.3.3.2	X		X	
MF2.7a	Pairnet Secure command frame	6.3.3a.2	O (S2a:M)		O (S2a:M)	
MF2.8	Non-secure data frame	6.3.4.1	X		X	
MF2.8a	Pairnet Non-secure Aggregate Data frame	6.3.4a.1	O		M	
MF2.9	Secure data frame	6.3.4.2	X		X	
MF2.9a	Pairnet Secure Aggregate Data frame	6.3.4a.2	O		O (S2a:M)	
MF2.10	Non-secure Multi-protocol Data frame	6.3.5.1	X		X	

Table E-3a—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
MF2.10a	Non-secure Pairnet Aggregated Multi-protocol Data frame	6.3.5a.1	O		O	
MF2.11	Secure Multi-protocol Data frame	6.3.5.2	X		X	
MF2.11a	Secure Pairnet Aggregated Multi-protocol Data frame	6.3.5a.2	O		O	
MF2.12	Sync frame	6.3.6	X		X	
MF3	IEs					
MF3.1	CTA IE	6.4.1	X		X	
MF3.2	BSID IE	6.4.2	M		M	
MF3.3	Parent Piconet IE	6.4.3	X		X	
MF3.4	DEV Association IE	6.4.4	X		X	
MF3.5	PNC shutdown IE	6.4.5	X		X	
MF3.6	Piconet Parameter Change IE	6.4.6	X		X	
MF3.7	AS IE	6.4.7	X		X	
MF3.8	Pending channel time map (PCTM) IE	6.4.8	X		X	
MF3.9	PNC handover IE	6.4.9	X		X	
MF3.10	CTA status IE	6.4.10	X		X	
MF3.11	Capability IE	6.4.11	X		X	
MF3.11a	PRC Capability IE	6.4.11a	O (FHD2:M)		O	
MF3.11b	PRDEV Capability IE	6.4.11b	M		M	
MF3.11c	Pairnet Operation Parameters IE	6.4.11c	O (FHD2:M)		O	
MF3.12	Transmit Power Parameters IE	6.4.12, 7.12.2.2	O		O	
MF3.13	PS status IE	6.4.13	X		X	
MF3.14	CWB IE	6.4.14	X		X	
MF3.15	Overlapping PNID IE	6.4.15	X		X	
MF3.16	Piconet Services IE	6.4.16	X		X	
MF3.17	Vendor Defined IE	6.4.17	X		X	
MF3.18	Group ID IE	6.4.18	X		X	

Table E-3a—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
MF3.19	Stream Renew IE	6.4.19	X		X	
MF3.20	Next PNC IE	6.4.20	X		X	
MF3.21	Piconet Channel Status IE	6.4.21	X		X	
MF3.22	Synchronization IE	6.4.22	X		X	
MF3.23	TSD IE	6.4.23	X		X	
MF3.24	UEP Specific IE	6.4.24	X		X	
MF3.25	IFS IE	6.4.25	X		X	
MF3.26	CTA Relinquish Duration IE	6.4.26	X		X	
MF3.27	Feedback IE	6.4.27	X		X	
MF3.28	Mapping IE	6.4.28	X		X	
MF3.29	BST Clustering IE	6.4.29	X		X	
MF3.30	PET Clustering IE	6.4.30	X		X	
MF3.31	Beam PET IE	6.4.31	X		X	
MF3.32	HRS Beam PET IE	6.4.32	X		X	
MF3.33	PET Amplitude IE	6.4.33	X		X	
MF3.34	PET Phase IE	6.4.34	X		X	
MF3.35	Sync Frame Frequency IE	6.4.35	X		X	
MF3.36	Vendor Specific IE	6.4.36	O		O	
MF3.37	MIMO Information IE	6.4.37	O		O	
MF3.38	Higher Layer Protocol Information IE	6.4.38	O		O	
MF4	Command Types					
MF4.1	Association Request command Payload format for piconet	6.5.1.1	X		X	
MF4.1a	Association Request command Payload format for pairnet	6.5.1.1	M		FHD2:M	
MF4.2	Association Response command Payload format for piconet	6.5.1.2	X		X	
MF4.2a	Association Response command Payload format for pairnet	6.5.1.2	FHD2:M		M	
MF4.3	Disassociation Request command	6.5.1.3	M		M	

Table E-3a—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
MF4.4	Request Key command	6.5.2.1	S2a:M		S3:M	
MF4.5	Request Key Response command	6.5.2.2	S3:M		S2a:M	
MF4.6	Distribute Key Request command	6.5.2.3	S3:M		S2a:M	
MF4.7	Distribute Key Response command	6.5.2.4	S2a:M		S3:M	
MF4.8	PNC Handover Request	6.5.3.1	X		X	
MF4.9	PNC Handover Response command	6.5.3.2	X		X	
MF4.10	PNC Handover Information command	6.5.3.3	X		X	
MF4.11	PNC Information Request command	6.5.4.1	X		X	
MF4.12	PNC Information command	6.5.4.2	X		X	
MF4.13	Security Information Request command	6.5.4.3	O		O	
MF4.14	Security Information command	6.5.4.4	O		O	
MF4.15	Probe Request command	6.5.4.5	O		O	
MF4.16	Probe Response command	6.5.4.6	M		M	
MF4.17	Piconet Services command	6.5.5.1	X		X	
MF4.18	Announce command	6.5.5.2	X		X	
MF4.19	Channel Time Request command	6.5.6.1	X		X	
MF4.20	Channel Time Response command	6.5.6.2	X		X	
MF4.21	Channel Status Request command	6.5.7.1	X		X	
MF4.22	Channel status response	6.5.7.2	X		X	
MF4.23	Remote Scan Request command	6.5.7.3	X		X	
MF4.24	Remote Scan Response command	6.5.7.4	X		X	
MF4.25	Transmit Power Change command	6.5.7.5	O		O	
MF4.26	PM Mode Change command	6.5.8.5	X		X	

Table E-3a—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
MF4.27	SPS Configuration Request command	6.5.8.3	X		X	
MF4.28	SPS Configuration Response command	6.5.8.4	X		X	
MF4.29	PS Set Information Request command	6.5.8.1	X		X	
MF4.30	PS Set Information Response command	6.5.8.2	X		X	
MF4.31	Security Message command	6.5.9.1	O		O	
MF4.32	Vendor Defined command	6.5.9.2	O		O	
MF4.33	Announce Response command	6.5.5.3	X		X	
MF4.34	PM Mode Change Response command	6.5.8.6	X		X	
MF4.35	AS IE Request command	6.5.9.3	X		X	
MF4.36	AS IE Response command	6.5.9.4	X		X	
MF4.37	Multicast Configuration Request command	6.5.10.1	X		X	
MF4.38	Multicast Configuration Response command	6.5.10.2	X		X	

Change the title of subclause E.7.3.2 as follows:

E.7.3.2 MAC sublayer functions <u>for piconet</u>

Change the title of Table E.4 as follows:

Table E-4—MAC sublayer functions <u>for piconet</u>

Insert new subclause E.7.3.2a after E.7.3.2:

E.7.3.2a MAC sublayer functions for pairnet

E.7.4 Security support

Change the title of Table E-5 as follows:

Table E-5—Security capabilities <u>for piconet</u>

Table E-4a—MAC sublayer functions for pairnet

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF1	Scanning capable	7.2.1	M			
MLF2	Starting capable	7.2.2	X			
MLF3	PNC handover capable	7.2.3	X			
MLF4	Child piconet support					
MLF4.1	Parent PNC supports request mechanism for creating child piconet	7.2.7	X			
MLF4.2	PNC capable DEV support of becoming a child PNC	7.2.7	X			
MLF5	Neighbor piconet support					
MLF5.1	Parent PNC supports request mechanism for creating neighbor piconet	6.5.1.2, 7.2.8	X			
MLF5.2	PNC capable DEV support of becoming a neighbor PNC	7.2.8	X			
MLF6	Stopping piconet operations	7.2.9.1	X			
MLF7	Association	7.3.1	X			
MLF7.1	Association for pairnet	7.3a.1	M			
MLF8	Piconet services support	7.3.2	X			
MLF9	Broadcasting of piconet information	7.3.3	X			
MLF9a	Higher layer protocol setup during association procedure	7.3a.3	O			
MLF10	DEV disassociation	7.3.4	X			
MLF10.1	Disassociation for pairnet	7.3a.2	M			
MLF11	PNC disassociation	7.3.4	X			
MLF12	Contention access methods					
MLF12.1	CAP channel access during piconet operations	7.4.2	X			
MLF12.2	Open and association MCTA operations	7.4.3.3, 7.4.3.4	X			
MLF12.3	Regular MCTA operations	7.4.3.3	X			
MLF12.4	PAP after association	7.4.4	FHD1:M			
MLF13	Asynchronous channel time reservation	7.5.2	X			
MLF14	Synchronization	7.6	X			
MLF14.1	Beacon frame generation	7.6.2	X			

Table E-4a—MAC sublayer functions for pairnet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF14.2	Extended beacon support, reception	7.6.2	X			
MLF14.3	Extended beacon support, generation	7.6.2	X			
MLF14.4	Synchronization for pairnet	7.6a	M			
MLF14.5	Beacon frame generation	7.6a.2	O FHD2:M			
MLF15	Fragmentation and defragmentation	7.7	X			
MLF15.1	Pairnet aggregation	7.8.3	FHD1:O			
MLF15.2	Pairnet Deaggregation	7.8.3	FHD1:M			
MLF16	Acknowledgment and retransmissions					
MLF16.1	No acknowledgment	7.9.1	M			
MLF16.2	Immediate acknowledgment	7.9.2	X			
MLF16.2a	Stk-ACK	7.9.2a.1	FHD1:M			
MLF16.2b	Recovery Process	7.9.2a.2	FHD1:M			
MLF16.3	Delayed acknowledgment	7.9.3	X			
MLF16.4	Retransmissions	7.9.6	X			
MLF16.5	Duplicate detection	7.9.7	X			
MLF16.6	Imp-ACK	7.9.4	X			
MLF17	Peer discovery					
MLF17.1	PNC information request	7.10.1	X			
MLF17.2	Probe request and response	7.10.2	X			
MLF17.3	Announce	7.10.3	X			
MLF17.4	Channel status request	7.10.4	X			
MLF17.5	Remote Scan	7.10.5	X			
MLF18	Changing piconet parameters					
MLF18.1	Moving beacon	7.11.1	X			
MLF18.2	Changing superframe duration	7.11.2	X			
MLF18.3	Setting the BSID and PNID	7.2.10, 7.11.3	X			

Table E-4a—MAC sublayer functions for pairnet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF18.4	Maintaining synchronization in dependent piconets	7.11.4	X			
MLF19	Multi-rate support	7.13	X			
MLF20	Dynamic channel selection	7.12.1	X			
MLF21	Power management					
MLF21.1	PSPS	7.14.1	X			
MLF21.2	SPS (support for at least one SPS set)	7.14.2	X			
MLF21.3	APS	7.14.3	X			
MLF21.4	LLPS	7.14a	FHD1:O			
MLF22	Transmit power control					
MLF22.1	Fixed maximum transmit power	7.12.2.1	X			
MLF22.2	Request transmitter power adjustment	7.12.2.2	O			
MLF22.3	Adjust transmitter power as requested	7.12.2.2	O			
MLF23	AS IE					
MLF23.1	Capable of requesting AS IEs	7.15	X			
MLF23.2	Capable of putting AS IEs in a beacon	7.15	X			
MLF24	Relinquish CTA					
MLF24.1	Capable of relinquishing CTA time	7.4.3.8	X			
MLF24.2	Capable of using CTA time released by other DEV	7.4.3.8	X			
MLF25	Multicast configuration					
MLF25.1	Request to join or leave a multicast group	7.5.3	X			
MLF25.2	Maintain list of multicast groups	7.5.3	X			
MLF26	Handover extensions					
MLF26.1	Preliminary handover capable	7.2.4	X			
MLF26.2	Next PNC capable	7.2.5	X			
MLF26.3	Report status with Piconet Channel Status IE	6.4.21	X			
MLF27	Aggregation	7.8	X			
MLF28	Block ACK	7.9.5	X			

Table E-4a—MAC sublayer functions for pairnet (*continued*)

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF29	Beam forming	Clause 12	X			
MLF30	Channel probing	7.10.7	X			
MLF31	UEP	7.17	X			
MLF32	TSD	12.8	X			

Insert Table E-5a after Table E-5:

Table E-5a—Security capabilities for pairnet

Item Number	Item Description	Reference	Status	Support		
				N/A	Yes	No
S1	Mode 0 support	8.2.1	M			
S2	Mode 1 support	8.2.2, 9.2, 9.3	X			
S2a	Mode 1 support for pairnet	8.2.2, 9a.	O			
S3	Supports acting as a key originator	8.3	FHD2 & S2a: M, S2a: O			
S4	Security info handover	8.4.1	X			

Consensus

WE BUILD IT.

Connect with us on:

Facebook: https://www.facebook.com/ieeesa

Twitter: @ieeesa

LinkedIn: http://www.linkedin.com/groups/IEEESA-Official-IEEE-Standards-Association-1791118

IEEE-SA Standards Insight blog: http://standardsinsight.com

YouTube: IEEE-SA Channel

IEEE
standards.ieee.org
Phone: +1 732 981 0060 Fax: +1 732 562 1571