IEEE STANDARDS ASSOCIATION

IEEE Standard for Local and metropolitan area networks—

Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

Amendment 1: MAC sublayer

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

16 April 2012

IEEE Std 802.15.4e™-2012
(Amendment to
IEEE Std 802.15.4™-2011)

IEEE Std 802.15.4e™-2012
(Amendment to
IEEE Std 802.15.4™-2011)

IEEE Standard for
Local and metropolitan area networks—

Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

Amendment 1: MAC sublayer

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 6 February 2012

IEEE-SA Standards Board

Abstract: IEEE Std 802.15.4-2011 is amended by this standard. The intention of this amendment is to enhance and add functionality to the IEEE 802.15.4 MAC to (a) better support the industrial markets and (b) permit compatibility with modifications being proposed within the Chinese WPAN.

Keywords: asynchronous multi-channel adaptation (AMCA), deterministic and synchronous multi-channel extension (DSME), enhanced beacon request (EBR), enhancements for security, fast association (FastA), IEEE 802.15.4, IEEE 802.15.4e, low energy, low latency deterministic networks (LLDN), MAC performance metrics (metrics), RFID Blink, timeslotted channel hopping (TSCH)

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

Print: ISBN 978-0-7381-7204-0 STD97211
PDF: ISBN 978-0-7381-7225-5 STDPD97211

Notice to users

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

This document is copyrighted by the IEEE. It is made available 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 this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE 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. 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://standards.ieee.org/index.html or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at http://standards.ieee.org/index.html.

Errata

Errata, if any, for this and all other standards can be accessed 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 amendment was submitted to the IEEE-SA Standards Board for approval, the IEEE P802.15 Working Group had the following voting members:

Robert F. Heile, *Chair*
Rick Alfvin, *Co-Vice Chair*
Patrick W. Kinney, *Co-Vice Chair, Secretary*
James P. K. Gilb, *Technical Editor*
Clint Chaplin, *Treasurer*

Seong-Soon Joo, *Task Group 4e Chair*
Patrick W. Kinney, *Task Group 4e past Chair and Secretary*
Ludwig Winkel, *Task Group 4e Chief-Editor*
Gahng-Seop Ahn, *Task Group 4e Co-Editor*
Jonathan Simon, *Task Group 4e Co-Editor*
Benjamin Rolfe, *Task Group 4e Co-Editor*
Wun-Cheol Jeong, *Task Group 4e Co-Editor*
Liang Li, *Task Group 4e Co-Editor*

Emad Afifi
Roberto Aiello
Arthur Astrin
Taehan Bae
Michael Bahr
John Barr
Anuj Batra
Tuncer Baykas
Philip Beecher
Ashutosh Bhatia
Ghulam Bhatti
Gary Birk
Mathew Boytim
Peter David Bradley
Nancy Bravin
David Britz
Monique Brown
Sverre Brubk
Brian Buchanan
John Buffington
Kiran Bynam
Brent Cain
Edgar Callaway
Chris Calvert
Ruben Cardozo
Douglas Castor
Jaesang Cha
Russell Chandler
Soo-Young Chang
Kuor-Hsin Chang
Hind Chebbo
Chang-Soon Choi
Sangsung Choi
Ciaran Connell
David Cypher
Matthew Dahl
David Davenport
Mark Dawkins
Hendricus De Ruijter
Upkar Dhaliwal
Gang Ding
Paul Dixon
Guido Dolmans
Igor Dotlic
Michael Dow
Dietmar Eggert
David Evans
Charles Farlow
John Farserotu
Jeffrey Fischbeck
Mike Fischer
George Flammer
Ryosuke Fujiwara
Noriyasu Fukatsu
Kiyoshi Fukui
John Geiger
Gregory Gillooly
Tim Godfrey
Paul Gorday
Elad Gottlib
Robert Hall
Shinsuke Hara
Hiroshi Harada
Timothy Harrington
Rodney Hemminger
Marco Hernandez
Garth Hillman
Jin-Meng Ho
Wei Hong
Srinath Hosur
David Howard
Taeho Hwang
Jung-Hwan Hwang
Ichirou Ida
Tetsushi Ikegami
Akio Iso
Yeong Min Jang
Adrian Jennings
Steven Jillings
Tae Kang
Noh-Gyoung Kang
Tae-Gyu Kang
Tatsuya Kato
Shuzo Kato
Jeritt Kent
Prithpal Khakuria
Dae Ho Kim
Dong-Sun Kim
Dukhyun Kim
Jaehwan Kim
Jeffrey King
Ryuji Kohno
Fumihide Kojima
Bruce Kraemer
Raymond Krasinski
Masahiro Kuroda
John Lampe
Zhou Lan
Khanh Le
Cheolhyo Lee
Hyungsoo Lee
Myung Lee
Daniel Lewis
Huan-Bang Li
Sang-Kyu Lim
Jeremy Link
Michael Lynch
Robert Mason
Tomokuni Matsumura
Jeff McCullough
Michael McGillan
Michael McInnis

Michael McLaughlin
Charles Millet
Siamak Mirnezami
Rishi Mohindra
Emmanuel Monnerie
Rajendra Moorti
Robert Moskowitz
Hamilton Moy
Peter Murray
Theodore Myers
Chiu Ngo
Paul Nikolich
Hirohito Nishiyama
David Olson
Okundu Omeni
Ryoji Ono
Laurent Ouvry
James Pace
Jahng Park
Seung-Hoon Park
Taejoon Park
Hyung-Il Park
Ranjeet Patro
Al Petrick
Dalibor Pokrajac
Daniel Popa
Stephen Pope
Richard Powell
Clinton Powell
Chang-Woo Pyo
Mohammad Azizur Rahman
Sridhar Rajagopal
Jayaram Ramasastry
Marc Reed
Ivan Reede
Richard Roberts
Craig Rodine
June Chul Roh
Seung-Moon Ryu
Didier Sagan
Kentaro Sakamoto
Filippo Will San
H. Sanderford
Kamran Sayrafian
Timothy Schmidl
Michael Schmidt
Jean Schwoerer
Cristina Seibert
Neal Seidl
Kunal Shah
Steve Shearer
Stephen Shellhammer
Shusaku Shimada
CheolHo Shin
Chang Sub Shin
Michael Sim
Jaeseung Son
Paul Stadnik
René Struik
Chin-Sean Sum
Hui-Hsia Sung
Gu Sungi
Kenichi Takizawa
Hirokazu Tanaka
Larry Taylor
Mark Thompson
James Tomcik
Ichihiko Toyoda
David Tracey
Khanh Tran
Jerry Upton
Jana van Greunen
Hartman van Wyk
Michel Veillette
Billy Verso
Bhupender Virk
Joachim Walewski
Junyi Wang
Quan Wang
Xiang Wang
Andy Ward
Scott Weikel
Nicholas West
Mark Wilbur
Eun Tae Won
Alan Chi Wai Wong
Tao Xing
Yang Yang
Wen-Bin Yang
Kazuyuki Yasukawa
Kamya Yazdandoost
Kaoru Yokoo
Mu Zhao
Bin Zhen

Major contributions were received from the following individuals:

Micheal Bahr
Ghulam Bhatti
Kuor-Hsin Chang
Tim Godfrey
Hiroshi Harada
Wei Hong
Chol Su Kang
Jeffrey King
Fumihide Kojima
Myung Lee
HaiTao Liu
Tae Rim Park
Kentaro Sakamoto
Chang Sub Shin
René Struik
Tao Xing
Yang Yang
Kazuyuki Yasukawa
Mu Zhao

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

Jon Adams
Emad Afifi
Rick Alfvin
Butch Anton
Lee Armstrong
Takamasa Asano
Arthur Astrin
Michael Bahr
Jonathan Bailey
Edward Baranski
Tuncer Baykas
Philip Beecher
Harry Bims
Mathew Boytim
Nancy Bravin
Vern Brethour
Monique Brown
Sverre Brubk
John Buffington
William Byrd
Edgar Callaway
Radhakrishna Canchi
Dave Cavalcanti
Kuor-Hsin Chang
Clint Chaplin
Keith Chow
Charles Cook
Joseph Decuir
Patrick Diamond
Thomas Dineen
Sourav Dutta
Peter Ecclesine
Richard Edgar
John Egan
Stanislav Filin
Will San Filippo
Kimberly Francisco
James P. K. Gilb
Gregory Gillooly
Tim Godfrey
Randall C. Groves
Michael Gundlach
Henrik Gustafsson
Jose Gutierrez
C. Guy
Rainer Hach
Hiroshi Harada
Timothy Harrington
Rodney Hemminger
Marco Hernandez
Ryoichi Higashi
Wei Hong
Srinath Hosur
David Howard
Heqing Huang
David Hunter
Ichirou Ida
Tetsushi Ikegami
Noriyuki Ikeuchi
Akio Iso
Atsushi Ito
Raj Jain
Oyvind Janbu
Junghoon Jee
Jorjeta Jetcheva
Tal Kaitz
Shinkyo Kaku
Ruediger Kays
Raymond H. Kelley
Jeritt Kent
Stuart J. Kerry
Yongbum Kim
Jeffrey King
Patrick W. Kinney
Fumihide Kojima
Bruce Kraemer
Yasushi Kudoh
Jeremy Landt
Khanh Tuan Le
David G. K. Li
Jan-Ray Liao
Arthur Light
Lu Liru
HaiTao Liu
William Lumpkins
Greg Luri
Michael Lynch
Elvis Maculuba
Stephen Makgill
Robert Mason
Michael McInnis
Paul Mobus
Apurva Mody
Emmanuel Monnerie
Kenichi Mori
Robert Moskowitz
Peter Murray
Michael S. Newman
Paul Nikolich
John Notor
Okundu Omeni
Masayuki Oodo
Chris Osterloh
Satoshi Oyama
Kevyn Pair
Dalibor Pokrajac
Daniel Popa
Clinton Powell
Venkatesha Prasad
Mohammad Azizur Rahman
Jayaram Ramasastry
Mark Ranta
Ivan Reede
Maximilian Riegel
Robert Robinson
Craig Rodine
Benjamin Rolfe
Carl Royer
Randall Safier
Kentaro Sakamoto
Kazuyuki Sakoda
Shigenobu Sasaki
Katsuyoshi Sato
Bartien Sayogo
Timothy Schmidl
Cristina Seibert
Kunal Shah
Jie Shen
Shusaku Shimada
Gil Shultz
Jonathan Simon
Kapil Sood
Amjad Soomro
Robert Soranno
Thomas Starai
René Struik
Walter Struppler
Mark Sturza
Chin-Sean Sum
Chen Sun
Larry Taylor
Ichihiko Toyoda
Ha Nguyen Tran
Anna Urra
Jana van Greunen
Dmitri Varsanofiev
Prabodh Varshney
John Vergis
Billy Verso
Dalton Victor
Bhupender Virk
Jerome Viviano
George Vlantis
Khurram Waheed
Junyi Wang
Stanley Wang
Xiang Wang
Andy Ward
Hung-Yu Wei
Scott Weikel
Mark Wilbur
Ludwig Winkel
Andreas Wolf
Tao Xing
Yang Yang
Kazuyuki Yasukawa
Jianming Yuan
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 6 February 2012, it had the following membership:

Introduction

This introduction is not part of IEEE Std 802.15.4e-2012, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)—Amendment 1: MAC sublayer.

This amendment to IEEE Std 802.15.4-2011 specifies additional media access control (MAC) behaviors and frame formats that allow IEEE 802.15.4 devices to support a wide range of industrial and commercial applications that were not adequately supported prior to the release of this amendment. Specifically, these changes to the MAC sublayer facilitate industrial applications such as Process Control and Factory Automation in addition to the MAC behaviors that support the Chinese Wireless Personal Area Network (CWPAN) standard.

Contents

IEEE Standard for
Local and metropolitan area networks—

Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)

Amendment 1: MAC sublayer

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

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

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.[1]

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]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

Insert in alphabetical order the following definitions:

blink: A multipurpose frame with a one-octet Frame Control field used by unassociated devices.

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

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

coordinated sampled listening (CSL): A low-energy mode of media access control specifying how receiving devices periodically monitor the channel(s) for incoming transmissions at low duty cycles.

coordinated sampled listening channel sample: The operation of performing energy detection on a channel and attempting to receive a wake-up frame if energy is detected.

coordinated sampled listening payload frame: A beacon, data, or command frame carrying the application payload following the wake-up sequence in a coordinated sampled listening period.

coordinated sampled listening period: The time interval in which receiving devices monitor the channel(s) for incoming transmissions.

coordinated sampled listening phase: The length of time between a given time and the next channel sample, as a fraction of the coordinated sampled listening period.

coordinated sampled listening rendezvous time: A timestamp in the coordinated sampled listening wake-up frame payload indicating the expected length of time in milliseconds between the end of the wake-up frame transmission and the beginning of the payload frame transmission.

coordinated sampled listening wake-up frame: A special short frame repeatedly transmitted before the payload frame to ensure the reception of the payload frame by a coordinated sampled listening receiving device.

coordinated sampled listening wake-up frame sequence: A sequence of back-to-back wake-up frames up to the duration of the coordinated sampled listening period.

downlink: Data communication from the personal area network (PAN) coordinator to the PAN device.

encapsulated service data unit: A unit of data that has been passed down from a higher layer that has been encapsulated.

enhanced active scan: An active scan using one or more enhanced beacon requests.

enhanced beacon (EB): A beacon that contains information elements (IEs) (rather than a superframe specification) to describe various properties of the network.

enhanced beacon request (EBR): A beacon request with optional filter elements to limit the number of coordinator responses.

information element (IE): A formatted data object consisting of an ID, a length, and a data payload used to pass data between layers or devices.

low latency deterministic network (LLDN): A personal area network (PAN) organized as a star network with a superframe structure and using LLDN frames.

low latency deterministic network (LLDN) device: A device in an LLDN that is associated to an LLDN coordinator.

receiver initiated transmission (RIT): An alternative low-energy mode to coordinated sampled listening in which receiving devices periodically broadcast data request frames, and transmitting devices only transmit to a receiving device upon receiving a data request frame.

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

slot owner: A low latency deterministic network (LLDN) device that is assigned exclusive access rights at the beginning of a timeslot in an LLDN.

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

uplink: data communication from the personal area network (PAN) device to the PAN coordinator.

3.2 Acronyms and abbreviations

Insert in 3.2 the following abbreviations and acronyms in alphabetical order:

ACK	positive acknowledgment
AMCA	asymmetric multi-channel adaptation
ASN	absolute slot number
CSL	coordinated sampled listening
CTS	clear to send
DSME	deterministic and synchronous multi-channel extension
EB	enhanced beacon
EBR	enhanced beacon request
ECFP	extended contention-free period
ESDU	encapsulated service data unit
ESOR	enhanced security and overhead reduction
EUI	extended unique identifier
FastA	fast association

GACK	group acknowledgment
IE	information element
LE	low energy
LL	low latency
LLDN	low latency deterministic network
NACK	negative acknowledgment
NHL	next higher layer
PA	process automation
RIT	receiver initiated transmission
RTS	request to send
SAB	slot allocation bitmap
SD	superframe duration
TSCH	timeslotted channel hopping
WLAN	wireless local area network

4. General description

4.1 General

Insert before 4.2 the following text:

Supplemental information for different industrial domains is contained in Annex I.

4.3 Network topologies

4.3.1 Star network formation

Insert before 4.3.2 the following paragraphs:

A low latency deterministic network (LLDN) operates in a star topology. More information on the star topology of LLDNs is given in Annex I.

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

4.3.2 Peer-to-peer network formation

Insert before 4.4 the following paragraphs:

Deterministic and synchronous multi-channel extension (DSME) supports both star and peer-to-peer network topologies.

TSCH devices support peer-to-peer topologies.

4.5 Functional overview

4.5.1 Superframe structure

Insert the following before the existing text:

4.5.1.1 General

This standard allows the optional use of different superframe structures, as follows:

- — Superframe structure described in 4.5.1.1 (as defined in IEEE Std 802.15.4™-2011).
- — Superframe structure described in 4.5.1.2 based on beacons defined in 5.2.2.1 (Enhanced Beacon) with an information element (IE) defined in 5.2.4.9 (DSME PAN Descriptor).
- — Superframe structure described in 4.5.1.3 based on LL-Beacons defined in 5.2.2.5.2.

A Slotframe structure, as described in 4.5.1.4, takes the place of superframes in a TSCH PAN.

Delete the following existing text:

~~This standard allows the optional use of a superframe structure.~~

Insert before 4.5.2 the following subclauses and figures:

4.5.1.2 DSME multi-superframe structure

The DSME-enabled (i.e., *macDSMEenabled* is TRUE) PANs use the DSME multi-superframe structure. The format of the DSME multi-superframe structure is defined by coordinators that periodically transmit an Enhanced Beacon (EB) with the DSME PAN descriptor IE. A multi-superframe is a cycle of repeated superframes, each of which consists of a beacon frame, a Contention Access Period (CAP) and a Contention-Free Period (CFP). An example of a multi-superframe structure is shown in Figure 4a.

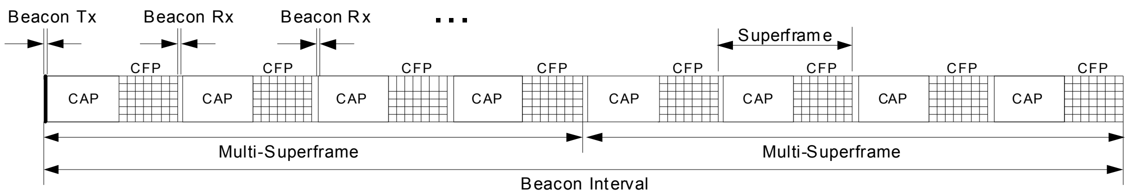

Figure 4a—General DSME Multi-superframe Structure

The single common channel, the logical channel number used in the successful association, is used to transmit the enhanced beacon frames and the frames transmitted during the CAP. The channel diversity method, either channel adaptation or channel hopping as selected by *macChannelDiversityMode*, is applied to transmit data frames and acknowledge frames during the CFP. Beacons are scheduled as defined in 5.1.10.6. Frames sent during the CFP are transmitted using the allocated channel for DSME-GTS. A DSME-GTS can be allocated on any of the available channels in the current ChannelPage.

Details on the DSME Multi-superframes structure are described in 5.1.10.1.

4.5.1.3 Superframe structure based on LL Beacons

LLDN PANs (i.e., *macLLenabled* is TRUE) use the LLDN superframe structure as described in 5.1.1.6. The superframe is divided into a beacon slot, 0 or 2 management timeslots (i.e., 2 if *macLLDNmgmtTS* is TRUE), and *macLLDNnum TimeSlots* number of timeslots of equal length as shown in Figure 4b.

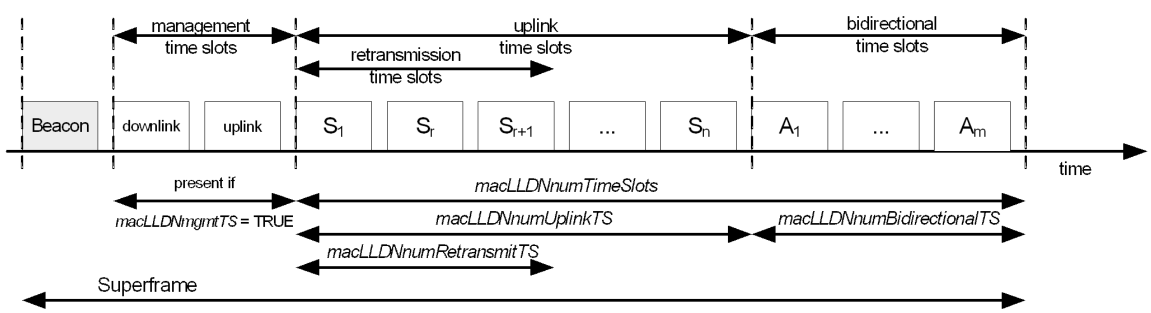

Figure 4b—LLDN Superframe with dedicated timeslots

The first timeslot of each superframe contains an LL-Beacon frame. The LL-Beacon frame is used for synchronization with the superframe structure. It is also used for re-synchronization of devices that, for instance, went into power save or sleep mode.

The beacon timeslot may be followed by two management timeslots, one for downlink and one for uplink.

The remaining timeslots are assigned to the LLDN devices in the network; there is no explicit addressing necessary inside the frames provided that there is exactly one device assigned to a timeslot as per 5.1.1.6.6. The determination of the sender is achieved through the indexing of timeslots. If there is more than one device assigned to a timeslot, the timeslot is referred to as shared group timeslot, and a simple addressing scheme with 8-bit addresses, *macSimpleAddress,* is used as described in 6.3.

4.5.1.4 Slotframes

In a TSCH PAN, the concept of the superframe is replaced with a slotframe. The slotframe also contains defined periods of communications between peers that may be either CSMA-CA or guaranteed, but the slotframe automatically repeats based on the participating devices' shared notion of time, and does not require beacons to initiate communications. Unlike the superframe, slotframes and a device's assigned timeslot(s) within the slotframe may initially be communicated by beacon, but are typically configured by a higher layer as the device joins the network. Because all devices share common time and channel information, devices may hop over the entire channel space to minimize the negative effects of multipath fading and interference, and do so in a slotted way to avoid collisions, minimizing the need for retransmissions. Both of these features are desirable for operation in a harsh industrial environment.

4.5.2 Data transfer model

4.5.2.1 Data transfer to a coordinator

Insert before 4.5.2.2 the following paragraph and figure:

When a device wishes to transfer data to a PAN coordinator in an LLDN, it first listens for the network beacon. When the beacon is found, the device synchronizes to the superframe structure. At its assigned timeslot, the device transmits its data frame to the LLDN PAN coordinator. If the device transmits its data frame in a dedicated timeslot or as slot owner of a shared group timeslot, the data frame is transmitted without using CSMA-CA. If the device transmits its data frame in a shared group timeslot and is not the slot owner, the data frame is transmitted using slotted CSMA-CA as described in 5.1.1.4.4, or ALOHA described in 4.5.4.3, depending on the used PHY. The LLDN PAN coordinator may acknowledge the successful reception of the data by transmitting an optional acknowledgment frame. Successful data transmissions in dedicated timeslots or by the slot owner are acknowledged by the LLDN PAN coordinator with a Group Acknowledgment either in the next beacon or as a separate group acknowledgment (GACK) frame. This sequence is summarized in Figure 4c.

.

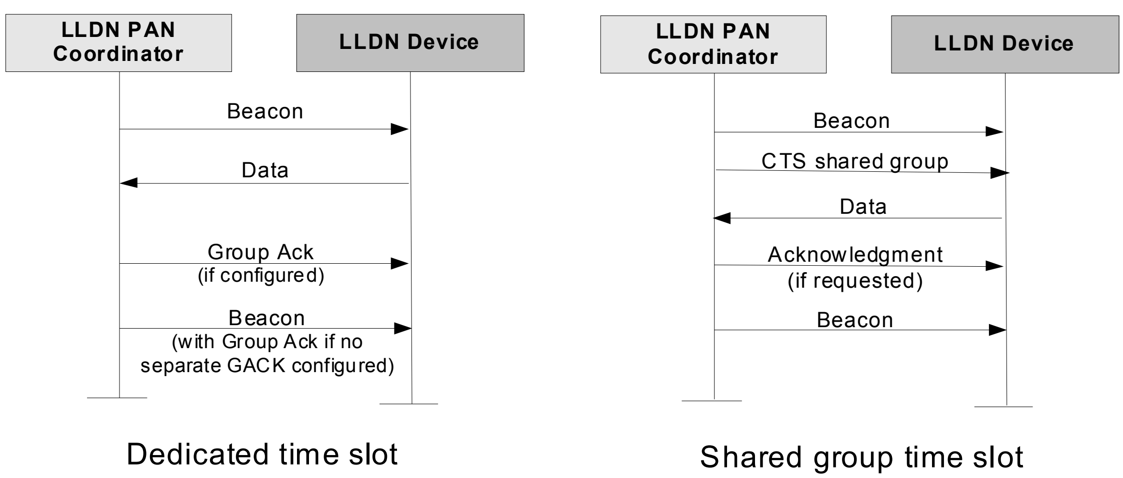

Figure 4c—Communication to a PAN coordinator in an LLDN

4.5.2.2 Data transfer from a coordinator

Insert before 4.5.2.3 the following paragraphs and figure:

A data transfer from an LLDN PAN coordinator is only possible in the *macLLDNnumBidirectionalTS* timeslots described in 5.1.1.6.5 and if the Transmission Direction field in the Flags field of the beacon indicates downlink direction.

When the LLDN PAN coordinator wishes to transfer data to an LLDN device assigned to a bidirectional timeslot in an LLDN, it indicates in the network beacon that the transmission direction is downlink. At the appropriate time, the LLDN PAN coordinator transmits its data frame to the device without using CSMA-CA. The device may acknowledge the successful reception of the data by transmitting an acknowledgment frame to the LLDN PAN coordinator in the same timeslot of the next superframe. In order to do so, the transmission direction has to be uplink in that superframe. This sequence is summarized in Figure 4d.

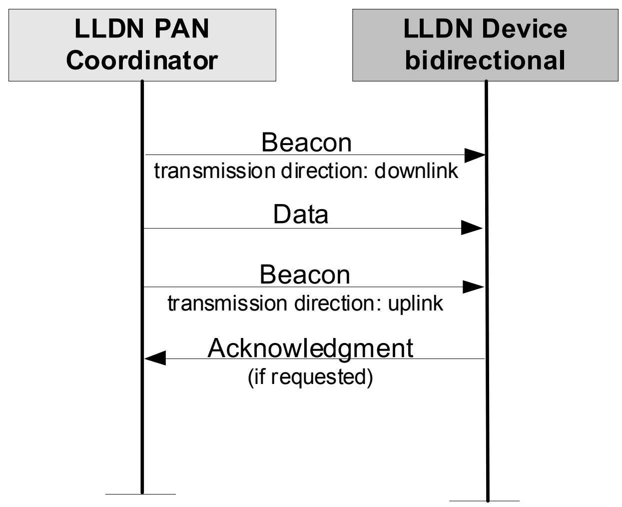

Figure 4d—Communication from a PAN coordinator to a device in an LLDN

4.5.3 Frame structure

Change in 4.5.3 the first paragraph and dashed list as follows:

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

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

— A ~~d~~Data frame, used for ~~all~~ transfers of data

— An unsecured and enhanced acknowledgment frame, used for confirming successful frame reception

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

— A multipurpose frame, used for transfers of data where additional MHR options are needed

— A LLDN frame, used to minimize overhead in latency critical networks

Insert at the end of 4.5.3 the following paragraph:

This standard makes use of IEs to transfer formatted data between layers and between devices. IEs consist of an identification (ID) (or tag), a length, and the IE content. Devices can accept or discard a particular element if the ID is known, and skip over unknown ID elements. IDs are categorized into managed spaces, whose content is defined in this standard, and unmanaged spaces, whose content is left to a higher layer to define. As defined in this standard, IEs provide extensibility of the MAC protocol: future versions of the protocol can add functionality by adding new IE types in a backwards compatible way, without redefining the structure of the MAC frame or new frame types. This removes the limitations of fixed length MAC Header fields such as the Frame Version and Addressing Mode fields.

4.5.4 Improving probability of successful delivery

4.5.4.1 CSMA-CA mechanism

Insert before 4.5.4.2 the following:

LLDNs use a slotted CSMA-CA channel access mechanism for management timeslots and shared group timeslots, where the backoff slots are aligned as follows:

- With the start of the beacon transmission in management timeslots
- With tSlotTxOwner in shared group timeslots

Each time a device wishes to transmit data frames with CSMA-CA at the appropriate places, it locates the boundary of the next backoff slot and then waits for a random number of backoff slots. If the channel is busy, following this random backoff, the device waits for another random number of backoff slots before trying to access the channel again. If the channel is idle, the device begins transmitting on the next available backoff slot boundary. Acknowledgment and beacon frames are sent without using a CSMA-CA mechanism.

Devices in a TSCH PAN also use a slotted CSMA-CA scheme for shared slots, but use a slotted backoff scheme as described in 5.1.1.4.3.

4.5.4.3 Frame acknowledgment

Insert in 4.5.4.3 after the first paragraph the following text:

The receiving device may insert additional content in an enhanced acknowledgment encapsulated as IEs. If the originator does not understand the IE content of the acknowledgment, it is ignored, but the transmission is considered successful.

Insert before 4.5.5 (after 4.5.4.4) the following subclause:

4.5.4.5 Asynchronous multi-channel adaptation

The single common channel approach may not be able to connect all devices in the PAN. The variance of channel condition can be large and channel asymmetry between two neighboring device can happen. Asynchronous multi-channel adaptation (AMCA) is a solution to handle such cases. The AMCA is performed in non-beacon mode, refer to 5.1.12 for details.

4.5.5 Power consumption considerations

Insert the following header before the existing text:

4.5.5.1 General

Insert before 4.5.6 the following new subclause:

4.5.5.2 Low-energy mechanisms

Two low-energy mechanisms are provided to further reduce energy consumption by allowing devices to communicate while maintaining low duty cycles. They are coordinated sampled listening (CSL) and receiver initiated transmission (RIT).

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

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

- Low data traffic rate and loose latency requirement, where a few seconds of latency is allowable by application.
- Local regulations restricting the duration of continuous radio transmissions (e.g., 950 MHz band in Japan).

4.5.6 Security

Insert before the last paragraph the following paragraph:

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

5. MAC protocol

5.1 MAC functional description

5.1.1 Channel access

Insert before 5.1.1.1 the following subclause (5.1.1a) and figure:

5.1.1a Channel hopping

Networks may support channel hopping where permitted or when required using any multichannel PHY. Hopping sequences are defined in the Hopping Sequence PIB (as defined in Table 52f) and may be of arbitrary length up to 511 elements, and cover all or a subset of channels available to the PHY. Devices may hop in a slotted mode (e.g., TSCH or DSME) or in an unslotted mode.

The Hopping Sequence PIB includes an ID (*macHoppingSequenceID*), fields describing operation of the PHY, the length of the sequence (*macHoppingSequenceLength*), an ordered list of channels (*macHoppingSequenceList*), and the dwell time for unslotted hopping, if applicable.

All hopping sequences are referred to by an ID, *macHoppingSequenceID*, with ID = 0 denoting the default sequence for a particular PHY (or PHY configuration if the PHY supports more than one channel list).

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

a) SHUFFLE is a *macHoppingSequenceLength*-sized array. The contents of this array are equivalent to the first *macHoppingSequenceLength* outputs of a 9-bit linear feedback shift register (LFSR) with polynomial $x^9 + x^5 + 1$ and a starting seed of 255. Each LFSR output is modulo *macHoppingSequenceLength*, so that each entry of SHUFFLE is between 0 and (*macHoppingSequenceLength* − 1), inclusive.

b) CHANNELS is a *macHoppingSequenceLength*-sized array that is initially populated with the monotonically increasing list of channels available to the PHY.

c) CHANNELS is shuffled as per Figure 7a. Elements may wind up being swapped multiple times in this process.

d) The default sequence (i.e., *macHoppingSequenceList for macHoppingSequenceID* = 0) is equivalent to the shuffled CHANNELS array.

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

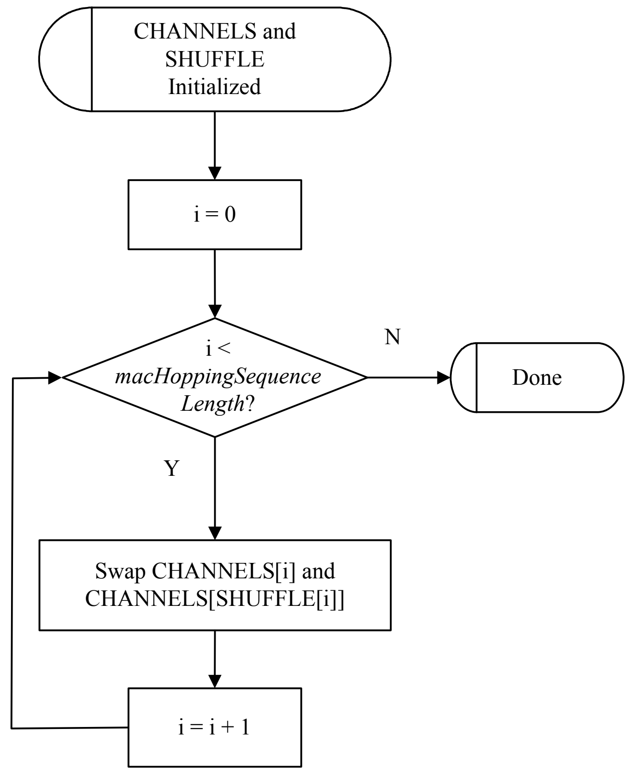

Figure 7a—CHANNELS shuffle algorithm

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

In general, any optional MAC mode as described in Annex I can calculate the channel as

CH = *macHoppingSequenceList* [COUNTER % *macHoppingSequenceLength*]

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

5.1.1.2 Superframe structure

Insert in 5.1.1.1 after the first paragraph the following text:

For LLDN applications an additional superframe structure with LLDN beacons is required, as described in 5.1.1.6.

For LE-applications an additional and optional superframe structure is adopted, as described in 5.1.1.7.

5.1.1.4 CSMA-CA algorithm

Insert after the heading of 5.1.1.4 the following subheading followed by the existing text:

5.1.1.4.1 General

Replace existing Figure 11 with the following new Figure 11:

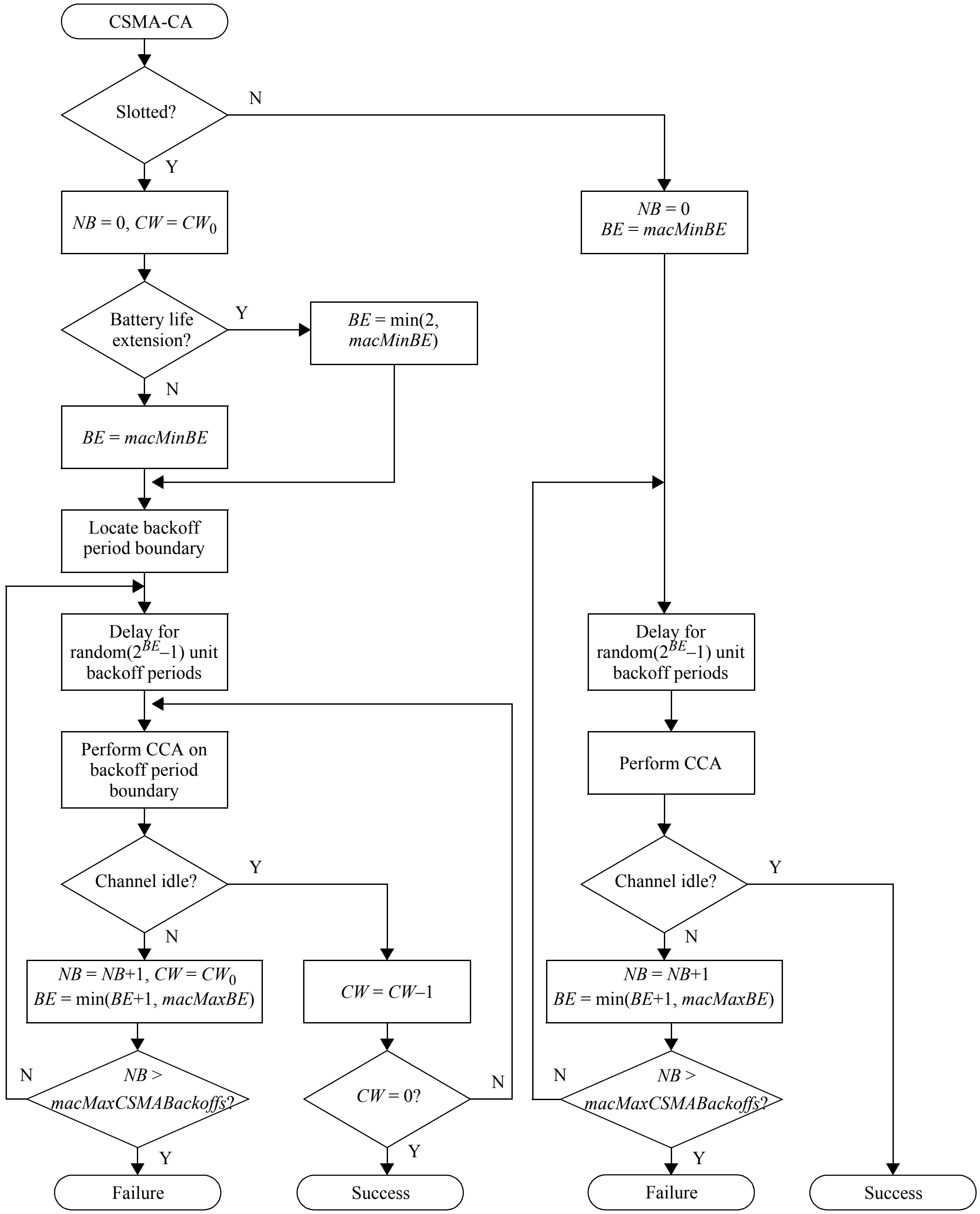

Figure 11—CSMA-CA Algorithm

Insert before 5.1.2 the following subclauses (5.1.1.4.2−5.1.1.7.4):

5.1.1.4.2 TSCH CCA algorithm

When a device is operating in TSCH mode as described in 5.1.2.6, CCA may be used to promote coexistence with other users of the radio channel. For other devices in the same network, the start time of transmissions, *macTsTxOffset*, is closely aligned making intra-network collision avoidance using CCA ineffective. TSCH devices also use channel hopping so there is no backoff period used when CCA prevents a transmission.

When a device has a packet to transmit, it shall wait for a link to the destination device. If CCA has been enabled, the MAC requests the PHY to perform a CCA at the designated time in the timeslot, *macTsCCAOffset*, without any backoff delays. Figure 11a extends Figure 11 for the TSCH mode.

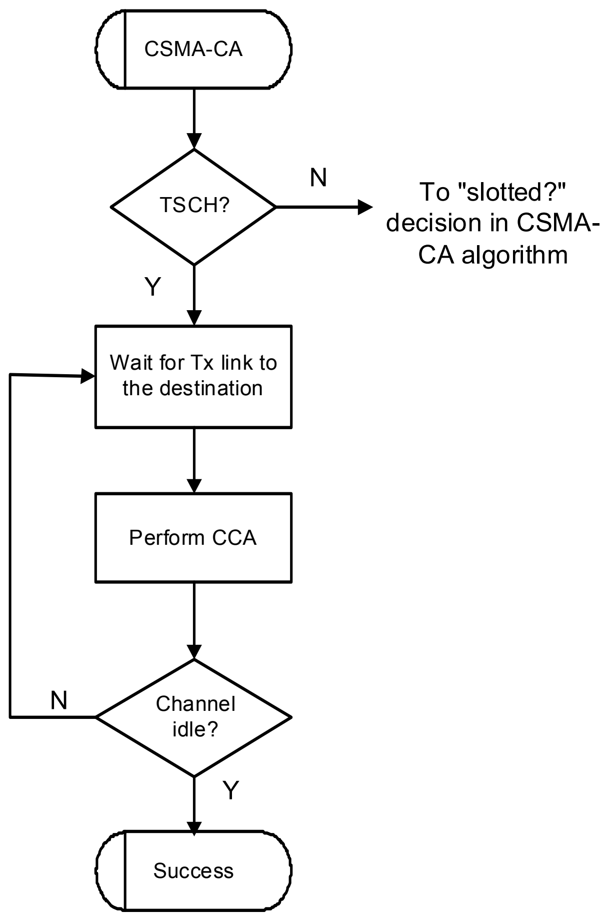

Figure 11a—TSCH CSMA-CA algorithm

5.1.1.4.3 TSCH CSMA-CA algorithm

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

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

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

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

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

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

For the *macMaxBE* and *macMinBE* values when the device is in TSCH mode refer to Table 52b.

The device shall use an exponential backoff mechanism analogous to that described in 5.1.1.4.1. A device upon encountering a transmission failure in a shared link shall initialize the BE to *macMinBE*. The MAC sublayer shall delay for a random number in the range 0 to $2^{BE} - 1$ shared links (on any slotframe) before attempting a retransmission on a shared link. Retransmission on a dedicated link may occur at any time. For each successive failure on a shared link, the device should increase the backoff exponent until the backoff exponent = *macMaxBE*. Successful transmission on a shared link resets the backoff exponent to *macMinBE*.

If an acknowledgment is still not received after *macMaxFrameRetries* retransmissions, the MAC sublayer shall assume the transmission has failed and notify the next higher layer of the failure. This is illustrated in Figure 11b.

5.1.1.4.4 LLDN simplified CSMA-CA

A simplified CSMA-CA algorithm is used during Management timeslots and Shared Group timeslots in LLDNs.

The simplified CSMA-CA is a slotted CSMA-CA mechanism and follows the same algorithm as described in 5.1.1.4.1.

The backoff slots of *aUnitBackoffPeriod* symbols are aligned with the start of the beacon transmission in management timeslots and with tSlotTxOwner in shared group timeslots.

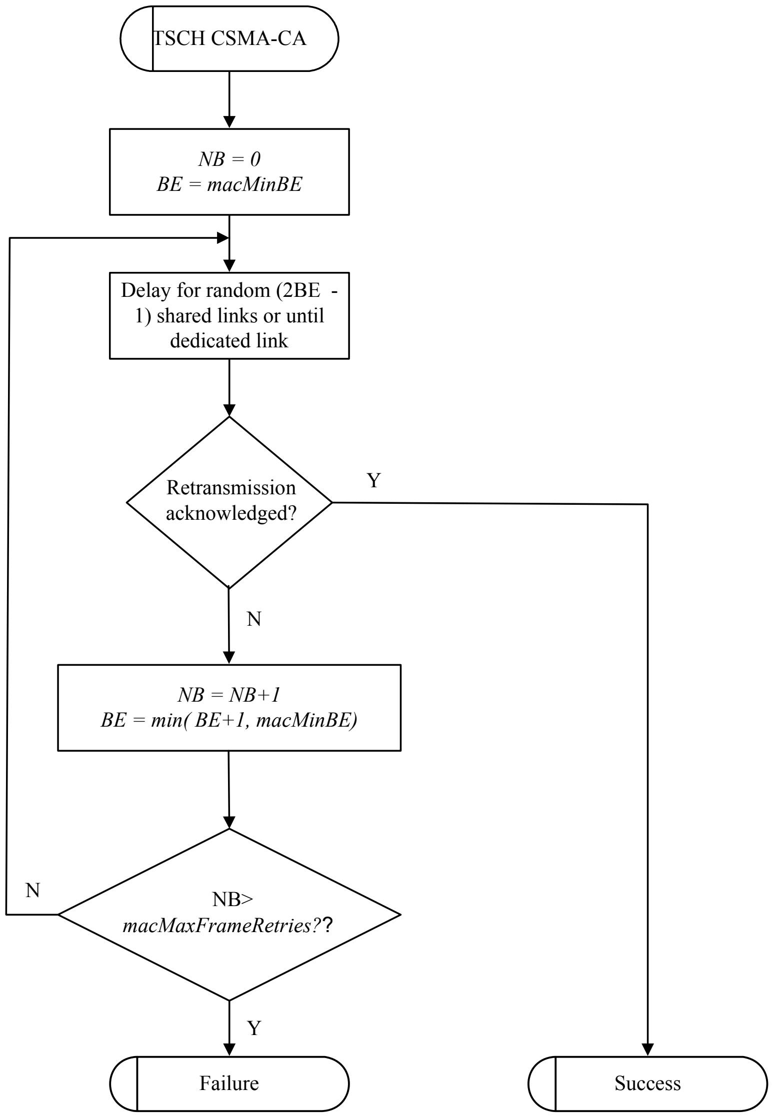

Figure 11b—TSCH CSMA-CA backoff

5.1.1.5 TSCH Slotframe structure

5.1.1.5.1 General

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

5.1.1.5.2 Absolute Slot Number (ASN)

The total number of timeslots that has elapsed since the start of the network or an arbitrary start time determined by the PAN coordinator is called the Absolute Slot Number (ASN). It increments globally in the network every *macTsTimeslotLength* µs (refer to Table 52e). It may be beaconed by devices already in a TSCH PAN, allowing new devices to synchronize. It is used globally by devices in a TSCH PAN as the frame counter (thus allowing for time-dependent security) and is used to compute the channel for any given pairwise communication as described in 5.1.1.5.3.

5.1.1.5.3 Links

Figure 11c shows how nodes may communicate in a sample three-timeslot slotframe. Nodes A and B communicate during timeslot 0, nodes B and C communicate during timeslot 1, and timeslot 2 is not being used. Every three timeslots, the schedule repeats, but note that ASN increments continuously. The pairwise assignment of a directed communication between devices in a given timeslot on a given channelOffset is a link. Physical channel, CH, in a link is made according to the following formula:

CH = *macHoppingSequenceList* [(*macASN* + channelOffset) % *macHoppingSequenceLength*]

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

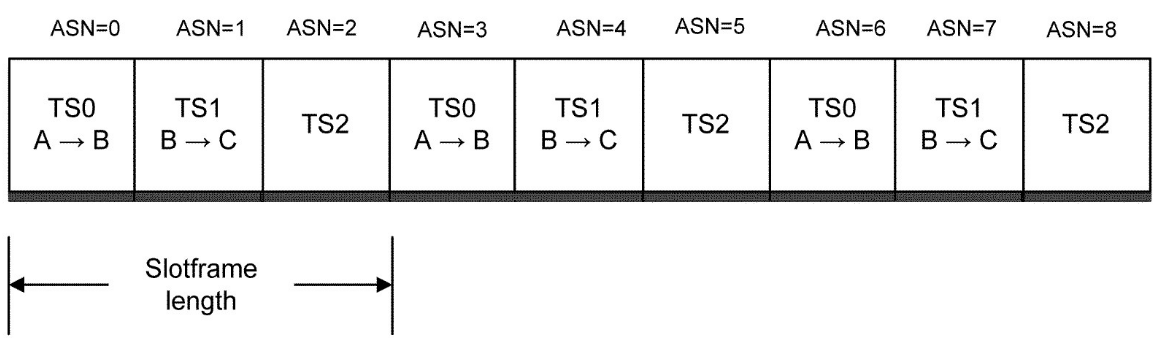

Figure 11c—Example of a three time-slot slotframe

5.1.1.5.4 Multiple slotframes

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

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

Slotframes can be added, removed, and modified while the network is running. Even though this is the case, all slotframes are aligned to timeslot boundaries, and timeslot 0 of the first repetition of every slotframe is projected back to *macASN* = 0, which is determined by the PAN coordinator (or other network device that starts the network). Because of this, timeslots in different slotframes are always aligned, even though the beginning and end of a particular repetition of that slotframe may not be as illustrated in Figure 11d. When, for any given timeslot, a device has links in multiple slotframes, transmissions take precedence over receives, and lower slotframeHandle slotframes takes precedence over higher slotframeHandle slotframes.

	ASN=0	ASN=1	ASN=2	ASN=3	ASN=4	ASN=5	ASN=6	ASN=7	
Slotframe 1 5 slots	TS 0	TS 1	TS 2	TS 3	TS 4	TS 0	TS 1	TS 2	...
Slotframe 2 3 slots	TS 0	TS 1	TS 2	TS 0	TS 1	TS 2	TS 0	TS 1	...

Figure 11d—Multiple slotframes in the network

5.1.1.6 LLDN Superframe structure

5.1.1.6.1 General structure of superframe

The LLDN superframe is divided into a beacon slot, management timeslots if present, and *macLLDNnumTimeSlots* base timeslots of equal length as illustrated in Figure 11e.

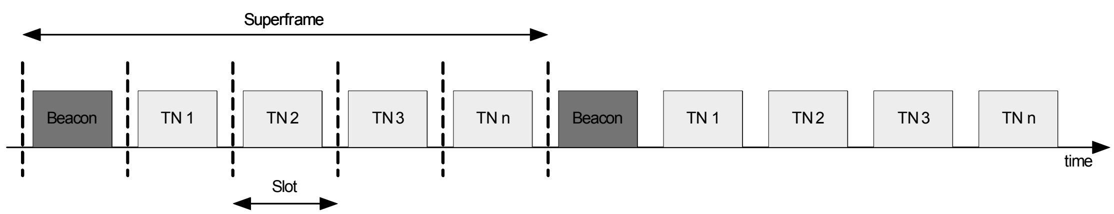

Figure 11e—Superframe with dedicated timeslots

The first timeslot of each superframe contains a beacon frame. The beacon frame is used for synchronization with the superframe structure. It is also used for re-synchronization of devices that went into power save or sleep mode.

The remaining timeslots are assigned to specific devices of the network. Each timeslot may have assigned a so-called slot owner. The slot owner has access privileges in the timeslot (dedicated timeslot). There is no explicit addressing necessary inside the frames if the slot owner transmits in its timeslot. The determination of the sender is achieved through the number of the timeslot. More than one device can be assigned to a timeslot (shared group timeslot). The devices use a contention-based access method (modified CSMA-CA as specified in 5.1.1.4.4) and a simple addressing scheme with 8-bit addresses in shared group timeslots.

Multiple adjacent base timeslots can be concatenated to a single, larger timeslot, as illustrated in Figure 11f.

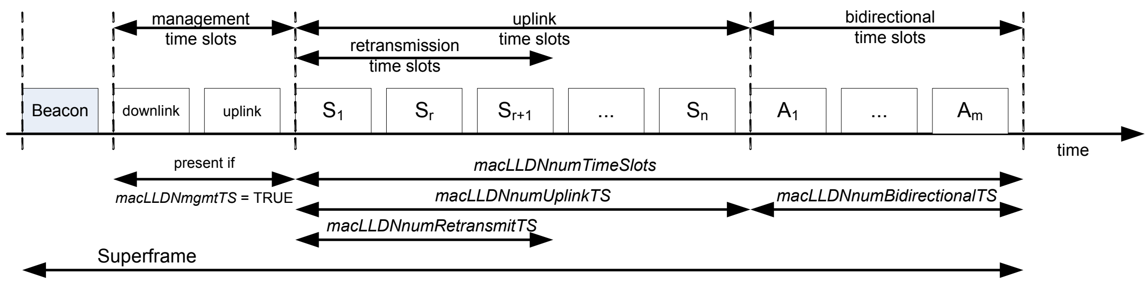

Figure 11f—Usage and order of slots in a superframe

As shown in Figure 11f, there is a specific order in the meaning or usage of the timeslots, as follows:

— Beacon Timeslot: always present.
— Management Timeslots: one timeslot downlink, one timeslot uplink, presence is configurable in *macLLDNmgmtTS* during the Configuration state.
— Uplink timeslots for LLDN devices: *macLLDNnumUplinkTS* timeslots uplink (unidirectional communication), *macLLDNnumRetransmitTS* timeslots at the beginning can be reserved for retransmissions according to the Group Acknowledgement field contained in the LL-beacon as described in 5.2.2.5.2 and 5.1.9.4.
— Bidirectional timeslots for LLDN devices: *macLLDNnumBidirectionalTS* timeslots uplink/downlink (bidirectional communication).

It is also possible to use a separate Group Acknowledgement (GACK) frame as described in 5.2.2.5.4 in order to facilitate retransmissions of failed transmissions in the uplink timeslots within the same superframe. The use of a separate GACK is configurable during configuration mode. If the use of a separate GACK is configured, the structure of the superframe is as depicted in Figure 11g.

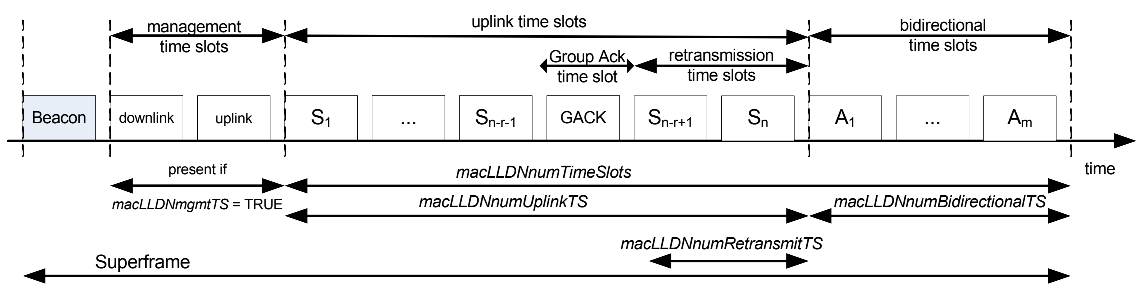

Figure 11g—Usage and order of slots in a superframe with configured use of separate GACK

Descriptions of the following configuration parameters and intervals for the superframe with a separate GACK are only different for the Uplink Timeslots:

— Beacon Timeslot
— Management Timeslots
— Uplink Timeslots: *macLLDNnumUplinkTS* denotes the total number of timeslots available for uplink (unidirectional) communication. Typically, one timeslot is allocated to each LLDN device. In this case, M denotes the number of LLDN devices, *macLLDNnumRetransmitTS* denotes the number of timeslots allocated for LLDN devices that failed their original transmissions prior to the GACK and

need to retransmit their message, and N denotes the number of LLDN devices that are allowed to retransmit. One timeslot is allocated for each retransmitting LLDN device.

— GACK: It contains an M bit bitmap to indicate successful and failed uplink transmissions in the same order as the uplink transmissions.

— Bidirectional Timeslots

The LL Beacon frame in the LLDN mode always carries the GACK bitmap even if a separate GACK frame is used. The GACK bitmap is used for acknowledging the successful retransmissions in timeslots R1, R2, ..., RN since some of the retransmitted frames (in R1, R2, …, RN timeslots) may fail.

5.1.1.6.2 Beacon timeslot

The beacon timeslot is reserved for the LLDN PAN coordinator to indicate the start of a superframe with the transmission of a beacon. The beacon is used to synchronize the devices and to indicate the current transmission mode. The beacon also contains acknowledgments for the data transmitted in the last superframe.

The beacon timeslot is available in every superframe.

5.1.1.6.3 Management timeslots

The first portion of a superframe after the beacon timeslot is formed by the management timeslots, i.e., the downlink and uplink management timeslots.

The downlink direction is defined as sending data to the LLDN device. The uplink direction is defined as sending data from the LLDN device to the LLDN Coordinator.

Management timeslots provide a mechanism for bidirectional transmission of management data in downlink and uplink direction. Downlink and uplink timeslots are provided in equal number in a superframe. There are two management timeslots per superframe at maximum. Management timeslots are implemented as shared group access timeslots.

Management downlink and uplink timeslots are used in the Discovery state and the Configuration state and are optional in the Online state. These states are described in 5.1.9.

5.1.1.6.4 Uplink timeslots

After the management timeslots, timeslots for the transmission of data are contained in a superframe. Uplink timeslots allow for unidirectional communication (uplink) only.

The first *macLLDNnumRetransmitTS* of the *macLLDNnumUplinkTS* uplink timeslots are dedicated timeslots for retransmissions of failed uplink transmission attempts in dedicated timeslots of the previous superframe. The dynamic assignment of nodes to retransmission timeslots is described in 5.1.9.4.

5.1.1.6.5 Bidirectional timeslots

Bidirectional timeslots allow for bidirectional communication between the LLDN PAN coordinator and the LLDN device. The direction of the communication is signaled in the beacon as described in 5.2.2.5.2. Bidirectional timeslots are used for the transmission of device data to the LLDN PAN coordinator (uplink) as well as of data from the LLDN PAN coordinator to the LLDN device (downlink).

5.1.1.6.6 Channel access within timeslots

Each timeslot is described by four time attributes as illustrated in Figure 11h and described in Table 0a.

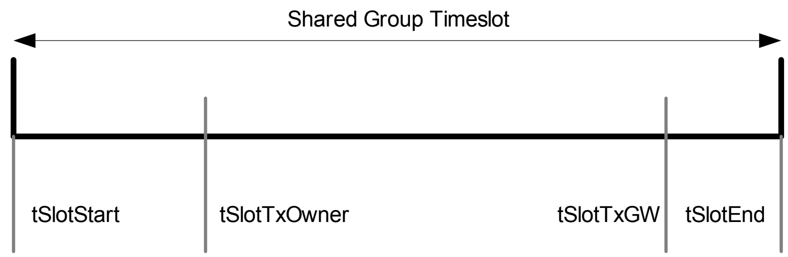

Figure 11h—Time attributes of timeslots

Table 0a—Time attributes of timeslots

Attribute	Description
tSlotStart	Starting time of timeslot
tSlotTxOwner	End time of privileged access by device that owns the timeslot
tSlotTxGW	If timeslot is unused, LLDN PAN coordinator can use the timeslot
tSlotEnd	End time of timeslot

From tSlotStart till tSlotTxOwner, the device that owns the slot, the slot owner, has exclusive access to the timeslot.

From tSlotTxOwner till tSlotTxGW, any device other than the LLDN PAN Coordinator may use the timeslot for data transmission with a modified CSMA-CA access scheme as described in 5.1.1.4.4 if the timeslot is not used by the slot owner. If the timeslot is not used by the slot owner, the LLDN PAN Coordinator shall indicate this by broadcasting a Clear To Send (CTS) Shared Group frame (5.3.10.4). To reduce the chances of collisions from other LLDN devices trying to use this timeslot, an LLDN device should send a Request To Send (RTS) frame (5.3.10.5) and wait for the receipt of the corresponding CTS frame (5.3.10.6) that identifies this LLDN device, from the LLDN PAN Coordinator before it transmits its data with a modified CSMA-CA access scheme as described in 5.1.1.4.4.

From tSlotTxGW till tSlotEnd, the LLDN PAN coordinator may use the timeslot, if the timeslot is still unused.

Dedicated timeslots are reserved for a single device (slot owner). This is achieved by setting tSlotTxOwner and tSlotTxGW to tSlotEnd. A dedicated timeslot allows the transmission of exactly one packet. Dedicated timeslots are only used during online mode as described in 5.1.9.4.

Shared group timeslots with contention-based access for every allowed device can be achieved by setting tSlotTxOwner to tSlotStart.

5.1.1.7 LE Functional description

This subclause specifies functionalities of devices supporting the following PIB attributes:

— *macCSLPeriod*
— *macRITPeriod*
— *macCSLMaxPeriod*
— *macLowEnergySuperframeSupported*
— *macLowEnergySuperframeSyncInterval*

5.1.1.7.1 LE Contention access period (LE CAP)

When *macCSLPeriod* is nonzero, CSL is deployed in CAP. CSL behavior is defined in 5.1.11.1. The *macRITPeriod* shall be set to zero in a beacon-enabled PAN.

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

5.1.1.7.2 LE Superframe structure

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

5.1.1.7.3 LE-incoming and outgoing superframe timing

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

5.1.1.7.4 LE scan

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

5.1.2 Starting and maintaining PANs

5.1.2.1 Scanning through the channels

Change the first paragraph as follows:

All devices shall be capable of performing passive and orphan scans across a specified list of channels. In addition, an FFD shall be able to perform energy detection (ED) and active scans. <u>Optionally an FFD may be able to perform an enhanced active scan.</u> The next higher layer should submit a scan request for a particular channel page containing a list of channels chosen only from the channels specified by *phyChannelsSupported* for that particular channel page.

5.1.2.1.2 Active and passive channel scan

Change the fourth paragraph as follows:

A~~n active or~~ passive scan or an active scan or an enhanced active scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate a~~n active or~~ passive scan, an active scan or enhanced active scan respectively. For each channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly. For an active scan, the device shall send a beacon request command, as described in 5.3.7. For UWB and CSS PHYs, the scan process shall be repeated for each mandatory preamble code, setting the *phyCurrentCode* appropriately. For an enhanced active scan the device shall send an enhanced beacon request as described in 5.3.7.2. Upon successful transmission of the beacon request command for an active scan or after switching to the channel for a passive scan, the device shall enable its receiver for at most [*aBaseSuperframeDuration* $\times (2^n + 1)$], where n is the value of the ScanDuration parameter. During this time, the device shall reject all nonbeacon frames and record the information contained in all unique beacons in a PAN descriptor structure, as described in Table 17, including the channel information and, if required, the preamble code.

Insert the following new text, table, and figure in 5.1.2.1.2 after the eighth paragraph:

If a coordinator capable of responding to the enhanced beacon request (EBR) command and capable of filtering as described in 5.3.7.2.2 receives an EBR command, it shall perform the filtering as indicated in the EBR. If the filter conditions are satisfied, it shall transmit the appropriate enhanced beacon as per Table 0b.

Table 0b—Channel access for EBR response

Mode of operation	Response required	Access method	When to respond	Notes
Beacon PAN	Y	Slotted CSMA-CA	Next available CAP	
Nonbeacon PAN	Y	Unslotted CSMA-CA	ASAP	E.G., unslotted hopping
DSME beacon	Y	Slotted CSMA-CA	Next available EB	
DSME non-beacon	Y	Unslotted CSMA-CA	ASAP	
TSCH	Y	Slotted CSMA-CA	Next available time slot on same channel	

The messages exchanged when using an Enhanced Active Scan with an Enhanced Beacon Request are shown in Figure 13a.

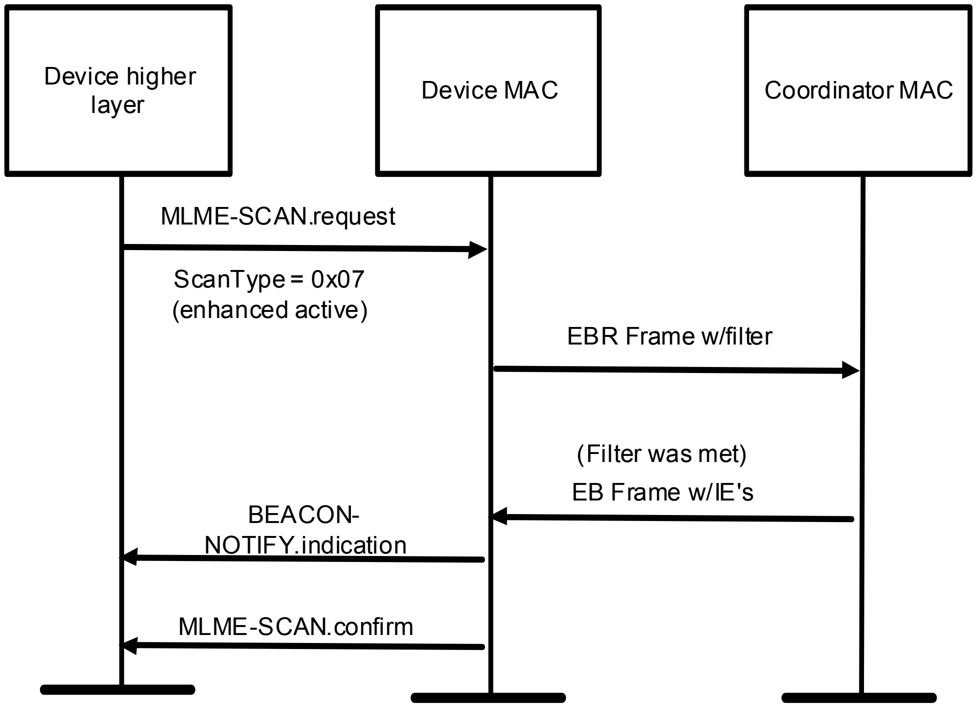

Figure 13a—Enhanced Active scan with Enhanced Beacon request

Insert before 5.1.3 (after 5.1.2.5) the following subclause and figure:

5.1.2.6 TSCH PAN formation

A TSCH PAN is formed when a device, usually the PAN coordinator, advertises the presence of the network by sending Enhanced Beacons upon receipt of a MLME BEACON.request from a higher layer. The Enhanced Beacons contain the following:

— Time information so new devices can synchronize to the network (as described in 5.2.4.13).
— Channel hopping information (as described in 5.1.1.6).
— Timeslot information describing when to expect a frame to be transmitted and when to send an acknowledgment (as described in 5.2.4.15)
— Initial link and slotframe information so new devices know when to listen for transmissions from the advertising device and when they can transmit to the advertising device (as described in 5.2.4.14).

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

At this point the device is synchronized to the network and may optionally send in an Association Request. If the device uses Association, it may request a 2-octet address. The sequence of messages exchanged to synchronize a device to the networks is shown in Figure 16a, and the process of synchronization is described in 5.1.4.2a.

Typically at this point the device will go through a procedure to allocate additional communication resources (slotframes and links) to the joining device. This procedure may include a security handshake to mutually authenticate the joining device, configure encryption keys, and configure routing information. The mechanism and rules for setting up these additional communication links and configure other policies would normally be defined in a higher layer standard—the content of these messages is beyond the scope of this document.

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

After joining, the device may receive additional slotframes and links from a higher layer management entity or peer as required by the application, or the device may be instructed to remove slotframes and links obtained from the enhanced beacon.

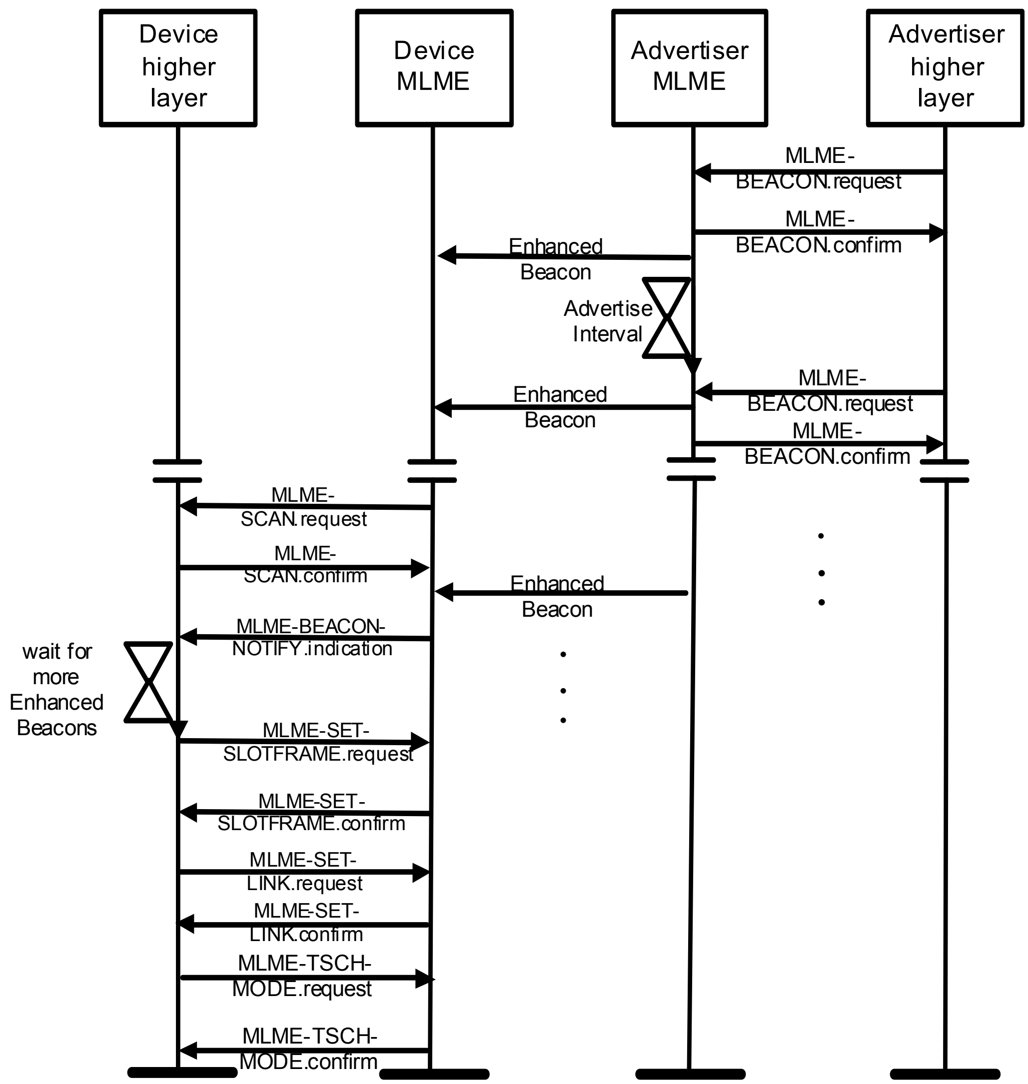

Figure 16a—Message sequence chart for TSCH procedure to find an advertising device

5.1.3 Association and disassociation

5.1.3.1 Association

Insert at end of 5.1.3.1 the following paragraph:

Association is optional for devices operating in TSCH mode.

5.1.3.2 Disassociation

Insert at end of 5.1.3.2 the following paragraphs:

For devices using the optional TSCH mode, additional disassociation behavior is required. A device shall only disassociate from the PAN if it receives a disassociation notification command from either the PAN coordinator, or all of its timekeeping neighbors, i.e., neighbors to whom the device has receive links marked with the Timekeeping bit set in the *maclinkOptions*.

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

Insert the following subclause 5.1.3.3 before 5.1.4:

5.1.3.3 Fast association (FastA)

Fast association (FastA) is optional and is not defined in IEEE 802.15.4 devices.

A device is instructed to Fast associate with a PAN through the MLME ASSOCIATE.request primitive with the Association Type field of the Capability Information field set to one.

The MAC sublayer of an unassociated device shall initiate the FastA procedure by sending an association request command with the Association Type field of the Capability Information field set to one to the coordinator of an existing PAN. Because the association request command contains an acknowledgment request, the coordinator shall confirm its receipt by sending an acknowledgment frame.

On receipt of the acknowledgment to the association request command, the device shall wait for at most *macResponseWaitTime* symbols for the coordinator to make its association decision.

If the coordinator has sufficient resources, the next higher layer should allocate a 16-bit short address to the device, and the MAC sublayer shall generate an association response command containing the new address and a status indicating a successful FastA. If the Association Type field of the Capability Information field of the association request command is set to one, the MAC sublayer of the coordinator shall send the association response command to the device directly as described in Figure 60a. If the Association Type field of the Capability Information field of the association request command is set to zero, the association response command shall be sent as 5.1.3.1.

Because the association response command contains an acknowledgment request, the device requesting FastA shall confirm its receipt by sending an acknowledgment frame. If the Association Status field of the command indicates “FastA successful” (0x80), the device shall store the address contained in the 16-bit Short Address field of the command in *macShortAddress* for its communication use in the PAN.

5.1.4 Synchronization

Insert before 5.1.4.1 the following paragraph:

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

5.1.4.2 Synchronization without beacons

Insert the following subclauses (5.1.4.2.a−5.1.4.2a.2) before 5.1.4.3:

5.1.4.2a Synchronization in TSCH PAN

In a TSCH PAN, all communication happens in timeslots as described in 5.1.4.2a.1. To remain synchronized, the devices should have the same notion of when each timeslot begins and ends, within ± *macTsRxWait*/2 μs. In a typical TSCH PAN, time propagates outwards from the PAN coordinator. A device shall periodically synchronize its network time to at least one other network device. It may also provide its network time to one or more network devices. A network device uses a neighbor as a time source when it has received links to that neighbor with the Timekeeping bit set in the *maclinkOptions*. A higher layer may add or change time source neighbors at any time.

Note that a device sending Enhanced Beacons to advertise a TSCH PAN should set the Timekeeping bit in the Link Option field (as described in 5.2.4.14) for the joining devices' receive link so that joining devices can maintain time synchronization until additional time source neighbors are configured by a higher layer.

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

Figure 22a shows an example of time propagation in a TSCH PAN. The arrows indicate the direction of time distribution. In this example, the PAN coordinator acts as the time source for the entire network. Device 1 synchronizes to the PAN coordinator only, and is the time source for device 3. Device 2 synchronizes its time to both 1 and the PAN coordinator, and device 4 synchronizes to the PAN coordinator, device 2, and device 3.

Synchronization is possible whenever a device exchanges a frame with a time source neighbor. This can either come from receiving an acknowledgment with time correction information (as described in 5.2.4.13) or from the arrival time of a frame from the time source neighbor. Both methods are described in 5.1.4.2a.2.

In order to ensure that it remains synchronized with the TSCH PAN, a network device shall ensure that it communicates with each of its timekeeping neighbors at least once per Keep Alive period. If a network device has not sent a packet to its time source neighbor within this interval, it shall send any empty acknowledged MAC frame and use the acknowledgement frame to perform acknowledgement based synchronization.

5.1.4.2a.1 Timeslot communication

During a timeslot in a slotframe, one node typically sends a frame, and another sends back an acknowledgment frame containing the ACK/NACK Time Correction IE as described in 5.2.4.11 if it successfully receives that frame and the frame passes authentication. The acknowledgment can be positive (ACK) or negative (NACK). A positive acknowledge indicates that the receiver has successfully received the frame and has taken ownership of it for further routing. A negative acknowledgment indicates that the receiver cannot accept the frame at this time, but has heard it with no errors. Both ACKs and NACKs carry timing information used by nodes to maintain network synchronization. Frames sent to a unicast node address require that a link-layer acknowledgment be sent in response during the same timeslot as shown in Figure 22b. If an acknowledgment is requested and not received within the time-out period, retransmission of the frame waits until the next assigned transmit timeslot to that address occurs.

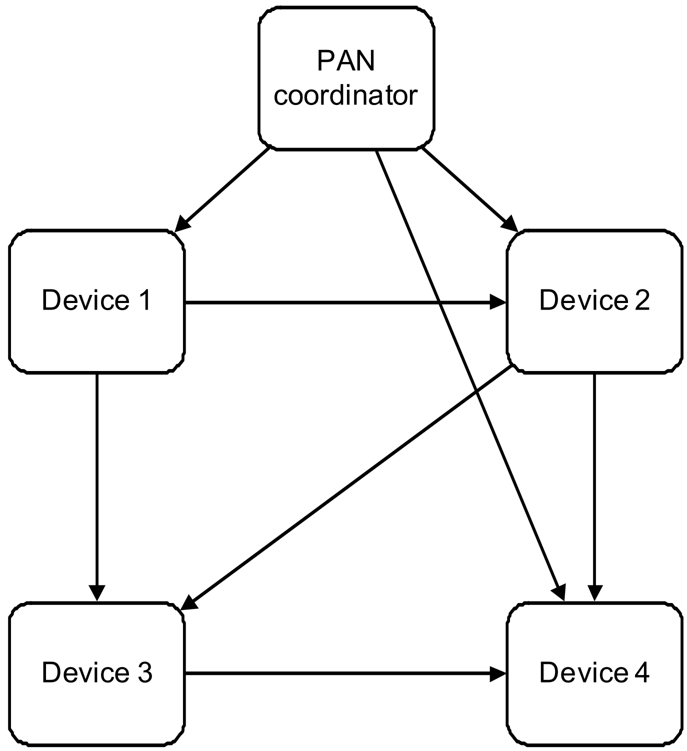

Figure 22a—Example of possible time propagation in a TSCH PAN

As shown in Figure 22b, the timeslot starts at time T = 0 from the transmitting device's perspective. The transmitter waits *macTsCCAOffset* µs, and then performs CCA (if active). At *macTsTxOffset* µs, the device begins transmitting the frame. If an acknowledgment is expected, the device waits *macTsRxAckDelay* µs, and then enables the receiver to await the acknowledgment. If the acknowledgment does not arrive within *macTsAckWait* µs the device may idle the radio and consider the transmission a failure. If no acknowledgment is expected, the transmitter may idle the radio after sending the frame.

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

The transmitter or receiver may resynchronize clocks as described in 5.1.4.2a.2.

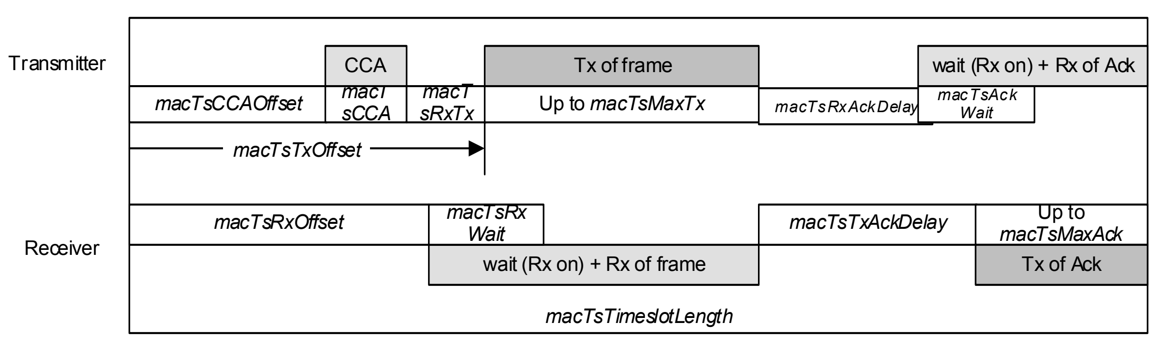

Figure 22b—Timeslot diagram of acknowledged transmission

5.1.4.2a.2 Node synchronization

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

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

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

— Originator sends a frame, timing the start symbol to be sent at *macTsTxOffset* according to its clock, which would correspond to *macTsRxOffset* + *macTsRxWait*/2 in the receiver's clock if both clocks were perfectly synchronized.

— Receiver records the timestamp (e.g., from the MCPS DATA.indication) when it receives the start symbol of the frame.

— Receiver calculates a time correction = *macTsRxOffset* + *macTsRxWait*/2 − arrival timestamp.

— Receiver sends back the time correction in the IE field in the corresponding enhanced acknowledgment frame (assuming the incoming frame passes validation).

— Originator receives the acknowledgment. If the receiver node is a time source neighbor, the originator adjusts its own clock by incorporating the difference into an average of the drift to all its time source neighbors. The averaging method is implementation dependent. If the receiver is not a clock source, the time correction is ignored.

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

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

— Receiver records the timestamp (e.g., from the MCPS DATA.indication) when it receives the start symbol of the frame.

— The receiver calculates a time correction = *macTsRxOffset* + *macTsRxWait*/2 − arrival timestamp.

— If the originator was a time source neighbor, the receiver adjusts its own clock by incorporating the difference into an average of the drift to all its time source neighbors. The averaging method is implementation dependent. If the receiver is not a clock source, the time correction is ignored.

Figure 22c illustrates both time synchronization mechanisms. In both cases, the receiver calculates its time adjustment to either send back to the transmitting device or to use locally.

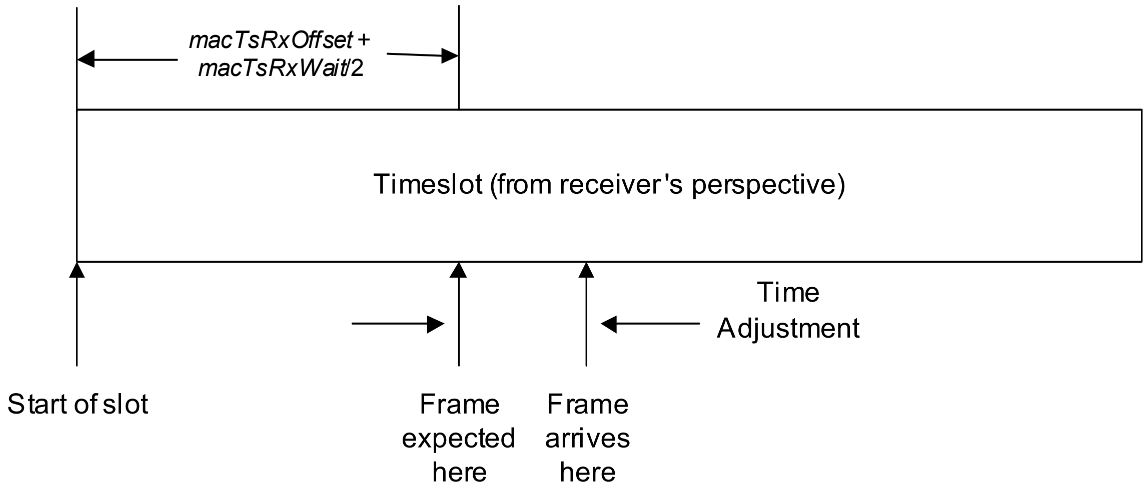

Figure 22c—Time synchronization

5.1.6 Transmission, reception, and acknowledgment

5.1.6.2 Reception and rejection

Change the sixth paragraph in 5.1.6.2 as indicated:

If any of the third-level filtering requirements are not satisfied, the MAC sublayer shall discard the incoming frame without processing it further. If all of the third-level filtering requirements are satisfied, the frame shall be considered valid and processed further. For valid frames that are not broadcast, if the Frame Type field indicates a data or MAC command frame with the Frame Version field set to 0b00-0b01 and the AR field is set to request an acknowledgment, the MAC sublayer shall send an acknowledgment frame. Prior to the transmission of the acknowledgment frame, the sequence number included in the received data or MAC command frame shall be copied into the Sequence Number field of the acknowledgment frame. This step will allow the transaction originator to know that it has received the appropriate acknowledgment frame.

Insert the following paragraph after the sixth paragraph in 5.1.6.2:

For valid frames that are not broadcast, if the Frame Type field indicates a multipurpose frame, or a data or MAC command frame with the Frame Version field set to 0b10, and the AR field is set to request an acknowledgment, the MAC sublayer shall send an enhanced acknowledgement as described in 5.1.6.4.2.

5.1.6.4 Use of acknowledgments and retransmissions

5.1.6.4.2 Acknowledgment

Insert in 5.1.6.4.2 after the second paragraph the following text:

For enhanced acknowledgment frames (which are only sent after the incoming frame passes security filtering, if applicable), additional time may be required after *aTurnaroundTime* to complete incoming and outgoing security processing. If the enhanced acknowledgment is not sent before *macEnhAckWaitDuration* μs, the sender will assume the frame was not successful.

The receiving device may include additional content in an enhanced acknowledgment encapsulated as IEs. If the originator does not understand the IE content of the acknowledgment, it is ignored, but the transmission is considered successful.

When returning Time Correction information in the acknowledgment (via the ACK/NACK Timing Correction IE as described in 5.2.4.11), the receiving device may acknowledge frames with an NACK to indicate that the frame could not be transferred to a higher layer. This is treated as a failure from the sender's perspective, and a notification of failure due to congestion is indicated to the higher layer through the MAC performance metrics.

LLDNs use several methods for the acknowledgment of data transmissions. The timings of these mechanisms are defined by the superframe structure of the LLDN. The transmission of an LL-Acknowledgment frame in response to an LL-data frame in an LLDN shall commence in the same bidirectional timeslot in the next superframe. The LL-Acknowledgment frame shall only be used with bidirectional timeslots.

Insert before 5.1.6.4.3 the following paragraph:

When in TSCH mode, incoming frames are acknowledged using the enhanced acknowledge frame as described in 5.2.2.3. Security of the acknowledgment frame should match that of the incoming frame. When operating in the TSCH mode (refer to 6.2.19.5), the enhanced acknowledgment frame is sent at the time specified by the *macTimeslotTemplate* being used (as described in 6.4.3.3.3 and 5.1.4.2a), i.e., *macEnhAckWaitDuration* should be set to a value corresponding to *macTsAckWait*.

5.1.6.4.3 Retransmissions

Change in 5.1.6.4.3 the second paragraph as follows:

A device that sends a data or MAC command frame with frame version 0b00-0b01 and with its AR field set to acknowledgment requested shall wait for at most *macAckWaitDuration* for the corresponding acknowledgment frame to be received. If an acknowledgment frame is received within *macAckWaitDuration* and contains the same DSN as the original transmission, the transmission is considered successful, and no further action regarding retransmission shall be taken by the device. If an acknowledgment is not received within *macAckWaitDuration* or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

Insert in 5.1.6.4.3 after the second paragraph the following paragraph:

A device that sends a data or MAC command frame with frame version 0b10 with its Acknowledgment Request field set to one shall wait for at most *macEnhAckWaitDuration* µs for the corresponding acknowledgment frame to be received, as additional security processing is required. If an acknowledgment frame is received within *macEnhAckWaitDuration* µs and contains the same DSN as the original transmission, the transmission is considered successful, and no further action regarding retransmission shall be taken by the device. If an acknowledgment is not received within *macEnhAckWaitDuration* µs or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

Insert before 5.1.6.5 the following subclause (5.1.6.4.4):

5.1.6.4.4 TSCH retransmissions

A device that sends a data or MAC command frame with its Acknowledgment Request field set to one shall wait for *macTsRxAckDelay* µs. If an acknowledgment frame begins within *macEnhAckWaitDuration* and contains the same DSN as the original transmission, and otherwise has no errors, the transmission is considered successful, and no further action regarding retransmission shall be taken by the device. If an acknowledgment is not received within the appropriate time-out or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

All transmissions in TSCH are direct. If a single transmission attempt has failed, the device shall repeat the process of transmitting the data or MAC command frame and waiting for the acknowledgment, up to a maximum of *macMaxFrameRetries* times. In TSCH mode (6.2.19.5), retransmissions only occur on subsequent transmit links to the same recipient on any slotframe assigned to the device. If an acknowledgment is still not received after *macMaxFrameRetries* retransmissions, the MAC sublayer shall assume the transmission has failed and notify the next higher layer of the failure.

5.1.6.6 Transmission scenarios

Replace in 5.1.6.6 each occurrence of "macAckWaitDuration" with "macAckWaitDuration or macEnhAckWaitDuration as appropriate."

Insert before 5.2 the following subclauses (5.1.9−5.1.12.5):

5.1.9 LLDN transmission states

5.1.9.1 General

The transitions between the different transmission states are illustrated in Figure 34a.

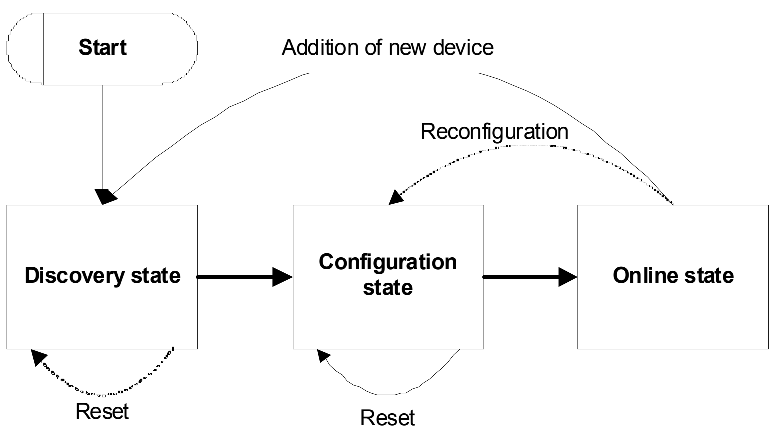

Figure 34a—Transitions between transmission states

The discovery state is the first step during network setup: the new devices are discovered and configured in the second step, the configuration state. After the successful completion of the configuration state, the network can go into online state. Data and readings from the devices can only be transmitted during online state. In order to reconfigure a network, the configuration state can be started again.

5.1.9.2 Discovery state

The Discovery state is the first step during network setup or for the addition of new devices to an existing network.

In the Discovery state, the superframe contains only the timeslot for the beacon described in 5.1.1.6.2 and two management timeslots, one downlink and one uplink (5.1.1.6.3).

A new device scans the different channels until it detects an LLDN PAN coordinator sending beacons that indicate Discovery state.

If a new device received a beacon indicating Discovery state, it attempts to access the medium in the uplink management timeslot in accordance with 5.1.1.4.4 in order to send a Discover Response frame to the LLDN PAN coordinator. The Discover Response frame is described in 5.3.10.1. The Discover Response frame contains the current configuration of the device. The new device shall repeat sending the Discover Response frame until it receives an acknowledgment frame for it or the Discovery state is stopped by the LLDN PAN coordinator. The acknowledgment frame is described in 5.2.2.5.4.

The LLDN coordinator changes from the Discovery state to the Configuration state if it did not receive any Discover Response frames within *macLLDNdiscoveryModeTimeout* seconds.

Figure 34b illustrates the Discovery state.

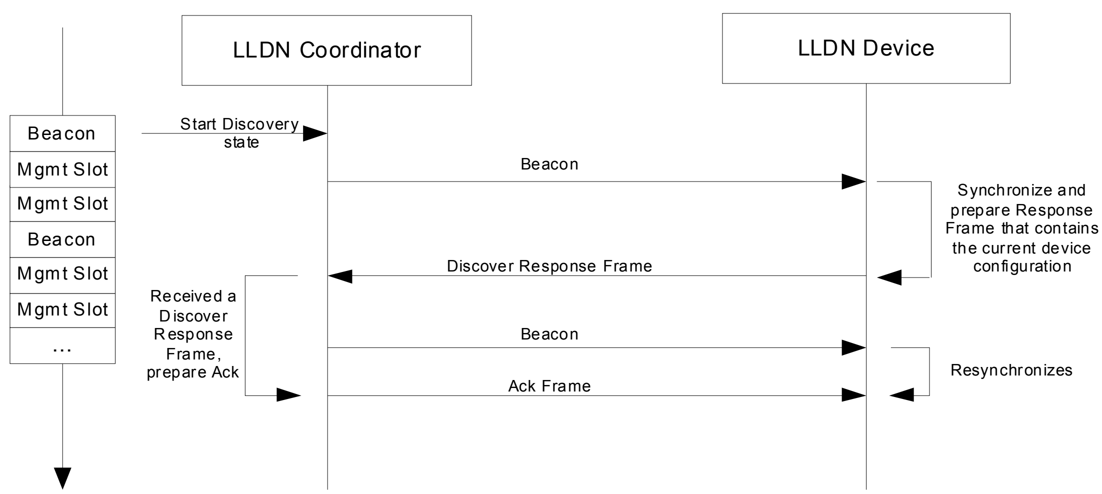

Figure 34b—Flow diagram of Discovery state

5.1.9.3 Configuration state

The Configuration state is the second step during network setup. It is also used for network reconfiguration.

In the Configuration state, the superframe contains only the timeslot for the beacon described in 5.1.1.6.2 and two management timeslots, one downlink and one uplink as described in 5.1.1.6.3.

If a device received a beacon indicating configuration state, it tries to get access to the transmission medium in the uplink management timeslot in order to send a Configuration Status frame to the LLDN PAN coordinator. The Configuration Status frame is described in 5.3.10.2. The Configuration Status frame contains the current configuration of the device. The new device shall repeat sending the Configuration Status frame until it receives a Configuration Request frame for it or the Configuration state is stopped by the LLDN PAN coordinator. The Configuration Request frame is described in 5.3.10.3. The Configuration Request frame contains the new configuration for the receiving device. After successfully receiving the

Configuration Request frame, the device sends an acknowledgment frame to the LLDN PAN coordinator. The acknowledgment frame is described in 5.2.2.5.4.

Figure 34c illustrates the Configuration state.

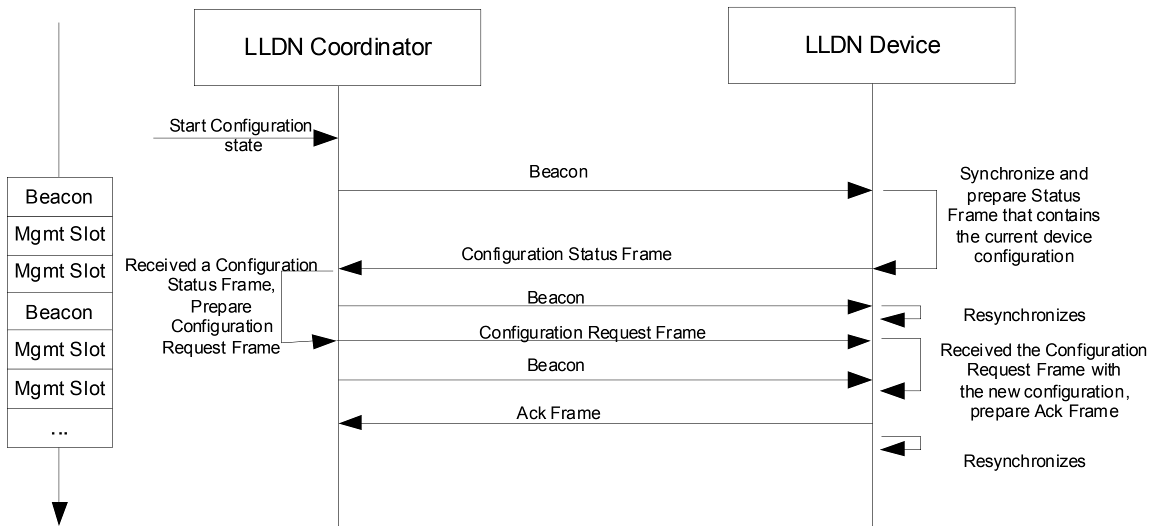

Figure 34c—Flow diagram of Configuration state

5.1.9.4 Online state

User data is only sent during Online state. The superframe starts with a beacon and is followed by several timeslots. The devices can send their data during the timeslots assigned to them during the Configuration state. The different types of timeslots are described in 5.1.1.6.

The existence and length of management timeslots in the Online state are contained in the Configuration Request frame.

The successful reception of data frames by the LLDN PAN coordinator is acknowledged in the Group Acknowledgment bitmap of the beacon frame of the next superframe described in 5.2.2.5.2 or in a separate Data Group Acknowledgment frame depicted in Figure 48h. This is the case for both uplink timeslots and bidirectional timeslots if the transmission direction is uplink. Figure 34d illustrates an example of the Online state for uplink transmissions. In this example, the network has three dedicated timeslots, and LLDN device 2 is assigned to timeslot 2.

If retransmission timeslots are configured (i.e., *macLLDNnumRetransmitTS* > 0), the retransmission slots are assigned to the owners of the first *macLLDNnumRetransmitTS* with the corresponding bit in the group acknowledgment bitmap set to zero. Each LLDN device shall execute the algorithm as illustrated in Figure 34e in order to determine its retransmission timeslot. The LLDN PAN coordinator has to execute a similar algorithm in order to determine the senders of the frames in the retransmission slots.

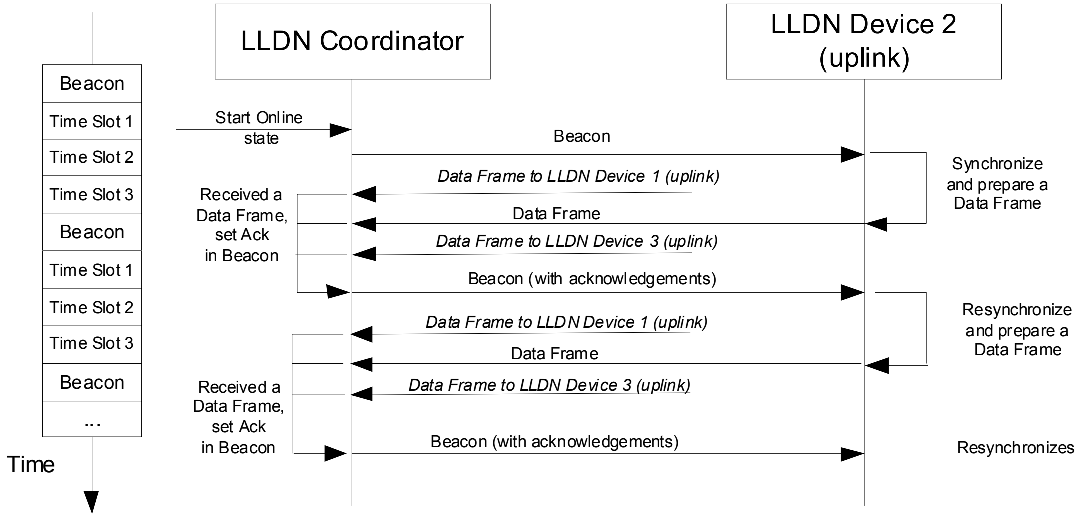

Figure 34d—Flow diagram of Online state for LLDN devices (uplink)

Ack[i] represents the uplink success and maps to the bit b(i−1) in the group acknowledgment bitmap as illustrated in Figure 48e. Assuming that the LLDN device has been assigned to uplink timeslottimeslot “s,” Ack[s] represents the uplink success of that LLDN device.

If the data transmission of the LLDN device has failed and has not been acknowledged, that is, ack[s] is zero (i.e., false), the LLDN device determines the number of failed transmissions in previous timeslots excluding retransmission timeslots. This number of failed transmissions, NFT, is the number of ack[i] equal to 0 (i.e., false) with (*macLLDNnumRetransmitTS*+1) ≤ i ≤ (s−1).

A retransmission is possible if the number of failed transmissions NFT is less than *macLLDNnumRetransmitT*S. The LLDN device retransmits its data in retransmission timeslot (NFT+1).

If the number of failed transmissions NFT is equal or greater than *macLLDNnumRetransmitTS*, a retransmission is not possible.

The successful reception of data frames by LLDN devices assigned to bidirectional timeslots (transmission direction is downlink) is acknowledged by an explicit acknowledgment frame by the corresponding LLDN devices in the following superframe. This means that after setting the Transmission Direction bit in the beacon described in 5.2.2.5 to downlink and sending a data frame to one or more LLDN devices, the LLDN PAN coordinator shall set the Transmission Direction bit to uplink in the directly following superframe. LLDN devices assigned to bidirectional timeslots that have successfully received a data frame from the LLDN PAN coordinator during the previous superframe shall send an acknowledgment frame to the LLDN PAN coordinator. LLDN devices that did not receive a data frame from the LLDN PAN coordinator may send data frames to the LLDN PAN coordinator during this superframe with Transmission Direction bit set to uplink. Figure 34f illustrates the Online state with LLDN devices assigned to bidirectional timeslots. In this figure, the network has three dedicated bidirectional timeslots, and LLDN device 2 is assigned to timeslot 2.

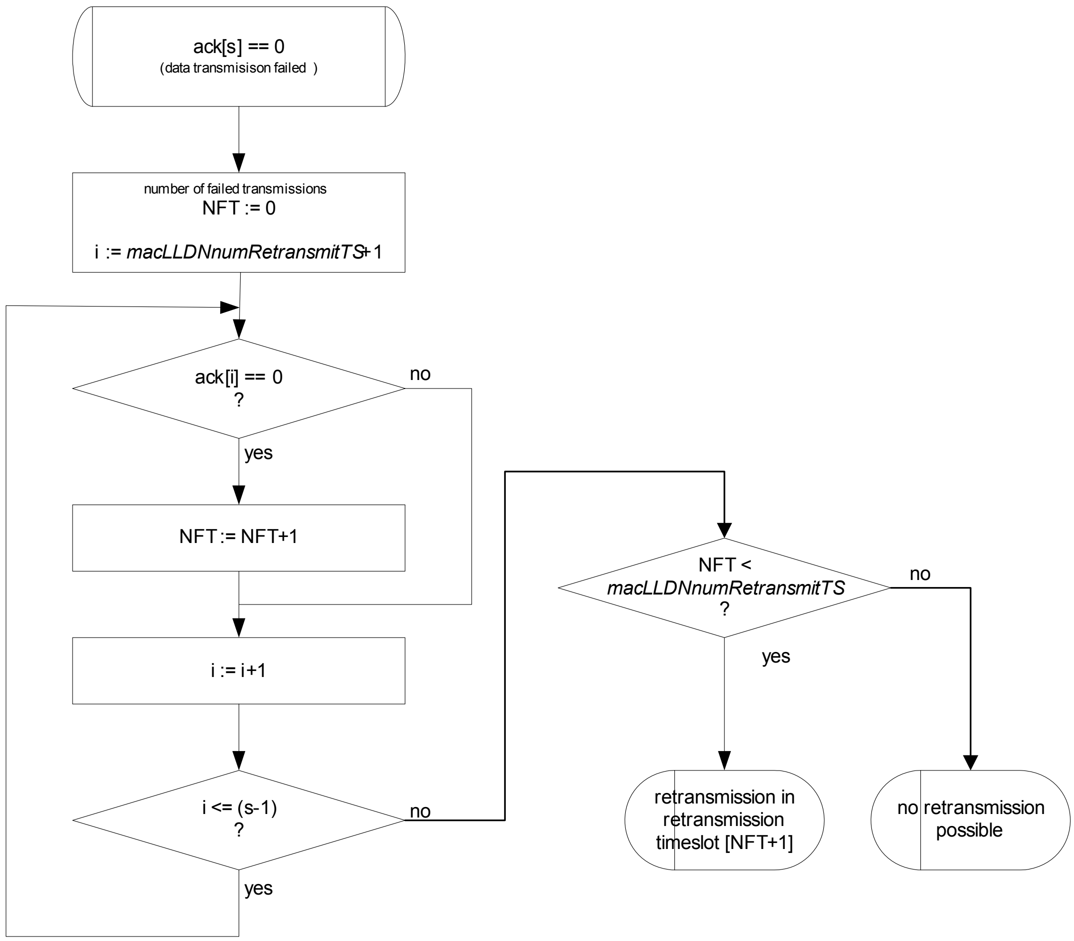

Figure 34e—Retransmission Slot Algorithm

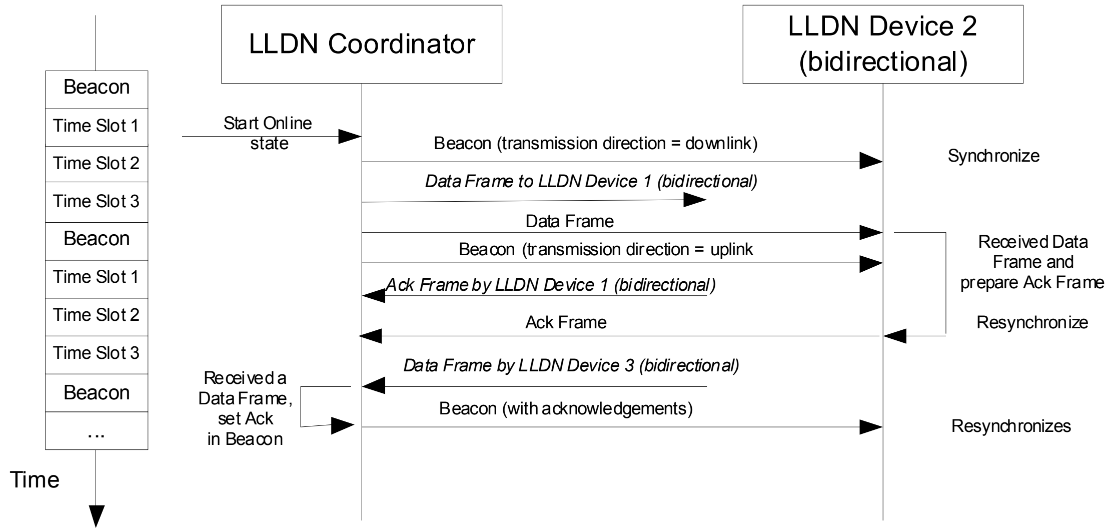

Figure 34f—Flow diagram of Online state for LLDN devices (bidirectional)

5.1.10 Deterministic and synchronous multi-channel extension (DSME)

5.1.10.1 DSME multi superframe structure

A coordinator in a DSME enabled PAN shall periodically transmit an enhanced beacon (illustrated in Figure 40a) with DSME PAN Descriptor IE as described in 5.2.4.9 to coordinate a DSME multi-superframe structure. A multi-superframe is a cycle of repeated superframes. A superframe consists of a beacon frame, a CAP, and a CFP.

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

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

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

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

Enhanced beacons shall be transmitted, without the use of CSMA-CA, at the start of slot 0 if *macDeferredBeaconUsed* is FALSE. The start of slot 0 is defined as the point at which the first symbol of the beacon PPDU is transmitted. If *macDeferredBeaconUsed* is TRUE, enhanced beacons shall be transmitted following the procedure described in 5.1.10.8.

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

An example of a multi-superframe structure is shown in Figure 34g.

The number of superframes in a multi -superframe: $N = 2^{(MO-SO)}$

The number of multi-superframes in a beacon interval: $N = 2^{(BO-MO)}$

Superframe duration: $SD = aBaseSuperframeDuration * 2^{SO}$ *Symbols*

Multi-superframe duration: $MD = aBaseSuperframeDuration * 2^{MO}$ *Symbols*

Beacon interval: $BI = aBaseSuperframeDuration * 2^{BO}$ *symbols*

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

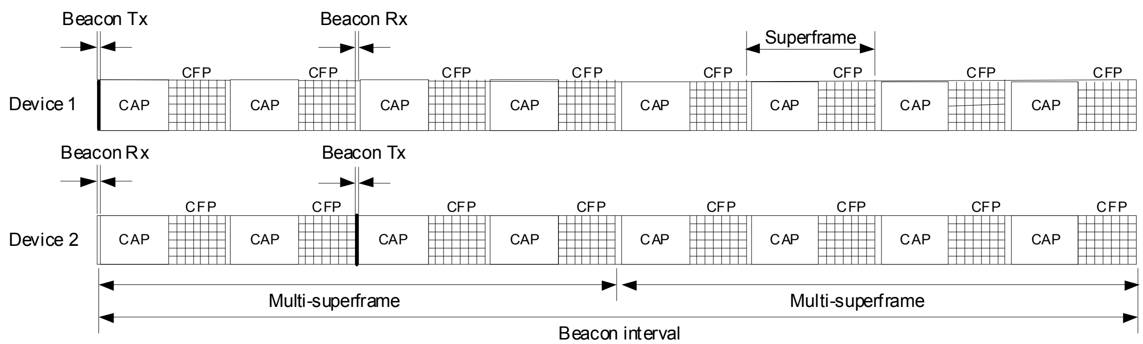

Figure 34g—DSME multi-superframe structure

5.1.10.2 Channel diversity

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

5.1.10.2.1 Channel adaptation

In Channel Adaptation (i.e., ChannelDiversityMode field of the DSME PAN Descriptor IE in an enhanced beacon is set to zero), the Source device may allocate DSME-GTSs in a single channel or in different channels to a Destination device based on the knowledge of current channel quality. If DSME-GTSs in different slots in different channels are successfully allocated for a pair of a Source device and a Destination device, the Source device shall transmit data frames according to the scheduled timeslots and channels specified in *macDSMEACT*.

An example of the schedule of channels and DSME-GTSs in Channel Adaptation is illustrated in Figure 34h. In this example, the device uses channel 11 from slot 0 to 2, and then it switches to channel 13 on slot 5.

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

5.1.10.2.2 Channel hopping

In channel hopping (i.e., ChannelDiversityMode field of the DSME PAN Descriptor IE in an enhanced beacon is set to one), each DSME-GTS hops over predefined frequency channels to receive. The series of channels used at each DSME-GTS is referred to as a *Hopping Sequence*. The same Hopping Sequence shall be repeated over whole DSME-GTSs.

An example of the schedule of channels and DSME-GTSs in Channel Hopping is illustrated in Figure 34i.

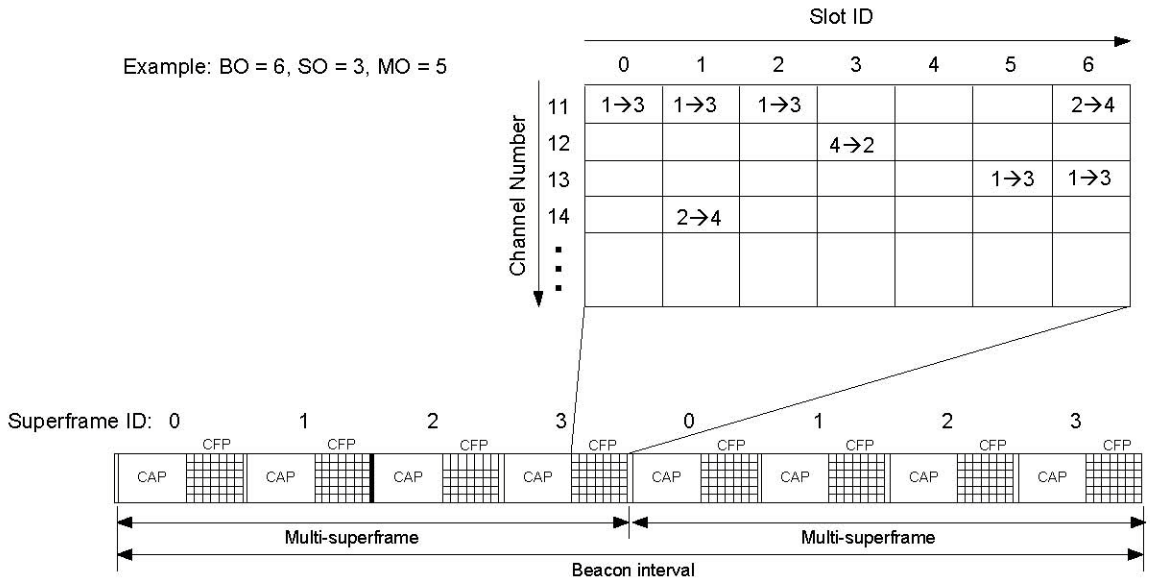

Figure 34h—Channel usage of DSME-GTSs in Channel Adaptation

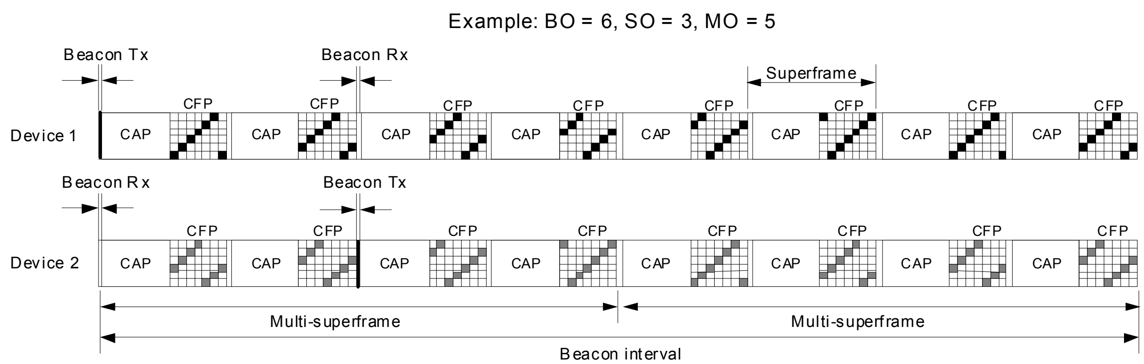

Figure 34i—Channel usage of DSME-GTSs in Channel Hopping

In this example, Hopping Sequence is {1, 2, 3, 4, 5, 6} and the Channel Offset values of two devices are 0 and 2 respectively. For the device with Channel Offset value of 0, DSME-GTSs (timeslot, channel) for this device are (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 1), (8, 2), (9, 3), and so on. Similarly, for the device with Channel Offset value of 2, DSME slots are given as (1, 3), (2, 4), (3, 5), (4, 6), (5, 1), (6, 2), and so on. The transmitting device shall switch to the channel used by the receiving device in order to send a data frame. If the receiving device receives the data frame successfully, it sends an ACK frame to the transmitting device on the same channel.

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

$$C(i) = macHoppingSequenceList[(j \times l + i + macChannelOffset + macPANCoordinatorBSN) \% macHoppingSequenceLength]$$

where l is 15 if *macCAPReductionFlag* is TRUE and SDIndex j is not zero, and 7 otherwise, *macHoppingSequenceList*[*m*] represents the $(m)^{th}$ channel number in *macHoppingSequenceList*, *macChanneloffset* is the channel offset value of the receiver device, *macHoppingSequenceLength* is the length of Hopping Sequence and *macPANCoordinatorBSN* is an enhanced beacon sequence number of a PAN coordinator.

5.1.10.3 Group ACK

In many application systems, it may be imperative to provide the DSME nodes with a retransmission opportunity, within the same multi-superframe, for a data frame that failed in its GTS transmission. To satisfy that crucial requirement, the MAC sublayer shall provide an optional feature of group acknowledgment whereby:

— multiple DSME-GTS frames received by the coordinator in the CFP shall be acknowledged by a single transmission of GACK frame by the receiver (i.e., the coordinator); and

— new DSME-GTS timeslots shall be allowed to be allocated to some of the transmitting DSME nodes for retransmission of their failed GTS transmission or for transmitting additional data frames.

If the value of *macGACKFlag* is TRUE, the group ACK shall be enabled. In a network that allows the use of the Group ACK, nodes shall use appropriate flags and fields in the enhanced beacon frame, data frame header, and the GACK frame in order to facilitate the use of the GACK feature. A coordinator shall decide if the GACK feature is activated for receiving data frames from its child or peer nodes by setting the GACK flag in its beacon frame.

The use of the group ACK in the DSME multi-superframe is shown in Figure 34j. A coordinator allocates two DSME-GTS slots; one for of GACK1 frame and the other for GACK2 frame. A coordinator announces the superframe ID and the slot ID of GACK1 and GACK2 in its enhanced beacons. The coordinator uses the GACK1 frame to acknowledge all data frames received by it during CFP until GACK1 timeslot. The GACK2 frame is used to acknowledge all data frames received after the GACK1 frame but before transmission of GACK2 frame. An enhanced beacon frame with GACK flag set to zero indicates that the coordinator is not using the GACK feature. In this case, all data transmissions to the coordinator are acknowledged individually.

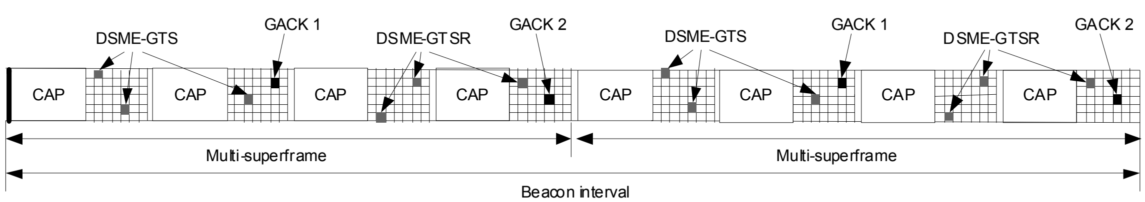

Figure 34j—GACK in DSME Multi-superframe structure

The devices that are sending their data frames to a coordinator with enabled GACK feature shall expect the acknowledgment of their DSME-GTS transmissions in the GACK 1 and the acknowledgment of their DSM-GTS retransmissions in the GACK2. A sender node, therefore, shall listen to the destination coordinator's beacon to determine the superframe ID, the slots ID, and the channel ID for GACK1 and GACK2. The information of GACK1 and GACK2 shall be stored in its allocation counter table (*macDSMEACT*). The devices shall allocate an additional DSME-GTSR (i.e., GTS for Retransmission) per each allocated DSME-GTS for transmission to that coordinator. The procedure for allocation and management of DSME-GTSRs is the same as the procedure for DSME-GTS as described in 5.1.10.5. The information of allocated DSME-GTSRs shall be stored in its allocation counter table (*macDSMEACT*).

After the transmission of a data frame in DSME-GTS, a sender node shall be able communicate with other nodes. It shall, however, switch back to pre-determined channel in the GACK1 or GACK2 timeslot to receive the acknowledgment for the previous DSME-GTS or DSME-GTSR transmission, respectively. The bitmap in the "Group Ack Flags" field of a GACK frame shall be used to determine if the GTS transmission by a sender node was successful. If the corresponding bit in the flag in the field indicates a failed DSME-GTS transmission, the sender node shall use DSME-GTSR slot to retransmit the data frame.

The application running on a coordinator shall enable or disable the GACK feature by using MLME_SET.request primitive in order to set the PIB attribute *macGACKFlag* to appropriate value. The application shall determine if the GACK feature is currently activated in the MAC by getting the current setting for macGACKFlag PIB attribute. The application shall use MLME_GET.request primitive to get the current value of *macGACKFlag* attribute.

5.1.10.4 CAP reduction

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

Figure 34k shows an example of the multi-superframe structure when CAP reduction is enabled. As shown in Figure 34k, every device has the CAP at the same time, which is the first superframe in a multi-superframe. The devices do not have the CAP in all other superframes.

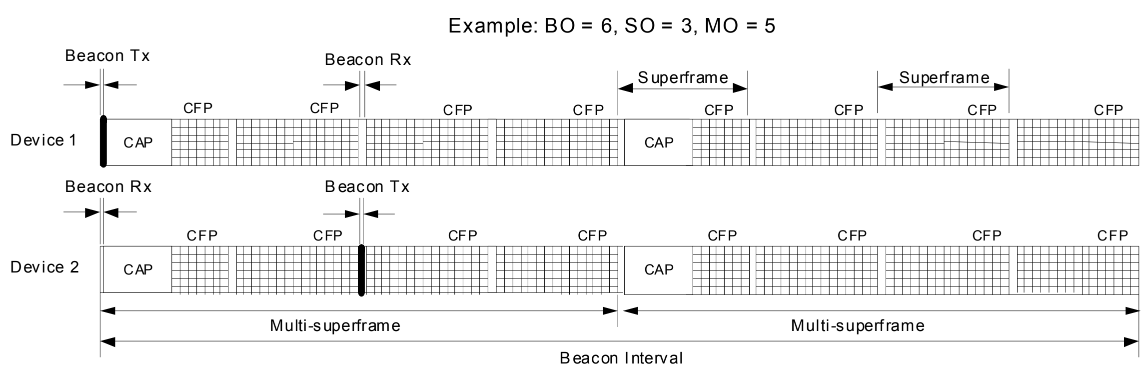

Figure 34k—CAP Reduction in DSME Multi superframe Structure

5.1.10.5 DSME-GTS allocation and management

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

A DSME-GTS shall be allocated before use, with the destination device deciding whether to allocate a DSME-GTS based on the requirements of the DSME-GTS request and the current slot availability. DSME-GTSs shall be allocated on a first-come-first-served basis, and all DSME-GTSs shall be placed contiguously at the end of the superframe and after the CAP (or after the beacon slot if CAP reduction is enabled). Each DSME-GTS shall be deallocated when the DSME-GTS is no longer required, and a DSME-GTS can be deallocated at any time by the destination device or the source device that originally requested the DSME-GTS.

A data frame transmitted in an allocated DSME-GTS shall use only short addressing.

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

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

5.1.10.5.1 DSME-GTS allocation

A DSME enabled device is instructed to request the allocation of new DSME-GTSs through the MLME-DSME-GTS.request primitive. The ManagementType parameter shall be set to 0b001 (DSME-GTS allocation).

On receipt of the primitive, the MLME of the device shall send a DSME-GTS request command frame as described in 5.3.11.4 to the value of DeviceAddress parameter of the primitive. The ManagementType, Direction, PrioritizedChannelAccess, NumSlot, PreferredSuperframeID, PreferredSlotID, DSMESABSubblockIndex, and DSMESABSubblockLength parameters shall be contained in the corresponding fields of the command frame. The DSME SAB Sub-block field shall contain a bitmap of a subsub-blockblock of *macDSMESAB*. The DSMESABSubblockIndex and DSMESABSubblockLength indicate the bitmap of which sub-block shall be contained in the command frame.

If the MLME successfully transmits the DSME-GTS request command frame, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a status of NO_ACK.

After sending the DSME-GTS request command frame, the device shall wait for at most *macMaxFrameTotalWaitTime* CAP symbols, if no DSME-GTS reply command frame is received, the MLME of the device shall notify the higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a status of NO_DATA.

On receipt of a DSME-GTS request command frame with management type one (allocation), the MLME of the device shall notify the receipt of the command frame to the higher layer through MLME-DSME-GTS.indication. The higher layer will make the decision on the allocation using the value of DSMESABSpecification parameter as the slot availability information. The zeroes (0) in *macDSMESAB* and DSME-GTS SAB Sub-block field of the command frame indicate the available slots. It is recommended that the preferred superframe ID and the preferred slot ID are considered in allocating slots. If the preferred slot is not available, the next slot can be used. Following this approach, slot allocation of a multihop flow can be allocated sequentially on each hop to reduce the end-to-end delay. After making the allocation decision, MLME-DSME-GTS.response primitive will be issued. The DSMESABSpecification parameter shall indicate the newly allocated slots.

On receipt of MLME-DSME-GTS.reponse primitive with Status parameter value of SUCCESS, the device shall send a DSME-GTS reply command frame to the requesting device (i.e., the destination address is set to 0xffff and the DSME-GTS destination address is set to the short address of the requesting device). The Management Type field shall be set to 0b001 (allocation). The Status field in the command frame shall be set to zero (SUCCESS). The DSMESABSpecification field shall be set to the same as the DSMESABSpecification parameter of the primitive. Also, the device shall update *macDSMEACT* according to the DSMESABSpecification parameter value.

On receipt of MLME-DSME-GTS.reponse primitive with Status parameter value of DENIED or INVALID_PARAMETER, the device shall send a DSME-GTS reply command to the requesting device (i.e., the destination address is set to the short address of the requesting device, and also the DSME-GTS destination address is set to the short address of the requesting device). The Management Type field shall be set to 0b001 (allocation). The Status field shall be set to one (DENIED) or two (INVALID_PARAMETER). The DSMESABSpecification field shall be set to the same as the DSMESABSpecification parameter of the primitive.

On receipt of a DSME-GTS reply with management type one (allocation), the device shall check the DSME-GTS Destination Address field of the received command frame.

If the DSME-GTS Destination address is the same as the *macShortAddress*, the device shall inform the next higher layer of the result of the DSME-GTS allocation request using MLME-DSME-GTS.confirm. If the value of the Status field in the command frame is zero (SUCCESS), the device shall broadcast a DSME-GTS notify command (i.e., the destination address is set to 0xffff, and the DSME-GTS destination address is set to the short address of the source device of the received command). The Management Type field shall be set to 0b001 (DSME-GTS allocation). The DSMESABSpecification field shall be set to the same as the DSMESABSpecification field in the received command frame. Also, the device shall update *macDSMEACT* according to the DSMESABSpecification in the received command frame.

If the DSME-GTS Destination address of the DSME-GTS reply command frame is not the same as the *macShortAddress*, the device shall check if the slots marked as one in this command frame is conflicting with the readily allocated slots in *macDSMEACT*. If there is no conflict, the device shall update macDSMESAB according to the DSMESABSpecification in this command frame to reflect the neighbor's newly allocated DSME-GTSs. If there is a conflict, the device shall send a DSME-GTS request command frame with Management type 0b001 duplicated allocation notification to the source device of the received command.

On receipt of a DSME-GTS notify with management type one (allocation), the device shall check the DSME-GTS Destination Address field of the received command frame.

If the DSME-GTS Destination address is the same as the *macShortAddress*, the device shall notify the next higher layer of the receipt of the DSME-GTS notify command frame using MLME-COMM-STATUS.indication.

If the DSME-GTS Destination address of the DSME-GTS notify command frame is not the same as the *macShortAddress*, the device shall check if the slots marked as one in this command frame is conflicting with the readily allocated slots in *macDSMEACT*. If there is no conflict, the device shall update *macDSMESAB* according to the DSMESABSpecification in this command frame to reflect the neighbor's newly allocated DSME-GTSs. If there is a conflict, the device shall send a DSME-GTS request command frame with management type 0b010 duplicated allocation notification to the source of the received command.

On receipt of a DSME-GTS request command frame indicating a DSME-GTS duplicate allocation notification, the device shall deallocate the duplicated DSME-GTSs as described in 5.1.10.5.2.

An example of DSME-GTS allocation is shown in Figure 34l.

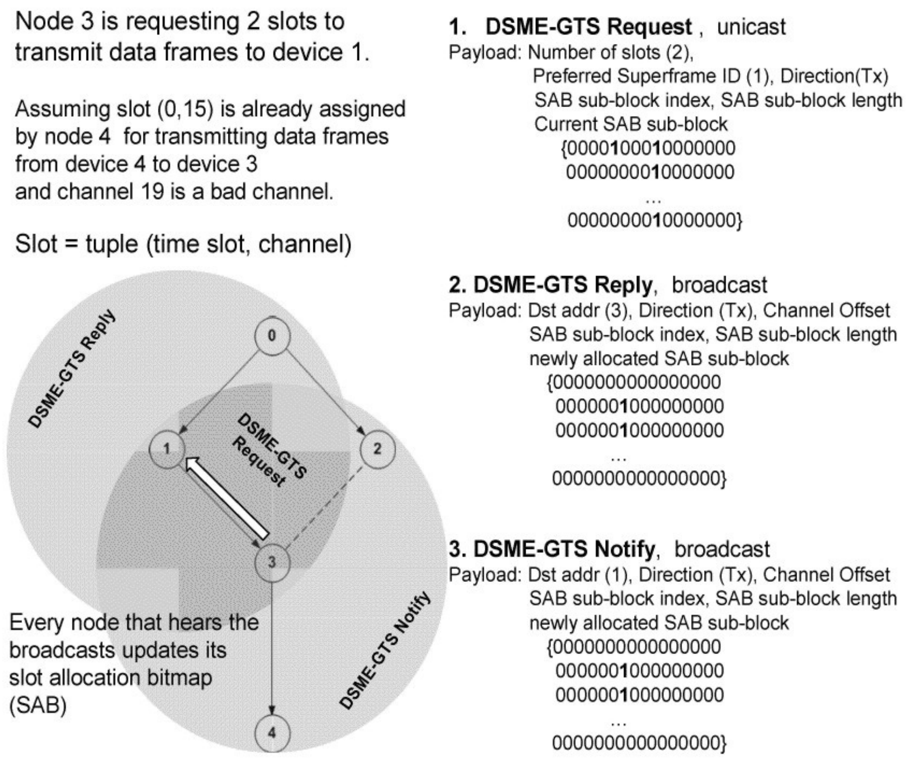

Figure 34l—Example of a handshake for DSME-GTS allocation

5.1.10.5.2 DSME-GTS deallocation

The DSME-GTS Source device is instructed to request the deallocation of existing DSME-GTSs through the MLME-DSME-GTS.request primitive. The Destination device can request the deallocation of existing DSME-GTSs if a deallocation request comes from the higher layer, or upon the expiration of the DSME-GTSs. From this point onward, the DSME-GTSs to be deallocated shall not be used by the device, and the *macDSMESAB* and the *macDSMEACT* shall be reset accordingly.

When a DSME-GTS deallocation is initiated by the higher layer of the device, the MLME shall receive the MLME-DSME-GTS.request primitive, the ManagementType parameter set to zero (deallocation) and the DSMESABSpecification set to indicate the DSME-GTSs to deallocate.

When a DSME-GTS deallocation is due to the DSME-GTS expiring, the MLME shall notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-DSME-GTS.indication primitive of which the Management Type field set to 0b101 (DSME-GTS Expiration). The higher layer shall initiate a DSME-GTS deallocation using MLME-DSME-GTS.request.

To deallocate existing DSME-GTSs, the MLME shall send the DSME-GTS request command to the corresponding device (the Source or Destination of which the DSME-GTS to be deallocated). The destination address of the command shall be set to the corresponding device. The Management Type field shall be set to zero (DSME-GTS deallocation) and the DSMESABSpecification field shall be set to indicate the DSME-GTSs to deallocate.

If the MLME successfully transmits the DSME-GTS request command frame, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a status of NO_ACK.

After sending the DSME-GTS request command frame with management type 0 (deallocation), the device shall wait for at most *macMaxFrameTotalWaitTime* CAP symbols. If no DSME-GTS reply command frame with management type 0 (deallocation) is received, the MLME of the device shall notify the higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a status of NO_DATA.

On receipt of a DSME-GTS request command frame with management type 0 (deallocation), the device shall check the DSME-GTSs specified for deallocation in the received command frame.

If the DSME-GTSs in the command frame match the allocated DSME-GTSs in *macDSMEACT*, the device shall notify the receipt of the command frame to the higher layer through MLME-DSME-GTS.indication. The higher layer shall make the decision on the deallocation, and inform MLME of its decision using MLME-DSME-GTS.response. Then, the device shall send a DSME-GTS reply command frame to the Source device of the received command frame (i.e., the destination address is set to 0xffff, and the DSME-GTS destination address is set to the short address of the Source device). The management type of the DSME-GTS reply command shall be set to zero (DSME-GTS deallocation), and the Status field shall be set to zero (SUCCESS). The DSMESABSpecification field shall be set to indicate the DSME-GTSs to deallocate.

If the DSME-GTSs in the command frame do not match any of the allocated DSME-GTSs in *macDSMEACT*, the device shall send a DSME-GTS reply command frame to the source device of the received command frame (i.e., the destination address is set to the short address of the source device, and also the DSME-GTS destination address is set to the short address of the source device). The Management Type field shall be set to zero (deallocation), and the Status field shall be set to two (INVALID_PARAMETER). The DSMESABSpecification field shall be set to the same as the received command frame.

On receipt of a DSME-GTS reply command with management type zero (deallocation), the device shall check the DSME-GTS Destination Address field of the received command frame.

If the DSME-GTS Destination address is the same as the *macShortAddress*, the device shall inform the next higher layer of the result of the DSME-GTS deallocation request using MLME-DSME-GTS.confirm. If the value of the Status field in the command frame is zero (SUCCESS), the device shall send a DSME-GTS notify command frame to the source device of the received command frame (i.e., the destination address is set to 0xffff, and the DSME-GTS destination address is set to the short address of the source device). The Management Type field of the DSME-GTS notify command frame shall be set to zero (deallocation), and the DSMESABSpecification field shall be set to indicate the DSME-GTSs to deallocate.

If the DSME-GTS Destination address is not the same as the *macShortAddress*, the device shall update *macDSMESAB* according to the DSMESABSpecification in this command frame to reflect the neighbor's deallocated DSME-GTSs.

On receipt of a DSME-GTS notify with management type 0 (deallocation), the device shall check the DSME-GTS Destination Address field of the received command frame.

If the DSME-GTS Destination address is the same as the *macShortAddress*, the device shall notify the next higher layer of the receipt of the DSME-GTS notify command frame using MLME-COMM-STATUS.indication.

If the DSME-GTS Destination address is not the same as the *macShortAddress*, the device shall update macDSMESAB according to the DSMESABSpecification in this command frame to reflect the neighbor's deallocated DSME-GTSs.

5.1.10.5.3 DSME-GTS expiration

The MLME of the device shall attempt to detect idle DSME-GTSs in the allocation counter table (*macDSMEACT* is described in Table 1a) using the following rules:

— The MLME of the device shall perform DSME-GTS deallocation when a DSME-GTS has expired (i.e., a device has stopped using the DSME-GTS).
— The MLME of the Destination device of DSME-GTS shall assume that the source device is no longer using its DSME-GTS if a data frame has not been received in the DSME-GTS for *macDSMEGTSExpirationTime* multi-superframes.
— The MLME of the Source device of DSME-GTS shall assume that the link quality is bad if no acknowledgment frame has been received for *macDSMEGTSExpirationTime* multi-superframes while the acknowledgment was requested.

Table 1a—Allocation counter Table (*macDSMEACT*) description

Field	Type	Valid range	Description
Superframe ID	Integer	0x0000–0xfffd	The superframe ID of the DSME-GTS in a multi-superframe.
Slot ID	Integer	0–14	The slot ID of the DSME-GTS in the superframe.
Channel ID	Integer	0x0000–0xffff	In channel adaptation, this field shall contain the Channel number of the DSME-GTS. In channel hopping, this field shall contain the Channel Offset.
Direction	Integer	0–1	The direction of the allocated DSME-GTS. 0: Transmission (TX), 1: Reception (RX)
Type	Integer	0x00–0x03	The type of the DSME-GTS. 0x00: (regular) DSME-GTS 0x01: DSME-GTSR 0x02: GACK1 0x03: GACK2
Prioritized channel access	Boolean	TRUE or FALSE	The priority level of the DSME-GTS. TRUE: High priority. FALSE: Low priority.
Source/Destination Address	Device address	0x0000–0xfffd	The 16-bit short address of the device that is the source (if RX) or the destination of the allocated DSME-GTS.
Counter	Integer	0x0000–0xffff	An idle counter, in other word, the number of idle multi-superframes since the allocated DSME-GTS was used.
LinkQuality	Integer	0x0000–0xffff	The link quality of the allocated DSME-GTS.

5.1.10.5.4 DSME-GTS retrieve

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

If a device has lost synchronization before its allocated DSME-GTSs of current superframe starts, the device can be instructed to retrieve synchronization through the MLME-DSME-INFO.request primitive as described in 6.2.21.2.1 with the INFO parameter "0x00" (i.e., timestamp). Also, a device can be instructed to retrieve the slot allocation and availability information of a neighboring DSME-enabled device through MLME-DSME-INFO.request primitive with INFO parameter "0x01" (i.e., DSMESABSpecification). In this case, the DSME SAB sub-block length and the DSME SAB sub-block index shall be set to specify the DSME SAB sub-block of which the device needs to find the slot availability. A device can be instructed to retrieve the superframe configuration information through MLME-DSME-INFO.request primitive with INFO parameter "0x02" (i.e., DSME PAN Descriptor).

On receipt the primitive, the MLME of the device shall send a DSME-Information request command frame to the Destination device. The Info Type field shall be set to the same value as INFO parameter of the primitive.

If the MLME successfully transmits the DSME-Information request command frame, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME INFO.confirm primitive with a status of NO_ACK.

After sending the DSME-Information request command frame, the Source device shall wait no longer than *macMaxFrameTotalWaitTime* symbols. If no DSME-Information reply command frame is received, the MLME of the source device shall notify the next higher layer of the failure by the MLME-DSME-INFO.confirm primitive with a status of NO_DATA.

On receipt of a DSME-Information request command with Info type 0x00, the device shall determine whether it has an allocated DSME-GTS to the requesting device. If it has an allocated DSME-GTS, the MLME of the Destination device shall send a DSME Information reply command frame in the DSME-GTS. The DSME-Information reply command contains the timestamp, superframe ID, and the slot ID. If it does not have an allocated DSME-GTS to the requesting device, the MLME of the Destination device shall ignore the request.

On receipt of a DSME-Information request command with Info type 0x01, the device shall send a DSME-Information reply command with the DSMESABSpecification. The DSMESABSpecification shall be set according to the current allocation status of all one-hop neighborhoods of the device. The zeroes (0) in the SAB sub-block indicate the vacant slots and the ones (1) indicate allocated or unavailable slots. The entire DSME SAB is stored in *macDSMESAB*. The DSME SAB sub-block index and the DSME SAB sub-block length indicate which part of the macDSMESAB shall be selected as the DSME SAB sub-block to be included in the DSME-Information reply.

On receipt of a DSME-Information request command with Info type 0x02, the device shall send a DSME-Information reply command with the DSME PAN Descriptor.

On receipt of a DSME-Information reply command frame containing the timestamp and the slot ID, the MLME of the device shall notify the higher layer of the result of the information request through MLME-DSME-INFO.confirm primitive. If the Info type of the received command frame is 0x00, the device can use this information to retrieve synchronization and continue to use DSME-GTSs.

5.1.10.5.5 DSME-GTS change

The Destination device allocates the DSME-GTSs to the Source device according to the first-come-first-served basis. If the Destination device receives a DSME-GTS request command from a source device with a higher priority of data transmission when there is no available DSME-GTSs, the Destination device shall reduce part or all of the DSME-GTSs that are being used for the lower priority data transmission and allocate the reduced DSME-GTSs for the higher priority data transmission. If the Destination device receives more than one DSME-GTS request command with the same priority of data transmission, the Destination device shall allocate the DSME-GTSs according to the first-come-first-served basis.

After the higher priority data transmission in the DSME-GTSs is finished, if there are no more DSME-GTS request commands with data transmissions of higher priority are received, the Destination device shall restart the DSME-GTSs for the lower priority data transmission that were reduced previously. Otherwise, the higher priority data transmission shall use the DSME-GTSs first. If the lower priority data transmission has been suspended for *n* multi-superframes, where *n* is an integer and its value is application dependent, the Destination device shall allocate the next available DSME-GTSs to the corresponding Source.

The procedure of DSME-GTS change shall be initiated when a Destination device wants to reduce or restart the allocated DSME-GTSs through the MLME-DSME-GTS.request primitive.

When a DSME-GTS change is initiated by the next higher layer of the Destination device, the MLME shall receive the MLME-DSME-GTS.request primitive with the management type set accordingly (i.e., 0b011 for DSME-GTS Reduce or 0b100 for DSME-MLME-GTS Restart)

To request the change of existing DSME-GTSs, the MLME of the device shall send the DSME-GTS request command frame to the corresponding device of the DSME-GTSs. The Management Type field shall be set accordingly (i.e., 0b011 for DSME-GTS Reduce or 0b100 for DSME-GTS Restart), and DSME-GTS SAB Specification field shall be set to indicate the DSME-GTSs to change.

On receipt of a DSME-GTS request command frame for DSME-GTS change, the device shall immediately change its DSME-GTS according to the Management Type field in the received command frame. Then the MLME of the device shall notify the higher layer of the change. This notification is achieved when the MLME issues the MLME-DSME-GTS.indication primitive as described in 6.2.21.1.2. Also, the device shall send DSME-GTS reply command frame to the source of the DSME-GTS request command.

On receipt of the DSME-GTS reply command frame, the MLME of the device shall notify the next higher layer of the DSME-GTS change. This notification is achieved when the MLME issues the MLME-DSME-GTS.confirm primitive (refer to 6.2.21.1.4) with a status of SUCCESS, the ManagementType parameter set to 0b011 for DSME Reduce or 0b100 for DSME-GTS Restart accordingly, and DSMEGTSSABSpecification parameter set to indicate the DSME-GTSs to change.

5.1.10.6 Beacon scheduling

The transmission of enhanced beacons from different devices shall be scheduled using the procedure described in this subclause.

When a new device wants to join a network, it uses the MLME SCAN.request primitive in order to initiate a channel scan over a given list of channels. It searches for all coordinators transmitting enhanced beacon frames within the maximum BI period. The neighboring devices (i.e., coordinators) would share their beacon schedule information by transmitting an enhanced beacon frame to the new device. The beacon schedule information is expressed in a bitmap sequence that represents the schedule of enhanced beacons transmitted from all neighboring devices. The corresponding bit in the bitmap shall be set to one if a beacon is occupied in that beacon slot. The new device shall search a vacant beacon slot from the bitmap in all the received beacon frames. Once the new device finds a vacant beacon slot, it uses the slot as its own beacon slot.

There can be a beacon slot collision when two or more devices are trying to compete for same beacon slot number. Figure 34m illustrates this circumstance; device D and E are new devices that join the network, and these devices receive the beacon bitmap from their neighboring devices. Since device A is a common neighboring device to device D and E, there can be a collision if both new devices E and D notify the use of the same vacant beacon slot within the CAP. This circumstance is well-known as a *hidden node problem.* This happens because device D and E are hidden to each other, and cannot listen to each other's transmission. DSME-Beacon allocation notification command is used to resolve this possible collision. If device D and E send DSME-Beacon allocation notification command with same SDIndex value (SDIndex of 1 in Figure 34m), device A determines which device has notified first. Device A shall reply with the beacon collision notification command frame to the device that has notified later.

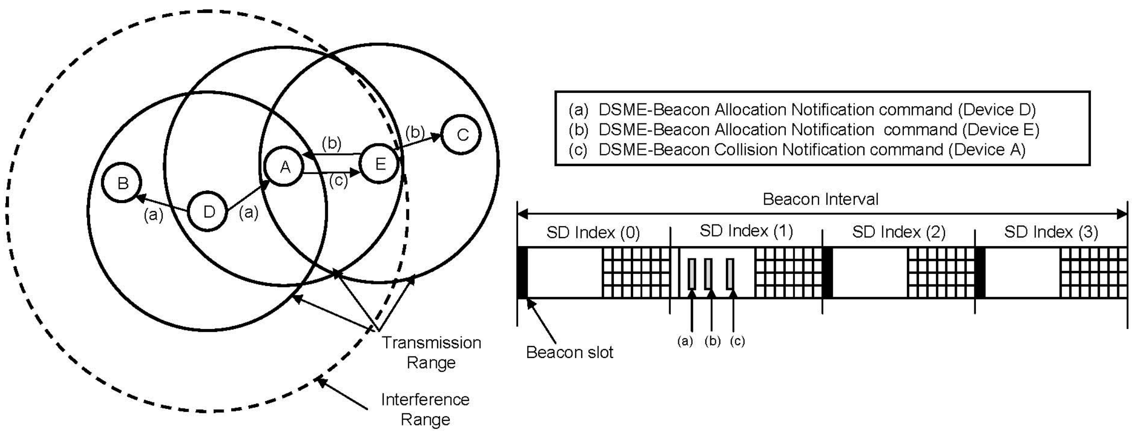

Figure 34m—Avoidance of beacon slot allocation collision

5.1.10.7 Time synchronization

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

Each device performs time synchronization using the values of the Timestamp fields in the beacon from its time synchronization parent in order to maintain global time synchronization in the PAN. As shown in Figure 34n, a device locates the start time of superframe using the value of the Beacon Timestamp field in the beacon frame. The device knows when the beacon frame is actually transmitted from the time synchronization parent using value of the Beacon Offset Timestamp field. The difference in timestamp values reflects time delay due to CCA or processing time at PHY to transmit a frame. Then, the device transmits the beacon frame. A device that has received the beacon frame calculates the actual reception time considering the value of the beacon offset timestamp, and then resets its local time accordingly. The PAN coordinator does not reset its local time according to the received beacon frame since the PAN coordinator does not have the time synchronization parent device.

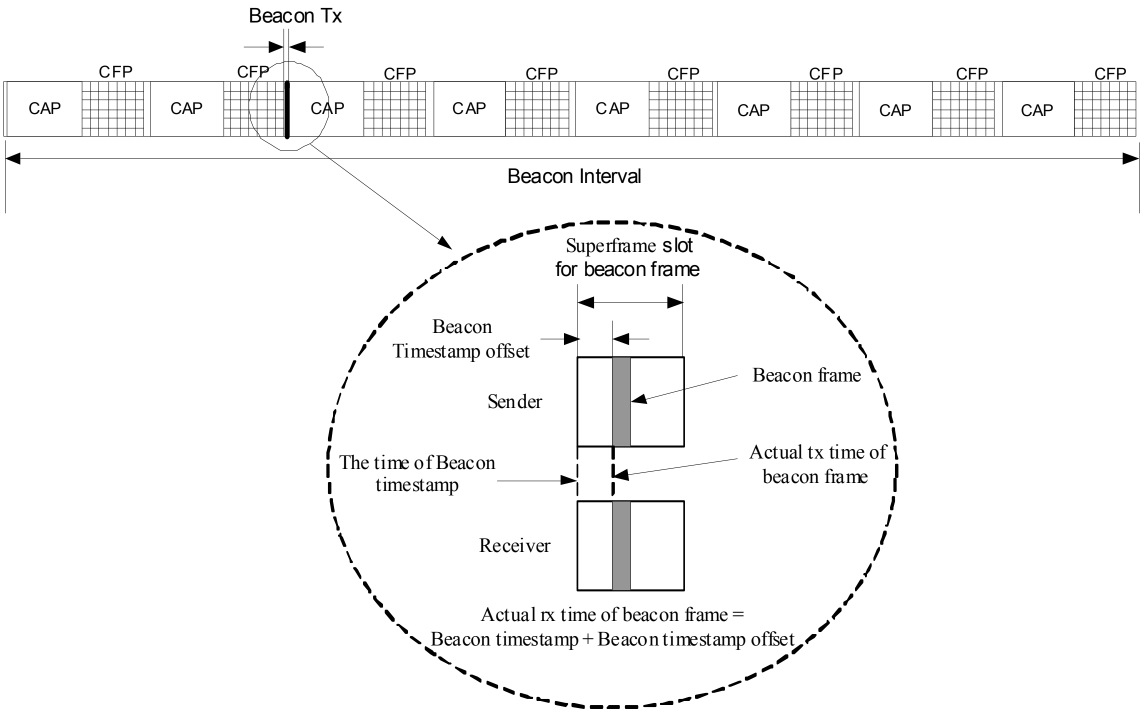

Figure 34n—Time synchronization in a DSME-enabled PAN

5.1.10.8 Deferred beacon

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

5.1.10.9 Passive channel scan

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

5.1.10.10 Coexistence of beacon-enabled and nonbeacon-enabled

PANs that contain both beacon-enabled and nonbeacon-enabled devices shall include the Connection Devices.

The beacon-enabled device shall either transmit periodic beacons or track the beacon for communication in the PAN. The nonbeacon-enabled device shall neither transmit periodic beacons nor track the beacon for communication in the PAN, and it shall either request the data from other nonbeacon-enabled devices or transmit the data upon receipt of the data request commands from the nonbeacon-enabled devices in the PAN. Connection devices shall operate in both beacon-enabled and nonbeacon-enabled at the same time, which means the Connection device can transmit or receive frames with either beacon-enabled devices or nonbeacon-enabled devices.

In order to maintain a consistent beacon order, the Connection device shall store the superframe structure parameters in the beacon it tracks and use those parameters in its own beacon to transmit. Moreover, the Connection device shall send the stored parameters to the neighbors actively or upon receipt of a DSME-Information request command from other devices. In order to avoid frame conflict, the Connection device shall not communicate with a nonbeacon-enabled device when it is tracking or transmitting beacons or communicating to a beacon-enabled device.

5.1.11 LE transmission, reception and acknowledgment

5.1.11.1 Coordinated sampled listening (CSL)

5.1.11.1.1 General

The coordinated sampled listening (CSL) mode is turned on when the PIB attribute *macCSLPeriod* is nonzero and turned off when macCSLPeriod is zero. In CSL mode, transmission, reception, and acknowledgment work as follows. Figure 34o illustrates the basic CSL operations.

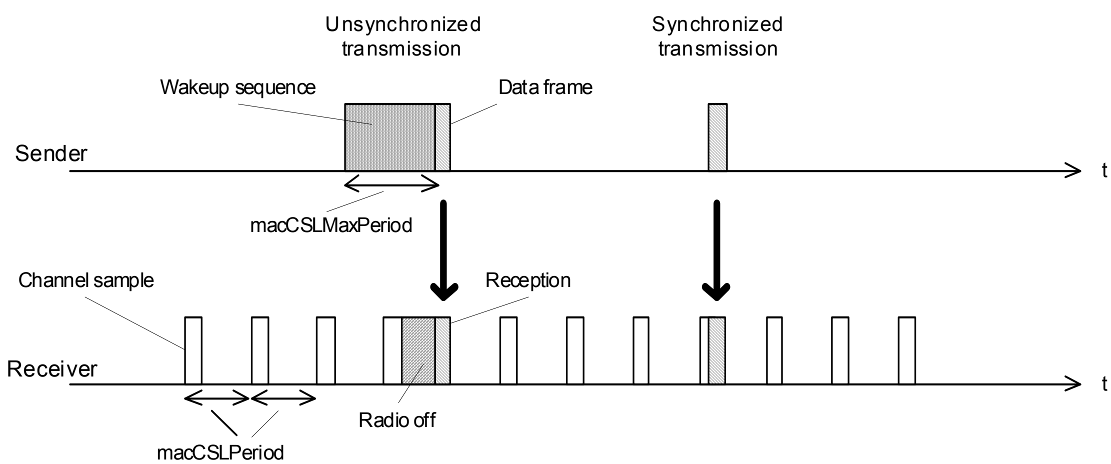

Figure 34o—Basic CSL operations

5.1.11.1.2 CSL idle listening

During idle listening, CSL performs a channel sample every *macCSLPeriod time*. If the channel sample does not detect energy on the channel, CSL disables the receiver until the next channel sample time, and then performs the next channel sample. If the channel sample receives a wake-up frame, CSL checks the destination address in the wake-up frame. Retransmissions follow the same process as defined in 5.1.6.4.3 except that each transmission follows the process above. If it matches *macShortAddress*, CSL disables the receiver until the Rendezvous Time (RZ Time) in the wake-up frame, and then enables the receiver to receive the payload frame. Otherwise, CSL disables the receiver until RZ Time plus the transmission time of the maximum length payload frame and the secure acknowledgment frame, and then resumes channel sampling.

5.1.11.1.3 CSL transmission

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

5.1.11.1.4 Unicast transmission

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

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

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

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

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

a) Perform CSMA-CA to acquire the channel
b) If the previous acknowledged payload frame to the destination has the frame pending bit set and is within *macCSLFramePendingWaitT* (defined in Table 52j), go to step e).
c) If it is a synchronized transmission, wait until the destination device's next channel sample.
d) For the duration of wake-up sequence length (short or long)
    1) Construct wake-up frame with the destination short address and remaining time to payload frame transmission (at the end of wake-up sequence)
    2) Transmit wake-up frame
e) Transmit payload frame
f) Wait for up to *macEnhAckWaitDuration* (defined in Table 52j) symbol time for the enhanced acknowledgment frame if the Acknowledge Request field in the payload frame is set to one.
g) If the enhanced acknowledgment frame is received, update CSL phase and period information about the destination device from the Acknowledgment CSL Sync field.
h) If the enhanced acknowledgment frame is not received, start retransmission process.

Retransmissions follow the same process as defined in 5.1.6.4.3 except that each transmission follows the process above.

5.1.11.1.5 Broadcast transmission

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

— It is always unsynchronized transmission.
— The destination address in wake-up frames is set to 0xffff.
— Optionally include LE CSL IE.

Selectively the next higher layer may add LE CSL IE in the frame header to propagate CSL phase and period information among the neighboring devices.

5.1.11.1.6 CSL reception

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

- Immediately sends back an enhanced acknowledgment frame with the destination address set as the transmitting device and its own CSL phase and period filled in the LE CSL IE. The acknowledgment frame can be optionally authenticated and/or encrypted depending on the current security mode.
- If the LE CSL IE is present in the payload frame, the CSL phase and period information about the transmitting device is updated with the information in the LE CSL IE.
- If the Frame Control field frame pending bit in the received payload frame is set to one, the receiver is kept on for macCSLFramePendingWaitT time before going back to CSL idle listening. Otherwise, CSL idle listening starts.

5.1.11.1.7 CSL over multiple channels

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

5.1.11.1.8 Turning off CSL mode to reduce latency

The next higher layer has the option to turn off sampled listening and stop sending wake-up sequences to reduce latency for urgent messages. This assumes that the higher layer manages the coordination between the sender and receiver in turning on and off sampled listening. To turn off sampled listening, the higher layer simply sets *macCSLPeriod* to zero. To turn on sampled listening, the high layer restores *macCSLPeriod* to its previous nonzero values. Similarly, to stop sending wake-up sequences, the higher layer sets *macCSLMaxPeriod* to zero and restores it to its previous value to return to normal CSL mode. To request a neighboring device to turn off sampled listening, the higher layer shall send a frame to the device with frame pending bit set to 1. This prevents CSL from turning off the radio before the request is processed.

5.1.11.2 Receiver initiated transmission (RIT)

5.1.11.2.1 General

The receiver initiated transmission (RIT) is an alternative low-energy MAC for nonbeacon-enabled PAN (BO = 15). RIT mode is turned on when PIB attribute *macRITPeriod* is nonzero and is turned off when *macRITPeriod* is zero. In RIT mode, transmission, reception, and acknowledgment work as follows.

macCSLPeriod (in coordinated sample listening) and *macRITPeriod* shall not be nonzero at the same time. Figure 34p illustrates the basic RIT operations. Figure 34q illustrates the RIT operations when the RIT data request command payload carries schedule information, refer to 5.3.12.1.4 for details.

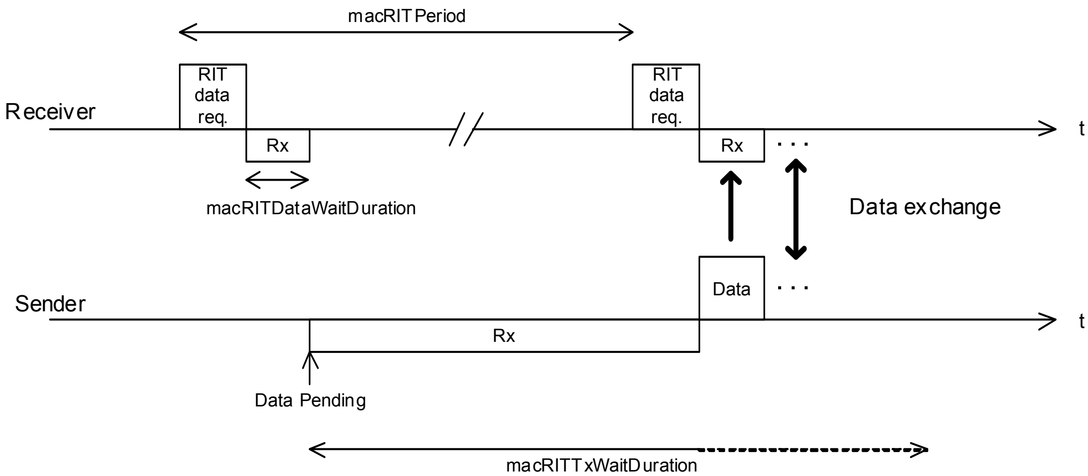

Figure 34p—Basic RIT operations

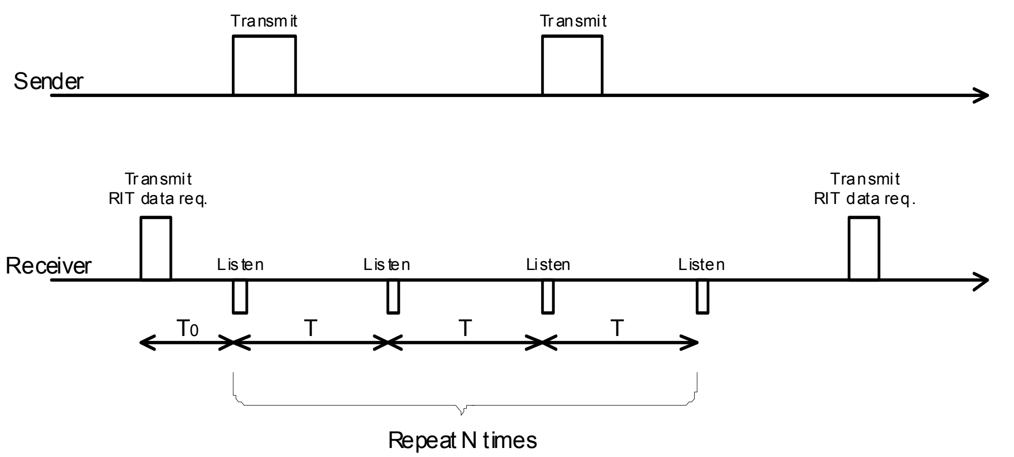

Figure 34q—RIT operations when data req carries schedule information

5.1.11.2.2 Periodic RIT data request transmission and reception

In RIT mode, a device transmits RIT data request command every *macRITPeriod* using unslotted CSMA-CA. The destination address of the command may be broadcast address (0xffff) or the address of the intended transmitter of data (associated coordinator). The command may optionally contain a 4-octet payload defined in 5.3.12.1.4. When the command carries no payload, after the transmission of RIT data request command frame, the device listens for *macRITDataWaitDuration* for incoming frame (except RIT data request frame) and goes back to idle state till the next periodic transmission of RIT data request command. When a device is in the receiving state after transmission of RIT data request command, RIT data request command frame from another device shall be discarded. When data request command carries a 4-octet payload (time to 1st listen *T*0, number of repeat *N*, repeat listens interval *T*), the device goes back to sleep for *T*0 period of time then listens for *macRITDataWaitDuration* before going back to sleep. The first listen on, it repeats a listen interval of *macRITDataWaitDuration* every *T* period of time for *N* times. The device shall start listening slightly before each scheduled listen time based on a guard time computed from possible clock skew since the last data request command transmission.

Upon reception of frame after the transmission of RIT data request command, it notifies its arrival to the next higher layer by initiating corresponding indication primitive. At the completion of frame reception, the device may set *macRITPeriod* parameter to zero (RIT off) at the discretion of the next higher layer. If this is the case, it shall stop periodic transmission of RIT data request command and become always active until *macRITPeriod* parameter is set to a nonzero value by the next higher layer again. During this period (*macRITPeriod* equals to zero), all transactions shall be handled as those of normal non beacon-enabled PAN (*macRxOnWhenIdle*: False).

Figure 34r shows the Message sequence chart for starting RIT mode.

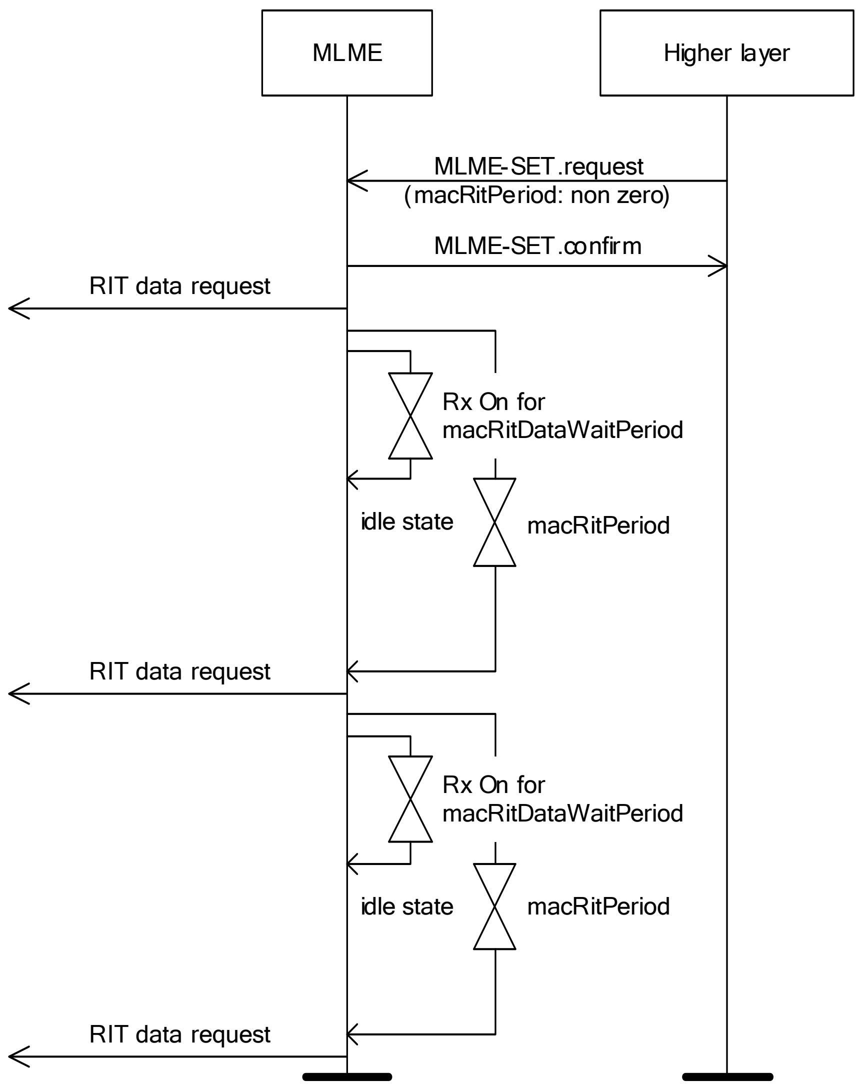

Figure 34r—Message sequence chart for starting RIT mode

5.1.11.2.3 RIT Transmission

In order to transmit frame in RIT mode, the device shall at first stop its periodic transmission of RIT data request, enable its receiver, and wait at most *macRITTxWaitDuration* for reception of RIT data request command frame from neighboring device. During this *macRITTxWaitDuration* period, all other frames except RIT data request command shall be discarded.

Upon reception of RIT data request command frame, the MAC sublayer sends the pending data with a use of unslotted CSMA-CA. In case that the Destination PAN Identifier field and the Destination Address field of the received RIT data request command are broadcast (0xffff) and the DstPANId and DstAddr parameters of initiated MCPS DATA.request are also broadcast, the Destination PAN Identifier field and the Destination Address field of the outgoing data frame shall be set as the Source PAN Identifier field and the Source Address field of the received RIT data request command, respectively.

At the completion of frame transmission, corresponding confirm primitive shall be initiated by the MAC sublayer to the next higher layer. At this point, the device shall restart its transmission of periodic RIT data request transmission. Also at this point, the device may set *macRITPeriod* parameter to zero (RIT off) at the discretion of the next higher layer. If this is the case, it shall continue to stop periodic transmission of RIT data request command and become always active until *macRITPeriod* parameter is set to nonzero value by the next higher layer again. During this period (*macRITPeriod* equals to zero), all transactions shall be handled as those of normal non beacon-enabled PAN (*macRxOnWhenIdle*: False).

When the data request commands carry the listen schedule payload, the device can either wait to receive a data request frame from the receiving device as described above, or sleep until the next scheduled listen time by the receiving device then wake up to transmit the intended frame.

Figure 34s shows the message sequence chart for data transmission in RIT mode.

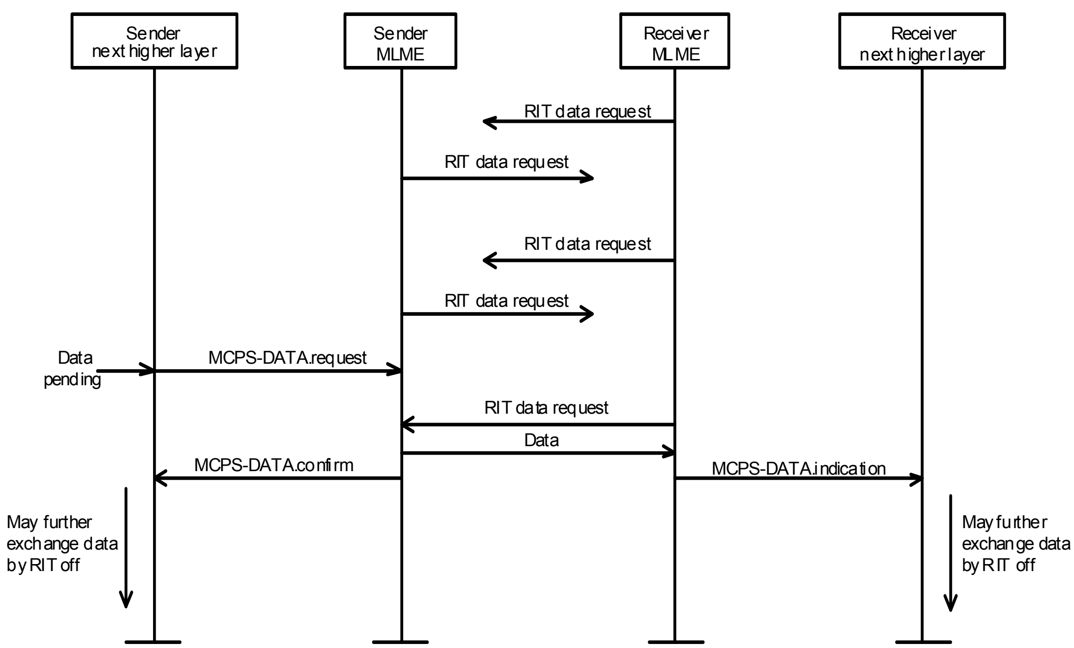

Figure 34s—Message sequence chart for data transmission in RIT mode

5.1.11.2.4 Broadcast transmission

Broadcast transmission shall not be supported in RIT mode.

5.1.12 Asynchronous multi-channel adaptation (AMCA)

5.1.12.1 General

Single common channel approach may not be able to connect all devices in the PAN. The variance of channel condition can be large and channel asymmetry between two neighboring device can happen. Multi-channel adaptation is a solution to handle such case.

Two types of multi-channel adaptation are specified: synchronous multi-channel adaptation and asynchronous multi-channel adaptation. The synchronous multi-channel adaptation is performed in beacon-enabled mode and is handled by DSME-GTS as described in 5.1.10.5. The asynchronous multi-channel adaptation is performed in non-beacon mode and is described in this subclause.

5.1.12.2 Receiver-based communication

It is possible that no common channel exists such that two devices can communicate in DSME-GTS mode even though there are many available channels. In that case, each device selects its designated listening channel based on its local link quality, and updates the value of *macDesignatedListeningChannel*. The device keeps listening to its designated listening channel (*macDesignatedListeningChannel*). When another device wants to communicate with it, the sender device shall switch to the designated listening channel (*macDesignatedListeningChannel*) of the receiver device and transmit a DATA frame. Then the sender device shall switch back to its own designated listening channel and keep listening. On receipt of the data frame from the sender device, the receiver device shall switch to the designated listening channel of the sender device and transmit an ACK frame (if requested). After sending the acknowledgment frame, the receiver device shall switch back to its own designated listening channel and keep listening to the channel.

Figure 34t illustrates the receiver-based communications.

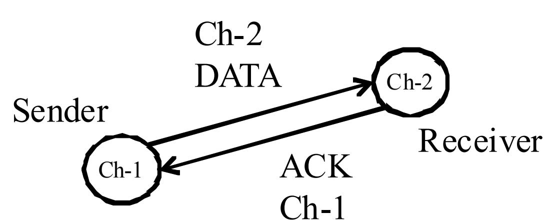

Figure 34t—Receiver-based communication

5.1.12.3 Asymmetric multi-channel active scan

An asymmetric multi-channel active scan allows device to detect the designated listening channel (*macDesignatedListeningChannel*) of each coordinator or detect the best channel for the device.

The asymmetric multi-channel active scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to 0x04.

For each channel, the device shall first switch to the channel, by setting *phyCurrentChannel* and *phyCurrentPage* accordingly, and send an AMCA-Beacon request command as described in 5.3.13.1. Upon successful transmission of the AMCA-Beacon request command, the device shall enable its receiver for [*aBaseSuperframeDuration* × (2^n + 1)] symbols, where n is the value of the *ScanDuration* parameter. During this time, the device shall reject all non-beacon frames and record the information contained in all unique beacons in a PAN descriptor structure (defined in Table 17). After this time, the device shall switch to the next channel and repeat the same procedure. The device shall repeat this procedure until visiting each channel twice to guarantee successful reception of the beacon request and the beacon under the asymmetric channel condition.

If *linkqualityscan* flag is FALSE, the device may stop after it receives a beacon and decide the current channel as its designated listening channel. If *linkqualityscan* flag is TRUE, the device makes decision on its designated listening channel comparing line quality indication (LQI) or receive signal strength indicator (RSSI) of the received beacons.

On receipt of the AMCA-Beacon request command, the coordinator shall transmit a beacon (refer to 5.2.2.1) over a set of channels specified in the asymmetric multi-channel beacon request command. Upon successful transmission of the beacon, the coordinator shall switch to the next channel after [*aBaseSuperframeDuration* × (2^n + 1)] symbols, where n is the value of the *ScanDuration* parameter, and send another beacon. The coordinator shall repeat the same procedure over all the channels specified in the AMCA-Beacon request command.

5.1.12.4 Multi-channel hello

The Multi-channel hello mechanism allows a device to announce its designated listening channel to its one-hop neighbor devices.

After successfully performing the asymmetric active scan and the association, the device shall transmit a multi-channel hello command on each channel of the channel page used in the scan and the association. The device can request multi-channel hello of neighbors by setting the Hello Request of the multi-channel hello command to one. When its neighbors receive the multi-channel hello command with Hello Request set to one, each neighbor shall transmit a multi-channel hello command on the designated listening channel of the requesting device with Hello Request set to zero.

5.1.12.5 Channel probe

Channel Probe can probe other channels and switch to a better channel. After switching to the new channel, the device shall broadcast a multi-channel hello command to its one-hop neighbors to notify them of the new channel.

The channel probe over a specified channel is requested using the MLME-SCAN.request primitive with the ScanType parameter set to 0x05.

The device can check the condition of its designated listening channel by using the handshake mechanism. The procedure of the handshake channel probing is described as follows.

The request device sends a channel probe request command frame to one of its neighbors on the designated listening channel of the neighbor. On receipt of the channel probe request command, the neighbor sends a channel probe reply frame back to the request device on the originator's channel indicating in the channel probe request command. The request device shall check the LQI or RSSI of the channel probe reply frame upon receiving it. The request device determines that the link quality of the channel is bad if the device have not received the channel probe reply frame after [*aBaseSuperframeDuration* × (2^n + 1)] symbols after sending the probe request, where n is the value of the *ScanDuration* parameter.

5.2 MAC frame formats

5.2.1 General MAC frame format

Replace Figure 35 with the following figure:

<table>
<tr><th>Octets: 1/2</th><th>0/1</th><th>0/2</th><th>0/1/2/8</th><th>0/2</th><th>0/1/2/8</th><th>0/1/5/6/10/14</th><th colspan="2">variable</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Frame Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="8">MHR</td><td colspan="2">MAC Payload</td><td>MFR</td></tr>
</table>

Figure 35—General MAC frame format

Change the second paragraph as indicated:

The fields of the MHR appear in a fixed order; however, some ~~the addressing~~ fields may not be included in all frames.

5.2.1.1 Frame Control field

Change 5.2.1.1 as indicated:

The Frame Control field contains information defining the frame type, addressing fields, and other control flags. The Frame Control field for the beacon, data, acknowledgment, and MAC command frames shall be formatted as illustrated in Figure 36. The Frame Control fields for Low Latency frames and Multipurpose frames are specified in 5.2.2.5.1 and 5.2.2.6, respectively.

Replace Figure 36 with the following figure:

Bits: 0–2	3	4	5	6	7	8	9	10–11	12–13	14–15
Frame Type	Security Enabled	Frame Pending	AR	PAN ID Compression	Reserved	Sequence Number Suppression	IE List Present	Dest. Addressing Mode	Frame Version	Source Addressing Mode

Figure 36—Format of the Frame Control field for beacon, data, acknowledgment, and MAC command frames

5.2.1.1.1 Frame Type field

Change Table 2 as indicated:

Table 2—Values of the Frame Type field

Frame Type value b_2 b_1 b_0	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100	LLDN
101	Multipurpose
110~~0~~–111	Reserved

Insert before 5.2.1.1.2 the following text:

Frame formats are specified in 5.2.2.

5.2.1.1.3 Frame Pending field

Insert the following paragraphs before the last paragraph in 5.2.1.1.3:

When operating in low-energy (LE) CSL mode, the frame pending bit may be set to one to indicate that the transmitting device has back-to-back frames to send to the same recipient and expects the recipient to keep the radio on until the frame pending bit is reset to zero.

When operating in TSCH mode, the frame pending bit can be set to one to indicate that the recipient should stay on in the next timeslot and on the same channel if there is no link scheduled.

5.2.1.1.5 PAN ID Compression field

Change in 5.2.1.1.5 text as follows and insert Table 2a:

The PAN ID Compression field specifies whether the MAC frame is to be sent containing ~~only~~ one or neither of the PAN identifier fields ~~when both source and destination addresses are present. If this field is set to one and both the source and destination addresses are present, the frame shall contain only the Destination PAN Identifier field, and the Source PAN Identifier field shall be assumed equal to that of the destination. If this field is set to zero, then the PAN Identifier field shall be present if and only if the corresponding address is present.~~ The PAN ID Compression field shall take one of the values shown in Table 2a.

Table 2a—PAN ID Compression

Source address	Destination address	Frame version	Frame types	Source PAN ID	Destination PAN ID	PAN ID com- pression
Not Present	Not Present	0b00 or 0b01	Acknowledgment	Not Present	Not Present	0
Not Present	Not Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Not Present	0
Not Present	Not Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Present	1
Present	Not Present	0b00 or 0b01	Beacon, Data, Acknowledgment, MAC	Present	Not Present	0
Present	Not Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Present	Not Present	0
Present	Not Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Not Present	1
Not Present	Present	0b00 or 0b01	Beacon, Data, Acknowledgment, MAC	Not Present	Present	0
Not Present	Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Present	0
Not Present	Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Not Present	1
Present	Present	0b00 or 0b01	Beacon, Data, Acknowledgment, MAC	Present	Present	0
Present	Present	0b00 or 0b01	Beacon, Data, Acknowledgment, MAC	Not Present	Present	1
Present	Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Present	0
Present	Present	0b10	Beacon, Data, Acknowledgment, MAC, Multipurpose	Not Present	Not Present	1

Insert before 5.2.1.1.6 the following new subclauses (5.2.1.1.5a−5.2.1.1.5b):

5.2.1.1.5a Sequence Number Suppression

When set to one, this field indicates suppression of the Sequence Number field in the frame, and the sequence number shall be omitted. When set to zero, the Sequence Number field is present. If the Frame Version field is 0b00 or 0b01, the Sequence Number Suppression field shall be zero.

5.2.1.1.5b IE List Present field

The IE List Present field shall be set to one if IEs are contained in the frame. This field shall be set to zero otherwise. If the Frame Version field is b00 or b01, the IE List Present field shall be zero.

Change 5.2.1.1.6 as follows:

5.2.1.1.6 Destination Addressing Mode field

The Destination Addressing Mode field shall be set to one of the values listed in Table 3.

If this field is equal to zero and the Frame Type field specifies a data or command frame and the Frame Version field is set to 0b00 or 0b01 ~~does not specify that this frame is an acknowledgment or beacon frame~~, the Source Addressing Mode field shall be nonzero, implying that the frame is directed to the PAN coordinator with the PAN identifier as specified in the Source PAN Identifier field.

If the Frame Type field does not specify an LLDN or multipurpose frame, and the Source Addressing and Destination Addressing Mode fields are set to zero, and the PAN ID Compression field is set to one, the Frame Version field (described in 5.2.1.1.7) shall be set to 0b10.

For frames with a Frame Version of 0b10, the destination address or destination PAN ID or both may be omitted. Upon reception of a frame with no destination address or PAN ID, if *macImplicitBroadcast* is TRUE, the destination address filtering steps in 5.1.6.2 shall be bypassed and the frame shall be accepted if the other filtering steps are successful.

Table 3—Possible values of the Destination Addressing Mode and Source Addressing Mode fields

Addressing mode value b_1 b_0	Description
00	PAN Identifier and Address fields are not present.
01	Address field contains an 8-bit simple address. ~~Reserved~~.
10	Address field contains a short address (16 bit).
11	Address field contains an extended address (64 bit).

5.2.1.1.7 Frame Version field

Change in 5.2.1.1.7 the second paragraph and insert Table 3a as follows:

The Frame Version field shall take one of the nonreserved values in Table 3a. ~~This field shall be set to 0x00 to indicate a frame compatible with IEEE Std 802.15.4-2003 and 0x01 to indicate an IEEE 802.15.4 frame. All other field values are reserved.~~ Details on frame compatibility are described in 5.2.3.

Table 3a—Possible values of the Frame Version field

Frame version value b_1 b_0	Description
00	Frame compatible with IEEE Std 802.15.4-2003
01	Frames introduced with IEEE Std 802.15.4-2006
10	Frame compatible with IEEE Std 802.15.4
11	Reserved

5.2.1.1.8 Source Addressing Mode field

Change the second paragraph of 5.2.1.1.8 as follows:

If this field is equal to zero and the Frame Type field ~~does not specify that this frame is an acknowledgment frame~~ specifies a data or command frame, and the Frame Version field is set to 0b00 or 0b01, the Destination Addressing Mode field shall be nonzero, implying that the frame has originated from the PAN coordinator with the PAN identifier as specified in the Destination PAN Identifier field.

5.2.1.4 Destination Address field

Change the first paragraph of 5.2.1.4 as follows:

The Destination Address field, when present, with a length specified in the Destination Addressing Mode field of the Frame Control field as described in 5.2.1.1.6, specifies the address of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid address by all devices currently listening to the channel.

5.2.1.7 Auxiliary Security Header field

Insert the following subclause (5.2.1.7a) and figure before 5.2.1.8:

5.2.1.7a Information Element field

The IE List field is variable length and contains one or more IE. This field is comprised of the Header IE and Payload IE subfields. This field shall be present only if the IE List Present field in the Frame Control field is set to one. The format of the IE List field is shown in Figure 5.2.2. Each IE consists of a descriptor and an optional payload as described in 5.2.4. This standard does not limit the number of IEs within the IE list.

Header IEs, if present, follow the Auxiliary Security Header and are part of the MHR.

Payload IEs, if present, follow the MHR and are considered part of the MAC payload, i.e., they may be encrypted. A list of payload IEs may require termination. See 5.2.4.22 for details on termination.

Octets: variable		variable	0/2	variable		variable	0/3
Header IE Number 1	…	Header IE Number n	Termination IE	Payload IE Number 1	…	Payload IE Number m	Termination IE

Figure 36u—Format of IE List field

5.2.2 Format of individual frame types

5.2.2.1 Beacon frame format

Change in 5.2.2.1 the second paragraph as follows:

The format of the beacon frame for Frame Versions 0b00 or 0b01 shall be as illustrated in Figure 38. If the Frame Version is 0b00 or 0b01, the Superframe Specification, GTS, and Pending Address fields are always present. If the frame version is 0b10, the IE List Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 5.2.1.1.5b, 5.2.1.7a, and 5.2.4. The Superframe Specification, GTS, and Pending Address fields are optional if IEs are used, but IE terminators shall be used if they are present. These fields shall be present if an unformatted (not payload IE) beacon payload is present. The GTS fields, if present, shall be formatted as illustrated in Figure 39, and the pending address fields, if present, shall be formatted as illustrated in Figure 40.

Insert the following paragraph and Figure 40a before 5.2.2.1.1:

The enhanced beacon is differentiated from the legacy beacon by the frame version being set to 0b10. The MAC frame for the Enhanced Beacon frame shall be formatted as illustrated in Figure 40a.

<table>
<tr><th>Octets: 1/2</th><th>0/1</th><th>variable</th><th>0/1/5/6/10/14</th><th colspan="2">variable</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td rowspan="2">Addressing fields</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Beacon Payload</td><td rowspan="2">FCS</td></tr>
<tr><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="5">MHR</td><td colspan="2">MAC payload</td><td>MFR</td></tr>
</table>

Figure 40a—Enhanced Beacon frame format

5.2.2.1.1 Beacon frame MHR fields

Change the text of 5.2.2.1.1 as follows:

The Frame Type field shall contain the value that indicates a beacon frame, as shown in Table 2, and the Source Addressing Mode field shall be set to indicate the beacon source addressing mode, as defined in 5.2.1.4. The Security Enabled field shall be set to one if security is enabled and the Frame Version field is not zero. If a broadcast data or command frame is pending, the Frame Pending field shall be set to one. If the Frame Version field is not 0b10, all other fields in the Frame Control field shall be set to zero and ignored on reception.

The Sequence Number field shall contain the current value of *macBSN* or *macEBSN* if sending an enhanced beacon. As a device may be sending both beacons and enhanced beacons, separate sequence numbers shall be maintained. If sending an enhanced beacon, the sequence number may be suppressed by setting the Sequence Number Suppression field in the Frame Control field. For DSME-enabled PANs, the Sequence Number field shall contain the value of *macEBSN*. When the frame version field is 0b00-0b01 the Sequence Number field shall be present.

When the Frame Version field is 0b00-0b01, the ~~The~~ addressing fields shall comprise only the source address fields. The Source PAN Identifier and Source Address fields shall contain the PAN identifier and address, respectively, of the device transmitting the beacon.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the beacon frame, as specified in 5.2.1.7.

When the Frame Version field is 0b10 (enhanced beacon frame), the MHR may or may not contain a sequence number, and may contain any addressing fields supported by the general frame format. The order of MHR fields of the enhanced beacon frame shall conform to the order of the general MAC frame as illustrated in Figure 35. The Source PAN Identifier field, when present, shall contain the PAN identifier. The Source Address field shall contain the MAC address of the device transmitting the beacon. When the beacon is generated in response to an EBR, the Frame Version field shall be set to 0b10 (enhanced beacon), and the Destination Address field shall contain the source address contained in the received EBR.

Insert a new subclause (5.2.2.1.1a) before 5.2.2.1.2 as follows:

5.2.2.1.1a Information Elements field

When the IE List Present field of the Frame Control field is set to one, the enhanced beacon contains header and/or payload IEs, refer to 5.2.4 for further details.

When generated in response to an EBR that contained PIB attributes in the EB Filter (as described in 5.2.4.18), the content of the EB shall include the IEs corresponding to the requested PIB attributes shown in Table 3b. A device supporting filtering by the PIB attribute listed in the first column shall also support the IEs in the “IEs to include” column in Table 3b. When generated in response to an EBR that contained a requested IE list as described in 5.3.7.2.2, the EB shall also contain the requested IEs. The EB frame includes termination IEs as needed.

Table 3b—EBR IEs per enabled attribute

Attribute Request Identifier	PIB attribute	IE type	IEs to include
1	*macTSCHenabled*	MLME Payload	TSCH Synchronization (5.2.4.13), TSCH Slotframe and Link (5.2.4.14), TSCH Timeslot (5.2.4.15), Channel Hopping Sequence (5.2.4.16)
2	*macDSMEenabled*	Header	DSME PAN Descriptor (5.2.4.9)
3	*macLEenabled*	Header	LE CSL or LE RIT (5.2.4.7, 5.2.4.8)
4	*macHopping-Enabled*	MLME Payload	Channel Hopping Sequence (5.2.4.16) and Hopping Timing (5.2.4.17)

5.2.2.2 Data frame format

Replace Figure 46 with the following new Figure 46:

<table>
<tr><th>Octets: 2</th><th>0/1</th><th>variable</th><th>0/1/5/6/10/14</th><th colspan="2">variable</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td rowspan="2">Addressing fields</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Data Payload</td><td rowspan="2">FCS</td></tr>
<tr><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="5">MHR</td><td colspan="2">MAC Payload</td><td>MFR</td></tr>
</table>

Figure 46—Data frame format

5.2.2.2.1 Data frame MHR fields

Change the text in 5.2.2.2.1 as follows:

The Frame Type field shall contain the value that indicates a data frame, as shown in Table 2. The Security Enabled field shall be set to one if security is enabled and the Frame Version field is not zero. All other fields in the Frame Control field shall be set appropriately according to the intended use of the data frame.

The Sequence Number field, if present, shall contain the current value of *macDSN*. If the frame version is 0b10, the sequence number may be suppressed by setting the Sequence Number Suppression Field in the Frame Control field.

The addressing fields ~~shall~~ comprise the destination address fields and/or the source address fields, dependent on the settings in the Frame Control field.

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the data frame, as specified in 5.2.1.7.

If the frame version is 0b10, the IE List Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 5.2.1.1.5b, 5.2.1.7a, and 5.2.4.

5.2.2.3 Acknowledgment frame format

Change in 5.2.2.3 the first paragraph as follows:

For frame version b00-b01, ~~T~~the acknowledgment frame shall be formatted as illustrated in Figure 47a.

Insert after Figure 47 the following paragraph and Figure 47a:

The enhanced acknowledgment frame format (frame version set to 0b10) shall be formatted as illustrated in Figure 47a.

<table>
<tr><th>Octet: 1/2</th><th>0/1</th><th>0/2</th><th>0/1/2/8</th><th>0/2</th><th>0/1/2/8</th><th>0/1/5/6/ 10/14</th><th colspan="2">variable</th><th>varia- ble</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="8">MHR</td><td colspan="2">MAC Payload</td><td>MFR</td></tr>
</table>

Figure 47a—Enhanced Acknowledgement frame format

Insert in 5.2.2.3 the following text before the last paragraph, and change the last paragraph as follows:

The IE List Present field shall be set if an enhanced acknowledgement contains IEs.

If the acknowledgment frame contains fields other than Frame Control field, Sequence Number field, and FCS field, the Frame Version field shall be set to 0b10.

The Source Addressing Mode field defines the presence of the Source Address field and shall be set as appropriate for the address of the device transmitting the acknowledgment frame as described in Table 3.

The Source Address field, when present, contains the address of the device originating the acknowledgment frame. The use of any other Addressing field (source or destination address in any format of Table 3) shall not be precluded.

If the Security Enable field in the Frame Control field is set to one, the Auxiliary Security Header field shall match the corresponding field of the frame that is being acknowledged except that the Frame Counter Suppression field in the Auxiliary Security Header should be set, and the frame counter omitted from the frame (although it is still used to construct the CCM* nonce).

If the IEs list present field in the Frame Control field is set to one, IEs corresponding to the optional functional modes enabled, per Table 52a, are to be included in the frame.

The Sequence Number field<u>, when present,</u> shall contain the value of the sequence number received in the frame for which the acknowledgment is to be sent.

5.2.2.4 MAC command frame format

Replace Figure 48 with the following new Figure 48:

<table>
<tr><th>Octets: 2</th><th>0/1</th><th>variable</th><th>0/1/5/6/
10/14</th><th colspan="2">variable</th><th>1</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td rowspan="2">Addressing fields</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Command Frame Identifier</td><td rowspan="2">Command Payload</td><td rowspan="2">FCS</td></tr>
<tr><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="5">MHR</td><td colspan="3">MAC Payload</td><td>MFR</td></tr>
</table>

Figure 48—MAC command frame format

5.2.2.4.1 MAC command frame MHR fields

Change in 5.2.2.4.1 the second and third paragraphs, and insert a new last paragraph, as follows:

The Sequence Number field<u>, if present,</u> shall contain the current value of macDSN. <u>If the Frame Version field is 0b10, the sequence number may be suppressed by setting the Sequence Number Suppression field in the Frame Control field. If the Frame Version field is 0b00 or 0b01 the Sequence Number field shall be present.</u>

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the Frame Control field. <u>If the Frame Version field is 0b10 then either address may be suppressed.</u>

The Auxiliary Security Header field, if present, shall contain the information required for security processing of the MAC command frame, as specified in 5.2.1.7.

<u>If the frame version is 0b10, the IE List Present field in the Frame Control field may be set to one to indicate the presence of IEs as described in 5.2.1.1.5b, 5.2.1.7a, and 5.2.4.</u>

Insert before 5.2.3 the following new subclauses (5.2.2.5−5.2.2.8.2):

5.2.2.5 Low latency Command Frame format

5.2.2.5.1 General LL frame format

The general LLDN frame shall be formatted as illustrated in Figure 48a.

Octets: 1	**0/1**	**0/1/5/6/10/14**	**variable**	**2**
Frame Control	Sequence Number	Auxiliary Security Header	Frame Payload	FCS
MHR			MAC payload	MFR

Figure 48a—General LL frame format

The order of the fields of the LLDN frame shall conform to the order of the general MAC frame as illustrated in Figure 35.

Four subframe types are defined: LL Beacon, LL-data, LL-Acknowledgment, and LL-MAC command. These subframe types are specified in 5.2.2.5.2, 5.2.2.5.3, 5.2.2.5.4, and 5.2.2.5.5, respectively.

The Frame Control field contains information defining the subframe type of the LLDN frame. The Frame Control field shall be formatted as illustrated in Figure 48b.

Bits: 0–2	**3**	**4**	**4**	**6–7**
Frame Type	Security Enabled	Frame Version	ACK Request	Sub Frame Type

Figure 48b—Format of the Frame Control field (LLDN frame)

NOTE 1—The LLDN frame will be rejected by devices compliant to IEEE Std 802.15.4-2011 since the Frame Type value is listed as “reserved” by IEEE Std 802.15.4-2011. The position of the Frame Type should not be changed in future versions of the protocol.

The Frame Type field shall contain the value that indicates an LLDN frame, as shown in Table 2.

NOTE 2—The Frame Type field corresponds to the Frame Type field of the general MAC frame format in 5.2.1 in meaning and position. The frame type for LLDN frames allows efficient recognition of LLDN frames with a Frame Control field of 1 octet, but allows the usage of all other MAC frames within the superframe structure of an LLDN.

The Security Enabled field is 1 bit in length, and it shall be set to one if the frame is protected by the MAC sublayer or set to zero otherwise. The Sequence Number field and the Auxiliary Security Header field of the MHR shall be present only if the Security Enabled field is set to one.

The Frame Version field specifies the version number corresponding to the frame. This field shall be set to zero to indicate a frame compatible with IEEE Std 802.15.4. A value of one shall be reserved for future use.

The ACK Request field specifies whether an acknowledgment is required from the recipient device on receipt of a data or MAC command frame. If this field is set to one, the recipient device shall send an acknowledgment frame only if, upon reception, the frame passes the third level of filtering as described in 5.1.6.2. If this field is set to zero, the recipient device shall not send an acknowledgment frame.

The Frame Subtype field indicates the type of the LLDN frame. It shall be set to one of the values listed in Table 3c.

Table 3c—Values of Frame Subtype field (LLDN frame)

Frame Subtype value b_7 b_6	Description
00	LL-Beacon
01	LL-Data
10	LL-Acknowledgment
11	LL-MAC command

The Sequence Number field specifies the sequence identifier for the frame. The Sequence Number field shall be present only if the Security Enabled field is set to one.

The Auxiliary Security Header field has a variable length and specifies information required for security processing, including how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (refer to 7.5). This field shall be present only if the Security Enabled field is set to one. For details on formatting, refer to 7.4.

The Frame Payload field has a variable length and contains information specific to individual subframe types of an LLDN frame.

5.2.2.5.2 LL Beacon frame format

The LL Beacon frame is sent during the beacon slot in every superframe. The LL Beacon frame shall be formatted as illustrated in Figure 48c.

Octets: 1	0/1	0/1/5/6/10/14	1	1	1	1	0/1	variable	2
Frame Control	Sequence Number	Auxiliary Security Header	Flags	LLDN PAN coordinator ID field	Configuration Sequence Number	Timeslot Size	Number of Base Timeslots in Superframe	Group Acknowledgment	FCS
MHR			MAC Payload						MFR

Figure 48c—Format of the LL Beacon Frame

The order of the fields of the LL Beacon frame shall conform to the order of the general LLDN frame as illustrated in Figure 48a.

The LL Beacon frame has a short MHR containing the Frame Control field of one octet.

In the Frame Control field, the Frame Type field shall contain the value that indicates an LLDN frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL Beacon frame, as shown in Table 3c.

The Flags field contains control information. The structure of the Flags field is shown in Figure 48d.

Bits: 0–2	3	4	5–7
Transmission State	Transmission Direction	Reserved	Number of Base Timeslots per Management Timeslot

Figure 48d—Structure of Flags field of LL Beacon frame

The Transmission State field defines the transmission state. The values for the different transmission states are specified in Table 3d.

Table 3d—Transmission State settings

Bits 0–2	Transmission State
000	Online State (described in 5.1.9.4)
100	Discovery State (described in 5.1.9.2)
110	Configuration State (described in 5.1.9.3)
111	State Reset: The devices reset their state of the discovery or configuration

The Transmission Direction field indicates the transmission direction of all bidirectional timeslots during this superframe. If the Transmission Direction field is set to zero, the direction of all bidirectional timeslots is uplink (from LLDN device to LLDN PAN coordinator). If the Transmission Direction field is set to one, the direction of all bidirectional timeslots is downlink (from LLDN PAN coordinator to LLDN device). The Transmission Direction field is only used in online state.

The Number of Base Timeslots per Management Timeslot field contains the number of base timeslots per management timeslot. This value applies to both the downlink and the uplink management timeslot. A value of zero indicates that there are no management timeslots available in the superframe.

The LLDN PAN coordinator ID field contains the 8-bit simple address (i.e., *macSimpleAddress*) of the LLDN PAN coordinator.

The Configuration Sequence Number field contains an integer number that identifies, together with the LLDN PAN coordinator ID, the current configuration of the LLDN.

The Timeslot Size field defines the length of a base timeslot through the maximum expected number of octets of the data payload of an LL-data frame. The actual timeslot size in octets is calculated as

$$tTS := (p \times sp + (m + n) \times sm + macMinSIFSPeriod \text{ symbols } \{\text{if } m + n \leq aMaxSIFSFrameSize \text{ octets}\} \text{ or } macMinLIFSPeriod \text{ symbols } \{\text{if } m + n > aMaxSIFSFrameSize \text{ octets}\}) / v$$

with the description and values for the 2 450 MHz PHY as an example as shown in Table 3e.

Table 3e—Example of a set of parameter and values

Variable	Description	Value for 2450 MHz PHY with no security enabled
p	Number of octets of PHY header	6 octets
m	Number of octets of MAC overhead (MHR + MFR)	3 octets for LL-Data frames
n	Maximum expected number of octets of data payload	Value of Timeslot Size field of LL-Beacon frame
sp	Number of symbols per octet in PHY header	2 symbols per octet
sm	Number of symbols per octet in PSDU	2 symbols per octet
v	Symbol rate	62 500 symbols/s

The Number of Base Timeslots in Superframe field contains an integer number that represents the number of base timeslots for LLDN devices immediately following the management timeslots of the superframe (corresponds to *macLLDNnumTimeSlots*). The Number of Base Timeslots in the Superframe field is only present in the Online state.

The Group Acknowledgment field is a bitmap of length (*macLLDNnumTimeSlots* – *macLLDNnumRetransmitTS*) bits, padded to a multiple of 8 bits, as shown in Figure 48e, to indicate successful transmissions by LLDN devices from the previous superframe. The size of the bitmap shall always be a multiple of 8 after padding with additional zeros at the end if necessary. In the separate group acknowledgment configuration, this field is not present in the LL Beacon. The Group Acknowledgment field is only present in online mode. The Group Acknowledgment field contains a bit field where each bit corresponds to a timeslot associated with an LLDN device excluding retransmission timeslots. Bit b0 of the Group Acknowledgement bitmap corresponds to the first timeslot after the *macLLDNnumRetransmitTS* retransmission timeslots, bit b1 of the Group Acknowledgment bitmap corresponds to the second timeslot, and so on. A bit value of one means the corresponding uplink transmission in the previous superframe was successful, and a bit value of zero means the corresponding uplink transmission in the previous superframe failed or there was no uplink transmission. In the latter case, the LLDN device is allocated a timeslot for retransmission in the current superframe. Because concatenated timeslots are multiples of base timeslots, a concatenated timeslot of length of *n* base timeslots shall have *n* bits in the group acknowledgment bitmap at the corresponding positions. If the data frame has been received during a shared group timeslot, all corresponding bits of this shared group timeslot shall be set accordingly in the Group Acknowledgment bitmap.

Bits: 0	1	…	(*macLLDNnumTimeSlots* – *macLLDNnumRetransmitTS* – 1)	… n*8–1
Acknowledgment of transmission in time slot *macLLDNnumRetransmitTS* + 1	Acknowledgment of transmission in time slot *macLLDNnumRetransmitTS* + 2	…	Acknowledgment of transmission in time slot *macLLDNnumTimeSlots*	Padding

Figure 48e—Structure of Group Acknowledgment bitmap

5.2.2.5.3 LL-Data frame format

The LL-Data frame is sent during online mode in device timeslots. The LL-data frame shall be formatted as illustrated in Figure 48f.

Octets: 1	0/1	0/1/5/6/10/14	variable	2
Frame Control	Sequence Number	Auxiliary Security Header	Data Payload	FCS
MHR			MAC Payload	MFR

Figure 48f—Format of LL-Data frame

The order of the fields of the LL-Data frame shall conform to the order of the general MAC frame as illustrated in Figure 35.

The LL-data frame has a short MHR containing the Frame Control field of one octet.

In the Frame Control field, the Frame Type field shall contain the value that indicates an LLDN frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-data frame, as shown in Table 3c.

The payload of an LL-data frame shall contain the sequence of octets that the next higher layer has requested the MAC sublayer to transmit.

5.2.2.5.4 LL-Acknowledgment frame format

The LL-Acknowledgment frame is sent during online mode in bidirectional timeslots. The LL-Acknowledgment frame shall be formatted as illustrated in Figure 48g.

Octets: 1	0/1	0/1/5/6/10/14	1	variable	2
Frame Control	Sequence Number	Auxiliary Security Header	Acknowledg-ment type	Acknowledgment payload	FCS
MHR			MAC payload		MFR

Figure 48g—Format of the LL-Acknowledgment frame

The order of the fields of the LL-Acknowledgment frame shall conform to the order of the general LLDN frame as illustrated in Figure 48a.

The LL-Acknowledgment frame has a short MHR containing the Frame Control field of one octet.

In the Frame Control field, the Frame Type field shall contain the value that indicates as LLDN frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-Acknowledgment frame, as shown in Table 3c.

The Acknowledgment Type field indicates the type of frame that is acknowledged or the type of acknowledgment. Possible values are listed in Table 3f.

Table 3f—Acknowledgment types

Acknowledged frame type/Acknowledgment type	Acknowledgment payload
Configuration Request	No
Data	No
Data Group ACK	Yes
Discover Response	No

The Acknowledgment Payload field is only available in certain acknowledgment types as depicted in Table 3f. The structure and the length of the Acknowledgment Payload field depends on the value of the Acknowledgment Type field.

The structure of the Acknowledgment Payload field of the Data Group ACK frame is shown in Figure 48h.

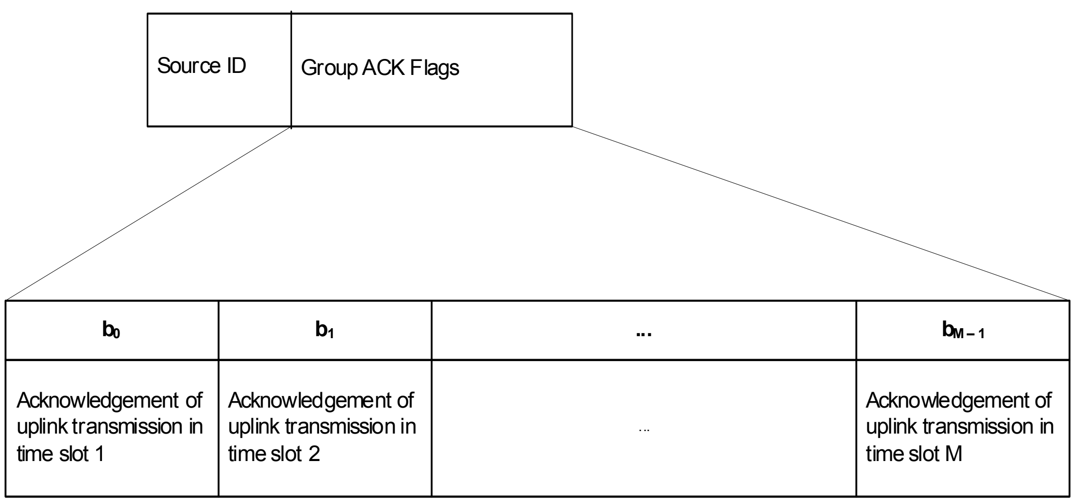

Figure 48h—Format of the Data Group ACK frame

The Source ID field shall be an 8-bit simple address that identifies the transmitting LLDN PAN coordinator.

The size of the bitmap shall be equal to the smallest multiple of 8 that is greater than or equal to the number of timeslots used for uplink transmissions by the LLDN devices.

The Group Ack Flags field is a bitmap of size equal to the smallest multiple of 8 that is greater than or equal to the number of uplink timeslots that indicates the states of transmissions of the LLDN devices in the uplink timeslots of the current superframe. A bit set to one indicates the fact that the coordinator received the data frame successfully in the corresponding timeslot. A value of zero means, that the coordinator failed in receiving a data frame in the corresponding slot from of the LLDN device.

5.2.2.5.5 LL-MAC Command frame format

There are different types of LL-MAC Command frames sharing a common, general structure, differing only at the Command Payload. The LL-MAC command frame shall be formatted as illustrated in Figure 48i.

Octets: 1	0/1	0/1/5/6/10/14	1	variable	2
Frame Control	Sequence Number	Auxiliary Security Header	Command Frame Identifier	Command Payload	FCS
MHR			MAC payload		MFR

Figure 48i—Format of the LL-MAC Command frame

The order of the fields of the LL-MAC Command frame shall conform to the order of the general LLDN frame as illustrated in Figure 48a.

The LL-MAC command frame has a short MHR containing the Frame Control field of one octet.

In the Frame Control field, the Frame Type field shall contain the value that indicates an LLDN frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

The Command Frame Identifier field identifies the MAC command being used. This field shall be set to one of the nonreserved values listed in Table 5.

The Command Payload field contains the MAC command itself. The formats of the individual commands are described in 5.3.10. The Command Payload field is of variable length and contains data specific to the different command frame types.

5.2.2.6 Multipurpose frame format

The multipurpose frame type provides a flexible format that may used for a variety of purposes. The format supports a short and long form of the Frame Control field, and allows for all MHR fields to be present or elided as specified by the generating service (via the MCPS-DATA.request primitive parameters). Some suggested uses are described in Annex I.

The multipurpose frame shall be formatted as illustrated in Figure 48j.

<table>
<tr><th>Octets: 1/2</th><th>0/1</th><th>0/2</th><th>0/1/2/8</th><th>0/1/2/8</th><th>0/1/2/8</th><th>0/1/5/6/10/14</th><th colspan="2">variable</th><th>variable</th><th>2</th></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td colspan="2">Information Elements</td><td rowspan="2">Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td><td>Header IEs</td><td>Payload IEs</td></tr>
<tr><td colspan="8">MHR</td><td colspan="2">MAC Payload</td><td>MFR</td></tr>
</table>

Figure 48j—Multipurpose frame format

The order of the fields of the multipurpose frame shall conform to the order of the general MAC frame as illustrated in Figure 35. Multipurpose frames are treated in the same manner as Data frames, and their content is passed to/from the next higher layer using the MCPS DATA primitives.

The Frame Control field in the multipurpose frame is either 1 or 2 octets. The base Frame Control field is 1 octet. It allows for configuring 0/1/2/8-octet source and destination addresses. By default sequence number is present, PAN ID is not present, and security is not enabled.

Frames that require security, want to specify acknowledgment request or frame pending, want to carry a PAN ID, suppress the sequence number, or specify IEs need to use a 2 octet frame control. Fields occur in the MHR in the order of their selector bits when present. The Frame Control field shall be formatted as illustrated in Figure 48k.

Bit: 0–2	3	4–5	6–7	8	9	10	11	12-13	14	15
Frame Type	Long Frame Control	Destination Address Mode	Source Address Mode	PAN ID Present	Security Enabled	Seq Number Suppression	Frame Pending	Frame Version	Ack Request	IEs List Present

Figure 48k—Format of the Frame Control field (multipurpose frame)

5.2.2.6.1 Frame Type field

The Frame Type field shall contain the value that indicates a multipurpose frame, as shown in Table 2.

5.2.2.6.2 Long Frame Control field

The Long Frame Control field shall be set to zero to indicate a 1-octet Frame Control field (only 0b0 to 0b7 in Figure 48k are carried in the frame), and to one to indicate a 2-octet Frame Control field (0b0 to 0b15 in Figure 48k are carried in the frame).

5.2.2.6.3 Destination Address Mode field

The Destination Address Mode field is 2 bits in length and shall be set to one of the values listed in Table 3g. The length of the Destination Address field in the MHR shall correspond to the value of the Destination Address Mode field.

Table 3g—Destination Address Mode field

Dst Addr Mode b_5 b_4	Description
00	No Destination Address in MHR
01	1-octet Destination Address
01	2-octet Destination Address
11	8-octet Destination Address

5.2.2.6.4 Source Address Mode field

The Source Address Mode field is 2 bits in length and shall be set to one of the values listed in Table 3h. The length of the source address field in the MHR shall correspond to the value of the Source Address Mode field.

Table 3h—Source Address Mode field

Src Addr Mode b_7 b_6	Description
00	No Source Address in MHR
01	1-octet Source Address
01	2-octet Source Address
11	8-octet Source Address

5.2.2.6.5 PAN ID Present field

The PAN ID Present field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if the destination PAN ID is present in the MHR, otherwise it is set to zero and the PAN ID is not present in the MHR.

5.2.2.6.6 Security Enabled field

The Security Enabled field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if the frame is protected by the MAC sublayer and shall be set to zero otherwise.

5.2.2.6.7 Sequence Number Suppression field

The Sequence Number Suppression field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if the Sequence Number field is not to be included in the MHR. Otherwise it shall be set to zero.

5.2.2.6.8 Frame Pending field

The Frame Pending field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if the device sending the frame has more data for the recipient. This field shall be set to zero otherwise.

5.2.2.6.9 Frame Version field

The Frame Version field is present only if the Long Frame Control field is set to one. It specifies the version number corresponding to the long frame control frame. This field shall be set to zero.

5.2.2.6.10 Acknowledgment Request field

The Acknowledgment Request field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if an acknowledgment is required from the recipient device on receipt of a valid frame. Otherwise it shall be set to zero.

5.2.2.6.11 IEs List Present field

The IEs List Present field is 1 bit in length and is present only if the Long Frame Control field is set to one. It shall be set to one if IEs (refer to 5.2.4 for details) are to be included in the frame payload. Otherwise it shall be set to zero.

5.2.2.6.12 Sequence Number field

The Sequence Number field specifies the sequence identifier for the frame. It is normally present, but can be suppressed using the Sequence Number Suppression field as described in 5.2.2.6.7.

5.2.2.6.13 Destination PAN Identifier field

The Destination PAN Identifier field specifies the unique PAN identifier of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast PAN identifier, which shall be accepted as a valid PAN identifier by all devices currently listening to the channel. The source PAN ID is assumed to match the Destination PAN ID when using a multipurpose frame.

5.2.2.6.14 Destination Address field

The Destination Address field, when present, specifies the address of the intended recipient of the frame. A 16-bit value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid 16-bit short address by all devices currently listening to the channel.

5.2.2.6.15 Source Address field

The Source Address field, when present, specifies the address of the originator of the frame. It can be 0, 1, 2, or 8 octets in length, as indicated by the Source Address Mode field in the Frame Control field. It specifies the address of the originator of the frame.

5.2.2.6.16 Auxiliary Security Header field

The Auxiliary Security Header field specifies information required for security processing, including how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (refer to 7.5). This field shall be present only if the Security Enabled field is set to one. For details on formatting, refer to 7.4.

5.2.2.6.17 IEs field

The IEs field is variable in length and is present if the IEs Present field is set to one. The format of IEs is specified in 5.2.4. It contains Header IEs, followed by Payload IEs. Each type of IE list is terminated as required per 5.2.4.22.

5.2.2.6.18 Payload field

The Payload field has a variable length and contains unformatted (not in an IE) payload.

5.2.2.7 Multipurpose blink frame

5.2.2.7.1 General

The blink is a periodic transmission that has been designed to be of minimal size to make its transmission require the lowest amount of energy to yield maximum battery life. The MHR for a blink shall typically only contain a 1-octet Frame Control field, the Sequence Number field, and the Source Address field.

Transmission and reception of the blink is controlled and signaled using the MCPS-DATA primitives. Application of the blink for RFID is described in I.5.

The format of a multipurpose blink is shown in Figure 48l.

Octets: 1/2	0/1	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Seq Number	Dest. PAN ID	Src address	Aux Sec	payload	FCS

Figure 48l—Multipurpose blink frame

In the Frame Control field, the Frame Type shall contain the value that indicates a multipurpose frame, as shown in Table 2.

Source and destination address can be configured using a 1-octet frame control.

If the Destination PAN Identifier field is present, the PAN ID field in the frame control shall be set to one and a 2-octet frame control is required. The Destination PAN Identifier field shall contain the PAN Identifier of the device receiving the blink.

If protection is used for the blink, the Security Enabled field shall be set to one and a 2-octet frame control is required.

The Sequence Number field (normally present) shall be set to the current value of *macDSN*. If suppressed, a 2-octet frame control is required.

The Source Address field, if present, shall contain the extended address of the device originating the blink. All other Addressing fields shall be omitted. The presence of these Addressing fields shall be indicated by the Destination PAN Identifier field and the Source Addressing field of the Frame Control field, respectively (present if set; absent otherwise).

5.2.2.7.2 Blink payload field

The Blink Payload field is an optional sequence specified to be transmitted by the next higher layer, specified by the MSDU field of the MCPS-DATA.request primitive. The Payload field, if present, may encode an alternative identifier for the originating device, and/or other data. The details of this encoding are outside scope of this specification.

5.2.2.8 LE-multipurpose Wake-up frame

5.2.2.8.1 General

The LE Wake-up frame is a multipurpose frame containing an RZ Time header IE as described in 5.2.4.10. The frame format is shown in Figure 48m.

The MHR for a wake-up frame shall contain the Frame Control field, the Sequence Number field, the Destination PAN ID, and the Destination Address field as shown in Figure 48m.

Octets: 1	1	2	2	4	2
Frame Control	Sequence Number	Dest. PAN ID	Dest. Address	RZ Time Header IE	IE List Terminator

Figure 48m—Multipurpose Wake-up Frame

In the Frame Control field, the Frame Type shall be set to 0b101, i.e., a multipurpose frame, as shown in Table 2. If protection is used for the wake-up frame, the Security Enabled field shall be set to one.

The Sequence Number field shall be set to the current value of *macDSN*.

The Addressing fields shall comprise only the Destination Addressing fields. The Destination PAN ID and the Destination Address fields shall contain the PAN Identifier and the short (2-octet) address of the device receiving the wake-up frame.

5.2.2.8.2 Wake-up frame payload

The LE Wake-up frame has no payload.

5.2.3 Frame compatibility

Change in 5.2.3 the second paragraph and Table 4 as follows:

All unsecured frames specified in this standard are compatible with unsecured frames compliant with IEEE Std 802.15.4-2003 and IEEE Std 802.15.4-2011, with ~~two~~ the following exceptions: a coordinator realignment command frame with the Channel Page field present, as defined in 5.3.8, ~~and~~ any frame with a MAC Payload field larger than *aMaxMACSafePayloadSize*, any frame with the Frame Version field set to 0b10, a multipurpose frame, and an LLDN frame.

Compatibility for secured frames is shown in Table 4, which identifies the security operating modes for IEEE Std 802.15.4-2003, IEEE Std 802.15.4-2011, and this standard.

Table 4—Frame compatibility between IEEE Std 802.15.4-2003, IEEE Std 802.15.4-2011, and this standard

Frame Control field bit assignments		Functionality
Security enabled b_3	**Frame version b_{13} b_{12}**	
0	00	No security. Frames are compatible between IEEE Std 802.15.4-2003 and this standard.
0	01	No security. Frames are not compatible between IEEE Std 802.15.4-2003 and this standard, but are compatible with IEEE Std 802.15.4-2011.
0	10	No security. Frames are not compatible with IEEE Std 802.15.4-2003 or IEEE Std 802.15.4-2011.
1	00	Secured frame formatted according to IEEE Std 802.15.4-2003. This type of frame is not supported in this standard.

Table 4—Frame compatibility between IEEE Std 802.15.4-2003, IEEE Std 802.15.4-2011, and this standard (continued)

Frame Control field bit assignments		Functionality
Security enabled b_3	Frame version b_{13} b_{12}	
1	01	Secured frame formatted according to IEEE Std 802.15.4-2011~~this standard~~.
1	10	Secured frame formatted according to this standard.

Insert before 5.3 the following subclauses (5.2.4−5.2.4.22):

5.2.4 Information Element

5.2.4.1 General

An information element (IE) provides a flexible, extensible, and easily implementable method of encapsulating information. The general format of an IE consists of an identification (ID) field, a length field, and a content field. Multiple IEs may be concatenated, and elements with unknown ID values in a list of IEs can be skipped since their length is known. IEs provide a flexible container for information that allows for adding new IE definitions in future versions of the standard in a backwards-compatible way.

IEs may be either part of the MAC Header or MAC payload. Header IEs are used by the MAC to process the frame immediately, i.e., they cover security, addressing, etc., and are part of the MAC Header. Payload IEs are destined for another layer or SAP and are part of the MAC payload (but like command IDs they may not be encrypted). If IEs are contained in a frame, they always occur before other unstructured MAC payload elements.

Each IE starts with an IE descriptor. Format of the descriptor is different for Header and Payload IEs.

5.2.4.2 Header Information Elements

The Header IE descriptor consists of a length field, an Element ID that uniquely identifies the IE, and a type field that is set to zero. It is followed by the IE content. The format of the Header IE is shown in Figure 48n. The length field specifies the number of octets in the IE Content. The Element ID space is broken into managed and unmanaged spaces as shown in Table 4a. The allocation of Managed Element IDs shall be according to Table 4b for Header IEs.

Bit: 0-6	7–14	15	Octets: 0 … 127
Length	Element ID	Type = 0	IE Content

Figure 48n—Format of Header IEs

Table 4a—Header Element ID namespace

ID range	Function
0x00–0x19	Unmanaged IDs (described in 5.2.4.21)
0x1a–0xff	Managed IDs (described in 5.2.4.6)

Table 4b—Element IDs, Header IEs

Element ID	Content length	Name	Description
0x00–0x19	—	—	Unmanaged (i.e., implementation specific) IEs
0x1a	4	LE CSL	Defined in 5.2.4.7
0x1b	4	LE RIT	Defined in 5.2.4.8
0x1c	Variable	DSME PAN Descriptor	Defined in 5.2.4.9
0x1d	2	RZ Time	Defined in 5.2.4.10
0x1e	2	ACK/NACK Time-Correction	Defined in 5.2.4.11
0x1f	4	Group ACK (GACK)	Defined in 5.2.4.12
0x20	Variable	LowLatencyNetworkInfo	Information for association specific to LLDNs from the next higher layer
0x21–0x7d	—	Reserved	—
0x7e	0	List Termination 1	Signals the end of the Header IEs when followed by payload IEs as described in 5.2.4.22
0x7f	0	List Termination 2	Signals the end of the Header IEs when followed by an unformatted payload as described in 5.2.4.22
0x80–0xff	—	Reserved	—

5.2.4.3 Payload Information Elements

The Payload IE descriptor consists of an length field, a Group ID field, and a type field, which is set to one. It is followed by the IE content that may contain additional IE formatting. The general format of the Payload IE is shown in Figure 48o.

Bit: 0-10	11-14	15	Octets: 0 … 2047
Length	Group ID	Type = 1	IE Content

Figure 48o—Payload IE general format

The Group ID space is broken into managed and unmanaged spaces as shown in Table 4c.

Table 4c—Payload IE Group ID namespace

Group ID value	Description
0x0	Encapsulated Service Data Unit (ESDU) as described in 5.2.4.4
0x1	MLME (Nested)
0x2–0x9	Unmanaged
0xa–0xe	Reserved
0xf	List termination

5.2.4.4 Encapsulated Service Data Unit Information Elements

An Encapsulated Service Data Unit (ESDU) IE encapsulates a higher layer payload, the content is transparent to the MAC, and it shall have the format shown in Figure 48p. Upper layer content is sent as received, i.e., no octet ordering changes shall be made.

Bit: 0-10	11-14	15	Octets: 0 … 2047
Length	Group ID = 0	Type = 1	IE Content

Figure 48p—ESDU IE

5.2.4.5 MLME Information Elements

The MLME IE transports MAC management information. The MLME IE uses a nested format as shown in Figure 48q.

Bit: 0-10	11-14	15	Octets:2			Variable	…
Length = 0-2047	Group ID = 1	Type = 1	Length	Sub-ID	Type	Sub-IE Content	
Outer IE Descriptor			Sub-IE descriptor				Additional Sub-IEs

Figure 48q—MLME IE

Each IE nested within an MLME IE consists of a nested Sub-IE descriptor (consisting of a length field, a Sub-ID field, and a type field) followed by the IE content. Short and long forms of the nested IEs are shown in Figure 48r and Figure 48s. The Sub-ID space for nested MLME IEs is broken into managed and unmanaged spaces as shown in Table 4d (short format) and Table 4e (long format).

Bit: 0-7	8-14	15	Octets: 0 … 255
Length	Sub-ID	Type = 0 (short)	IE Content

Figure 48r—Format of the nested MLME IE of type short

Bit: 0-10	11-14	15	Octets: 0 … 2047
Length	Sub-ID	Type = 1 (long)	IE Content

Figure 48s—Format of the nested MLME IE of type long

Table 4d—Sub-ID allocation for short format

Sub-ID value	Content length	Name	Description
0x00–0x19	—	Reserved	—
0x1a	6	TSCH Synchronization	Information to synchronize to a TSCH PAN. ASN of TSCH device and its join priority, defined in 5.2.4.13).
0x1b	variable	TSCH Slotframe and Link	Slotframe and link information for joining a TSCH device, defined in 5.2.4.14.
0x1c	variable	TSCH Timeslot	Timeslot template being used by the TSCH device, defined in 5.2.4.15.
0x1d	5	Hopping Timing	Timing information used to synchronize to an unslotted hopper, defined in 5.2.4.17.
0x1e	variable	EB Filter	Response filter for EBR, defined in 5.2.4.18.
0x1f	5	MAC Metrics 1	MAC Metrics counters 0x91 through 0x99, defined in 5.2.4.19.
0x20	40	MAC Metrics 2	All MAC Metrics counters in Figure 48nn, defined in 5.2.4.20
0x21–0x3f	—	Reserved	—
0x40–0x7f	—	Unmanaged.	—

Table 4e—Sub-ID allocation for long format

Sub-ID value	Content length	Name	Description
0x0–0x8	—	Unmanaged	—
0x9	Variable	Channel Hopping Sequence	The Hopping Sequence being used by the device, defined in 5.2.4.16
0xa–0xf	—	Reserved	—

5.2.4.6 Defined information Elements

The IEs that follow this subclause are used by the optional functional organization modes indicated in Table 52a. When sending, an appropriate header descriptor shall be used. All multi-octet fields within IEs are sent LSB first, except where otherwise indicated.

5.2.4.7 LE CSL IE

The LE CSL IE shall be used in all enhanced acknowledgements if *macLEenabled* is TRUE.

The structure of the LE CSL element as illustrated in Figure 48t.

Octets: 2	**2**
CSL Phase	CSL Period

Figure 48t—Format of the LE CSL element

The CSL Phase field specifies CSL phase information. This field shall be set to one of the nonreserved values in Table 4f.

Table 4f—Values of the CSL Phase field for LE CSL

Range	Description
0x0000–0xffff	CSL phase, in 10 symbols

The CSL Period field specifies CSL period information. This field shall be set to one of the nonreserved values in Table 4g.

Table 4g—Values of the CSL Period field for LE CSL

Range	Description
0x0000–0xffff	CSL period, in 10 symbols

5.2.4.8 LE RIT IE

The structure of the LE RIT IE is illustrated in Figure 48w.

Octets : 1	**1**	**2**
Time to 1st Listen	Number of Repeat Listen	Repeat Listen Interval

Figure 48u—Format of the LE RIT element

Time to 1st Listen and Repeat Listen Interval are in the same units as *macRITPeriod*, and Number of Repeat Listen < ((*macRITPeriod* − Time to 1st Listen)/(RepeatListenInterval)).

5.2.4.9 DSME PAN descriptor IE

The DSME PAN Descriptor IE shall be included in enhanced beacons that are sent every beacon interval in a DSME-enabled PAN.

The format of the DSME PAN Descriptor element shall be as illustrated in Figure 48v.

Octets:2	variable	1	8	variable	Variable	0/7
Superframe Specification	Pending Address	DSME Superframe Specification	Time Synchronization Specification	Beacon Bitmap	Channel Hopping Specification	Group ACK Specification

Figure 48v—Format of DSME PAN Descriptor IE

The Superframe Specification field is described in 5.2.2.1.2.

The Pending Address field is described in 5.2.2.1.6.

The DSME Superframe Specification field is described in 5.2.4.9.1.

The Time Synchronization Specification field is described in 5.2.4.9.2.

The Beacon Bitmap field is described in 5.2.4.9.3.

The Channel Hopping Specification field is described in 5.2.4.9.4. This field is valid only in the channel hopping mode (i.e., the value of Channel Diversity Mode field in DSME Superframe Specification is set to one).

The Group ACK Specification field is described in 5.2.4.9.5. This field is valid only if the group ACK is enabled (i.e., the value of GACK Flag field in DSME Superframe Specification is set to one).

5.2.4.9.1 DSME superframe specification

The DSME Superframe Specification field shall be formatted as illustrated in Figure 48w.

Bits: 0-3	4	5	6	7
Multi-superframe Order (MO)	Channel Diversity Mode	GACK Flag	CAP Reduction Flag	Deferred Beacon Flag

Figure 48w—Format of the DSME Superframe Specification field

The Multi-superframe Order field shall specify the length of time during which a group of superframes that is considered as one multi-superframe (i.e., receiver enabled). See 5.1.10.1 for an explanation of the relationship between the Multi-superframe Order and the multi-superframe duration.

The Channel Diversity Mode field shall indicate the type of channel diversity. If this value is zero, the DSME-enabled PAN operates on channel adaptation mode. If this value is one, the DSME-enabled PAN operates on channel hopping mode.

The GACK Flag field shall indicate whether the group ACK is enabled on the DSME-enabled device. If the GACK Flag field is set to one, the superframe of the transmitting device shall use group ACK mechanism as described in 5.1.10.3. If the GACK Flag field is set to zero, the group ACK is disabled.

The CAP Reduction Flag field shall be set to one if the CAP reduction is enabled, otherwise the CAP Reduction Flag field shall be set to zero.

The Deferred Beacon Flag shall be set to one if the device uses CCA before transmitting beacon frame, otherwise the bit shall be set to zero if the device shall not use CCA before transmitting beacon.

5.2.4.9.2 Time synchronization specification

The Time Synchronization Specification field shall be formatted as illustrated in Figure 48x.

Octets: 6	2
Beacon Timestamp	Beacon Offset Timestamp

Figure 48x—Format of the Time Synchronization Specification field

The Beacon Timestamp field shall specify the time of beacon transmission in units of microseconds for time synchronization. Its value is the start time of the beacon slot.

The Beacon Offset Timestamp field specifies that the time difference between the start time of a superframe and the actual time of transmitting a beacon frame. It reflects the delay due to CCA or processing time of the PHY to transmit a frame.

5.2.4.9.3 Beacon bitmap

The Beacon Bitmap field shall be formatted as illustrated in Figure 48y.

Octets: 2	2	Variable
SD Index	SD Bitmap Length	SD Bitmap

Figure 48y—Format of the Beacon Bitmap field

The SD Index field specifies the index of current Superframe Duration (SD) in a beacon interval. The superframe in which the PAN coordinator sends its beacons serves as the reference point (SD Index 0). When this field is contained in an enhanced beacon, this field specifies the index of the superframe that is allocated to the Source device of the beacon for beacon transmission.

The SD Bitmap Length shall specify the length of SD Bitmap field in octets.

The SD Bitmap field contains a bitmap indicating the beacon frame allocation information of all $2^{(BO–SO)}$ superframes within one beacon interval. Each corresponding bit in the bitmap shall be set to one if a beacon of a neighbor device is allocated in that SD, otherwise it is set to zero. It should be noted that the length of the beacon frame allocation information bitmap, $2^{(BO–SO)}$, may be different than the value of the SD Bitmap Length field since the SD Bitmap Length field is represented in octets. In this case, the value of the SD Bitmap Length field shall be chosen to be the smallest integer that is greater than the length of this bitmap, and those bits after the first $2^{(BO–SO)}$ bits in the SD Bitmap field shall be padded as zeros.

5.2.4.9.4 Channel hopping specification

The Channel Hopping Specification field shall be formatted as illustrated in Figure 48z.

Octets: 1	1	2	1	variable
HoppingSequenceID	PANCoordinatorBSN	ChannelOffset	ChannelOffsetBitmapLength	ChannelOffset Bitmap

Figure 48z—Format of the Channel Hopping Specification field

The HoppingSequenceID indicates the ID of the channel hopping sequence in use. HoppingSequenceID set to zero indicates that a default hopping sequence shall be used. A HoppingSequenceID set to one indicates that a hopping sequence generated by the PAN coordinator shall be used. The other values of HoppingSequenceID denote the sequence is set by a higher layer. If the HoppingSequenceID is one, a device shall request a channel hopping sequence from its coordinator when it associates to a PAN.

The PANCoordinatorBSN field shall specify the beacon sequence number of the PAN coordinator.

The ChannelOffset field shall specify the channel hopping offset value of the device.

The ChannelOffsetBitmapLength field shall specify the length of ChannelOffsetBitmap field in octets.

The ChannelOffsetBitmap field shall indicate the occupancy of the channel hopping offset values among neighbor devices and be represented in that bitmap. Each bit shall be set to one if the corresponding channel hopping offset value is already occupied by the neighbor devices, otherwise it shall be set to zero if the corresponding channel hopping value is not occupied. For instance, a ChannelOffsetBitmap of 0b1100100..0 indicates that channel hopping offset values of 0, 1, and 4 are being used by neighbor devices. Note that the $(i)^{th}$ bit in the ChannelOffsetBitmap corresponds to $(i-1)^{th}$ channel offset value. The length of ChannelOffsetBitmap is defined by the values specified in ChannelOffsetBitmapLength field.

5.2.4.9.5 Group ACK specification

The Group ACK Specification field shall be formatted as illustrated in Figure 48aa.

Bits: 0-15	16-19	20-27	28-43	44-47	48-55
GACK1 Superframe ID	GACK1 Slot ID	GACK1 Channel ID	GACK2 Superframe ID	GACK2 Slot ID	GACK2 Channel ID

Figure 48aa—Format of the Group ACK Specification field

The GACK1 Superframe ID field indicates the superframe in which GACK1 is reserved. In the same manner, the GACK2 Superframe ID field indicates the superframe in which GACK2 is reserved. The superframe ID is the sequence number of the superframe in a multi-superframe starting with zero. The superframe in which the PAN coordinator sends its beacons serves as the reference point (Superframe ID 0). An example of superframe ID is illustrated in Figure 34h.

The GACK1 Slot ID field indicates the superframe slot in which GACK1 is reserved. Also, the GACK2 Slot ID field indicates the superframe slot in which GACK2 is reserved. The slot ID is the sequence number of the DSME-GTSs (not including beacon or CAP slots) in a superframe starting with zero. An example of slot ID is illustrated in Figure 34h.

The GACK1 Channel ID field indicates the channel number in which GACK1 is reserved. Also, the GACK2 Channel ID field indicates the channel number in which GACK2 is reserved.

5.2.4.10 Rendezvous Time IE

The Rendezvous Time (RZ Time) IE is 2 octets. The RZ Time is the expected length of time in units of 10 symbols between the end of the transmission of the wake-up frame and the beginning of the transmission of the payload frame. The RZ Time shall be set by the next higher layer when requesting the MAC sublayer to transmit. The last wake-up frame in a wake-up sequence shall have RZ Time set to the value zero.

5.2.4.11 ACK/NACK time correction IE

The ACK/NACK Time Correction IE shall be used in all enhanced acknowledgements if *macTSCHenabled* is TRUE. A NACK may be indicated if the incoming frame successfully passed incoming frame security processing as stated in 7.2.3, but the frame could not be handled due to resource constraints (e.g., insufficient buffer space). A higher layer may signal lack of buffers by setting *macNoHLBuffers* to TRUE.

The ACK/NACK Timing IE shall be formatted as illustrated in Figure 48bb.

Octets: 2
Time Sync Info

Figure 48bb—Format of the ACK/NACK time correction element

The Time Sync Info field shall specify time synchronization information and ACK/NACK status. This field is constructed by taking a signed 16-bit 2s compliment time correction in the range of −2048 µs to 2047 µs, AND'ing it with 0xfff, and OR'ing again with 0x8000 to indicate a NACK. This field shall be set as indicated in Table 4h.

Table 4h—Values of the Time Sync Info field for ACK with timing Information

Range	Description
0x000–0x7ff	ACK with positive time correction in µs
0x800–0xfff	ACK with negative time correction in µs
0x8000–0x87ff	NACK with positive time correction in µs
0x8800–0x8fff	NACK with negative time correction in µs

5.2.4.12 Group ACK IE

A Group ACK header IE is meant for reducing the communication overhead in a network. Instead of transmitting an ACK frame for individual data frames received in each GTS, a coordinator device, if using the GACK mechanism, shall acknowledge the reception of multiple data frames by transmitting only a single enhanced ACK with the Header IE value 0x1f. In a normal operational mode, the receiver node individually acknowledges each successfully received GTS frame. However, repeated switching over between RX and TX states by the device is costly and time consuming. The concept of group acknowledgment allows the receiver to receive all scheduled GTS frames, while withholding individual acknowledgments, and then transmit a single ACK frame to acknowledge all successfully received frames. Additionally, a GACK frame can also specify additional timeslots for the failed GTS transmissions to be retransmitted in the extended CFP. That is, a group acknowledgment not only indicates the reception status for a group of GTS data frames but it can also specify new slot allocations, if any, for retransmission of failed GTS transmissions. Note that some of these fields are unavailable for the short frame structure used for the LLDN mode. The format of the GACK element is shown in Figure 48cc.

Octets: 1	1/2	variable	variable	variable
Ack. Control	GACK Bitmap	GACK Device List	GACK Index	GTS Directions

Figure 48cc—GACK element

5.2.4.12.1 Group ACK Control field

The Group ACK Control field for a GACK frame shall have the format according to Figure 48dd.

Bits: 0–2	3–4	5–7
Payload Size	Bitmap Size	Reserved

Figure 48dd—GACK Control field

The Payload Size field shall specify the length of the payload of a GACK frame. It is measured in number of octets.

The Bitmap Size field shall specify the size of bitmap in terms of octets, therefore the maximum size of the bitmap is 32 bits.

5.2.4.12.2 Remaining fields

The GACK Bitmap field shall contain a bitmap of size as specified by “Bitmap Size” field in the Group ACK Control field as described in 5.2.4.12.1. The bitmap shall indicate the state of transmissions during the CFP or extended contention-free period (ECFP) of the current superframe. A bit set to one shall indicate the fact that the coordinator received the data frame successfully in the corresponding GTS. A zero shall indicate that the coordinator failed in receiving a data frame in the corresponding timeslot. It is important to note that this field may have some unused bits that were actually padded to make the bitmap size a multiple of eight.

The GACK Device List field identifies the devices that are being allocated the timeslots in the ECFP portion of the superframe. This field shall not exist in the GACK frame in the LLDN mode.

The GACK Index field specifies the start of each GTS for the allocated nodes in the same order as in DSME device list. This field is applicable only in those systems that allow a GTS to consist of multiple timeslots. This field shall not exist the LLDN mode or PANs that do not allow frequency channel hopping.

The GTS Directions field specifies the direction of transmission for each GTS as uplink or downlink. This field is applicable only to PANs that allow the coordinator to transmit a frame to its devices by using a GTS. This field shall not exist in the GACK frame in the LLDN mode.

5.2.4.13 TSCH synchronization IE

The TSCH Synchronization payload IE is formatted as shown in Figure 48ee.

Octets: 5	1
ASN	Join Priority

Figure 48ee—Slotframe and Link IE

The ASN field contains the 5-octet Absolute Slot Number corresponding to the timeslot in which the enhanced beacon is sent. The ASN is used as the Frame Counter for security operations if enabled. The 1-octet Join Priority field can be used by a joining device to select among beaconing devices when multiple beacons are heard. The PAN coordinator's join priority is zero. A lower value of join priority indicates that the device is the preferred one to connect to. The beaconing device's join priority is the lowest join priority heard when it joined the network plus one.

The TSCH Synchronization IE is used to construct enhanced beacons that allow new devices to synchronize to a TSCH PAN.

5.2.4.14 TSCH slotframe and link IE

The TSCH slotframe and link IE contains one or more slotframes and their respective links that a beaconing device advertises to allow other devices to synchronize to the network. Refer to 5.1.2.6 for details on TSCH PAN formation.

The format of TSCH Slotframe and Link element shall be as illustrated in Figure 48ff.

Octets: 1	Variable
Number of Slotframes	Slotframe and link information

Figure 48ff—Slotframe and Link IE

The Number of Slotframes field is set to the total number of slotframe descriptors contained in the enhanced beacon.

For each slotframe included, the content of the slotframe and link descriptor is shown in Figure 48gg.

Octets: 1	2	1	5 x Number of Links
Slotframe Handle	Slotframe Size	Number of Links	Link Information for each Link

Figure 48gg—Slotframe and link fields (per slotframe)

The Slotframe Handle field shall be set to the slotframeHandle that uniquely identifies the slotframe.

The Slotframe Size field shall be set to the size of the slotframe in number of timeslots.

The Number of Links field shall be set to the number of links that belong to the specific slotframe identified by the slotframeHandle.

The Link Information for each Link field describes the attributes of each link. The format of Link Information field is depicted as shown in Figure 48hh.

Octets: 2	2	1
Timeslot	Channel Offset	Link Options

Figure 48hh—Link Information fields (per link)

The Timeslot field shall be set to the timeslot of this link.

The Channel Offset Information field shall be set to the channelOffset of this link.

The Link Options field indicates whether this link is a TX link, an RX link, or a SHARED TX link, and whether the device to which it is being linked is to be used for clock synchronization. The field corresponds to the Link Options field in the link table for the receiving device. SHARED TX links are used for a joining device to send an Association Request command or other higher layer message. RX links are used for a joining device to receive an Association Response command or other higher layer message from an advertising device (i.e., a device sending Enhanced Beacons with TSCH information). RX links shall have the Timekeeping linkOptions bit set. It is possible for one link to be used as both SHARED_TX and RX link.

The TSCH Slotframe and Link IE is used to construct enhanced beacons that allow new devices to synchronize to a TSCH PAN.

5.2.4.15 TSCH timeslot IE

The TSCH Timeslot IE contains the *macTimeslotTemplate* being used by the beaconing device as shown in Figure 48ii.

The full time-slot template may be omitted from the enhanced beacon to ensure that the beacon does not exceed *aMaxPHYPacketSize*. In this case only the template ID is carried, and the length of the IE is 1 octet. If the full time-slot template is carried inline, the element is 25 octets and corresponds to the PIB attributes in Table 52e.

Octets: 1	0/2	0/2	0/2	0/2	0/2	0/2	0/2
Timeslot ID	*macTsCCAOffset*	*macTsCCA*	*macTsTxOffset*	*macTsRxOffset*	*macTsRxAckDelay*	*macTsTxAckDelay*	*macTsRxWait*

0/2	0/2	0/2	0/2	0/2
macTsAckWait	*macTsRxTx*	*macTsMaxAck*	*macTsMaxTx*	*macTsTimeslotLength*

Figure 48ii—Format of TSCH Timeslot IE

The TSCH Timeslot IE is used in constructing enhanced beacons to allow new devices to synchronize to a TSCH PAN.

5.2.4.16 Channel hopping IE

The Channel Hopping IE, as shown in Figure 48jj, contains the Hopping Sequence being used by the device (and corresponds to the PIB attributes in Table 52f).

The full Hopping Sequence may be omitted to ensure that the frame does not exceed *aMaxPHYPacketSize*. In this case only the Hopping Sequence ID is carried, and the length of the IE is 1 octet. When the length is not equal to one, the additional fields are present. The element varies in length depending upon the *numberOfChannels* in use by the PHY. The *channelPage* and *numberOfChannels* attributes can be used to determine the size of the *extendedBitmap*, and the *macHoppingSequenceLength* can be used to determine the size of the *macHoppingSequenceList*.

Octets: 1	0/1	0/2	0/4	0/variable	0/2	0/variable	0/2
Hopping Sequence ID	Channel Page	Number of Channels	Phy Configuration	Extended bitmap	Hopping sequence length	Hopping sequence list	Current hop

Figure 48jj—Format of Channel Hopping IE

The Channel Hopping IE may be used by any channel hopping PAN to synchronize devices.

5.2.4.17 Hopping timing IE

The Hopping Timing IE is used in unslotted channel hopping systems to allow devices to synchronize with the channel hopping timing parameters of the network.

The format of the Hopping Timing IE shall be as illustrated in Figure 48kk.

Octets: 3	2
Present Hop Time Offset	Hop dwell time

Figure 48kk—Format of Hopping Timing IE

The Present Hop Time Offset field contains the amount of time (in units of 1 μs) that has passed at the time of frame transmission since the transmitting device hopped to the present channel, i.e., the relative time offset into the hopDwellTime as defined in Table 52f for the present channel.

The Hopping Timing IE is used to synchronize to an unslotted channel hopping network.

5.2.4.18 EB Filter IE

The EB Filter IE is a variable length payload IE that indicates the optional response filters to be used by coordinators to determine whether they shall respond to an EBR, if *macEBFilteringEnabled* is TRUE. The first octet contains filters and is followed by additional optional fields. Only devices matching the filters are required to respond. The EB Filter is as shown in Figure 48ll.

Bit:0	1	2	3 – 4	5 – 7	Octets: 0/1	0/1	0/1/2/3/4
Permit Joining On	Include Link Quality Filter	Include Percent Filter	Number of entries in PIB Identifier list	Reserved	Link Quality	Percent Filter	List of PIB IDs

Figure 48ll—EB Filter IE

5.2.4.18.1 Permit Joining On

If the Permit Joining On bit (b0) is set, only devices supporting EBR/EB with permit joining on shall respond to the enhanced beacon request.

5.2.4.18.2 Include Link Quality Filter

If the Include Link Quality Filter bit (b1) is set, a 1-octet Link Quality field follows the filter octet. The receiving device shall respond (if supporting EBR/EB) to the enhanced beacon request if the mpduLinkQuality is equal or higher than this value (where higher values represent higher quality links).

5.2.4.18.3 Include Percent filter

If the Include Percent filter bit (b2) is set, a 1-octet Percent Filter follows the Link Quality field (if present). It contains a scaled value from 0x00 to 0x64 representing zero to 100% probability for a given device to respond to the enhanced beacon request. The receiving device shall then randomly determine if it is to respond to the enhanced beacon request (if supporting EBR/EB) based on meeting this probability. For example, if the probability is set to 10%, then 1 of 10 devices would randomly be expected to respond.

5.2.4.18.4 PIB Identifier List

The PIB Identifier List bits b3 and b4 indicate the number of Boolean PIB attribute IDs that follow the Percent Filter field (if present).

When a PIB Identifier List is present, the enhanced beacon capable device shall respond to the enhanced beacon request (if supporting EBR/EB) when all of the capabilities identified in the list are supported.

5.2.4.19 MAC Metrics IE

The MAC Metrics IE, as shown in Figure 48mm, encapsulates one of the counters used for MAC performance metrics.

Octets: 1	4
Metric ID	Count

Figure 48mm—Format of MAC metrics IE

The Metric ID field shall be set to one of the values in Table 4i.

Table 4i—Metric Count IDs

Attribute name	Metric ID
macCounterOctets	0x01
macRetryCount	0x02
macMultipleRetryCount	0x03
macTXFailCount	0x04
macTXSuccessCount	0x05
macFCSErrorCount	0x06
macSecurityFailure	0x07
macDuplicateFrameCount	0x08
macRXSuccessCount	0x09
macNACKcount	0x0a

5.2.4.20 AllMAC Metrics IE

The AllMAC Metrics payload IE, as shown in Figure 48nn, encapsulates all of the counters used for MAC performance metrics. The metrics count fields appear in the order given in Table 4i.

Octets: 4	4	...	4
macCounterOctets	*macRetryCount*	...	*macNACKcount*

Figure 48nn—Format of AllMAC metrics IE

5.2.4.21 Unmanaged ID space IEs

Unmanaged ID spaces are left to the implementer to manage. The format of unmanaged IEs is beyond the scope of this document. Since multiple implementers may reuse the same IDs, use of a mechanism to ensure correct identification of a specific implementer's IE is encouraged.

5.2.4.22 IE List Termination IE

The Header IE list is terminated with an IE List Termination IE (ID = 0x7e or 0x7f) that has a content length of zero. Explicit termination is required after a Header IE if there is one or more Payload IEs (0x7e), or MAC payload (0x7f), following the Header IE list. If an unformatted payload follows the Payload IE list, then the payload IE list is terminated with a list termination IE (ID = 0xf) that has a content length of zero. Otherwise the terminator may be omitted

5.3 MAC Command frames

Change in 5.3 the first paragraph as follows:

The command frames defined by the MAC sublayer are listed in Table 5. An FFD shall be capable of transmitting and receiving all command frame types 0x01–0x08, with the exception of the GTS request command, while the requirements for an RFD are indicated by an "X" in the table. An FFD supporting one of the optional functional groups listed in Table 52a shall support the associated command frames in the range 0x0d–0x1e as identified by the associated functional group prefix, e.g., "LL-". MAC commands shall only be transmitted in the CAP for beacon-enabled PANs, in management timeslots of LLDNs, or at any time for nonbeacon-enabled PANs.

5.3.1 Association request command

Change in 5.3.1 the first paragraph and Table 5 as follows:

The association request command allows a device to request association with a PAN through the PAN coordinator or a coordinator. The association request command also allows a FastA device to request fast association with a PAN.

Table 5—MAC Command frames

Command frame identifier	Command name	RFD		Subclause
		TX	RX	
0x01	Association request	X		5.3.1
0x02	Association response		X	5.3.2
0x03	Disassociation notification	X	X	5.3.3
0x04	Data request	X		5.3.4
0x05	PAN ID conflict notification	X		5.3.5
0x06	Orphan notification	X		5.3.6
0x07	Beacon request			5.3.7
0x08	Coordinator realignment		X	5.3.8
0x09	GTS request			5.3.9
~~0x0a–0xff~~	~~Reserved~~			
0x0d	LL-Discover response	X		5.3.10.1
0x0e	LL-Configuration status	X		5.3.10.2
0x0f	LL-Configuration request		X	5.3.10.3
0x10	LL-CTS shared group		X	5.3.10.4
0x11	LL-Request To Send (RTS)	X	X	5.3.10.5

Table 5—MAC Command frames *(continued)*

Command frame identifier	Command name	RFD		Subclause
		TX	RX	
0x12	LL-Clear to send (CTS)		X	5.3.10.6
0x13	DSME-Association request	X		5.3.11.2
0x14	DSME-Association response		X	5.3.11.3
0x15	DSME-GTS request	X	X	5.3.11.4
0x16	DSME-GTS reply	X	X	5.3.11.5
0x17	DSME-GTS notify	X	X	5.3.11.6
0x18	DSME-Information request	X	X	5.3.11.7
0x19	DSME-Information reply	X	X	5.3.11.8
0x1a	DSME-Beacon allocation notification			5.3.11.9
0x1b	DSME-Beacon collision notification	X		5.3.11.10
0x1c	DSME-Link status report	X	X	5.3.11.11
0x1d	AMCA-Beacon request	X	X	5.3.13.1
0x1e	AMCA-Hello	X	X	5.3.13.2
0x1f	AMCA-Channel probe	X	X	5.3.13.3
0x20	LE-RIT data request	X	X	5.3.12.1
0x21–0x3f	Reserved			—
0x44–0x5f	Reserved			—
0x61–0x62	Reserved			—
0x64–0xff	Reserved			—

5.3.1.2 Capability Information field

Insert at the end of 5.3.1.2 the following text and figure:

For a FastA device, the Capability Information field shall be formatted as illustrated in Figure 50a.

bits: 0	1	2	3	4	5	6	7
Alternate PAN Coordinator	Device Type	Power Source	Receiver On When Idle	Association Type	Reserved	Security Capability	Allocate Address

Figure 50a—Capability Information field format for FastA device

The Association Type field shall be set to one if the FastA device requests fast association and zero if the FastA device requests association.

5.3.2 Association response command

Change in 5.3.2 the first paragraph as indicated:

The association response command allows the PAN coordinator or a coordinator to communicate the results of an association attempt back to the device requesting association. The association response command also allows the PAN coordinator or a coordinator to communicate the results of a FastA attempt back to the FastA device requesting fast association.

5.3.2.3 Association Status field

Change Table 6 as indicated:

Table 6—Valid values of the Association Status field

Association status	Description
0x00	Association successful.
0x01	PAN at capacity.
0x02	PAN access denied.
0x03	Hopping Sequence offset duplication (refer to 5.3.11.3.5)
0x0~~3~~4–0x7f	Reserved.
0x80	FastA successful.
0x8~~0~~1–0xff	Reserved for MAC primitives enumeration values.

5.3.7 Beacon request command

Insert the following header in 5.3.7 before the existing text:

5.3.7.1 General

Insert before 5.3.8 the following subclauses (5.3.7.2–5.3.7.2.2):

5.3.7.2 Enhanced beacon request

5.3.7.2.1 General

The enhanced beacon request (EBR) command is used by an EBR-capable device to locate a subset of coordinators during an enhanced active scan. The enhanced beacon request is a command frame with version 0b10, and with a command frame identifier indicating a beacon request.

This command is optional for an FFD and an RFD.

The enhanced beacon request command shall be formatted as illustrated in Figure 56a.

Octets: variable (refer to 5.2.2.4.1)		**variable**		**1**	**2**
MHR fields		IEs		Command Frame Identifier (defined in Table 5)	FCS

Figure 56a—EBR Enhanced Beacon request command

The Sequence Number field, if present (as described in 5.2.2.4.1) shall be set to the current DSN.

The Addressing fields in the MHR may be set to any valid combination of source and destination addresses and PAN IDs (as described in 5.2.2.4.1).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception.

The Acknowledgment Request field and Security Enabled field shall be set to zero. If the enhanced beacon request is being sent on a particular PAN Identifier that is not the broadcast PAN identifier, the Security Enable field may be set to one, otherwise it shall be set to zero.

The Destination PAN Identifier field, if present, shall contain either a specific PAN identifier or a broadcast PAN identifier if *macImplicitBroadcast* is FALSE.

Frames intended for all coordinators shall set the Destination Address field to the broadcast short address (i.e., 0xffff). Frames intended for a specific coordinator shall set the Destination Address field to the coordinator's short address.

5.3.7.2.2 IE field

The EBR may contain an EB Filter IE and/or a list of IE IDs. The EB Filter allows a device to request Beacons or Enhanced beacons from a subset of coordinators who hear the EBR and match various criteria. Refer to 5.2.4.18 for details on the EB Filter IE.

The optional list of IE IDs allows a device to request that a coordinator respond with an Enhanced Beacon containing the requested IEs. Either header or payload IEs may be requested. IEs are requested by sending the header descriptor with a length set to zero. If no IEs are included in the EB Filter and the IE list is not included, then the responding coordinator shall return a beacon. If either are present, the receiving device shall return an enhanced beacon (if filter criteria are met) containing IEs as appropriate. If necessary (to check filtering or return requested IEs), the EB Filter IE and other IEs may be passed to a higher layer using the MLME-BEACON-REQUEST.indication primitive provided that the recipient has *macBeaconAutoRespond* set to FALSE. This IE request mechanism is only available in an EBR.

Insert before Clause 6 the following subclauses (5.3.10–5.3.13.3.2):

5.3.10 LL commands

5.3.10.1 LL-Discover Response command

5.3.10.1.1 General

The LL-Discover Response command contains the configuration parameters that have to be transmitted to the LLDN PAN coordinator as input for the configuration process in an LLDN.

This command shall only be sent by a device that has received an LL-Beacon (refer to 5.2.2.5.2) indicating discovery mode as determined through the procedures of the Discovery state as described in 5.1.9.2.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The command payload of the discover response frame shall be formatted as illustrated in Figure 59a.

Octets: 1	variable
Command Frame Identifier (defined in Table 5)	Discovery parameters

Figure 59a—LL-Discover response command MAC payload

5.3.10.1.2 MHR fields

The LL-Discover Response command can be sent using both MAC Command frames described in 5.2.2.4 or LL-MAC Command frames described in 5.2.2.5.5.

The Frame Type field of the Frame Control field shall contain the value that indicates a MAC command frame, as shown in Table 2.

The Source Addressing Mode field of the Frame Control field shall be set to three (64-bit extended addressing).

The Source Address field shall contain the value of *aExtendedAddress*.

In the Frame Control field of the LL-MAC Command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Sub Frame Type field shall contain the value that indicates an LL-MAC Command frame, as shown in Figure 48i.

5.3.10.1.3 Command Frame Identifier field

The Command Frame Identifier field contains the value for the discover response command frame as defined in Table 5.

5.3.10.1.4 Discovery Parameters field

The Discovery Parameters field contains the configuration parameters that have to be transmitted to the LLDN PAN coordinator as input for the configuration process. The discovery parameters consist of the following:

— Full MAC address
— Required timeslot duration, this is defined by the application of the device (e.g., size of payload data)
— Uplink/bidirectional type indicator

5.3.10.2 Configuration Status Frame

5.3.10.2.1 General

The Configuration Status command contains the configuration parameters that are currently configured at the device as input for the configuration process in an LLDN.

This command shall only be sent by a device that has received an LL-Beacon (described in 5.2.2.5.2) indicating configuration mode as determined through the procedures of the configuration mode described in 5.1.9.3.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The command payload of the Configuration Status frame shall be formatted as illustrated in Figure 59b.

Octets: 1	variable
Command Frame (defined in Table 5)	Configuration Parameters

Figure 59b—Configuration Status command MAC payload

5.3.10.2.2 MHR fields

The configuration status command can be sent using both MAC Command frames described in 5.2.2.4 or LL-MAC Command frames described in 5.2.2.5.5.

5.3.10.2.3 Using MAC Command frames

The Frame Type field of the Frame Control field shall contain the value that indicates a MAC command frame, as shown in Table 2.

The Source Addressing Mode field of the Frame Control field shall be set to one (8-bit short addressing) or three (64-bit extended addressing).

The Source Address field shall contain the value of *macSimpleAddress* if the Source Addressing Mode field is set to one or *aExtendedAddress* if the Source Addressing Mode field is set to three.

In the Frame Control field of LL-MAC Command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

5.3.10.2.4 Command Frame Identifier field

The Command Frame Identifier field contains the value for the configuration status frame as defined in Table 5.

5.3.10.2.5 Configuration Parameters field

The Configuration Parameters field contains the configuration parameters that are currently configured at the device. The configuration parameters consist of the following:

- Full MAC address
- Short MAC address
- Required timeslot duration, this is defined by the application of the device (e.g., size of payload data)
- Uplink/bidirectional data communication
- Assigned timeslots

5.3.10.3 Configuration Request Frame

5.3.10.3.1 General

The Configuration Request command contains the configuration parameters that the receiving device shall use during the Online state. This command shall only be sent by an LLDN PAN coordinator in response to a received Configuration Status frame of a device during the Configuration state. Only LLDN PAN coordinators are requested to be capable of transmitting this command; RFD are required to be capable of receiving it.

The command payload of the Configuration Request Frame shall be formatted as illustrated in Figure 59c.

Octet: 1	variable
Command Frame Identifier (defined in Table 5)	Configuration Parameters

Figure 59c—Configuration request command MAC payload

5.3.10.3.2 MHR fields

The configuration request command can be sent using both MAC Command frames described in 5.2.2.4 or LL-MAC Command frames described in 5.2.2.5.5.

The Frame Type field of the Frame Control field shall contain the value that indicates a MAC command frame, as shown in Table 2.

The Source Addressing Mode field of the Frame Control field shall be set to one (8-bit short addressing) or three (64-bit extended addressing).

The Destination Address field shall contain the value of source address of the corresponding Configuration Status frame.

In the Frame Control field of LL-MAC Command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Sub Frame Type field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

5.3.10.3.3 Command Frame Identifier field

The Command Frame Identifier field contains the value for the configuration request frame as defined in Table 5.

5.3.10.3.4 Configuration Parameters field

The Configuration Parameters field contains the new configuration parameters that are sent to the device in order to either configure it or reconfigure it. The configuration parameters consist of the following:

- Full MAC address
- Short MAC address
- Transmission channel
- Existence of management frames
- Timetimeslotslot duration
- Assigned timeslots

5.3.10.4 Clear To Send Shared Group frame

5.3.10.4.1 General

The Clear To Send (CTS) Shared Group command indicates to the devices of the star network that they now may use the timeslot for transmitting their own data with a simplified CSMA-CA.

This command shall only be sent by an LLDN PAN coordinator in a timeslot after tSlotTxOwner has been elapsed and the slot owner is not transmitting. For further information on channel access within timeslots refer to 5.1.1.6.6.

Only LLDN PAN coordinators shall be capable of transmitting this command, all other devices shall be capable of receiving it.

The command payload of the CTS Shared Group frame shall be formatted as illustrated in Figure 59d.

Octet: 1	1
Command Frame Identifier (defined in Table 5)	Network ID

Figure 59d—Clear to send shared group command MAC payload

5.3.10.4.2 MHR fields

The CTS Shared Group command shall be sent using LL-MAC command frames.

In the Frame Control field of LLDN MAC Command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Sub Frame Type field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

5.3.10.4.3 Command Frame Identifier field

The Command Frame Identifier field contains the value for the CTS shared group frame as defined in Table 5.

5.3.10.4.4 Network ID field

The Network ID field contains an identifier specific to the LLDN PAN coordinator.

5.3.10.5 Request To Send Frame

5.3.10.5.1 General

The Request To Send (RTS) command may be used by a device to indicate to the LLDN PAN coordinator and to the other devices of the star network that it wants to transmit data with a simplified CSMA-CA. The RTS frame is transmitted using a simplified CSMA-CA.

This command shall only be sent by a device in a timeslot after tSlotTxOwner has been elapsed and a CTS shared group frame has been received from the LLDN PAN coordinator.

Devices shall be capable of transmitting and receiving this command.

The command payload of the RTS frame shall be formatted as illustrated in Figure 59e.

Octet: 1	1	1
Command Frame Identifier (defined in Table 5)	Short Originator Address	Network ID

Figure 59e—Request To Send command MAC payload

5.3.10.5.2 MHR fields

The RTS command can be sent using LL-MAC command frames.

In the Frame Control field of LL-MAC command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

5.3.10.5.3 Command Frame Identifier field

The Command Frame Identifier field contains the value for the RTS frame as defined in Table 5.

5.3.10.5.4 Short Originator Address field

The Short Originator Address field contains the 1-octet simple address of the device sending this RTS frame.

5.3.10.5.5 Network ID field

The Network ID field contains an identifier specific to the LLDN PAN coordinator. It has to be identical to the Network ID of the corresponding received CTS shared group frame.

5.3.10.6 Clear To Send Frame

5.3.10.6.1 General

The Clear To Send (CTS) command indicates to a specific device of the star network that it may now use the timeslot for transmitting its own data with a simplified CSMA-CA. The CTS command is broadcast by the LLDN PAN coordinator in response to a received RTS command.

LLDN PAN coordinators shall be capable of transmitting this command, other LLDN devices shall be capable of receiving it.

The command payload of the CTS frame shall be formatted as illustrated in Figure 59f.

Octet: 1	1	1
Command Frame Identifier (defined in Table 5)	Short Destination Address	Network ID

Figure 59f—Clear to send command MAC payload

5.3.10.6.2 MHR fields

The CTS command can be sent using LL-MAC command frames.

In the Frame Control field of LL-MAC Command frames, the Frame Type field shall contain the value that indicates an LL-MAC frame, as shown in Table 2, and the Frame Subtype field shall contain the value that indicates an LL-MAC command frame, as shown in Table 3c.

5.3.10.6.3 Command Frame Identifier field

The Command Frame Identifier field contains the value for the CTS frame as defined in Table 5.

5.3.10.6.4 Short Destination Address field

The Short Destination Address field contains the 1-octet simple address of the device to which this CTS frame is directed.

5.3.10.6.5 Network ID field

The Network ID field contains an identifier specific to the LLDN PAN coordinator that shall be identical to the Network ID of the corresponding received RTS frame.

5.3.11 DSME commands

5.3.11.1 General

An FFD device in a DSME-enabled PAN shall be capable of transmitting and receiving all command frame types defined in 5.3.11.

5.3.11.2 DSME-Association request command

The DSME-association request command allows a device to request association with a DSME-enabled PAN through the PAN coordinator or a coordinator.

This command shall only be sent by an unassociated device that wishes to associate with a PAN. A device shall only associate with a PAN through the PAN coordinator or a coordinator allowing association, as determined through the scan procedure.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The DSME-association request command shall be formatted as illustrated in Figure 59g.

Octets: variable (refer to 5.2.2.4.1)	**1**	**1**	**1**	**2**
MHR fields	Command Frame Identifier (defined in Table 5)	Capability Information	Hopping Sequence ID	Channel Offset

Figure 59g—Association request command format

5.3.11.2.1 MHR fields

The Source Addressing Mode field of the Frame Control field shall be set to three (64-bit extended addressing). The Destination Addressing Mode field shall be set to the same mode as indicated in the beacon frame to which the association request command refers.

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to one.

The Destination PAN Identifier field shall contain the identifier of the PAN to which to associate. The Destination Address field shall contain the address from the beacon frame that was transmitted by the coordinator to which the association request command is being sent. The Source PAN Identifier field shall contain the broadcast PAN identifier (i.e., 0xffff). The Source Address field shall contain the value of *aExtendedAddress*.

5.3.11.2.2 Capability Information field

The Capability Information field shall be formatted as described in 5.3.1.2.

5.3.11.2.3 Hopping Sequence ID field

The Hopping Sequence ID field shall indicate the ID of channel hopping sequence in use. The Hopping Sequence ID of zero indicates that a default hopping sequence shall be used. The HoppingSequenceID of one indicates that a hopping sequence generated by PAN coordinator shall be used. The other value of the Hopping Sequence ID denotes the sequence set by a higher layer shall be used. A device is requesting a channel hopping sequence to its coordinator if the Hopping Sequence ID is one.

5.3.11.2.4 Channel Offset field

The Channel Offset field shall be set to the offset value of the unassociated device that wished to associate with a PAN, this value is specified by the next higher layer.

5.3.11.3 DSME-Association response command

The DSME-association request command allows a device to request association with a PAN through the PAN coordinator or a coordinator.

This command shall only be sent by an unassociated device that wishes to associate with a PAN. A device shall only associate with a PAN through the PAN coordinator or a coordinator allowing association, as determined through the scan procedure.

All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The DSME association response command shall be formatted as illustrated in Figure 59h.

Octets: variable (refer to 5.2.2.4.1)	**1**	**2**	**1**	**1**	**Variable**
MHR fields	Command Frame Identifier (defined in Table 5)	Short Address	Association Status	Hopping Sequence Length	Hopping Sequence

Figure 59h—DSME Association response command format

5.3.11.3.1 MHR fields

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall each be set to three (i.e., 64-bit extended addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to one.

The PAN ID Compression field of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression field, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. The Destination Address field shall contain the extended address of the device requesting association. The Source Address field shall contain the value of *aExtendedAddress*.

5.3.11.3.2 Short Address field

If the coordinator was not able to associate this device to its PAN, the Short Address field shall be set to 0xffff, and the Association Status field shall contain the reason for the failure. If the coordinator was able to associate the device to its PAN, this field shall contain the short address that the device may use in its communications on the PAN until it is disassociated.

A Short Address field value equal to 0xfffe shall indicate that the device has been successfully associated with a PAN, but has not been allocated a short address. In this case, the device shall communicate on the PAN using only its 64-bit extended address.

5.3.11.3.3 Association Status field

The Association Status field shall contain one of the nonreserved values listed in Table 6.

5.3.11.3.4 Hopping Sequence Length field

The Hopping Sequence Length field shall specify the length of the Hopping Sequence used in the PAN if the PAN runs in both beacon-enabled mode and Channel Hopping mode, i.e., ChannelDiversityMode of one. When the value of HoppingSequenceID is other than one, this field shall be set to zero. Hopping Sequence field shall be present only if the value of Hopping Sequence Length field is not zero.

5.3.11.3.5 Hopping Sequence field

The Hopping Sequence field shall contain the current value of *macHoppingSequenceList*. The size of the Hopping Sequence field is defined by the Hopping Sequence Length field and the Hopping Sequence field contains the current value of *macHoppingSequenceList*. This field shall be present only if the PAN runs in both beacon-enabled mode and Channel Hopping mode, i.e., ChannelDiversityMode of one, and the value of Hopping Sequence Length field is not zero.

5.3.11.4 DSME-GTS request command

5.3.11.4.1 General

The DSME-GTS request command allows an associated device to request the allocation of a new DSME or the deallocation, or change of an existing DSME from the corresponding device. Only devices that have a 16-bit short address less than 0xfffe shall send this command.

The DSME-GTS request command shall be formatted as illustrated in Figure 59i.

Octets: variable (refer to 5.2.2.4.1)	**1**	**1**	**0/1**	**0/2**	**0/1**	**Variable**
MHR fields	Command Frame Identifier (defined in Table 5)	DSME-GTS Management	Number of Slots	Preferred Superframe ID	Preferred Slot ID	DSMESABSpec-ification

Figure 59i—DSME-GTS request command format

5.3.11.4.2 MHR fields

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to two (i.e., 16-bit short addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledge Request field shall be set to one.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN Identifier field shall contain the value of *macPANId*, and the Destination Address field shall be set to the short address of the destination device.

5.3.11.4.3 DSME-GTS Management field

The DSME-GTS Management field shall be formatted as illustrated in Figure 59j.

bit: 0-2	**3**	**4**	**5-7**
Management Type	Direction	Prioritized Channel Access	Reserved

Figure 59j—DSME-GTS Management field format

The Management Type field shall be set to one of the nonreserved values listed in Table 7a.

Table 7a—Values of the Management Type field

Management Type value $b_2b_1b_0$	Description
000	Deallocation
001	Allocation
010	Duplicated Allocation Notification
011	Reduce
100	Restart
101	DSME-GTS Expiration
110–111	Reserved

The Direction field shall indicate the DSME-GTSs are being allocated for TX (the data transmission) or for RX (the data reception) of the requesting device. The value of this field is set to zero if the allocation is for TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not 0b001 (allocation).

The Prioritized Channel Access field shall be set to one if DSME-GTS should be reserved as high priority, or set to zero if DSME-GTS should be reserved as low priority. When the DSME-GTS request command is used in the DSME-GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME-GTS.

5.3.11.4.4 Number of slots field

The number of slots field shall contain the number DSME-GTSs that this command is requesting.

This field is valid only if the management type is 0b001 (allocation).

5.3.11.4.5 Preferred Superframe ID field

The Preferred Superframe ID field shall contain the index of the preferred superframe in which the DSME-GTSs need be allocated. The superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero. The superframe in which the PAN coordinator sends its beacons serves as the reference point (Superframe ID 0). An example of superframe IDs are illustrated in Figure 34h.

This field is valid only if the management type is 0b001 (allocation).

5.3.11.4.6 Preferred Slot ID field

The Preferred slot ID shall contain the index of the preferred slot from which the DSME-GTSs need to be allocated. The slot ID is the sequence number of the DSME-GTSs (not including beacon or CAP slots) in a superframe beginning from zero. An example of slot IDs are illustrated in Figure 34h.

This field is valid only if the management type is 0b001 (allocation).

5.3.11.4.7 DSMESABSpecification field

The DSMESABSpecification field shall be formatted as illustrated in Figure 59k.

Octets: 1	2	Variable
DSME SAB sub-block length	DSME SAB sub-block index	DSME SAB sub-block

Figure 59k—Format of the DSMESABSpecification field

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB sub-block in the entire SAB as illustrated in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as described in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode. However, the meaning of zeroes and ones in the DSME SAB sub-block has different meaning from the descriptions in those figures if the management type is not 0b001 (allocation). If the management type is 0b001 (allocation), the ones in the DSME SAB Sub-block field indicate readily allocated or unavailable slots and the zeroes indicate vacant and available slots. If the management type is not 0b001 (allocation), the ones in the DSME SAB Sub-block field indicate the slots that are being requested for deallocation, duplicated allocation notification, reduce, or restart.

5.3.11.5 DSME-GTS reply command

5.3.11.5.1 General

The DSME-GTS reply command allows an associated device to announce the result of a request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME-GTSs to the neighboring devices including the requesting device. Only devices that have a 16-bit short address less than 0xfffe shall send this command.

The DSME-GTS reply command shall be formatted as illustrated in Figure 59l.

Octets: variable (refer to 5.2.2.4.1)	1	1	2	0/2	Variable
MHR fields	Command Frame Identifier (defined in Table 5)	DSME-GTS Management	Destination Address	Channel Offset	DSMESAB-Specification

Figure 59l—DSME-GTS reply command format

5.3.11.5.2 MHR fields

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to two (i.e., 16-bit short addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledge Request field shall be set to one.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN Identifier field shall contain the value of *macPANId*, and the Destination Address field shall be set to 0xffff (i.e., broadcast address).

5.3.11.5.3 DSME-GTS Management field

The DSME-GTS Management field shall be formatted as illustrated in Figure 59m.

bit: 0-2	3	4	5-7
Management Type	Direction	Prioritized Channel Access	Status

Figure 59m—DSME-GTS Management field format

The Management Type field shall be set to one of the nonreserved values listed in Table 7a.

The Direction field shall indicate the DSME-GTSs are being allocated for TX (the data transmission) or for RX (the data reception) of the requesting device. The value of this field is set to 0 if the allocation is for TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not 0b001 (allocation).

The Prioritized Channel Access field shall be set to one if DSME-GTS should be reserved as high priority, or set to zero if DSME-GTS should be reserved as low priority. When the DSME-GTS request command is used in the DSME-GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME-GTS.

The Status field length is 3 bits and shall indicate the result of the DSME-GTS request. This field is set to 0b000 if the request was approved. This field is set to 0b001 if a request for allocation is disapproved because of the lack of availability. This field is set to 0b010 if a request for either deallocation, duplicated allocation notification, reduce, or restart is disapproved because the indicated DSME-GTSs are unknown.

5.3.11.5.4 DSME-GTS Destination Address fields

The DSME-GTS Destination Address shall contain the short address of the Destination device.

5.3.11.5.5 Channel Offset fields

The Channel Offset field shall contain the channel offset of the RX device of the allocated DSME-GTSs. This field is not valid in the channel adaptation mode.

5.3.11.5.6 DSMESABSpecification fields

The DSMESABSpecification field shall be formatted as illustrated in Figure 59k.

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB Sub-block in the entire SAB as illustrated in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as described in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode. However, the meaning of zeroes and ones in the DSME SAB sub-block has different meaning from the descriptions in those figures. The ones in the DSME SAB Sub-block field indicate the slots that are selected for allocation, deallocation, duplicated allocation notification, reduce, or restart.

5.3.11.6 DSME-GTS notify command

5.3.11.6.1 General

The DSME-GTS notify command allows an associated device to announce the result of its request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME-GTSs to the neighboring devices. Only devices that have a 16-bit short address less than 0xfffe shall send this command.

The DSME-GTS notify command shall be formatted as illustrated in Figure 59n.

Octets: variable (as per 5.2.2.4.1)	**1**	**1**	**2**	**0/2**	**Variable**
MHR fields	Command Frame Identifier (defined in Table 5)	DSME-GTS Management	Destination Address	Channel Offset	DSMESAB-Specification

Figure 59n—DSME-GTS notify command format

5.3.11.6.2 MHR fields

The Destination Addressing Mode and the Source Addressing Mode fields of the Frame Control field shall both be set to two (i.e., 16-bit short addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledge Request field shall be set to one.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

The Destination PAN Identifier field shall contain the value of *macPANId*, and the Destination Address field shall be set to 0xffff (i.e., broadcast address).

5.3.11.6.3 DSME-GTS Management field

The DSME-GTS Management field shall be formatted as illustrated in Figure 59m.

The Management Type field shall be set to one of the nonreserved values listed in Table 7a.

The Direction field shall indicate the DSME-GTSs are being allocated for TX (the data transmission) or for RX (the data reception) of the requesting device. The value of this field is set to zero if the allocation is for

TX. The value of this field is set to one if the allocation is for RX. This field is ignored if the management type is not 0b001 (allocation).

The Prioritized Channel Access field shall be set to one if DSME-GTS should be reserved as high priority, or set to zero if DSME-GTS should be reserved as low priority. When the DSME-GTS request command is used in the DSME-GTS change procedure, the Prioritized Channel Access shall be set according to the original DSME-GTS.

The Status field length is 3 bits and shall indicate the result of the DSME-GTS request. This field is set to 0b000 if the request was approved. This field is set to 0b001 if a request for allocation is denied because of the lack of availability. This field is set to 0b010 if a request for deallocation, duplicated allocation notification, reduce, or restart is denied because the indicated DSME-GTSs are unknown.

5.3.11.6.4 DSME-GTS Destination Address fields

The DSME-GTS Destination Address shall contain the short address of the Destination device.

5.3.11.6.5 Channel Offset fields

The Channel Offset field shall contain the channel offset of the RX device of the allocated DSME-GTSs. This field is not valid in the channel adaptation mode.

5.3.11.6.6 DSMESABSpecification fields

The DSMESABSpecification field shall be formatted as illustrated in Figure 59k.

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB sub-block in the entire SAB as illustrated in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as described in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode. However, the meaning of zeroes and ones in the DSME SAB sub-block has different meaning from the descriptions in those figures. The ones in the DSME SAB Sub-block field indicate the slots that are selected for allocation, deallocation, duplicated allocation notification, reduce, or restart.

5.3.11.7 DSME-Information request command

The DSME-Information request command is used to request the timestamp, DSMESABSpecification, or DSME PAN Descriptor information of the destination device.

The DSME-Information request command shall be formatted as illustrated in Figure 59o.

Octets: variable (refer to 5.2.2.4.1)	**1**	**1**	**1**	**2**
MHR fields	Command Frame Identifier (defined in Table 5)	Info type	DSME SAB sub-block length	DSME SAB sub-block index

Figure 59o—DSME-Information request command format

The Info Type field shall indicate the type of DSME information that are requested by the device. If this value is zero, this command is for requesting the timestamp. If this value is one, this command is for requesting DSMESABSpecification. If this value is two, this command is for requesting DSME PAN Descriptor.

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode. This field is valid only if the value of the Info type field is zero.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB Sub-block in the entire SAB as illustrated in Figure 59q for channel adaptation mode or in Figure 59r for channel hopping mode. This field is valid only if the value of the Info Type field is zero.

5.3.11.8 DSME-Information reply command

The DSME-Information reply command frame is used by a destination device that is replying the timestamp and the DSME information to the source device.

The DSME-Information reply command frame shall be formatted as illustrated in Figure 59p.

This command is mandatory for DSME devices.

Octets: variable (refer to 5.2.2.4.1)	**1**	**1**	**6**	**2**	**1**	**Variable**	**Variable**
MHR fields	Command Frame Identifier (defined in Table 5)	Info type	Timestamp	Superframe ID	Slot ID	DSMESABSpecification	DSME PAN Descriptor

Figure 59p—DSME-Information reply command format

The Info Type field indicates the type of DSME information that was requested. If this value is zero, this command contains the timestamp, the superframe ID, and the slot ID. If this value is one, this command contains DSMESABSpecification. If this value is two, this command contains DSME PAN Descriptor.

The Timestamp field shall contain the time, in symbols, at which this command will be transmitted. The time can be predicted because this command will be transmitted in the DSME-GTS. This field is valid only if the value of the Info Type field is zero.

The Superframe ID field shall contain the ID of the superframe in which this command will be transmitted. The superframe ID is the sequence number of the superframe in a multi-superframe beginning from zero. This field is valid only if the value of the Info Type field is zero.

The Slot ID field shall contain the ID of the superframe slot in which this command will be transmitted. The slot ID is the sequence number of the DSME-GTSs (not including beacon or CAP slots) in a superframe beginning from zero. This field is valid only if the value of the Info Type field is zero.

The DSMESABSpecification field shall be set as described in 5.3.11.4.7. This field is valid only if the value of the Info Type field is one.

The DSME SAB Sub-block Length field shall contain the length of the DSME SAB sub-block in units defined in Figure 59q.

The DSME SAB Sub-block Index field shall indicate the beginning of the DSME SAB Sub-block in the entire SAB as illustrated in Figure 59q.

The DSME SAB sub-block shall contain the sub-block of the DSME Slot Allocation Bitmap as illustrated in Figure 59q.

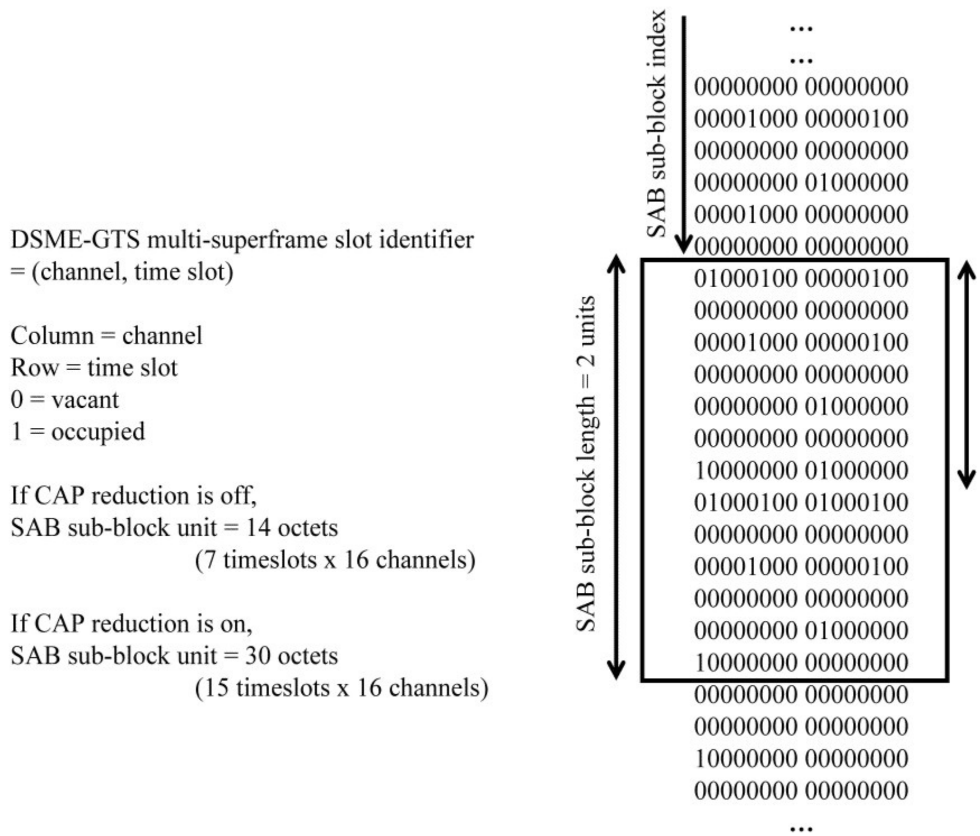

Figure 59q—SAB sub-block in channel adaptation

When the channel hopping mode is used to obtain channel diversity gain, i.e., ChannelDiversityMode bit shall be set to one. SAB represents the usage of corresponding DSME-GTS, a bit shall be set to one if the corresponding slot is allocated to transmit or receive, or zero if the slot is available. Similarly in channel adaptation, DSME SAB sub-block Index and DSME SAB Sub-block length shall indicate the start position and the length of SAB Sub-block. Thus, only sub block of whole SAB is exchanged for scheduling. This is illustrated in Figure 59r.

The DSME PAN Descriptor field shall be set as described in 5.2.4.9. This field is valid only if the value of the Info Type field is two.

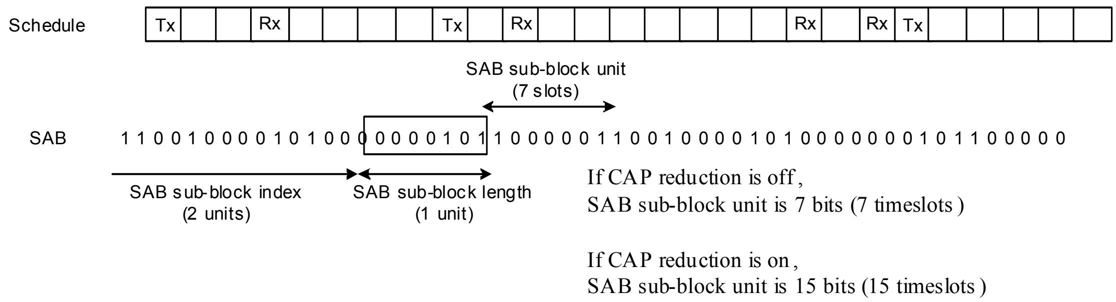

Figure 59r—SAB sub-block in channel hopping

5.3.11.9 DSME-Beacon allocation notification command

The DSME beacon allocation notification command is used by a device that selects vacant Superframe Duration (SD) for using transmission of beacon frame.

The DSME beacon allocation notification command shall be formatted as illustrated in Figure 59s.

This command is mandatory for DSME devices.

Octets: variable (refer to 5.2.2.4.1)	1	2
MHR fields	Command Frame Identifier (defined in Table 5)	Allocation Beacon SD Index

Figure 59s—DSME Beacon allocation notification command format

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall both be set to two (e.g., 16-bit short addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to zero.

The PAN ID Compression field of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression field, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. The Destination Address field shall be set to 0xffff. The Source Address field shall contain the value of *macShortAddress*.

The Allocation Beacon SD Index field shall contain the allocating SD index number for beacon frame that is allocated to the Source device.

5.3.11.10 DSME-Beacon collision notification command

The DSME beacon collision notification command is used by a device that detects the collision of beacon frame.

The DSME beacon collision notification command shall be formatted as illustrated in Figure 59t.

This command is mandatory for DSME devices.

Octets: variable (refer to 5.2.2.4.1)	1	2
MHR fields	Command Frame Identifier (defined in Table 5)	Collision SD Index

Figure 59t—DSME Beacon collision notification command format

The Destination Addressing Mode and Source Addressing Mode fields of the Frame Control field shall both be set to two (e.g., 16-bit short addressing).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to one.

The PAN ID Compression field of the Frame Control field shall be set to one. In accordance with this value of the PAN ID Compression field, the Destination PAN Identifier field shall contain the value of *macPANId*, while the Source PAN Identifier field shall be omitted. The Destination Address field shall be set to the node address that has requested later. The Source Address field shall contain the value of *macShortAddress*.

The Conflict SD Index field shall contain the SD index number of collision beacon frame.

5.3.11.11 DSME-Link status report command

5.3.11.11.1 General

The DSME link status report command allows a source device to report its link quality parameters to a destination device.

This command shall only be sent by an associated device that wishes to report the link quality. All devices shall be capable of transmitting this command, although an RFD is not required to be capable of receiving it.

The DSME link status report command shall be formatted as illustrated in Figure 59u.

Octets: variable (refer to 5.2.2.4.1)	1	Variable
MHR fields	Command Frame Identifier (defined in Table 5)	DSME-Link Status Specification

Figure 59u—DSME Link status report command format

5.3.11.11.2 MHR fields

The Source Addressing Mode field of the Frame Control field shall be set to two (16-bit extended addressing), and the Destination Addressing Mode field shall be set to the same mode as the destination device to which the status report command refers.

The Frame Pending field of the Frame Control field shall be set to one, and the Acknowledgment Request field shall be set to one.

The Destination PAN Identifier field shall contain the identifier of the PAN of the destination device to which to report the link status. The Destination Address field shall contain the address of the destination device to which the status report command is being sent.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

The Frame Type field in MHR shall be set to 0b100 and the Frame Version field should be set to 0b10.

5.3.11.11.3 Link Status Specification field

The DSME Link Status Specification fields shall be formatted as illustrated in Figure 59v.

Octets: 1	variable
Link Status Descriptor Count	Link Status List

Figure 59v—DSME Link Status Specification field format

The Link Status Descriptor Count field specifies the number of the Link Status Descriptors in the Link Status List field.

The size of the Link Status List fields is defined by the values specified in the Link Status Descriptor Count field and contains the list of Link Status Descriptors that represents the link status of each link. The Link Status Descriptor shall be formatted as illustrated in Figure 59w.

Octets: 1	1	1	1
Channel	avgLQI	avgRSSI	Reserved

Figure 59w—Link status Descriptor format

The Channel field specifies the channel index reported by the source device.

The avgLQI field contains the average received LQI of the channel specified in Channel field within LinkStatusStatisticPeriod symbols.

The avgRSSI field contains the average received signal power by ED measurement during a period of LinkStatusStatisticPeriod symbols. The avgRSSI measurement shall be performed for each received packet, and the use of the avgRSSI result by the next higher layer is not specified in this standard.

5.3.12 LE commands

5.3.12.1 RIT data request command

5.3.12.1.1 General

The RIT data request command allows a device to request data from its neighboring devices in the RIT mode. This command shall only be sent and received by a device supporting RIT mode. This command is optional and applicable for FFD only. The RIT data request command shall be formatted as illustrated in Figure 59x.

Octets: variable (refer to 5.2.2.4.1)	**1**	**0 or 4**
MHR fields	Command Frame Identifier (defined in Table 5)	Optional command payload

Figure 59x—Format of RIT data request command

5.3.12.1.2 MHR fields

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall also be set to zero. All other fields shall be set appropriately according to the intended use of the command frame.

5.3.12.1.3 Command Frame Identifier

Refer to Table 5.

5.3.12.1.4 Optional Command Payload

The command payload can be either 0 or 4 octets. The 4-octet payload is defined in Figure 59y.

Octets : 1	**1**	**2**
Time to 1st Listen (*T*0)	Number of Repeat (*N*)	Repeat Listen Interval (*T*)

Figure 59y—Format of Optional Command Payload

Time to 1st Listen ($T0$) and Repeat Listen Interval (T) are in the same units as *macRITPeriod*. Number of Repeat Listen (N) is constrained by $T0 + N \times T < macRITPeriod$.

5.3.13 AMCA commands

5.3.13.1 AMCA-Beacon Request command

The AMCA-Beacon request command is used by a device that is performing asymmetric multi-channel active scan.

The AMCA-Beacon request command shall be formatted as illustrated in Figure 59z.

Octets: variable (refer to 5.2.2.4.1)	**1**	**4**
MHR fields	Command Frame Identifier (defined in Table 5)	Scan Channels

Figure 59z—AMCA-Beacon Request command format

The Destination Addressing Mode field of the Frame Control field shall be set to two (e.g., 16-bit short addressing), and the Source Addressing Mode field shall be set to zero (e.g., source addressing information not present).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to zero.

The Destination PAN Identifier field shall contain the broadcast PAN identifier (i.e., 0xffff). The Destination Address field shall contain the broadcast short address (i.e., 0xffff).

The Scan Channels field is represented in 27-bit bitmaps. The 27 bits (b0, b1,... b26) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 channels supported by the ChannelPage parameter.

5.3.13.2 AMCA-Hello command

The AMCA-Hello command is used to inform neighboring devices of the device's designated listening channel (*macDesignatedListeningChannel*). The AMCA-Hello command shall be formatted as illustrated in Figure 59aa.

Octets: variable (refer to 5.2.2.4.1)	1	1
MHR fields	Command Frame Identifier (defined in Table 5)	Hello Specification

Figure 59aa—AMCA-Hello command format

5.3.13.2.1 MHR fields

The Destination Addressing Mode field of the Frame Control field shall be set to two (i.e., 16-bit short addressing), and the Source Addressing Mode field shall be set to zero (i.e., source addressing information not present).

The Frame Pending field of the Frame Control field shall be set to zero and ignored upon reception, and the Acknowledgment Request field shall be set to zero.

The Destination PAN Identifier field shall contain the broadcast PAN identifier (i.e., 0xffff). The Destination Address field shall contain the broadcast short address (i.e., 0xffff).

5.3.13.2.2 Hello Specification Field

The Hello Specification field shall be formatted as illustrated in Figure 59bb.

Bits: 5	1	2
Designated listening channel Index	Hello Request	Reserved

Figure 59bb—Hello Specification field format

The Designated listening channel Index field shall contain the *macDesignatedListeningChannel* attribute value of the device.

The Hello Request field shall indicate whether the AMCA-multi-channel hello command needs an AMCA-multi-channel hello from its neighbors. When a device receives the AMCA-multi-channel hello command with the Hello Request bit set to one, the device shall transmit a AMCA-multi-channel hello command with the Hello Request set to zero.

5.3.13.3 AMCA-Channel probe command

The AMCA-channel probe command is used to check the link quality of the specified channel. The AMCA-channel probe command shall be formatted as illustrated in Figure 59cc.

Octets: variable (refer to 5.2.2.4.1)	1	2
MHR fields	Command Frame Identifier (defined in Table 5)	Channel Probe Specification

Figure 59cc—AMCA-Channel Probe command format

5.3.13.3.1 MHR fields

The Source Addressing Mode field of the Frame Control field shall be set to two (16-bit extended addressing), and the Destination Addressing Mode field shall be set to the same mode as the destination device to which the channel probe command refers.

The Frame Pending field of the Frame Control field shall be set to zero, and the Acknowledgment Request field shall be set to one.

The Destination PAN Identifier field shall contain the identifier of the PAN of the destination device to which to check the link quality. The Destination Address field shall contain the address of the destination device to which the channel probe command is being sent.

The Source PAN Identifier field shall contain the value of *macPANId*, and the Source Address field shall contain the value of *macShortAddress*.

5.3.13.3.2 Channel Probe Specification Field

The Channel Probe Specification field shall be formatted as illustrated in Figure 59dd.

Bits: 2	5	5	4
Channel Probe Subtype	Designated listening channel	Probe Channel	Reserved

Figure 59dd—Channel Probe specification format

The Channel Probe Subtype field shall be set to one of the nonreserved values listed in Table 7b.

Table 7b—Values of the Channel Probe Subtype field

Channel Probe subtype value b_1b_0	Description
00	Request
01	Reply
10	Probe
11	Reserved

The Designated listening channel field indicates the originator's *macDesignatedListeningChannel* attribute value.

The Probe Channel field indicates the channel that needs to be probed.

6. MAC services

6.2 MAC management service

Insert after the heading of 6.2 the following subclauses (6.2.0a−6.2.0d):

6.2.0a General

6.2.0b TSCH MAC management service

When the optional TSCH mode is implemented (i.e., *macTSCHenabled* = TRUE), the MAC management services shall comply with Table 8a. The primitives are discussed in the subclauses referenced in Table 8a.

Table 8a—Summary of the primitives accessed through the MLME-SAP for TSCH

Name	Request	Indication	Response	Confirm
MLME-SET-SLOTFRAME	6.2.19.1	—	—	6.2.19.2
MLME-SET-LINK	6.2.19.3	—	—	6.2.19.4
MLME-TSCH-MODE	6.2.19.5	—	—	6.2.19.6
MLME-KEEP-ALIVE	6.2.19.7	—	—	6.2.19.8

6.2.0c LL-MAC management service

When the optional LLDN mode is implemented (i.e., *macLLenabled* = TRUE), the services shown in Table 8b shall be implemented. The primitives are discussed in the subclauses referenced in Table 8b.

Table 8b—Summary of the primitives accessed through the MLME-SAP for LLDN

Name	Request	Indication	Response	Confirm
MLME-LLDN-DISCOVERY	6.2.20.2	—	—	6.2.20.3
MLME-LLDN-CONFIGURATION	6.2.20.4	—	—	6.2.20.5
MLME- LLDN-ONLINE	6.2.20.6	6.2.20.7	—	—

6.2.0d DSME MAC management service

In a DSME-enabled PAN, the MAC management services shall comply with Table 46 and Table 8c. The DSME specific primitives are discussed in the subclauses referenced in Table 8c.

Table 8c—Summary of the primitives accessed through the MLME-SAP for DSME

Name	Request	Indication	Response	Confirm
MLME-DSME-GTS	6.2.21.1.1	6.2.21.1.2	6.2.21.1.3	6.2.21.1.4
MLME-DSME-INFO	6.2.21.2.1	6.2.21.2.2	—	6.2.21.2.3
MLME-DSME-LINKSTATUSRPT	6.2.21.3.1	6.2.21.3.2	—	6.2.21.2.3

6.2.1 Common requirements for MLME primitives

6.2.2 Association primitives

6.2.2.1 MLME-ASSOCIATE.request

Change 6.2.2.1 and Table 9 as indicated:

The MLME-ASSOCIATE.request primitive is used by a device to request an association with a coordinator and allows a FastA device to request a fast association with a coordinator.

The semantics of this primitive are:

```
MLME-ASSOCIATE.request      (
                            ChannelNumber,
                            ChannelPage,
                            CoordAddrMode,
                            CoordPANId,
                            CoordAddress,
                            CapabilityInformation,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex,
                            LowLatencyNetworkInfo,
                            ChannelOffset,
                            HoppingSequenceID
                            )
```

The primitive parameters are defined in Table 9.

Table 9—MLME-ASSOCIATE.request parameters

Name	Type	Valid range	Description
ChannelNumber	Integer	Any valid channel number	The channel number on which to attempt association.
ChannelPage	Integer	Any valid channel page	The channel page on which to attempt association.
CoordAddrMode	Enumeration	SHORT_ADDRESS, EXTENDED_ADDRESS	The coordinator addressing mode for this primitive and subsequent MPDU.
CoordPANId	Integer	0x0000–0xffff	The identifier of the PAN with which to associate.
CoordAddress	Device address	As specified by the CoordAddrMode parameter.	The address of the coordinator with which to associate.
CapabilityInformation	Bitmap	As defined in 5.3.1.2	Specifies the operational capabilities of the associating device and the association type of FastA-device.
SecurityLevel	Integer	As defined in Table 46	As defined in Table 46
KeyIdMode	Integer	As defined in Table 46	As defined in Table 46
KeySource	Set of octets	As defined in Table 46	As defined in Table 46
KeyIndex	Integer	As defined in Table 46	As defined in Table 46
LowLatencyNetwork-Info	Set of octets of variable length	—	Information for association specific to LLDN networks from the next higher layer. Only available if *macLLenabled* is TRUE.
ChannelOffset	Integer	0x00–0xffff	Specifies the offset value of Hopping Sequence.
HoppingSequenceID	Integer	0x00–0x0f	Indicate the ID of channel hopping sequence in use: 0x00: a default hopping sequence 0x01: a hopping sequence generated by PAN coordinator 0x02–0x0f: a hopping sequence set by NHL If a coordinator receives an association request command with HoppingSequenceID of 1, it replies with a channel hopping sequence in an association response command.

On receipt of the MLME-ASSOCIATE.request primitive, the MLME of an unassociated device first updates the appropriate PHY and MAC PIB attributes, as described in 5.1.3.1 and then generates an association request command, as defined in 5.3.1.

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

The MLME-ASSOCIATE.request is also used to request a fast association by FastA device.

If the LowLatencyNetworkInfo parameter has a nonzero length and *macLLenabled* is FALSE, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of UNSUPPORTED_FEATURE.

6.2.2.2 MLME-ASSOCIATE.indication

Change in 6.2.2.2 the semantics of this primitive and Table 10:

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

Table 10—MLME-ASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended IEEE address.	The address of the device requesting association.
CapabilityInformation	Bitmap	As defined in 5.3.1.2	The operational capabilities of the device requesting association and the association type of FastA device.
SecurityLevel	Integer	As defined in Table 48	As defined in Table 48
KeyIdMode	Integer	As defined in Table 48	As defined in Table 48
KeySource	Set of 0, 4, or 8 octets	As defined in Table 48	As defined in Table 48
KeyIndex	Integer	As defined in Table 48	As defined in Table 48
LowLatencyNetwork-Info	Set of octets of variable length	—	Information for association specific to LLDN networks from the next higher layer. Only available if *macLLenabled* is TRUE.

Table 10—MLME-ASSOCIATE.indication parameters *(continued)*

Name	Type	Valid range	Description
ChannelOffset	Integer	0x0000–0xffff	Specifies the offset value of Hopping Sequence.
HoppingSequenceID	Integer	0x00–0x0f	Indicate the ID of channel hopping sequence in use: 0x00: a default hopping sequence 0x01: a hopping sequence generated by PAN coordinator 0x02-0x0f: a hopping sequence set by NHL If a coordinator receives an association request command with HoppingSequenceID of 1, it replies with a channel hopping sequence in an association response command.

Insert at the end of 6.2.2.2 the following paragraph:

The FastA device requesting fast association receives the association response command frame from the coordinator directly.

6.2.2.3 MLME-ASSOCIATE.response

Change in 6.2.2.3 the semantics of this primitive and Table 11 as indicated:

The semantics of this primitive are:

```
MLME-ASSOCIATE.response      (
                             DeviceAddress,
                             AssocShortAddress,
                             status,
                             SecurityLevel,
                             KeyIdMode,
                             KeySource,
                             KeyIndex,
                             LowLatencyNetworkInfo,
                             ChannelOffset,
                             HoppingSequenceLength,
                             Hopping Sequence
                             )
```

The primitive parameters are defined in Table 11.

Table 11—MLME-ASSOCIATE.response parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting association.
AssocShortAddress	Integer	0x0000–0xffff	The short device address allocated by the coordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful.
status	Enumeration	As defined in 5.3.2.3	The status of the association attempt and the FastA attempt.
SecurityLevel	Integer	As defined in Table 46	As defined in Table 46
KeyIdMode	Integer	As defined in Table 46	As defined in Table 46
KeySource	Set of 0, 4, or 8 octets	As defined in Table 46	As defined in Table 46
KeyIndex	Integer	As defined in Table 46	As defined in Table 46
LowLatencyNetworkInfo	Set of octets of variable length	—	Information for association specific to LLDNs to the next higher layer. Only available if *macLLenabled* is TRUE.
ChannelOffset	Integer	0x0000–0xffff	Specifies the offset value of Hopping Sequence.
HoppingSequence-Length	Integer	0x0000–0xffff	Specifies the length of Hopping Sequence as described in 5.1.1a. When *macDSMEenabled* is TRUE, the *macSequenceLength* shall be set to zero for all *macHoppingSequenceID* values other than one.
Hopping Sequence	Set of octets	—	Specifies the sequence of channel numbers that is set by a higher layer as described in 5.1.1a. This parameter shall be present only if *macHoppingSequenceLength* is nonzero.

Insert the following text at the end of 6.2.2.3:

If the Status field of MLME ASSOCIATE.response primitive is set to 0x80 (i.e., FastA successful), the association response command shall be sent to the FastA device requesting fast association directly.

6.2.2.4 MLME-ASSOCIATE.confirm

Change in 6.2.2.4 the semantics of this primitive:


```
MLME-ASSOCIATE.confirm    (
                          AssocShortAddress,
                          status,
                          SecurityLevel,
                          KeyIdMode,
                          KeySource,
                          KeyIndex,
                          LowLatencyNetworkInfo,
                          ChannelOffset,
                          HoppingSequenceLength,
                          Hopping
                          Sequence
                          )
```

Change row "status" of Table 12 as indicated:

Table 12—MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	The value of the Status field of the association response command as defined in 5.3.2.3, SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY, UNSUPPORTED_FEATURE, INVALID_PARAMETER	The status of the association attempt.

Insert at the end of Table 12 the following new rows:

Table 12—MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
LowLatencyNetworkInfo	Set of octets of variable length	—	Information for association specific to LLDNs to the next higher layer. Only available if *macLLenabled* is TRUE.
ChannelOffset	Integer	0x0000–0xffff	Specifies the offset value of Hopping Sequence.
HoppingSequenceLength	Integer	0x0000–0xffff	Specifies the length of Hopping Sequence. This parameter shall be set to zero when the value of *macHoppingSequenceID* is other than one.
Hopping Sequence	Set of octets	—	Specifies the sequence of channel numbers that is set by NHL. This parameter shall be present only if *macHoppingSequenceLength* is nonzero.

Insert before 6.2.3 the following subclauses (6.2.2.5–6.2.2.6):

6.2.2.5 FastA message sequence charts

Figure 60a illustrates a sequence of messages for a successful FastA.

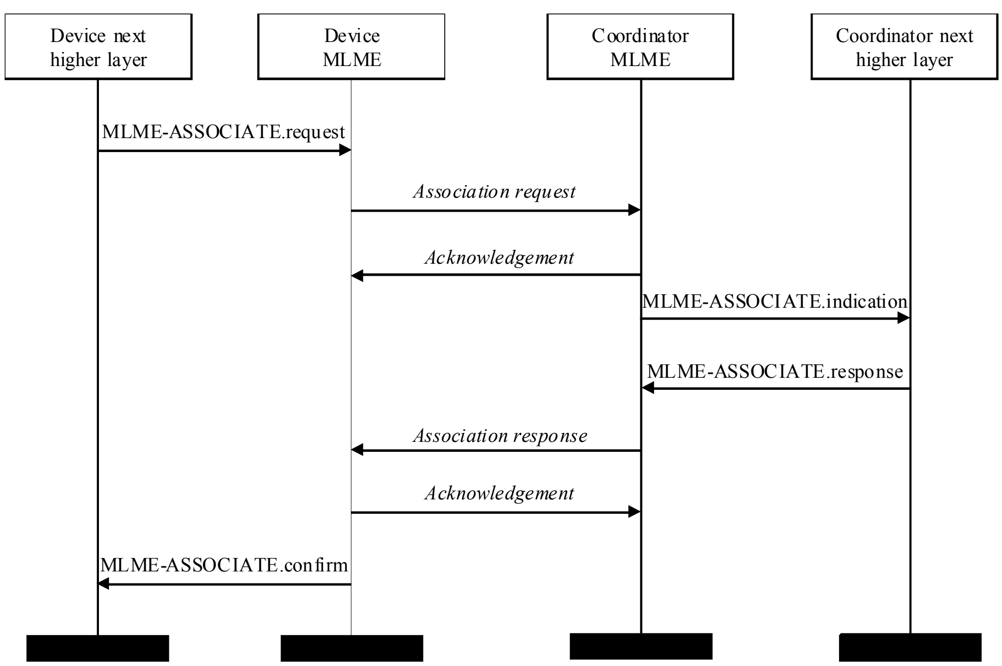

Figure 60a—Message sequence chart of fast association

6.2.2.6 Association message sequence charts for DSME devices

Figure 60b illustrates a sequence of messages that may be used by a DSME device that is not tracking the beacon of the coordinator to successfully associate with a PAN. On receipt of the association response command, a DSME device associating with a coordinator generates a DSME-Beacon allocation notification command frame. The command frame is sent to the neighbor devices to notify its beacon allocation information.

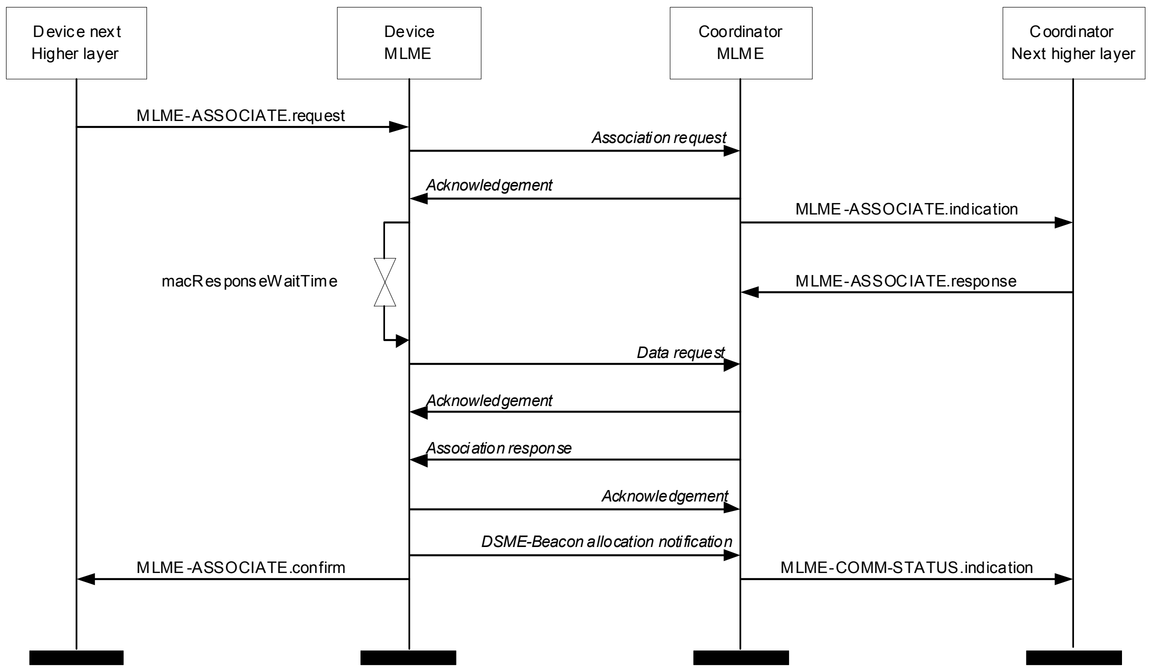

Figure 60b—Message sequence chart for DSME device association

6.2.4 Communications notification primitives

Insert a new paragraph and change existing text in 6.2.4 as follows:

The MLME-SAP beacon primitives define how a coordinator in a nonbeacon-enabled PAN may send a beacon or an enhanced beacon, or respond to beacon/enhanced beacon requests. Coordinators operating in a beaconing PAN use MLME-START to configure beacons and enhanced beacons. Only FFDs capable of acting as a coordinator are required to provide the MLME-BEACON SAPs.

~~This subclause defines two primitives.~~ The ~~first~~ MLME BEACON NOTIFY.indication primitive is used to notify the next higher layer when a beacon or enhanced beacon is received during normal operating conditions. The ~~second~~ MLME-COMM-STATUS.indication primitive is used to notify the next higher layer that an error has occurred during the processing of a frame that was instigated by a response primitive.

6.2.4.1 MLME BEACON NOTIFY.indication

Change in 6.2.4.1 the first paragraph as follows:

The MLME-BEACON-NOTIFY.indication primitive is used to send parameters contained within a beacon frame or an enhanced beacon frame received by the MAC sublayer to the next higher layer when either *macAutoRequest* is set to FALSE or when the beacon frame contains one or more octets of payload. The primitive also sends a measure of the LQI and the time the beacon frame was received. When an enhanced beacon is received, the SDU contains a list of IEs, and Superframe Specification, GTS fields, PendingAdd, and beacon Payload parameters are not present.

Change the semantics of this primitive:

```
MLME BEACON NOTIFY.indication   (
                                BSN,
                                PANDescriptor,
                                PendAddrSpec,
                                AddrList,
                                sduLength,
                                sdu,
                                EBSN,
                                beaconType,
                                )
```

Change the end of Table 16 as follows:

Table 16—MLME-BEACON-NOTIFY.indication parameters

Name	Type	Valid range	Description
EBSN	Integer	0x00–0xff	Beacon sequence number used for enhanced beacon frames.
sdu	Set of octets	—	The set of octets comprising the beacon payload ~~to be transferred from the MAC layer entity to the next higher layer~~ including Payload IEs if present.
beacon Type	Integer	0x00–0x01	Indicates a beacon (0x00) or enhanced beacon (0x01) was received.

Insert at the end of Table 17 the following new rows:

Table 17—Elements of PANDescriptor

Name	Type	Valid range	Description
DSMESuper-frame-Specification	Bitmap	Defined in 5.2.4.9	The DSME superframe specification as specified in the received enhanced frame. This element is present when *macDSMEenabled* is TRUE.
TimeSynchroni-zation-Specification	Bitmap	Defined in 5.2.4.9.2	The time synchronization specification as specified in the received enhanced beacon frame. This element is present when *macDSMEenabled* is TRUE.
BeaconBitmap	Bitmap	Defined in 5.2.4.9.3	The beacon bitmap as specified in the received enhanced beacon frame. This element is present when *macDSMEenabled* is TRUE.

Table 17—Elements of PANDescriptor *(continued)*

Name	Type	Valid range	Description
ChannelHopping-Specification	Bitmap	Defined in 5.2.4.9.4	The channel hopping specification specified in the received enhanced beacon frame. This element is present when *macDSMEenabled* is TRUE and the value of Channel Diversity Mode field in the received enhanced beacon frame is one.
GroupACK-Specification	Bitmap	Defined in 5.2.4.9.5	The group ACK specification specified in the received enhanced beacon frame. This element is present when *macDSMEenabled* is TRUE and the value of GACK Flag field in the received enhanced beacon frame is one.

6.2.10 Primitives for channel scanning

6.2.10.1 MLME SCAN.request

Change in 6.2.10.1 the semantics as follows:

```
MLME SCAN.request          (
                           ScanType,
                           ScanChannels,
                           ScanDuration,
                           ChannelPage,
                           SecurityLevel,
                           KeyIdMode,
                           KeySource,
                           KeyIndex,
                           LinkQualityScan,
                           frameControlOptions,
                           headerIElist,
                           payloadIElist,
                           )
```

Change Table 30 as indicated:

Table 30—MLME-SCAN.request parameters

Name	Type	Valid range	Description
ScanType	Enumeration	ED, ACTIVE, PASSIVE, ORPHAN, ASYMMETRIC MULTI-CHANNEL ACTIVE, CHANNEL PROBE, MULTI-CHANNEL HELLO, ENHANCED ACTIVE SCAN (optional for RFD)	Indicates the type of scan performed, as described in 5.1.2.1. The coding is specified in Table 31.
ScanChannels	List of integers	Any list of valid channel numbers	The channel numbers to be scanned. For pages 0 to 6, the 27 bits (b0, b1,... b26) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 channels supported by the ChannelPage parameter. For pages 7 and 8, a bitmap corresponding to the number of channels, *n*, where (b0, b1,…bn) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the n channels supported by the ChannelPage and PHY configuration.
ScanDuration	Integer	0–14	A value used to calculate the length of time to spend scanning each channel for ED, active, and passive scans. This parameter is ignored for orphan scans. The time spent scanning each channel is $[aBaseSuperframeDuration \times (2^n + 1)]$ symbols, where n is the value of the ScanDuration parameter.
ChannelPage	Integer	Any valid channel page	The channel page on which to perform the scan.
SecurityLevel	Integer	As defined in Table 46	As defined in Table 46
KeyIdMode	Integer	As defined in Table 46	As defined in Table 46
KeySource	Set of 0, 4, or 8 octets	As defined in Table 46	As defined in Table 46
KeyIndex	Integer	As defined in Table 46	As defined in Table 46
LinkQualityScan	Boolean	TRUE or FALSE	If TRUE, Link Quality Scan should be enabled, otherwise FALSE.
frameControl-Options	List of Boolean	PAN_ID_SUPPRESSED, IES_INCLUDED, SEQ_#_SUPPRESSED	Corresponding to the options in frame control for frame version b10. Used with enhanced active scan.

Table 30—MLME-SCAN.request parameters *(continued)*

Name	Type	Valid range	Description
headerIElist	List of IEs. Refer to 5.2.4.2	Refer to Table 4b	Determines which Header IEs are sent in the frame. IES_INCLUDED shall be set in frameControlOptions if present.
payloadIElist	List of IEs. Refer to 5.2.4.3	Refer to Table 4c	Determines which Payload IEs are sent in the frame. IES_INCLUDED shall be set in frameControlOptions if present.

Insert at the end of 6.2.10.1 the following paragraphs:

In addition, when the ScanType parameter is set to 0x04, the MLME SCAN.request primitive is used to initiate a channel scan over a given list of channels for asynchronous multi-channel adaptation of DSME. When the ScanType is set to 0x05, this primitive is used to probe a channel. When the ScanType is set to 0x06, this primitive is used to transmit a multi-channel hello. When the ScanType is set to 0x07, this primitive is used to initiate and enhanced active scan.

The active scan is performed on each channel by the MLME first sending a beacon request command as described in 5.3.7 or an enhanced beacon request command as described in 5.3.7.2 if using an enhanced active scan. The MLME then enables the receiver and records the information contained in each received beacon in a PAN descriptor structure as described in Table 17. The active scan on a particular channel terminates when the number of PAN descriptors stored equals an implementation-specified maximum or when [*aBaseSuperframeDuration* × $(2^n + 1)$] symbols, where *n* is the value of the ScanDuration parameter, have elapsed, whichever comes first. Refer to 5.1.2.1.2 for more detailed information on the active channel scan procedure.

The asymmetric multi-channel active scan is performed on each channel by the MLME first sending an AMCA multi channel beacon request command as described in 5.3.13.1. The MLME then enables the receiver and records the information contained in the received beacon in a PAN descriptor structure as described in Table 17. If the LinkQualityScan flag is FALSE, the asymmetric multi-channel active scan terminates when the device receives a beacon and then chooses the current channel as its designated listening channel (*macDesignatedListeningChannel*). Otherwise, if the LinkQualityScan flag is TRUE, the asymmetric multi-channel active scan on a particular channel terminates when [*aBaseSuperframeDuration* × $(2^n + 1)$] symbols, where *n* is the value of the ScanDuration parameter, have elapsed after successful transmission of the DSME multi channel beacon request command, then switch to the next channel and repeat the same procedure. In this case, the whole asymmetric multi-channel active scan shall terminate when the device has scanned every channel twice, and the device shall choose its designated listening channel according to the LQI or RSSI of the received beacons. Refer to 5.1.12.3 for more detailed information on the asymmetric multi-channel active scan.

The channel probe scan is performed on the channel specified by the ScanChannels parameter by the MLME first sending an AMCA channel probe command with its subtype set to request (refer to 5.3.13.3) to one of its neighbors using the designated listening channel of the neighbor.

The multi-channel hello is performed by the MLME first sending an AMCA-multi-channel hello command on each channel. Refer to 5.1.12.4 for more detailed information on the multi-channel hello.

The MLME then switches to the device own designated listening channel and enables the receiver and waits the AMCA channel probe command with reply subtype. The channel probe scan terminates when [*aBaseSuperframeDuration* × $(2^n + 1)$] symbols, where *n* is the value of the ScanDuration parameter, have elapsed after successful transmission of the AMCA channel probe command with request subtype. The device shall check the LQI or RSSI of the AMCA channel probe command with reply subtype upon receiving it. Refer to 5.1.12.5 for more detailed information on the channel probe scan.

The results of an asymmetric multi-channel active scan are reported to the next higher layer through the MLME SCAN.confirm primitive. If the scan is successful and *macAutoRequest* is TRUE, the primitive results shall include a set of PAN descriptor values. If the scan is successful and *macAutoRequest* is FALSE, the primitive results shall contain a null set of PAN descriptor values, and each PAN descriptor value shall be sent individually to the next higher layer using separate MLME BEACON NOTIFY primitives. In both cases, the MLME SCAN.confirm primitive shall contain a list of unscanned channels and a status of SUCCESS.

If, during an asymmetric multi-channel active scan, the MLME is unable to transmit a AMCA multi channel beacon request command on a channel specified by the ScanChannels parameter due to a channel access failure, the channel shall appear in the list of unscanned channels returned by the MLME SCAN.confirm primitive. If the MLME was able to send an AMCA multi channel beacon request command on at least one of the channels but no beacons were found, the MLME SCAN.confirm primitive shall contain a null set of PAN descriptor values, regardless of the value of *macAutoRequest*, and a status of NO_BEACON.

The results of a channel probe scan are reported to the next higher layer through the MLME SCAN.confirm primitive. If the scan is successful the primitive results shall include a status of SUCCESS.

If, during a channel probe scan, the MLME is unable to transmit an AMCA channel probe command with request subtype on a channel specified by the ScanChannels parameter due to a channel access failure, the channel shall appear in the list of unscanned channels returned by the MLME SCAN.confirm primitive. If the MLME was able to send an AMCA channel probe command with request subtype but no channel probe reply was found, the MLME SCAN.confirm primitive shall contain a status of BAD_CHANNEL.

6.2.10.2 MLME SCAN.confirm

Change in Table 31 the first two rows:

Table 31—MLME-SCAN.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, LIMIT_REACHED, NO_BEACON, SCAN_IN_PROGRESS, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, BAD_CHANNEL, INVALID_PARAMETER	The status of the scan request.
ScanType	Integer	0x00–0x0~~3~~7	Indicates the type of scan performed: 0x00 = ED scan (optional for RFD). 0x01 = active scan (optional for RFD). 0x02 = passive scan. 0x03 = orphan scan. 0x04 = asymmetric multi-channel active scan. 0x05 = channel probe. 0x06 = multi-channel hello. 0x07 = enhanced active scan (optional for RFD).

6.2.12 Primitives for updating the superframe configuration

6.2.12.1 MLME START.request

Change in 6.2.12.1 the semantics as indicated:

```
MLME START.request          (
                            PANId,
                            ChannelNumber,
                            ChannelPage,
                            StartTime,
                            BeaconOrder,
                            SuperframeOrder,
                            PANCoordinator,
                            BatteryLifeExtension,
                            CoordRealignment,
                            CoordRealignSecurityLevel,
                            CoordRealignKeyIdMode,
                            CoordRealignKeySource,
                            CoordRealignKeyIndex,
                            BeaconSecurityLevel,
                            BeaconKeyIdMode,
                            BeaconKeySource,
                            BeaconKeyIndex,
                            DSMESuperframeSpecification,
                            BeaconBitmap,
                            HoppingDescriptor
                            )
```

Change Table 34 as indicated:

Table 34—MLME-START.request parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000–0xffff	The PAN identifier to be used by the device.
ChannelNumber	Integer	Any valid channel number	The channel number to use.
ChannelPage	Integer	Any valid channel page	The channel page to use.
StartTime	Integer	0x000000–0xffffff	The time at which to begin transmitting beacons. If this parameter is equal to 0x000000, beacon transmissions will begin immediately. Otherwise, the specified time is relative to the received beacon of the coordinator with which the device synchronizes. This parameter is ignored if either the BeaconOrder parameter has a value of 15 or the PANCoordinator parameter is TRUE. This parameter is ignored in a DSME-enabled PAN. The time is specified in symbols and is rounded to a backoff period boundary. The precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.

Table 34—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
BeaconOrder	Integer	0–15	Indicates the frequency with which the beacon is transmitted, as defined in 5.1.1.1
SuperframeOrder	Integer	0–*BO* or 15	The length of the active portion of the superframe, including the beacon frame, as defined in 5.1.1.1
PANCoordinator	Boolean	TRUE, FALSE	If this value is TRUE, the device will become the PAN coordinator of a new PAN. If this value is FALSE, the device will begin using a new superframe configuration on the PAN with which it is associated.
BatteryLife-Extension	Boolean	TRUE, FALSE	If this value is TRUE, the receiver of the beaconing device is disabled *macBattLifeExtPeriods* full backoff periods after the interframe spacing (IFS) period following the beacon frame. If this value is FALSE, the receiver of the beaconing device remains enabled for the entire CAP. This parameter is ignored if the BeaconOrder parameter has a value of 15.
CoordRealign-ment	Boolean	TRUE, FALSE	TRUE if a coordinator realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.
CoordRealign-SecurityLevel	Integer	0x00–0x07	The security level to be used for coordinator realignment command frames, as described in Table 58.
CoordRealign-KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used, as described in 7.4.1.2. This parameter is ignored if the CoordRealignSecurityLevel parameter is set to 0x00.
CoordRealign-KeySource	Set of octets	As specified by the CoordRealign-KeyIdMode parameter	The originator of the key to be used, as described in 7.4.1.3. This parameter is ignored if the CoordRealignKeyIdMode parameter is ignored or is set to 0x00.
CoordRealign-KeyIndex	Integer	0x01–0xff	The index of the key to be used, as described in 7.4.3.2. This parameter is ignored if the CoordRealignKeyId Mode parameter is ignored or is set to 0x00.
BeaconSecurity-Level	Integer	0x00–0x07	The security level to be used for beacon frames, as described in Table 58.
BeaconKeyId-Mode	Integer	0x00–0x03	The mode used to identify the key to be used, as described in Table 59. This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.
BeaconKey-Source	Set of octets	As specified by the BeaconKeyId-Mode parameter	The originator of the key to be used, as described in 7.4.3.1. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.
BeaconKeyIndex	Integer	0x01–0xff	The index of the key to be used, as described in 7.4.3.2. This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.
DSME-Superframe-Specification	DSME-Super-frame-Specifi-cation	As specified by 5.2.4.9.1	Specifies the superframe configuration in the DSME-enabled PAN. Refer to 5.2.4.9.1.

Table 34—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
BeaconBitmap	Beacon Bitmap	As specified by 5.2.4.9.3	Specifies beacon bitmap. Refer to 5.2.4.9.3.
HoppingDescriptor	HoppingDescriptor Value	Refer to 5.2.4.9.4	Specifies channel hopping information. Refer to Table 34a.

Insert after Table 34 a new table (Table 34a):

Table 34a—Elements of HoppingDescriptor

Name	Type	Valid range	Description
HoppingSequenceID	Integer	0x00–0x0ff	Indicate the ID of Hopping Sequence in use: 0x00: the default hopping sequence. 0x01: a hopping sequence generated by PAN coordinator. 0x02–0xff: a hopping sequence set by NHL. If a coordinator receives an association request command with HoppingSequenceID of one, it replies with the channel hopping sequence in an association response command.
HoppingSequenceLength	Integer	0x0000–0xffff	Specifies the length of Hopping Sequence. When *macDSMEenabled* is TRUE, the *macSequenceLength* shall be set to zero for all *macHoppingSequenceID* values other than one.
Hopping Sequence	Set of octets	0x0000–0x01ff for each channel	Specifies the sequence of channel numbers, which is set by a higher layer. PAN coordinator may select the sequence to use when it establishes a PAN. In such case, the HoppingSequenceID shall be set to one
ChannelOffset	Integer	0x0000–0xffff	Specifies the offset value of Hopping Sequence.
ChannelOffsetBitmapLength	Integer	0x0000–0xffff	Specifies the length of ChannelOffsetBitmap in octets.
ChannelOffsetBitmap	Set of octets	—	Bit value of ChannelOffsetBitmap sequence represents whether the corresponding channel offset is used. If the corresponding channel offset is used, the bit value shall be set to one. Otherwise, it shall be set to zero. For instance, if the 1st, 2nd, 4th channels offset are used with ChannelOffsetBitmapLength of 16, ChannelOffsetBitmap shall be 0b0110100000000000. The number of effective bits in ChannelOffsetBitmap is the same as the number of available channels in current channel page.

Insert before 6.2.12.2 the following text:

If the SDIndex parameter is nonzero and the MLME is not currently tracking the beacon of the coordinator through which it is associated, the MLME shall issue the MLME START.confirm primitive with a status of TRACKING_OFF.

If *macUseEnhancedBeacon* is TRUE, the MLME shall use enhanced beacons rather than standard beacons containing the IEs provided or *macEBIEList* if present. The enhanced beacon shall be secured according to the value of the BeaconSecurityLevel parameter.

Insert before 6.3 the following subclauses (6.2.18–6.2.21.3.4):

6.2.18 Primitives for Beacon Generation

6.2.18.1 MLME-BEACON.request

The MLME-BEACON.request primitive requests the generation of a beacon or enhanced beacon in a non-beacon-enabled PAN, either in response to a beacon request command when *macBeaconAutoRespond* is FALSE, or on demand, e.g., to send beacons to enable a TSCH passive scan.

The semantics of this primitive are:

```
MLME-BEACON.request          (
                             BeaconType,
                             Channel,
                             ChannelPage,
                             SuperFrameOrder,
                             BeaconSecurityLevel,
                             BeaconKeyIdMode,
                             BeaconKeySource,
                             BeaconKeyIndex,
                             DstAddrMode
                             DstAddr,
                             BSNSuppression
                             )
```

The primitive parameters are defined in Table 44a.

The MLME-BEACON.request primitive may be generated by a higher layer when a beacon or enhanced beacon is to be sent in a non-beacon enabled PAN, either on demand, or in response to a MLME-BEACON-REQUEST.indication when *macBeaconAutoRespond* is FALSE.

On receipt of the MLME-BEACON.request primitive, the MAC sublayer entity constructs and transmits a beacon or enhanced beacon depending on the value of the BeaconType parameter.

The MAC sublayer builds an MPDU to transmit from the supplied arguments. The flags in the DstAddrMode parameters correspond to the Addressing fields in the Frame Control field (refer to 5.2.1.1) and are used to construct both the Frame Control and Addressing fields of the MHR.

The frame control and addressing mode options for beacon and enhanced beacons are described in 5.2.2.1.1.

The address used by the coordinator in its beacon frames is determined by the current value of *macShortAddress*, which is set by the next higher layer before issuing this primitive.

Table 44a—MLME-BEACON.request parameters

Name	Type	Valid range	Description
BeaconType	Integer	0x00–0x01	Indicates whether to send a beacon (0x00) or enhanced beacon (0x01).
Channel	Integer	Any valid channel number	The channel number to use.
ChannelPage	Integer	Any valid channel page	The channel page to use.
SuperframeOrder	Integer	0–15	The length of the active portion of the superframe, including the beacon frame.
BeaconSecurity-Level	Integer	0x00–0x07	The security level to be used for beacon frames (refer to Table 95).
BeaconKeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (refer to Table 96). This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.
BeaconKeySource	Set of 0, 4, or 8 octets	As specified by the BeaconKeyIdMode parameter	The originator of the key to be used (refer to 7.4.3.1). This parameter is ignored if the Beacon-KeyIdMode parameter is ignored or set to 0x00.
BeaconKeyIndex	Integer	0x01–0xff	The index of the key to be used (refer to 7.4.3.2). This parameter is ignored if the BeaconKeyId-Mode parameter is ignored or set to 0x00.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive and subsequent beacon. This value can take one of the following values: 0x00 = no address (Addressing fields omitted, refer to 5.2.1.1.6). 0x01 = 8-bit address. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddr	Device Address	As specified by the DstAddrMode parameter	If sent in response to an MLME-BEACON-REQUEST.indication, the device who sent the beacon request, otherwise the short broadcast address (0xffff).
BSNSuppression	Boolean	TRUE, FALSE	If BeaconType = 0x01, then if BSNSuppression is TRUE, the EBSN is omitted from the frame and the Sequence Number Suppression field of the Frame Control field is set to one.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled field of the Frame Control field to one. The MAC sublayer will perform outgoing processing on the frame, as described in 7.2.1.

If BeaconType is set to zero, the *macBSN* is used for the sequence number. If BeaconType is set to one, and BSNSuppression is FALSE, the *macEBSN* is used. If BeaconType is set to one, and BSNSuppression is TRUE, then the sequence number is omitted and the Sequence Number Suppression field of the Frame Control field is set to one.

If BeaconType is set to one, then an enhanced beacon is constructed using the list of IEs in *macEBIEList*, and the IE List field of the Frame Control field is set to one.

If the length of the beacon frame exceeds *aMaxPHYPacketSize* (e.g., due to the additional overhead required for security processing), the MAC sublayer shall discard the beacon frame and issue the MLME-BEACON.confirm primitive with a status of FRAME_TOO_LONG.

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

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

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

6.2.18.2 MLME-BEACON.confirm

The semantics of this primitive are:

```
MLME-BEACON.confirm        (
                           status
                           )
```

The primitive parameters are defined in Table 44b.

The MLME-BEACON.confirm primitive is generated by the MAC sublayer entity in response to an MLME-BEACON.request primitive. The MLME-BEACON.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or the appropriate error code. The status values are fully described in 6.3.1.

Table 44b—MLME-BEACON.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, CHANNEL_ACCESS_FAILURE, FRAME_TOO_LONG, INVALID_PARAMETER	The result of the attempt to send the beacon or enhanced beacon

On receipt of the MLME-BEACON.confirm primitive, the higher layer is notified of the result of its request to transmit a beacon or enhanced beacon. If the transmission attempt was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter will indicate the error. If an enhanced beacon was requested, and the device does not support enhanced beacons, INVALID_PARAMETER will be returned.

6.2.18.3 MLME-BEACON-REQUEST.indication

The MLME-BEACON-REQUEST.indication primitive indicates the receipt of a beacon request. It is only available when *macBeaconAutoRespond* is FALSE.

The semantics of this primitive are:


```
MLME-BEACON-REQUEST.indication    (
                                  BeaconType,
                                  SrcAddrMode,
                                  SrcAddr,
                                  dstPANID,
                                  IEList
                                  )
```

The primitive parameters are defined in Table 44c.

Table 44c—MLME-BEACON-REQUEST.indication parameters

Name	Type	Valid range	Description
BeaconType	Enumera-tion	BEACON, ENHANCED BEACON	BEACON = 0: send a beacon. ENHANCED BEACON = 1: send an enhanced beacon.
SrcAddr Mode	Enumera-tion	0x00–0x03	The source addressing mode for device from whom the beacon request was received. This value can take one of the following values: 0x00 = no address (Addressing fields omitted, refer to 5.2.1.1.8). 0x01 = 8-bit address. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcAddr	Device Address	As specified by the SrcAddrMode parameter	The device who sent the beacon request, if present, otherwise the short broadcast address (0xffff)
DstPANID	Integer	0x0000–0xffff	The PANID contained in the beacon request, or the broadcast PAN ID (0xffff) if PAN ID not present.
IEList	List of IEs	As described in 5.2.4	If BeaconType = 0x01, the EB Filter IE and/or other IEs are contained in the beacon request. Otherwise it is empty.

The MLME-BEACON-REQUEST.indication primitive is generated by the MAC sublayer and issued to a higher layer on receipt of a beacon request or enhanced beacon request. The MLME-BEACON-REQUEST.indication is only generated when *macBeaconAutoRespond* is FALSE.

A higher layer may use the information contained in the MLME-BEACON-REQUEST.indication to construct and send a beacon or enhanced beacon using the MLME-BEACON.request primitive.

6.2.19 Primitives for TSCH

6.2.19.1 MLME SET SLOTFRAME.request

The MLME-SET-SLOTFRAME.request primitive is used to add, delete, or modify a slotframe at the MAC sublayer. The slotframeHandle is supplied by a higher layer.

The semantics of this primitive are:

```
MLME-SET-SLOTFRAME.request          (
                                    slotframeHandle,
                                    operation,
                                    size
                                    )
```

The primitive parameters are defined in Table 44e.

Table 44d—MLME-SET-SLOTFRAME.request parameters

Name	Type	Valid range	Description
slotframeHandle	Integer	0x00–0xff	Unique identifier of the slotframe.
Operation	Enumeration	ADD, DELETE, MODIFY	Operation to perform on the slotframe. ADD = 0, DELETE = 2, MODIFY = 3.
Size	Integer	0x0000–0xffff	Number of timeslots in the new slotframe.

An MLME-SET-SLOTFRAME.request may be used by a higher layer to add, delete, or modify a slotframe at the MAC sublayer.

On receipt of an MLME-SET-SLOTFRAME.request, the MLME shall verify the parameters passed with the primitive. If the requested operation is set to ADD, the MLME shall attempt to add an entry into the *macSlotframeTable*. A *macSlotframeTable* entry shall be stored for each slotframe. If the operation is set to DELETE, all parameters except slotframeHandle and operation shall be ignored, and the slotframe record shall be deleted from the *macSlotFrameTable*. If there are links in the slotframe that are being deleted, the links shall be deleted from the MAC layer. If the device is in the middle of using a link in the slotframe that is being updated or deleted, the update shall be postponed until after the link operation completes either through a successful unacknowledged transmission, time-out for receipt of an expected acknowledgment, receipt of an invalid or unacknowledged frame, or transmission of an acknowledgment upon receipt of a valid frame. If the operation is set to MODIFY, it shall attempt to update an existing slotframe record in the table.

6.2.19.2 MLME-SET-SLOTFRAME.confirm

The MLME-SET-SLOTFRAME.confirm primitive reports the results of the MLME-SET-SLOTFRAME.request command. The slotframeHandle is that which was supplied by a higher layer in the prior call to the MLME-SET-SLOTFRAME.request.

The semantics of this primitive are:

```
MLME-SET-SLOTFRAME.confirm          (
                                    slotframeHandle,
                                    status
                                    )
```

The primitive parameters are defined in Table 44e.

The MLME-SET-SLOTFRAME.confirm primitive is generated by the MLME when the MLME-SET-SLOTFRAME.request is completed.

Table 44e—MLME-SET-SLOTFRAME.confirm parameters

Name	Type	Valid range	Description
slotframe-Handle	Integer	0x00–0xff	Unique identifier of the slotframe to be added, deleted, or modified.
Status	Enumeration	SUCCESS, INVALID_PARAMETER, SLOTFRAME_NOT_FOUND, MAX_SLOTFRAMES_EXCEEDED	Indicates results of the MLME-SET-SLOTFRAME.request.

If any of the arguments fail a range check, the status shall be INVALID_PARAMETER. If a new slotframe is being added and the *macSlotFrameTable* is already full, the status shall be MAX_SLOTFRAMES_EXCEEDED. If an update or deletion is being requested and the corresponding slotframe cannot be found, the status shall be SLOTFRAME_NOT_FOUND. If an add is being requested with a slotframeHandle corresponding to an existing slotframe, the status shall be INVALID_PARAMETER. Otherwise the status code shall be set to SUCCESS.

6.2.19.3 MLME SET LINK.request

The MLME-SET-LINK.request primitive requests to add a new link, or delete or modify an existing link at the MAC sublayer. The slotframeHandle and linkHandle are supplied by a higher layer.

The semantics of this primitive are:

```
MLME-SET-LINK.request          (
                               operation,
                               linkHandle,
                               slotframeHandle,
                               timeslot,
                               channelOffset,
                               linkOptions,
                               linkType,
                               nodeAddr
                               (
```

The primitive parameters are defined in Table 44f.

MLME-SET-LINK.request primitive may be used by the device management layer to add, delete, or modify a link in a slotframe.

When operationType is set to ADD_LINK, the MAC layer shall attempt to add the link to a new *macLinkTable* associated with the indicated slotframe. When operationType is set to DELETE_LINK, all parameters except linkHandle and slotframeHandle shall be ignored, and the indicated link shall be deleted from the associated *macLinkTable*. When operationType is set to MODIFY_LINK, the MAC layer shall attempt to update the indicated link. If the link is currently in use, the delete or modify operation shall be postponed until the link operation completes, either through a successful unacknowledged transmission, time-out for receipt of an expected acknowledgment, receipt of an invalid or unacknowledged frame, or transmission of an acknowledgment upon receipt of a valid frame. Upon completion, the result of the operation shall be reported through the corresponding MLME-SET-LINK.confirm primitive.

Table 44f—MLME-SET-LINK.request parameters

Name	Type	Valid range	Description
Operation	Enumeration	ADD_LINK, DELETE_LINK, MODIFY_LINK	Type of link management operation to be performed. ADD_LINK = 0, DELETE_LINK = 1, MODIFY_LINK = 2,
linkHandle	Integer	0x0000–0xffff	Unique identifier (local to specified slotframe) for the link.
slotframeHandle	Integer	0x00–0xff	The slotframeHandle of the slotframe to which the link is associated.
Timeslot	Integer	0x0000–0xffff	Timeslot of the link to be added, as described in 5.1.1.5.
channelOffset	Integer	As defined in 5.1.1.5.3	The Channel offset of the link
linkOptions	Bitmap	0x00–0x0ff	b0 = Transmit, b1 = Receive, b2 = Shared, b3= Timekeeping, b4–b7 reserved.
linkType	Enumeration	NORMAL, ADVERTISING	Type of link. NORMAL = 0. ADVERTISING = 1, and indicates the link may be used to send an Enhanced beacon.
nodeAddr	Integer	0x0000–0xffff	Address of neighbor device connected by the link. 0xffff indicates the link may be used for frames destined for the broadcast address.

The use of the Shared bit in the linkOptions bitmap indicates that if the link is also a transmit link that the device shall back off according to the method described in 5.1.5. Shared behavior is not defined for receive links. The default linkType is NORMAL. ADVERTISE links may be used to send Enhanced beacons as the result of the MAC receiving a MLME-BEACON.request. The higher layer may indicate that a neighbor is to be used for synchronization by setting the Timekeeping bit in the linkOptions on receive links from the neighbor. The Timekeeping bit in the linkOptions is not defined for transmit links.

6.2.19.4 MLME-SET-LINK.confirm

The MLME-SET-LINK.confirm primitive indicates the result of add, delete, or modify link operation. The linkHandle and slotframeHandle are those that were supplied by a higher layer in the prior call to MLME-SET-LINK.request

The semantics of this primitive are:

```
MLME-SET-LINK.confirm         (
                              status,
                              linkHandle,
                              slotframeHandle
                              )
```

The primitive parameters are defined in Table 44g.

Table 44g—MLME-SET-LINK.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, INVALID_PARAMETER, UNKNOWN_LINK, MAX_LINKS_EXCEEDED	Result of the request operation.
linkHandle	Integer	0x0000–0xffff	Unique (local to specified slotframe) identifier for the link.
slotframeHandle	Integer	0x00–0xff	The slotframeHandle of the slotframe to which the link is associated.

The status of the primitive shall indicate SUCCESS if the operation completed successfully. If any of the arguments fail a range check, the status shall be INVALID_PARAMETER. If a new link is being added and the *macLinkTable* is already full, the status shall be MAX_LINKS_EXCEEDED. A *macLinkTable* shall be stored for each link in a slotframe. If an update or deletion is being requested and the corresponding link cannot be found, the status shall be UNKNOWN_LINK. If an add is being requested with a linkHandle corresponding to an existing link, the status shall be INVALID_PARAMETER.

6.2.19.5 MLME-TSCH-MODE.request

The MLME-TSCH-MODE.request requests to put the MAC into or out of the TSCH mode.

The semantics of this primitive are:

```
MLME-TSCH-MODE.request        (
                              TSCHMode
                              )
```

The primitive parameters are defined in Table 44h.

Table 44h—MLME-TSCH-MODE.request parameters

Name	Type	Valid range	Description
TSCHMode	Enumeration	ON, OFF	Target mode ON = 1: TSCH mode is to be started, OFF = 0: TSCH mode is to be stopped.

The MLME-TSCH-MODE.request may be generated by a higher layer after the device has received advertisements from the network and is synchronized to a network, i.e., in response to an MLME BEACON NOTIFY.indication.

Upon receipt of the request with TSCHMode set to ON, the MAC shall start operating its TSCH state machine using slotframes and links already contained in its *macSlotframeTable* and *macLinkTable* MAC PIB attributes. To successfully complete this request the device shall already be synchronized to a network. The MAC shall stop using slotframes and links upon receipt of the request with TSCHMode set to off.

6.2.19.6 MLME-TSCH-MODE.confirm

The MLME-TSCH-MODE.confirm primitive reports the result of the MLME-TSCH-MODE.request primitive.

The semantics of this primitive are:

```
MLME-TSCH-MODE.confirm        (
                              TSCHMode,
                              status
                              )
```

The primitive parameters are defined in Table 44i.

Table 44i—MLME-TSCH-MODE.confirm parameters

Name	Type	Valid range	Description
TSCHMode	Enumeration	ON, OFF	Target mode ON = 1: TSCH mode is to be started, OFF = 0: TSCH mode is to be stopped.
Status	Enumeration	SUCCESS, NO_SYNC	Indicates results of the MLME-TSCH-MODE.request.

The MLME-TSCH-MODE.confirm is generated by the MAC layer to indicate completion of the corresponding request. If the corresponding request was to turn on its TSCH state machine, but the MAC layer has not been synchronized to a network, the status shall be NO_SYNC. Otherwise, the status shall be SUCCESS.

If the corresponding request was to turn off its TSCH state machine, the status shall be SUCCESS, and the MAC layer shall stop its TSCH state machine if in use.

6.2.19.7 MLME-KEEP-ALIVE.request

The MLME-KEEP-ALIVE.request primitive requests that frames be sent to a device with a minimum period.

The semantics of this primitive are:

```
MLME-KEEP-ALIVE.request       (
                              dstAddr,
                              keepAlivePeriod
                              )
```

The primitive parameters are defined in Table 44j.

Table 44j—MLME-KEEP-ALIVE.request parameters

Name	Type	Valid range	Description
dstAddr	Integer	0x0000–0xffff	Address of the neighbor device with which to maintain timing. Keep-alives with dstAddr of 0xffff do not expect to be acknowledged and cannot be used for timekeeping.
keepAlivePeriod	Integer	0x0001–0xffff	Period in timeslots after which a frame is sent if no frames have been sent to dstAddr.

Upon receipt of the request, the MAC layer shall monitor for frames sent to the destination node specified in the dstAddr parameter. If no frame is sent to the destination node in keepAlivePeriod, the MAC shall send any valid frame to the node dstAddr with an acknowledgment requested. A higher layer is responsible for determining which nodes should be sent Keep-alives, and can indicate a device by setting the Timekeeping bit in the linkOptions on any transmit link from that device.

6.2.19.8 MLME-KEEP-ALIVE.confirm

The MLME-KEEP-ALIVE.confirm primitive reports the results of a request that frames be sent to a device with a minimum period.

The semantics of this primitive are:

```
MLME-KEEP-ALIVE.confirm        (
                               status
                               )
```

The primitive parameters are defined in Table 44k.

Table 44k—MLME-KEEP-ALIVE.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER	Indicates results of the MLME-KEEP-ALIVE.request.

The MAC layer shall generate MLME-KEEP-ALIVE.confirm to acknowledge that it received MLME-KEEP-ALIVE request. If the dstAddr of the MLME-KEEP-ALIVE.request does not exist in the devices neighbor table and is not 0xffff, the status of the primitive shall indicate INVALID_PARAMETER. Otherwise it shall return SUCCESS.

6.2.20 Primitives for LLDN

6.2.20.1 General

These primitives control the different modes for the configuration and operation of the superframe in an LLDN.

6.2.20.2 MLME-LLDN-DISCOVERY.request

This primitive switches the LLDN into the Discover state.

The semantics of this primitive are:

```
MLME-LLDN-DISCOVERY.request          (
                                     LowLatencyNetworkConfiguration
                                     )
```

Table 44l specifies the parameters for the MLME-LLDN-DISCOVERY.request primitive.

Table 44l—MLME-LLDN-DISCOVERY.request parameters

Name	Type	Valid range	Description
LowLatencyNet-workConfiguration	Set of octets of variable length	—	Contains the necessary configuration parameters from the next higher layer for the LLDN in Discovery state

The MLME-LLDN-DISCOVERY.request primitive is generated by the next higher layer of an LLDN coordinator and issued to its MLME to switch the LLDN into the Discovery state as described in 5.1.9.2.

When the MLME of an LLDN coordinator receives the MLME-LLDN-DISCOVERY.request primitive, it sets the Transmission State field in the Flags field of the payload of the 1 octet MHR Beacons to the value for Discovery state as indicated in 5.2.2.5 and follows the procedures as defined for Discovery state in 5.1.9.2.

6.2.20.3 MLME-LLDN-DISCOVERY.confirm

This primitive indicates the end of the Discover state and gives the status of the Discovery state to a higher layer.

The semantics of this primitive are:

```
MLME-LLDN-DISCOVERY.confirm    (
                               status,
                               DiscoveredDevices,
                               LowLatencyNetworkConfiguration
                               )
```

Table 44m specifies the parameters for the MLME-LLDN-DISCOVERY.confirm primitive.

The MLME-LLDN-DISCOVERY.confirm primitive is generated by the MLME of the LLDN coordinator and issued to its next higher layer to indicate the end of the Discovery state in the LLDN. It returns the number of discovered devices and the collected information about the discovered devices in the LLDN to the next higher layer. The MLME-LLDN-DISCOVERY.confirm primitive will either return a status SUCCESS, indicating that all devices with *macLLenabled* set to TRUE within range have been discovered, or an error code of NO_DEVICE (expected to discover device, but none found) or ABORTED (Discovery state finished before all devices had been discovered).

Table 44m—MLME-LLDN-DISCOVERY.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_DEVICE, ABORTED	The status of the Discovery state when finished.
DiscoveredDevices	Integer	0 .. 128	Number of discovered devices.
LowLatencyNetwork-Configuration	Set of octets of variable length	—	Discovered information of the discovered devices of the LLDN for the next higher layer

After the next higher layer of an LLDN coordinator receives the MLME-LLDN-DISCOVERY.confirm primitive, the LLDN coordinator determines a configuration of the LLDN based on the status and the information about the discovered devices received in DiscoveredDevices and LowLatencyNetworkConfiguration DiscoveryModeStatus. It uses an algorithm outside the scope of this standard. The next higher layer of the LLDN coordinator should then issue the MLME-LLDN-CONFIGURATION.request primitive to its MLME.

6.2.20.4 MLME-LLDN-CONFIGURATION.request

This primitive switches the LLDN into the Configuration state.

The semantics of this primitive are:

```
MLME-LLDN-CONFIGURATION.request    (
                                    LowLatencyNetworkConfiguration
                                    )
```

Table 44n specifies the parameters for the MLME-LLDN-CONFIGURATION.request primitive.

Table 44n—MLME-LLDN-CONFIGURATION.request parameters

Name	Type	Valid range	Description
LowLatencyNet-workConfiguration	Set of octets of variable length	—	Contains the necessary configuration parameters for the LLDN in the Configuration state

The MLME-LLDN-CONFIGURATION.request primitive is generated by the next higher layer of an LLDN coordinator and issued to its MLME to switch the LLDN into the Configuration state as described in 5.1.9.3.

When the MLME of an LLDN coordinator receives the MLME-LLDN-CONFIGURATION.request primitive, it sets the Transmission State field in the Flags field of the payload of the 1 octet MHR Beacons as described in 5.2.2.5.2 to the value for the Configuration state as indicated in 5.2.2.5 and follows the procedures as defined for Configuration state described in 5.1.9.3.

6.2.20.5 MLME-LLDN-CONFIGURATION.confirm

This primitive indicates the end of the Configuration state and gives the status of the Configuration state to the next higher layer.

6.2.20.5.1 Semantics of the service primitive

The semantics of this primitive are:

```
MLME-LLDN-CONFIGURATION.confirm      (
                                     status,
                                     ConfiguredDevices,
                                     LowLatencyNetworkConfiguration
                                     )
```

Table 44o specifies the parameters for the MLME-LLDN-CONFIGURATION.confirm primitive.

Table 44o—MLME-LLDN-CONFIGURATION.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_DEVICE, ABORTED	The status of the Configuration state when finished.
ConfiguredDevices	Integer	0 .. 128	Number of configured devices.
LowLatencyNetwork-Configuration	Set of octets of variable length	—	Configuration of the configured devices of the LLDN for the next higher layer.

The MLME-LLDN-CONFIGURATION.confirm primitive is generated by the MLME of the LLDN coordinator and issued to its next higher layer to indicate the end of the Configuration state in the LLDN. It returns the number of configured devices and the collected configuration information about the configured devices in the LLDN to the next higher layer. The MLME-LLDN-CONFIGURATION.confirm primitive will either return a status SUCCESS, indicating that all devices with *macLLenabled* set to TRUE within range have been configured, or an error code of NO_DEVICE (expected to configure device, but none found) or ABORTED (Discovery state finished before all discovered devices had been configured).

When the next higher layer of an LLDN coordinator receives the MLME-LLDN-CONFIRMATION.confirm primitive, the next higher layer of the LLDN coordinator should issue the MLME-LLDN-ONLINE.request (status is SUCCESS), the MLME-LLDN-CONFIGURATION.request (status is ABORTED), or the MLME-LLDN-DISCOVERY.request (status is NO_DEVICE) primitive to its MLME.

6.2.20.6 MLME-LLDN-ONLINE.request

This primitive switches the LLDN into the Online state.

The semantics of this primitive are:

```
MLME-LLDN-ONLINE.request       (
                               )
```

The MLME-LLDN-ONLINE.request primitive is generated by the next higher layer of an LLDN coordinator and issued to its MLME to switch the LLDN into the Online state (5.1.9.4).

When the MLME of an LLDN coordinator receives the MLME-LLDN-ONLINE.request primitive, the coordinator shall switch over to Online state by setting appropriate flags in its beacon payload, as described 5.2.2.5.2, and follows the procedures as defined for Online state in 5.1.9.4.

6.2.20.7 MLME-LLDN-ONLINE.indication

This primitive indicates any problems during the Online state to the next higher layer.

The semantics of this primitive are:

```
MLME-LLDN-ONLINE.indication    (
                               status,
                               AdditionalInformation
                               )
```

Table 44p specifies the parameters for the MLME-LLDN-ONLINE.indication primitive.

Table 44p—MLME-LLDN-ONLINE.indication parameters

Name	Type	Valid range	Description
status	Enumeration	NONE, UNSPECIFIED	Contains the status in the LLDN including any discovered problems.
AdditionalInformation	Set of octets of variable length	—	Additional supporting information

The MLME-LLDN-ONLINE.indication primitive is generated by the MLME of any LLDN device and issued to its next higher layer to indicate the status and any problems that occurred in the LLDN during the operation in online mode. It returns the indication of the problem (NONE or UNSPECIFIED) and the additional supporting information to the higher layer.

When the next higher layer of an LLDN device receives the MLME-LLDN-ONLINE.indication primitive, the LLDN device determines appropriate countermeasures using an algorithm outside the scope of this standard.

6.2.21 Primitives for DSME

6.2.21.1 Primitives for DSME-GTS management

The MLME-SAP DSME-GTS management primitives define how DSME-GTSs are requested and maintained. A device wishing to use these primitives and DSME-GTSs in general will already be tracking the beacons of its coordinator.

6.2.21.1.1 MLME-DSME-GTS.request

This primitive allows a DSME-enabled device to request an allocation of new DSME-GTS or deallocation, duplicated allocation notification, reduce, or restart of existing DSME-GTSs.

The semantics of this primitive are:

```
MLME-DSME-GTS.request          (
                               DeviceAddress,
                               ManagmentType,
                               Direction,
                               PrioritizedChannelAccess,
                               NumSlot,
                               PreferredSuperframeID,
                               PreferredSlotID,
                               DSMESABSpecification,
                               SecurityLevel,
                               KeyIdMode,
                               KeySource,
                               KeyIndex
                               )
```

Table 44q specifies the parameters for the MLME-DSME-GTS.request primitive.

Table 44q—MLME-DSME-GTS.request parameters

Name	Type	Valid range	Description
DeviceAddress	Integer	0x0000–0xfffd	The short address of the neighboring device to request the management of DSME-GTSs.
ManagementType	Integer	0x00–0x04	The type of the management request 0b000: deallocation 0b001: allocation 0b010: duplicated allocation notification 0b011: reduce 0b100: restart
Direction	Integer	0x00–0x01	The direction of DSME-GTSs. 0x00: TX (Transmission) 0x01: RX (Reception)
PrioritizedChannelAccess	Integer	0x00–0x01	The priority level. 0x00: low priority. 0x01: high priority.
NumSlot	Integer	0x00–0xff	The number of slots to be requested for allocation. This parameter is ignored if the ManagementType is not 0b001 (allocation).
PreferredSuperframeID	Integer	0x0000–0xffff	The index of the preferred superframe in a multi-superframe. This parameter is ignored if the ManagementType is not 0b001 (allocation).
PreferredSlotID	Integer	0x00–0x0e	The index of the preferred slot in the preferred superframe. This parameter is ignored if the ManagementType is not 0b001 (allocation).

Table 44q—MLME-DSME-GTS.request parameters *(continued)*

Name	Type	Valid range	Description
DSMESABSpecification	DSMESAB-Specification Value	Refer to 5.3.11.4.7	If the ManagementType is 0b001, this parameter contains the information of the current DSME-GTS allocation status and slot availability in one hop neighborhood of the requesting device. If the ManagementType is not 0b001, this parameter indicates the DSME-GTSs to deallocate, notify duplicated allocation, reduce, or restart.
SecurityLevel	Integer	0x00–0x07	The security level to be used (refer to Table 95).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (refer to Table 96). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used, as described in 7.4.3.1. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used as described in 7.4.3.2. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

The MLME-DSME-GTS.request primitive is generated by the higher layer of a device and issued to its MLME to request the allocation of new DSME-GTSs or to request the deallocation, duplicated allocation notification, reduce, or change of existing DSME-GTSs.

On receipt of the MLME-DSME-GTS.request primitive, the MLME of the device shall send a DSME-GTS request command frame as described in 5.3.11.4 to the DeviceAddress. The ManagementType, Direction, PrioritizedChannelAccess, NumSlot, PreferredSuperframeID, PreferredSlotID, DSMESABSubblockIndex, and DSMESABSubblockLength parameters shall be contained in the corresponding fields of the command frame. The DSME SAB Sub-block field shall contain a bitmap of a sub-block of *macDSMESAB*. The DSMESABSubblockIndex and DSMESABSubblockLength indicate the bitmap of which sub-block shall be contained in the command frame.

If *macShortAddress* is equal to 0xfffe or 0xffff, the Source device is not permitted to request a DSME-GTS allocation. In this case, the MLME issues the MLME-DSME-GTS.confirm primitive containing a status of NO_SHORT_ADDRESS.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME shall set the Security Enabled field of the Frame Control field to one. The MAC sub-layer shall perform outgoing processing on the frame based on SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.3.1. If any error occurs during outgoing frame processing, the MLME shall discard the frame and issue the MLME-DSME-GTS.confirm primitive with the error status returned by outgoing frame processing.

If the DSME-GTS request command frame cannot be sent due to the channel condition, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a DSME-GTS request command, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-GTS.confirm primitive with a status of NO_ACK.

If the DSME-GTS request command frame is being sent, the source device shall wait for at most *macMax-FrameTotalWaitTime* CAP symbols, if no DSME-GTS reply command frame from the destination device appears within this time, the MLME of the source device shall notify the next higher layer of the failure by the MLME-DSME-GTS.confirm primitive with a status of NO_DATA.

6.2.21.1.2 MLME-DSME-GTS.indication

This primitive reports the reception of a DSME-GTS request command.

The semantics of this primitive are:

```
MLME-DSME-GTS.indication        (
                                DeviceAddress,
                                ManagmentType,
                                Direction,
                                PrioritizedChannelAccess,
                                NumSlot,
                                PreferredSuperframeID,
                                PreferredSlotID,
                                DSMESABSpecification,
                                )
```

Table 44r specifies the parameters for the MLME-DSME-GTS.indication primitive.

Table 44r—MLME-DSME-GTS.indication parameters

Name	Type	Valid range	Description
Device Address	Device Address	0x0000–0xfffd	The 16-bit short address of the device that has transmitted the received DSME-GTS request command.
ManagementType	Enumeration	0x00–0x04	The type of the management request 0b000: deallocation 0b001: allocation 0b010: duplicated allocation notification 0b011: reduce 0b100: restart 0b101: DSME-GTS expiration
Direction	Enumeration	0x00–0x01	The direction of DSME-GTSs. 0x00: TX (Transmission) 0x01: RX (Reception)
PrioritizedChanne-lAccess	Enumeration	0x00–0x01	The priority level. 0x00: low priority. 0x01: high priority.
NumSlot	Integer	0x00–0xff	The number of slots to be requested for allocation. This parameter is ignored if the ManagementType is not 0b001 (allocation).

Table 44r—MLME-DSME-GTS.indication parameters *(continued)*

Name	Type	Valid range	Description
PreferredSuper-frameID	Integer	0x0000–0xffff	The index of the preferred superframe in a multi-superframe. This parameter is ignored if the ManagementType is not 0b001 (allocation).
PreferredSlotID	Integer	0x00–0x0e	The index of the preferred slot in the preferred superframe. This parameter is ignored if the ManagementType is not 0b001 (allocation).
DSMESAB-Specification	DSMESAB-Specification Value	As described in 5.3.11.4.7	If the ManagementType is 0b001, this parameter contains the information of the current DSME-GTS allocation status and slot availability in one hop neighborhood of the requesting device. If the ManagementType is not 0b001, this parameter indicates the DSME-GTSs to deallocate, notify duplicated allocation, reduce, or restart.

This primitive is generated by the MLME of a device and issued to its next higher layer upon the reception of a DSME-GTS request command frame.

On receipt of the MLME-DSME-GTS.indication primitive, the higher layer is notified of the reception of a DSME-GTS request.

6.2.21.1.3 MLME-DSME-GTS.response

This primitive allows the next higher layer of a device to respond to the MLME-DSME-GTS.indication primitive.

The semantics of this primitive are:

```
MLME-DSME-GTS.response     (
                           DeviceAddress,
                           ManagmentType,
                           Direction,
                           PrioritizedChannelAccess,
                           ChannelOffset,
                           DSMESABSpecification,
                           Status
                           )
```

Table 44s specifies the parameters for the MLME-DSME-GTS.response primitive.

The MLME-DSME-GTS.response primitive can be generated by the next higher layer and issued to its MLME to respond to the allocation, deallocation, duplicated allocation notification, reduce, or restart of DSME-GTS.

Table 44s—MLME-DSME-GTS.response parameters

Name	Type	Valid range	Description
Device Address	Device Address	0x0000–0xfffd	The 16-bit short address of the device that has transmitted the received DSME-GTS request command.
ManagementType	Enumeration	0x00–0x04	The type of the management request 0b000: deallocation 0b001: allocation 0b010: duplicated allocation notification 0b011: reduce, 0b100: restart
Direction	Enumeration	0x00–0x01	The direction of DSME-GTSs. 0x00: TX (Transmission) 0x01: RX (Reception)
PrioritizedChannel Access	Enumeration	0x00–0x01	The priority level. 0x00: low priority. 0x01: high priority.
ChannelOffset	Integer	0x0000–0xffff	This parameter specifies the offset value of Hopping Sequence.
DSMESAB-Specification	DSMESAB-Specification Value	Described in 5.3.11.4.7	This parameter indicates the DSME-GTSs to allocate, deallocate, notify duplicated allocation, reduce, or restart.
Status	Enumeration	SUCCESS, DENIED, INVALID_PARAMETER	The status of the DSME-GTS request.

On receipt of the MLME-DSME-GTS.response primitive, the MLME of the device shall generate a DSME-GTS reply command frame as described in 5.3.11.5. The ManagmentType, Direction, Prioritized-ChannelAccess, ChannelOffset, and DSMESABSpecification parameters shall be contained in the corresponding fields of the command frame. The DSME-GTS Destination Address field shall be set to the value of DeviceAddress parameter. The Status field of the command frame shall be set to zero if the Status parameter value is SUCCESS. The Status field shall be set to one if the Status parameter value is DENIED. The Status field shall be set to two if the Status parameter value is INVALID_PARAMETER. Then the device shall broadcast the DSME-GTS reply command to its one-hop neighbors (i.e., the destination address set to 0xffff).

6.2.21.1.4 MLME-DSME-GTS.confirm

This primitive reports the results of a request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME-GTSs to the higher layer of the device.

The semantics of this primitive are:

```
MLME-DSME-GTS.confirm       (
                            DeviceAddress,
                            ManagmentType,
                            Direction,
                            PrioritizedChannelAccess,
                            ChannelOffset,
                            DSMESABSpecification,
                            Status
                            )
```


Table 44t specifies the parameters for the MLME-DSME-GTS.confirm primitive.

Table 44t—MLME-DSME-GTS.confirm parameters

Name	Type	Valid range	Description
Device Address	Device Address	0x0000–0xfffd	The 16-bit short address of the device that has transmitted the received DSME-GTS reply command.
ManagementType	Enumeration	0x00–0x04	The type of the management request 0b000: deallocation 0b001: allocation 0b010: duplicated allocation notification 0b011: reduce 0b100: restart
Direction	Enumeration	0x00–0x01	The direction of DSME-GTSs. 0x00: TX (Transmission) 0x01: RX (Reception)
Prioritized-ChannelAccess	Enumeration	0x00–0x01	The priority level. 0x00: low priority. 0x01: high priority.
ChannelOffset	Integer	0x0000–0xffff	This parameter specifies the offset value of Hopping Sequence.
DSMESAB-Specification	DSMESAB-Specification Value	Described in 5.3.11.4.7	This parameter indicates the DSME-GTSs to allocate, deallocate, notify duplicated allocation, reduce, or restart.
Status	Enumeration	SUCCESS, DENIED, INVALID_PARAMETER, NO_ACK, NO_DATA, CHANNEL_ACCESS_FAILURE	The status of the DSME-GTS request.

On receipt of a DSME-GTS reply command, the device shall check the DSME-GTS Destination Address field of the command frame.

If the DSME-GTS Destination Address in this command frame is the same as *macShortAddress*, the device shall check the Status field of the command frame. If the value of the Status field in the command frame is zero (SUCCESS), the device shall generate a DSME-GTS notify command frame as described in 5.3.11.6. The ManagmentType, Direction, PrioritizedChannelAccess, ChannelOffset, and DSMESABSpecification parameters shall be contained in the corresponding fields of the command frame. The DSME-GTS Destination Address field shall be set to the value of DeviceAddress parameter. The Status field of the command frame shall be set to zero if the Status parameter value is SUCCESS. The Status field shall be set to one if the Status parameter value is DENIED. The Status field shall be set to two if the Status parameter value is INVALID_PARAMETER. Then the device shall broadcast the DSME-GTS notify command to its one-hop neighbors (i.e., the destination address set to 0xffff).

Also, the MLME of the device shall notify the higher layer with the result of the request to allocate, deallocate, notify duplicated allocation, reduce, or restart DSME-GTSs. If the value of the Status field in the command frame is zero, the Status in this primitive shall be set to SUCCESS. If the value of the Status field is one, the Status parameter shall be set to DENIED. If the value of the Status field is two, the Status parameter shall be set to INVALID_PARAMETER.

If the DSME-GTSDestinationAddress is different from *macShortAddress*, the device shall update its DSME SAB according to the DSME-GTS SAB Specification of the received command frame.

On receipt of the MLME-DSME-GTS.confirm primitive, the higher layer is notified of the result of its request to notify duplicated allocation, reduce, or restart a DSME-GTS.

6.2.21.1.5 DSME management message sequence charts

Figure 60c depicts the message flow for the case in which a Source device initiates the DSME-GTS allocation, deallocation, duplicated allocation notification, reduce, or restart.

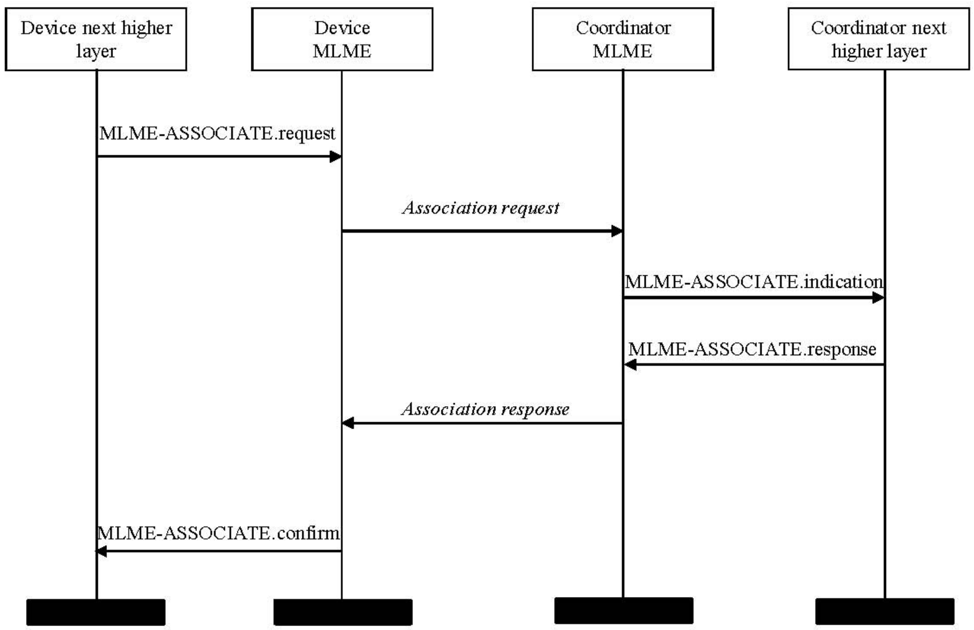

Figure 60c—Message sequence chart for DSME-GTS allocation and management

6.2.21.2 Primitives for requesting DSME information

MLME SAP DSME information primitives define how a device can acquire DSME information.

6.2.21.2.1 MLME-DSME-INFO.request

The MLME-DSME-INFO.request primitive allows a Source device to request the timestamp and the DSMESABSpecification of the Destination device or the DSME PAN Descriptor of the Connection device.

The semantics of this primitive are:

```
MLME-DSME-INFO.request     (
                           DstAddrMode,
                           DstAddr,
                           INFO,
                           DSMESABSubblockLength,
                           DSMESABSubblockIndex
                           )
```


The parameters for the MLME-DSME-INFO.request primitive are specified in Table 44u.

Table 44u—MLME-DSME-INFO.request parameters

Name	Type	Valid range	Description
DstAddr-Mode	Integer	0x02–0x03	The addressing mode of the Destination device to which the request is intended. This parameter can take one of the following values: 0x02 = 16-bit short address, 0x03 = 64-bit extended address.
DstAddr	Device-Address	As specified by DstAddrMode parameter	The address of the Destination device to which the request is intended.
INFO	Integer	0x00–0x02	The type of DSME information that are requested by the device. 0x00 = Timestamp 0x01 = DSMESABSpecification 0x02 = DSME PAN Descriptor
DSME SAB sub-block length	Integer	0x00–0x0ff	The length of the DSME SAB sub-block.
DSME SAB sub-block index	Integer	0x00–0xff	The index that indicates the beginning of the DSME SAB sub-block.

The MLME-DSME-INFO.request primitive is generated by the higher layer of a Source device and issued to its MLME to request the timestamp, DSMESABSpecification, or DSME PAN Descriptor of the Destination device.

On receipt of the MLME-DSME-INFO.request primitive, the MLME of the device sends a DSME-Information request command as described in 5.3.11.7.

If the DSME-Information request command cannot be sent due to a CSMA-CA algorithm failure, the MLME shall issue the MLME-DSME-INFO.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a DSME-Information request command, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-INFO.confirm primitive with a status of NO_ACK as described in 5.1.6.4. If an acknowledgment is received, the MLME shall wait for the DSME-Information reply command.

If a DSME-Information reply command is received, the MLME of the source device shall issue the MLME-DSME-INFO.confirm primitive with a status of SUCCESS.

And if a DSME-Information reply command is not received within *macMaxFrameTotalWaitTime* CAP symbols in a beacon-enabled PAN, or symbols in a non-beacon-enabled PAN, the MLME of the source device shall issue the MLME-DSME-INFO.confirm primitive with a status of NO_DATA.

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

6.2.21.2.2 MLME-DSME-INFO.indication

The MLME-DSME-INFO.indication primitive is used to indicate the reception of a DSME-Information request command frame.

The semantics of this primitive are:

```
MLME-DSME-INFO.indication      (
                               DeviceAddress,
                               INFO
                               )
```

The parameters for the MLME-DSME-INFO.indication primitive are specified in Table 44v.

Table 44v—MLME-DSME-INFO.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting DSME information.
INFO	Integer	0x00–0x02	The type of DSME information that are requested by the device. 0x00 = Timestamp 0x01 = DSMESABSpecification 0x02 = DSME PAN Descriptor

The MLME-DSME-INFO.indication primitive is generated by the MLME of the Destination device or the Connection device and issued to its next higher layer to indicate the reception of a DSME-Information request command. On receipt of the DSME-Information request command, the device shall send a DSME-Information reply command with requested information to the Source device.

6.2.21.2.3 MLME-DSME-INFO.confirm

The MLME-DSME-INFO.confirm primitive reports the results of a request for the timestamp, the DSME-GTS-SAB specification, or the DSME PAN Descriptor.

The semantics of this primitive are:

```
MLME-DSME-INFO.confirm         (
                               INFO,
                               Timestamp,
                               Superframe ID,
                               Slot ID,
                               DSMESABSpecification,
                               DSME PAN Descriptor,
                               status
                               )
```

Table 44w specifies the parameters for the MLME-DSME-INFO.confirm primitive.

Table 44w—MLME-DSME-INFO.confirm parameters

Name	Type	Valid range	Description
INFO	Integer	0x00–0x02	The type of DSME information that are requested by the device. 0x00 = Timestamp 0x01 = DSMESABSpecification 0x02 = DSME PAN Descriptor
Timestamp	Integer	0x000000–0xffffff	The time, in symbols, at which the DSME-Information reply command (refer to 5.3.11.8) was transmitted. This parameter is considered valid only if the value of the status parameter is SUCCESS. The symbol boundary is described by *macSyncSymbolOffset* as defined in Table 86. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
Superframe ID	Integer	0x0000–0xffff	The ID of the superframe in which the DSME-Information reply command was transmitted. The superframe ID is the sequence number of the superframe in a multi-superframe beginning from 0.
Slot ID	Integer	0x00–0xff	The ID of the superframe slot in which the DSME-Information reply command was transmitted. The slot ID is the sequence number of the DSME-GTSs (not including beacon or CAP slots) in a superframe beginning from 0.
DSMESAB-Specifica-tion	Set of octets of variable length	Refer to 5.3.11.4.7	The information of the current DSME-GTS allocation status and slot availability in one hop neighborhood of the replying device.
DSME PAN Descriptor	Set of octets of variable length	Refer to 5.2.4.9	The information of the configurations of the DSME-enabled PAN.
Status	Enumer-ation	SUCCESS, CHANNEL_ACCESS_ FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER.	The status of the DSME information request.

The MLME-DSME-INFO.confirm primitive is generated by the MLME and issued to its higher layer in response to an MLME-DSME-INFO.request primitive. If the INFO parameter is set to 0x00, the DSMES-ABSpecification and the DSME PAN Descriptor shall be ignored. If the INFO parameter is set to 0x01, the timestamp, the slot ID, and the DSME PAN Descriptor shall be ignored. If the INFO parameter is set to 0x02, the timestamp, the slot ID and the DSMESABSpecification shall be ignored.

On receipt of the MLME-DSME-INFO.confirm primitive, the next higher layer is notified of the result of the DSME information request.

6.2.21.2.4 DSME information sequence chart

Figure 60d illustrates the sequence of messages necessary for successful DSME information request.

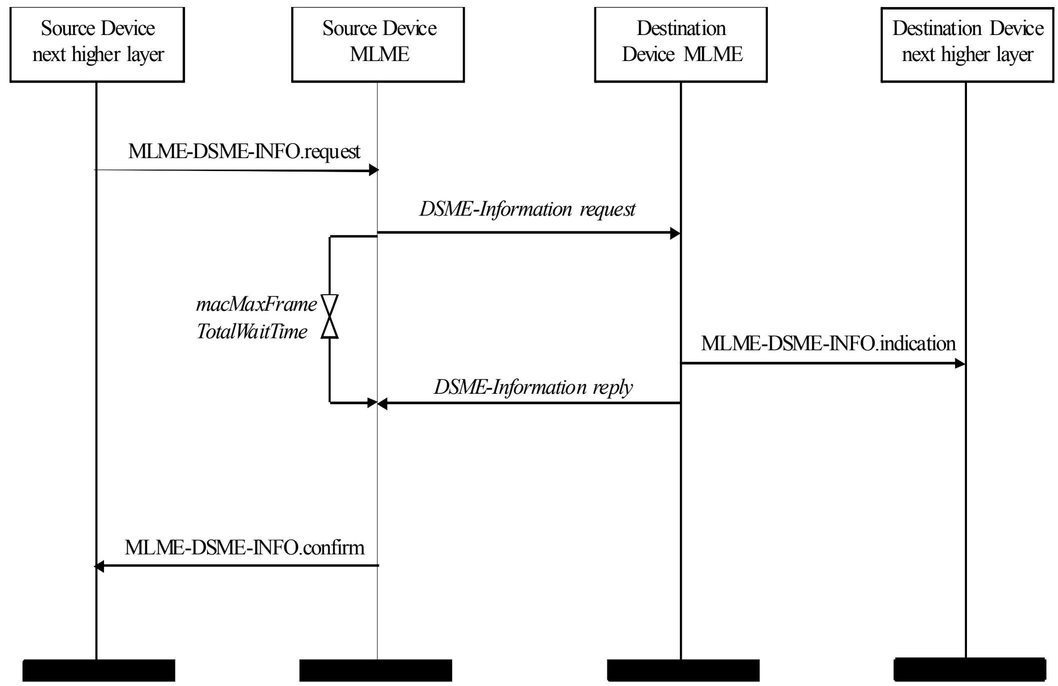

Figure 60d—Message sequence chart for DSME information request

6.2.21.3 Primitives for reporting the link status

The MLME-DSME-LINKSTATUSRPT primitives define how a source device reports the communication link status between the source device and the destination device.

6.2.21.3.1 MLME-DSME-LINKSTATUSRPT.request

The MLME-DSME-LINKSTATUSRPT.request primitive is generated by the higher layer of a source device and is issued to its MLME to request a device start a link quality statistic and periodically report the statistic results to the destination device.

The semantics of this primitive are:

```
MLME-DSME-LINKSTATUSRPT.request     (
                                    DstAddr,
                                    ReportPeriod
                                    )
```

Table 44x specifies the parameters for the MLME-DSME-LINKSTATUSRPT.request primitive.

Table 44x—MLME-DSME-LINKSTATUSRPT.request parameters

Name	Type	Valid range	Description
DstAddr	Integer	0x0000–0xffff	16-bit address of the Destination device to which the DSME-link status report request is intended.
Report-Period	Integer	0x000000–0xffffff	The time interval between two DSME-link status report command frames is defined as ReportPeriod × *aBaseSuperframeDuration* × 2^{MO} symbols. If the parameter equals to 0x000000, DSME-link status report command frame is not allowed to be sent.

The MLME-DSME-LINKSTATUSRPT.request primitive is generated by the higher layer of a device, and issued to its MLME to initiate a link status statistic.

On receipt of MLME-DSME-LINKSTATUSRPT.request primitive by a device, the MLME of the device attempts to generate a DSME link status report command as described in 6.2.21.3 with the information contained in this primitive, and if successful, sends it to the destination device according to the DstAddress parameter.

If the DSME link status report command frame cannot be sent due to a CSMA-CA algorithm failure, the MLME shall issue the MLME-DSME-LINKSTATUSRPT.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a DSME link status report command frame, the MLME expects an acknowledgment in return. If an acknowledgment is not received, the MLME shall issue the MLME-DSME-LINKSTATUSRPT.confirm primitive with a status of NO_ACK.

If the DSME link status report command frame has been acknowledged, the device shall send another DSME link status report command frame again in the interval defined in the parameter ReportPeriod.

If a device received a DSME link status report command frame from another device in the PAN, the destination device shall get the link status, and notify the result to its higher layer by the primitive MLME-DSME-LINKSTATUSRPT.indication.

6.2.21.3.2 MLME-DSME-LINKSTATUSRPT.indication

The MLME-DSME-LINKSTATUSRPT.indication primitive indicates the transfer of a DSME link status report of a device from the MAC sublayer to the local next higher layer.

The semantics of this primitive are:

```
MLME-DSME-LINKSTATUSRPT.indication    (
                                       DstAddr,
                                       LinkStatusSpecification
                                       )
```

Table 44y specifies the parameters for the MLME-DSME-LINKSTATUSRPT.indication request primitive.

Table 44y—MLME-DSME-LINKSTATUSRPT.indication parameters

Name	Type	Valid range	Description
DstAddr	Integer	0x0000–0xffff	16-bit address of the Destination device to which the DSME link status report request is intended.
LinkStatusSpecification	Link Status Specifica-tion	Refer to 5.3.11.11.3	The link status specification.

The MLME-DSME-LINKSTATUSRPT.indication primitive is generated by the MAC sublayer and issued to the next higher layer on receipt of a DSME link status report command.

On receipt of the MLME-DSME-LINKSTATUSRPT.indication primitive, the next higher layer is notified of the arrival of a DSME link status report command frame from a DSME device. The usage of the DSME link status report by the next higher layer is beyond the scope of this document.

6.2.21.3.3 MLME-DSME-LINKSTATUSRPT.confirm

The MLME-DSME-LINKSTATUSRPT.confirm primitive reports the results to start a DSME link status report process.

The semantics of this primitive are:

```
MLME-DSME-LINKSTATUSRPT.confirm     (
                                    status
                                    )
```

Table 44z specifies the parameters for the MLME-DSME-LINKSTATUSRPT.confirm primitive.

Table 44z—MLME-DSME-LINKSTATUSRPT.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	CHANNEL_ACCESS_FAILURE, NO_ACK, SUCCESS	The status of starting DSME link status report.

The MLME-DSME-LINKSTATUSRPT.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME LINKSATUSRPT.request primitive.

The MLME-DSME-LINKSTATUSRPT.confirm primitive returns a status of either SUCCESS, indicating the MAC sublayer has started reporting its statistic results periodically, or the appropriate error code.

On receipt of the MLME-DSME-LINKSTATUSRPT.confirm primitive by a device, the next higher layer is notified of the result of its request to start reporting link status in the PAN. If the request was successful, the status parameter indicates a successful DSME link status report operation. Otherwise, the status parameter shall indicate the error.

6.2.21.3.4 MLME-DSME-LINKSTATUSRPT message sequence charts

Figure 60e illustrates the sequence of messages necessary for DSME link status report initiated by a source device.

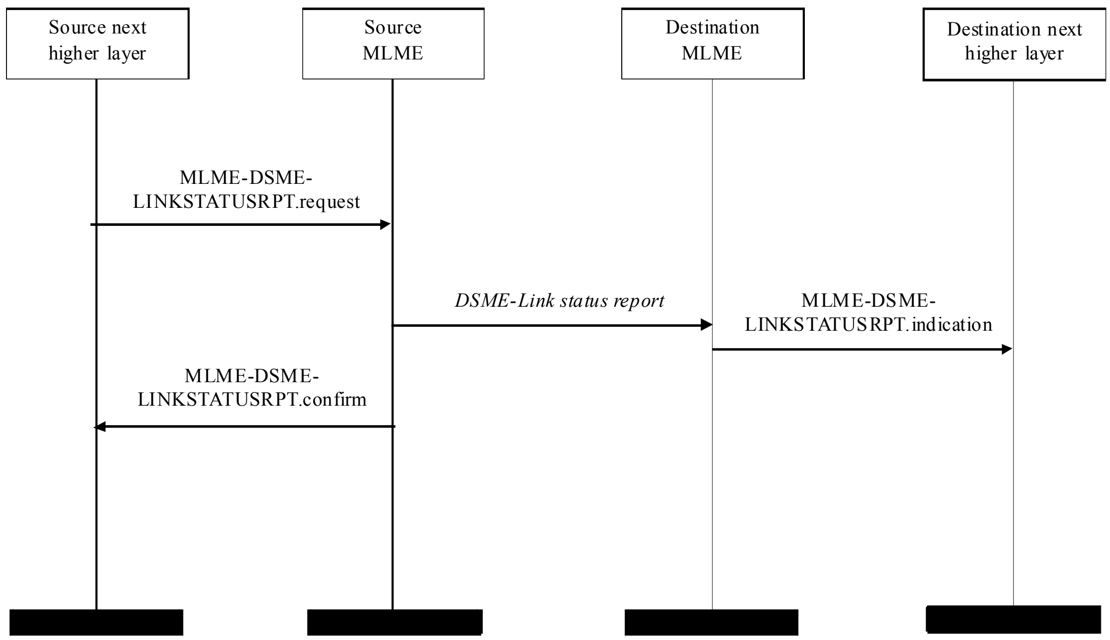

Figure 60e—Message sequence chart for DSME link status report

The message sequence of DSME link status report initiated by destination device is the same as sequence initiated by source device.

6.3 MAC data service

6.3.1 MCPS-DATA.request

Insert additional parameters to the primitive:

…
frameControlOptions,
headerIElist,
payloadIElist,
sendMultipurpose
)

Change elements of Table 46 as follows:

Table 46—MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Enumeration	NO_ADDRESS, SIMPLE_ADDRESS, SHORT_ADDRESS, EXTENDED_ADDRESS	The source addressing mode for this MPDU.
DstAddrMode	Enumeration	NO_ADDRESS, SIMPLE_ADDRESS, SHORT_ADDRESS, EXTENDED_ADDRESS	The destination addressing mode for this MPDU.
…			
frameControlOptions	List of Boolean	PAN_ID_SUPPRESSED, IES_INCLUDED, SEQ_#_SUPPRESSED	Corresponding to the options in the multipurpose frame control or general frame control for frame version 0b10.
headerIElist	List of IE IDs as described in Table 4b	Refer to Table 3a	Determines which Header IEs are sent. IES_INCLUDED shall be set in frameControlOptions if present.
payloadIElist	List of IE IDs as described in 5.2.4.3	Refer to Table 4c	Determines which Payload IEs are sent in the frame. IES_INCLUDED shall be set in frameControlOptions if present.
sendMultipurpose	Boolean	TRUE, FALSE	If TRUE, use the multipurpose frame type, with Frame Control fields as indicated in frameControlOptions. If FALSE, use data frame version 0b10 with applicable Frame Control fields as indicated in frameControlOptions.

Change the last paragraph as follows:

If the TxOptions parameter specifies that an indirect transmission is not required, the MAC sublayer will transmit the MSDU using CSMA-CA either in the CAP for a beacon-enabled PAN or immediately for a nonbeacon-enabled PAN, or at the next timeslot to the destination address if in TSCH mode.

Insert after the last paragraph the following text:

Two additional mutually exclusive options are available to use the MCPS-DATA.request other than to generate a version 0b00 or 0b01 data frame.

If sendMultipurpose is TRUE, then the frame is to be sent using the multipurpose frame type and the frameControlOptions parameter is used to specify the frame control settings and MHR fields present. If PAN_ID_PRESENT enumeration is included, then the frame is to include the PAN ID (normally suppressed), and the Frame Control fields set accordingly. If IES_INCLUDED enumeration is included, then the IEs Present field is set to indicate the presence of IEs and the IE field is populated from the list of IE ID's supplied. If the SEQ_#_SUPPRESSION enumeration is included, the Sequence Number is suppressed from the MHR and Sequence Number Suppression field is set accordingly. Refer to 5.2.2.6 for details on Frame Control fields for the multipurpose frame.

If sendMultipurpose is false, then data frame type with version = 0b10 is used if the MHR indicates the inclusion of IEs and/or the suppression of the sequence number. If IES_INCLUDED enumeration is included, then the IEs Present field is set to indicate the presence of IEs, and the IE field is populated from the list of IE IDs supplied. If the SEQ_#_SUPPRESSION enumeration is included, the Sequence Number is suppressed from the MHR and Sequence Number Suppression field is set accordingly. Refer to 5.2.1.1 for details on general Frame Control fields.

6.3.2 MCPS-DATA.confirm

Insert three additional parameters to this primitive:

…
NumBackoffs,
DSN,
AckPayload
)

Change the fourth paragraph as indicated:

If the generated MAC frame is a data frame with the Frame Version subfield set to 0b00 or 0b01, and both the SrcAddrMode and the DstAddrMode parameters are set to NO_ADDRESS in the MCPS-DATA.request primitive, the status shall be set to INVALID_ADDRESS.

Insert after the last paragraph as follows:

If an Acknowledgment is received without a DSN or Source Address field, the status will be set to ACK_RCVD_NODSN_NOSA.

If an Acknowledgement is received that contains user data encapsulated in payload IEs or included in the Payload field, the received AckPayload will contain the received data.

Change Table 47 as shown:

Table 47—MCPS-DATA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER, ACK_RCVD_NODSN_NOSA	The status of the last MSDU transmission.
NumBackoffs	Integer	0x00–0xff	The number of times the CSMA-CA algorithm was required to backoff as described in 5.1.1.4 while attempting the current transmission. If "Status" is anything other than "SUCCESS," this value is undefined.
AckPayload	Set of octets	—	The set of octets received in the Acknowledgement payload by the MAC sublayer entity including payload IEs if present.

6.3.3 MCPS-DATA.indication

Change elements of Table 48 as shown:

Table 48—MCPS-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Enumeration	NO_ADDRESS, SIMPLE_ADDRESS, SHORT_ADDRESS, EXTENDED_ADDRESS	The source addressing mode for this primitive corresponding to the received MPDU.
SrcPANId	Integer	0x0000–0xffff	The PAN identifier of the entity from which the MSDU was received. Valid only when a source PAN identifier is included in the received frame.
…			

Table 48—MCPS-DATA.indication parameters *(continued)*

Name	Type	Valid range	Description
DstAddrMode	Enumeration	NO_ADDRESS, SIMPLE_ADDRESS, SHORT_ADDRESS, EXTENDED_ADDRESS	The destination addressing mode for this primitive corresponding to the received MPDU.
DstPANId	Integer	0x0000–0xffff	The PAN identifier of the entity to which the MSDU is being transferred. Set to the receiver's PAN ID if the PAN ID is not carried in the received frame.
…			
msdu	Set of octets	—	The set of octets forming the MSDU being indicated by the MAC sublayer entity including payload IEs if present.
…			
DSN	Integer	0x00–0xff	The DSN of the received data frame if one was present.
…			

Change the last paragraph as indicated:

The MCPS-DATA.indication primitive is generated by the MAC sublayer and issued to the next higher layer on receipt of a data frame or a multipurpose frame at the local MAC sublayer entity that passes the appropriate message filtering operations as described in 5.1.6.2. If the primitive is received while the device is in promiscuous mode, the parameters will be set as specified in 5.1.6.5.

6.4 MAC constants and PIB attributes

6.4.2 MAC PIB attributes

Add the following rows to the end of Table 52:

Table 52—MAC PIB attributes

Attribute	Type	Range	Description	Default
macEnhAckWaitDuration	Integer	0x0000–0xffff	The maximum time (in μs) to wait for the PHY header of an enhanced acknowledgment frame to arrive following a transmitted data frame.	0x360 (864 μs)
macImplicitBroadcast	Boolean	TRUE, FALSE	Indicates whether frames without a destination PAN ID or destination address are to treated as though they are addressed to the broadcast PANID (0xffff) and broadcast short address (0xfffff)	FALSE

Add the following row to Table 52 before "macShortAddress":

Attribute	Type	Range	Description	Default
macSimpleAddress	Integer	0x00–0xff	The 8-bit address that the device uses to communicate in the PAN. If the device is the PAN coordinator, this value shall be chosen before a PAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfe indicates that the device has associated but has not been allocated an address. A value of 0xff indicates that the device does not have a simple address.	0xff

Change the default entries in the following rows for Table 52:

Attribute	Type	Range	Description	Default
macMaxBE	Integer	3–8	The maximum value of the backoff exponent, BE, in the CSMA-CA algorithm, as defined in 5.1.1.4.	5 except for LLDN mode = 3
macMaxCSMABackoffs	Integer	0–5	The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure.	4 except for LLDN mode = 0

6.4.3 Calculating PHY dependent MAC PIB values

Insert after the heading of 6.4.3, the following new subclause header:

6.4.3.1 General

Insert the following paragraph as the last text paragraph of 6.4.3.1:

If *macLLenabled* is TRUE, the attribute *macAckWaitDuration* is dependent on the acknowledgment method used and the timings of the superframe of the LLDN.

Insert before Clause 7 the following new subclauses (6.4.3.2–6.4.3.11):

6.4.3.2 General MAC PIB attributes for functional organization

Table 52a provides the General MAC PIB attributes for functional organization.

Table 52a—General MAC PIB attributes for functional organization

Attribute	Type	Range	Description	Default
macTSCHcapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to TSCH	Implementation specific
macLLcapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to LLDNs	Implementation specific
macDSMEcapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to DSME	Implementation specific
macLEcapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to low energy	Implementation specific
macRFIDcapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to RFID	Implementation specific
macHoppingCapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of unslotted channel hopping	Implementation specific
macAMCACapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of functionality specific to AMCA	Implementation specific
macMetricsCapable	Boolean	TRUE or FALSE	If TRUE, the device is capable of providing additional MAC metrics	Implementation specific
macTSCHenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to TSCH	Implementation specific
macLLenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to LLDNs	Implementation specific
macDSMEenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to DSME	Implementation specific
macLEenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to low energy	Implementation specific
macRFIDenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to RFID	Implementation specific
macHoppingEnabled	Boolean	TRUE or FALSE	If TRUE, the device is using unslotted channel hopping	Implementation specific

Table 52a—General MAC PIB attributes for functional organization *(continued)*

Attribute	Type	Range	Description	Default
macAMCAenabled	Boolean	TRUE or FALSE	If TRUE, the device is using functionality specific to AMCA	Implementation specific
macMetricsEnabled	Boolean	TRUE or FALSE	If TRUE, the device is providing additional MAC metrics	Implementation specific

6.4.3.3 TSCH specific MAC PIB attributes

Subclause 6.4.3.1 applies except that the attributes *macMinBE* and *macMaxBE* in Table 52 shall be according to Table 52b and an additional attribute *macDisconnectTime* is required as defined in Table 52b.

Table 52b—TSCH-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macMinBE	Integer	0–*macMaxBE*	The minimum value of the backoff exponent (BE) in the CSMA-CA algorithm or the TSCH-CA algorithm. Refer to 5.1.1.4 for a detailed explanation of the backoff exponent. Refer to 5.1.1.4.3 for use of the backoff exponent in TSCH mode.	3—CSMA-CA 1—TSCH-CA
macMaxBE	Integer	3–8	The maximum value of the BE in the CSMA-CA algorithm or the TSCH- CA algorithm. Refer to 5.1.1.4 for a detailed explanation of the backoff exponent. Refer to 5.1.1.4.3 for use of the backoff exponent in TSCH mode.	5—CSMA-CA 7—TSCH-CA
macDisconnectTime	Integer	0x0000–0xffff	Time (in Timeslots) to send out Disassociate frames before disconnecting.	0xff
macJoinPriority	Integer	0x00–0x3f	The lowest join priority from the TSCH Synchronization IE in an Enhanced beacon, described in 5.2.4.13, when the device joined the network + 1.	1

Table 52b—TSCH-specific MAC PIB attributes *(continued)*

Attribute	Type	Range	Description	Default
macASN	Integer	0x0000000000–0xffffffffff	The Absolute Slot Number, i.e., the number of slots that has elapsed since the start of the network. Used for nonce construction when security is enabled.	0x0000000000
macNoHLBuffers	Boolean	TRUE or FALSE	If the value is TRUE, the higher layer receiving the frame payload cannot buffer it, and the device should acknowledge frames with a NACK as described in 5.2.4.11. If FALSE, the higher layer can accept the frame payload.	FALSE

6.4.3.3.1 TSCH MAC PIB attributes for macSlotframeTable

The attributes contained in the MAC PIB for *macSlotframeTable* are presented in Table 52c. Each slotframe requires a *macSlotframeTable* to be stored.

Table 52c—TSCH-MAC PIB attributes for *macSlotframeTable*

Attribute	Type	Range	Description	Default
macSlotframeHandle	Integer	0x00–0xff	Identifier of the slotframe	—
macSlotframeSize	Integer	0x0000–0xffff	Number of timeslots in the slotframe	—

6.4.3.3.2 TSCH MAC PIB attributes for macLinkTable

The attributes contained in the MAC PIB for *macLinkTable* are presented in Table 52d. Each link requires a *macLinkTable* to be stored.

Table 52d—TSCH-MAC PIB attributes for *macLinkTable*

Attribute	Type	Range	Description	Default
macLinkHandle	Integer	0x0000–0xffff	Identifier of Link	—
macLinkOptions	Bitmap	0x00–0x0f	Flags indicating how the link is used. b0 = Transmit, b1 = Receive, b2 = Shared, b3 = Timekeeping, b4–b7 reserved. Some bit combinations are undefined.	—
macLinkType	Enumeration	NORMAL, ADVERTISING	Type of link. NORMAL = 0. ADVERTISING = 1, and indicates the link may be used to send an enhanced beacon.	—
SlotframeHandle	Integer	0x00–0xff	Identifier of Slotframe to which this link belongs	—

Table 52d—TSCH-MAC PIB attributes for *macLinkTable* (continued)

Attribute	Type	Range	Description	Default
macNodeAddress	IEEE address	16-bit address	Address of the node connected to this link	—
macTimeslot	Integer	0x0000–0xffff	Timeslot for this link	—
macChannelOffset	Integer	0x0000–0xffff	Channel offset for this link	—

6.4.3.3.3 TSCH MAC PIB attributes for macTimeslotTemplate

The attributes contained in the MAC PIB for *macTimeslotTemplate* are presented in Table 52e.

Table 52e—TSCH-MAC PIB attributes for *macTimeslotTemplate*

Attribute	Type	Range	Description	Default
macTimeslotTemplateId	Integer	0x0–0xf	Identifier of Timeslot Template	0
macTsCCAOffset	Integer	0x0000–0xffff	The time between the beginning of timeslot and start of CCA operation, in µs	1800
macTsCCA	Integer	0x0000–0xffff	Duration of CCA, in µs	128
macTsTxOffset	Integer	0x0000–0xffff	The time between the beginning of the timeslot and the start of frame transmission, in µs	2120
macTsRxOffset	Integer	0x0000–0xffff	Beginning of the timeslot to when the receiver shall be listening, in µs	1120
macTsRxAckDelay	Integer	0x0000–0xffff	End of frame to when the transmitter shall listen for Acknowledgment, in µs	800
macTsTxAckDelay	Integer	0x0000–0xffff	End of frame to start of Acknowledgment, in µs	1000
macTsRxWait	Integer	0x0000–0xffff	The time to wait for start of frame, in µs	2200
macTsAckWait	Integer	0x0000–0xffff	The minimum time to wait for start of an Acknowledgment, in µs	400
macTsRxTx	Integer	0x0000–0xffff	Transmit to Receive turnaround, in µs	192
macTsMaxAck	Integer	0x0000–0xffff	Transmission time to send Acknowledgment, in µs	2400
macTsMaxTx	Integer	0x0000–0xffff	Transmission time to send the maximum length frame, in µs	4256
macTsTimeslotLength	Integer	0x0000–0xffff	The total length of the timeslot including any unused time after frame transmission and Acknowledgment, in µs	10000

6.4.3.4 MAC PIB attributes for hopping sequence

The attributes contained in the MAC PIB for Hopping Sequence are presented in Table 52f.

Table 52f—MAC PIB attributes for Hopping Sequence

Attribute	Type	Range	Description	Default
macHoppingSequenceID	Integer	0x00–0x0f	Each hopping sequence has a unique ID.	0
macChannelPage	Integer	0x00–0x1f	Corresponds to the 5 MSBs (b27, ..., b31) of a row in phyChannelsSupported. Note this may not correspond to the current channelPage in use.	—
macNumberOfChannels	Integer	0x0000–0x01ff	Number of channels supported by the PHY on this channelPage.	—
macPhyConfiguration	Integer	0x00000000–0x7fffffff	For channel pages 0 to 6, the 27 LSBs (b0, b1, ..., b26) indicate the status (1 = to be used, 0 = not to be used) for each of the up to 27 valid channels available to the PHY. For pages 7 and 8, the 27 LSBs indicate the configuration of the PHY, and the channel list is contained in the extendedBitmap.	—
macExtendedBitmap	Bitmap	varies	For pages 7 and 8, a bitmap of numberOfChannels bits, where b_k shall indicate the status of channel k for each of the up to numberOfChannels valid channels supported by that channel page and phyConfiguration. Otherwise field is empty.	—
macHoppingSequenceLength	Integer	0x0000–0xffff	The number of channels in the Hopping Sequence. Does not necessarily equal numberOfChannels.	—
macHoppingSequenceList	List of Integers	0x0000–0x01ff for each channel	A *macHoppingSequenceLength*-element list of channels to be hopped over.	—
macCurrentHop	Integer	0x0000–0xffff	Index of the current position in the hopping sequence list.	—
hopDwellTime	Integer	0x0000–0xffff	For unslotted channel hopping modes, this field is the channel dwell time, in units of 10 μs. For other modes, the field is empty.	—

6.4.3.5 LLDN MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required as stated in Table 52g.

Table 52g—LLDN MAC PIB attributes

Attribute	Type	Range	Description	Default
macLLDNnum-TimeSlots	Integer	0 … 254	Number of timeslots within superframe excluding timeslot for beacon frame and management timeslots	20
macLLDNnum-UplinkTS	Integer	0 … *macLLDNnum-TimeSlots*	Number of uplink timeslots within superframe for unidirectional communication (uplink) as defined in 5.1.1.6.4.	20
macLLDNnum-RetransmitTS	Integer	0 … *macLLDNnum-UplinkTS/2*	Number of uplink timeslots reserved for retransmission	0
macLLDNnum-BidirectionalTS	Integer	0 … *macLLDNnum-TimeSlots*	Number of bidirectional timeslots as defined in 5.1.1.6.5 within superframe for bidirectional communication	0
macLLDNmgmtTS	Boolean	TRUE or FALSE	Indicates presence of management timeslots in Online state	FALSE
macLLDNlow-LatencyNWid	Integer	0x00–0xff	The 8-bit identifier of the LLDN on which the device is operating. If this value is 0xff, the device is not associated.	0xff
macLLDNtime-SlotInfo	Imple-men-tation specific		Information related to a timeslot, for instance, MAC addresses mapped to the timeslot. For the LLDN PAN coordinator, there are *macLLDNnumTimeSlots* attributes of *macLLDNtimeSlotInfo*.	Set during Configu-ration state
macLLDN-discoveryMode-Timeout	Integer	0…256	The LLDN coordinator switches from the Discovery state into the Configuration state after it did not receive a Discover Response Frame within the last *macLLDNdiscoveryModeTimeout* seconds.	256
macLLDN-coordinator	Boolean	TRUE or FALSE	Indicates whether the LLDN device is the LLDN PAN coordinator	FALSE

6.4.3.6 DSME specific MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required as described in Table 52h and Table 52i.

Table 52h—DSME-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macChannelIndex	Integer	0–31	Specifies the channel index of the channel's DSME link status reported by the source device.	—
macAvgLQI	Integer	0x00–0xff	A characterization of the link quality between a source device and a destination device on the channel defined by Channel Index, the measurement shall be performed for each received packet during a period of *LinkStatusStatisticPeriod*.	—

Table 52h—DSME-specific MAC PIB attributes *(continued)*

Attribute	Type	Range	Description	Default
macAvgRSSI	Integer	0–255	Average RSSI.	—
macLinkStatusStatisticPeriod	Integer	0x000000–0xffffff	The time interval between two times of link status statistics, which is defined as LinkStatusStatisticPeriod = *aBaseSuperframeDuration* × 2^{MO} symbols. If the parameter equals to 0x000000, link status statistic is not allowed.	Implementation specific
macGACKFlag	Boolean	TRUE or FALSE	This flag indicates if the coordinator is currently using the group acknowledge mechanism for DEMS-GTS frame receptions.	0x00
macCAPReductionFlag	Boolean	TRUE or FALSE	Indication of whether the CAP reduction is enabled. A value of TRUE indicates that the CAP reduction is enabled.	FALSE
macChannelDiversityMode	Integer	0–1	Indicates the method of channel diversity: 0x00 = Channel Adaptation 0x01 = Channel Hopping This value is not valid for a nonbeacon-enabled PAN.	0x00
macMultisuperframeOrder	Integer	0–15	The length of a multi-superframe, which is a cycle of the repeated superframes.	15
macConnecDev	Boolean	TRUE or FALSE	Indication of whether the device is a Connection Device or not. If this attribute is TRUE, the device is a Connection Device. This attribute shall be set to FALSE if the device is not a Connection Device.	FALSE
macDSMESAB	Bitmap	Refer to Figure 59q and Figure 59r	The slot allocation bitmap table of the DSME-GTS schedule.	0
macDSMEACT	Bitmap	Refer to Table 1a	The allocation counter table of the DSME-GTS allocated to the device.	0
macSDindex	Integer	0x0000–0xffff	Specifies the allocating SD index number for beacon frame.	0x0000
macSDBitmap	Bitmap	Refer to 5.2.4.9.3	Indicates the beacon frame allocation information of neighbor nodes. This field is expressed in bitmap format that orderly represents the schedule of beacons, with corresponding bit shall be set to one if a beacon is allocated in that SD.	Dependent on currently beacon allocation
macChannelOffset	Integer	0x0000–0xffff	ChannelOffset is the offset value of Hopping Sequence.	0
macDeferredBeaconUsed	Boolean	TRUE or FALSE	Indication of whether the device uses CCA before transmitting beacon frame. A value of TRUE indicates that the device uses CCA before transmitting beacon frame.	FALSE

Table 52h—DSME-specific MAC PIB attributes *(continued)*

Attribute	Type	Range	Description	Default
macSyncParentExtended-Address	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the coordinator through which the device is synchronized.	—
macSyncParentShortAddress	Integer	0x0000–0xffff	The 16-bit short address assigned to the coordinator through which the device is synchronized. A value of 0xfffe indicates that the coordinator is only using its 64-bit extended address. A value of 0xffff indicates that this value is unknown.	0xffff
macSyncParentSDIndex	Integer	0x0000–0xffff	Indication of SD index the synchronized parent used.	0
macChannelStatus	List of Link-Status entries as described in Table 44y	Refer to Table 44y	Channel status for each used channel.	—
macBeaconSlotLength	Integer	0x0000–0xffff	The number of symbols forming a beacon slot.	60
macDSMEGTSExpiration-Time	Integer	0x00–0xff	The number of idle incidents before expiring a DSME-GTS.	7
macChannelOffsetBitmap-Length	Integer	0x00–0xff	Specifies the length of ChannelOffsetBit-map in octets.	—
macChannelOffsetBitmap	Set of octets	—	Bit value of ChannelOffsetBitmap sequence represents whether the corre-sponding channel offset is used. If the corresponding channel offset is used, the bit value shall be set to one. Otherwise, it shall be set to zero. For instance, if the 1st, 2nd, 4th channels offset are used with ChannelOffsetBitmapLength of 16, ChannelOffsetBitmap shall be 0b0110100000000000. The number of bits set to one in the ChannelOffsetBitmap is the number of available channels in current channel page.	—
macPANCoordinatorBSN	Integer	0x00–0xff	The sequence number added to the transmitted beacon frame of a PAN coordinator.	—
macNeighborInformation-Table	List of Neighbor Informa-tion entries as described in Table 52i	—	A table of the neighbor device's informa-tion entries.	Null

Table 52i—Elements of Neighbor Information

Attribute	Type	Range	Description
ShortAddress	Integer	0x0000–0xffff	The 16-bit address of the neighbor device.
ExtendedAddress	IEEE address	An extended 64-bit IEEE address	The 64-bit (IEEE) address of the neighbor device.
SDIndex	Integer	0x0000–0xffff	The allocating SD index number for beacon frame.
ChannelOffset	Integer	0x0000–0xffff	The offset value of ChannelHoppingSequence.
TrackBeacon	Boolean	TRUE or FALSE	TRUE if the MLME is to track all future beacons of the neighbor device. FALSE if the MLME is not to track beacon of the neighbor device.
BeaconLostCount	Integer	0x00–0xff	The number of consecutive lost beacons.

6.4.3.7 LE specific MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required, as described in Table 52j.

Table 52j—LE-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macCSLPeriod	Integer	0 … 65535	CSL sampled listening period in unit of 10 symbols. Zero means always listening, i.e., CSL off.	0
macCSLMax-Period	Integer	0 … 65535	Maximum CSL sampled listening period in unit of 10 symbols in the entire PAN. This determines the length of the wake-up sequence when communicating to a device whose CSL listen period is unknown. NHL may set this attribute to zero to stop sending wake-up sequences with proper coordination with neighboring devices.	*macCSLPeriod*
macCSLChannel Mask	Bitmap	0x00000000 … 0xffffffff	32-bit bitmap relative to phyCurrentPage of channels. It represents the list of channels CSL operates on. Zero means CSL operates on phyCurrentChannel of phyCurrentPage.	0x00000000
macCSLFrame-PendingWaitT	Integer	*macMinLIFSPeriod* + max number of symbols per PPDU … 65535	Number of symbols to keep the receiver on after receiving a payload frame with Frame Control field frame pending bit set to one.	—

Table 52j—LE-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macRITPeriod	Integer	0x000000–0xffffff	The interval (in unit periods) for periodic transmission of RIT data request command in RIT mode. The unit period is *aBaseSuperframeDuration*. Zero means RIT is off.	0
macRITDataWaitDuration	Integer	0x00–0xff	The maximum time (in unit period) to wait for Data frame after transmission of RIT data request command frame in RIT mode. The unit period is *aBaseSuperframeDuration*.	0
macRITTxWait-Duration	Integer	*macRITPeriod*–0xffffff	The maximum time (in unit periods) that a transaction is stored by a device in RIT mode. The unit period is *aBaseSurperframeDuration*.	0
macLowEnergy-Superframe-Supported	Boolean	TRUE or FALSE	Indication of whether the low-energy superframe is operational or not. If this attribute is TRUE, the coordinator shall not transmit beacon frames regardless of BO value. This attribute shall be set to FALSE if the device is aware of the existence of allocated GTS.	Implementation specific
macLowEnergy-SuperframeSync-Interval	Integer	0x0000–0xffff	Indication of beacon transmission interval when *macLowEnergySuperframeSupported* is TRUE. The beacon transmission interval shall be indicated by *macLowEnergySuperframeSyncInterval* times the beacon interval in the case this attribute is not zero. The beacon shall be transmitted only when requested in the case this attribute is zero.	Implementation specific

6.4.3.8 AMCA specific MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required, as described in Table 52k.

Table 52k—AMCA-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macDesignated-ListeningChannel	Integer	0x00–0xff	The channel number of the designated listening channel of the device.	0

6.4.3.9 MAC-performance-metrics-specific MAC PIB attributes

Additional attributes are required for metrics to enable the assessment of network performance and behavior by higher layers, and aid in MSDU segmentation decisions. If macMetricsEnabled is TRUE, the MAC shall collect the metrics listed in Table 52l. The MAC PIB attributes in 6.4.3.1 apply.

MAC PIB attribute *macCounterOctets* defines the size of the counters providing attributes 0x91 through 0x98. The values of 1 though 4 correspond to counters of 8, 16, 24, or 32 bits. Attribute *macCounterOctets* is read-only and set by the implementer depending on the PHY characteristics and other considerations.

The counters implementing MAC PIB attributes 0x91 through 0x98 shall wrap to 0 when incremented beyond their maximum value ($2^{8 \times macCounterOctets}-1$). Attributes 0x91 through 0x98 are read/write and may be reset by higher layers by writing a 0 value.

The attributes *macRetryCount*, *macMultipleRetryCount*, *macTXFailCount*, and *macTXSuccessCount* relate to data frame transmission. Each MSDU transferred into the MAC layer through the MCPS-DATA.request primitive shall increment exactly one of these four attribute counters depending on the final disposition of the frame as described in Table 86.

The attributes *macFCSErrorCount*, *macSecurityFailure*, *macDuplicateFrameCount*, and *macRXSuccessCount* relate to data frame reception. Each MSDU transferred out of the MAC layer through the MCPS-DATA.indication primitive shall increment at least one of these four attribute counters based on the status of the frame as described in Table 86. To compute *macDuplicateFrameCount*, storing the most recently received frame's address and DSN is necessary to detect duplicate received frames. The number of unique device addresses to be stored is up to the implementer

To create a list of PHY related metrics, it is recommended that higher layers store relevant parameters of MCPS-DATA.indication such as mpduLinkQuality and DataRate for each unique Source Address (SrcAdr). The higher layers may also query PLME ED.request to establish an idle channel noise measurement.

Table 52l—Metrics-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macCounterOctets	Integer	1–4	Defines the counter size in octets for attributes 0x91 through 0x99.	Implementation dependent
macRetryCount	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of transmitted frames that required exactly one retry before acknowledgment.	0
macMultipleRetryCount	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of transmitted frames that required more than one retry before acknowledgment.	0
macTXFailCount	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of transmitted frames that did not result in an acknowledgment after *macMaxFrameRetries*.	0
macTXSuccessCount	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of transmitted frames that were acknowledged within *macAckWaitDuration* after the initial data frame transmission.	0
macFCSErrorCount	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of received frames that were discarded due to an incorrect FCS.	0
macSecurityFailure	Integer	$0–2^{8 \times macCounterOctets}-1$	The number of received data frames that were returned from the procedure described in 7.2.3 with any status other than "SUCCESS."	0

Table 52l—Metrics-specific MAC PIB attributes *(continued)*

Attribute	Type	Range	Description	Default
macDuplicateFrameCount	Integer	$0–2^{8 \times macCounterOctets}–1$	The number of received data frames that contained the same sequence number as a frame previously received (accounting for wrap-around of *macDSN*).	0
macRXSuccessCount	Integer	$0–2^{8 \times macCounterOctets}–1$	The number of received data frames that were received correctly.	0
macNACKcount	Integer	$0–2^{8 \times macCounterOctets}–1$	The number of transmitted frames that were acknowledged with a ACK/NACK timing correction IE indicating a NACK. Frames acknowledged in this manner also should increment *macRetryCount* or *macMultipleRetryCount.*	0

6.4.3.10 EBR specific MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required, as described in Table 52m.

Table 52m—EBR-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
EBRPermitJoining	Boolean	TRUE, FALSE	When TRUE, the Permit Joining request will be included in the EBR.	—
macEBRFilters	List of Booleans		Contains which EBR filter field bits should be set.	—
macEBRLinkQuality	Integer	0x00–0xff	Link quality level to be transmitted in the EBR.	—
macEBRPercentFilter	Integer	0–100	Percent filter threshold value to be transmitted in the EBR.	—
macEBRattributeList	List of PIB IDs	0x00–0xff	Contains zero to four attribute IDs. Each ID shall identify a Boolean PIB attribute, refer to 5.3.12.	—
macBeaconAutoRespond	Boolean	TRUE, FALSE	When TRUE, device responds to beacon requests and enhanced beacon requests automatically. When FALSE, device passes beacon/enhanced beacon payload up to higher layer using MLME-BEACONREQUEST.indication.	TRUE

6.4.3.11 EB specific MAC PIB attributes

Subclause 6.4.3.1 applies and additional attributes are required, as described in Table 52n.

Table 52n—EB-specific MAC PIB attributes

Attribute	Type	Range	Description	Default
macUseEnhancedBeacon	Boolean	TRUE, FALSE	When TRUE, in a beacon enabled PAN the device should use Enhanced Beacons rather than standard beacons.	—
macEBIEList	List of IEs	Refer to 5.2.4	A list of IEs to include in the enhanced beacon.	—
macEBFilteringEnabled	Boolean	TRUE, FALSE	Indicates if devices should perform filtering in response to EBR.	—
macEBSN	Integer	0x00–0xff	Beacon Sequence Number used for Enhanced Beacon Frames (separate from BSN).	—
macEBAutoSA	Enumeration	NONE, SHORT, FULL	Indicates if beacons generated by the MAC in response to EB include Source Address field.	FULL
macEAckIElist	List	—	List of additional IEs to be included in enhanced ACKs generated by the device.	Empty

7. Security

7.3 Security operations

7.3.2 CCM* Nonce

Insert at the end of 7.3.2 the following text:

When the *macFrameCounterMode* = 0x05, or the incoming frame counter size is 5 octets as described in 7.4.1.4, (such as when operating in TSCH mode, including when in active or passive scan to join a TSCH PAN), the nonce shall be formatted as shown in Figure 61a.

Octets: 8	5
Source Address	Frame Counter

Figure 61a—CCM* Nonce in TSCH mode

The Source Address shall be set to match the source address of the device originating the frame, i.e., the extended address *aExtendedAddress* if an 8-octet source is used, or the *macShortAddress* if a 2-octet source address is used. If no source address is used in the frame, then the *macShortAddress* of the device originating the frame is used.

In TSCH mode, the Frame Counter shall be set to the global frame counter value, i.e., the Absolute Slot Number as described in 5.1.1.5.2, regardless of whether the frame counter is carried in the auxiliary security header.

7.4 Auxiliary security header

Replace Figure 62 with the following new Figure 62:

Octets: 1	0/4/5	0/1/5/9
Security Control	Frame Counter	Key Identifier

Figure 62—Format of the auxiliary security header

7.4.1 Security Control field

Replace Figure 63 with the following new Figure 63:

Bits: 0–2	3–4	5	6	7
Security Level	Key Identifier Mode	Frame Counter suppression	Frame counter size	Reserved

Figure 63—Security Control field format

7.4.1.2 Key Identifier Mode field

Insert before 7.4.2 the following new subclauses (7.4.1.3–7.4.1.4):

7.4.1.3 Frame Counter Suppression field

The Frame Counter Suppression field is 1 bit in length. When set to zero, the frame counter is carried in the frame. When set to one, the frame counter is not carried in the frame, and the frame counter used to construct the nonce defined in 7.3.2 is either an incrementing shared global frame counter such as ASN, or in the case of an enhanced acknowledgment, the frame counter of the frame being acknowledged.

7.4.1.4 Frame Counter Size field

The Frame Counter Size field is 1 bit in length. When set to zero, the frame counter is 4 octets, and the nonce is constructed as shown in Figure 61. When set to one, the frame counter is 5 octets, and the nonce is constructed as shown in Figure 61a.

7.4.2 Frame Counter field

Insert before 7.4.3 the following new subclause (7.4.2.1):

7.4.2.1 Frame counter

The frame counter is 4-octet or 5-octet (as determined by *macFrameCounterMode*) long.

The frame counter may be included in each secured frame and is one of the elements required for the unsecuring operation at the recipient(s). The frame counter is incremented each time an outgoing frame is secured, as described in the outgoing frame security procedure 7.2.1. When the frame counter reaches its maximum value of 0xffffffff or 0xffffffffff for a 5-octet counter, the associated keying material can no longer be used, thus requiring all keys associated with the device to be updated. This provides a mechanism for ensuring that the keying material for every frame is unique and, thereby, provides for sequential freshness.

7.5 Security-related MAC-PIB attributes

Insert the following row at the end of Table 60:

Table 60—Security-related MAC PIB attributes

Attribute	Type	Range	Description	Default
macFrameCounterMode	Integer	0x04–0x05	Size of the frame counter: 0x04 = 4 octets 0x05 = 5 octets	0x04

Change "macFrameCounter" in Table 60 as follows:

Attribute	Type	Range	Description	Default
macFrameCounter	Integer	For *macFrameCounterMode* = 0x04 or *macFrameCounterMode* not implemented, 0x00000000–0xffffffff For *macFrameCounterMode* = 0x05, 0x0000000000–0xffffffffff	The outgoing frame counter for this device.	0x00000000

10. O-QPSK PHY

10.1 PPDU format

10.1.3 Frame Length field

Change 10.1.3 as follows and delete Table 72:

The Frame Length field specifies the total number of octets contained in the PSDU (i.e., PHY payload). It is a value between 0 and *aMaxPHYPacketSize*, as described in 9.2. Table 72 summarizes the type of payload versus the frame length value.

Annex A

(informative)

Bibliography

Insert the following references in alphanumeric order:

IEC 62591:2010, Industrial communication networks—Wireless communication network and communication profiles—*Wireless*HART®.[2]

IEEE EUI-64 Guidlines, Guidelines for 64-bit Global Identifier (EUI-64™) Registration Authority, March 1997.[3]

IEEE Std 802.11™-2007, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

IEEE Std 802.15.1™-2005, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs).

IEEE Std 802.15.3™-2003, IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY).

ISA 100.11a-2011, Wireless systems for industrial automation: Process control and related applications.

[2]*Wireless*HART is a registered trademark of the HART Communication Foundation.
[3]Available at http://standards.ieee.org/develop/regauth/tut/eui64.pdf.

Annex D

(informative)

Protocol implementation conformance statement (PICS) proforma[4]

D.7 PICS proforma tables

D.7.3 Major capabilities for the MAC sublayer

D.7.3.1 MAC sublayer functions

Insert the following rows in Table D.5:

Table D.5—MAC sublayer functions

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF15	TSCH Capability	Table 8a	O			
MLF15.1	TSCH MAC Management Services	6.2.19	MLF15:M			
MLF15.2	TSCH commands	6.2.19.1, 6.2.19.3, 6.2.19.5, 6.2.19.7	MLF15:M			
MLF15.3	TSCH Channel Access	5.1.1.4.2, 5.1.1.4.3, 5.1.1.5	MLF15:M			
MLF15.4	TSCH PAN formation	5.1.2.6	MLF15:M			
MLF15.5	Synchronization in TSCH PAN	5.1.4.2a	MLF15:M			
MLF15.6	TSCH Retransmissions	5.1.6.4.4	MLF15:M			
MLF15.7	TSCH IEs	5.2.4.11, 5.2.4.13, 5.2.4.14, 5.2.4.15, 5.2.4.16	MLF15:M			
MLF15.8	MTSCH Frame counter	7.4.2.1	MLF15:M			
...						
MLF16	LL Capability	Table 8b	O			

[4]Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

Table D.5—MAC sublayer functions *(continued)*

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF16.1	LL-MAC Management Services	6.2, 6.2.20	MLF16:M			
MLF16.2	LL commands	5.3.10	MLF16:M			
MLF16.2.1	LL: LL-Discover Response command, LL-Configuration Status command, LL-Configuration Request command, Clear To Send (CTS) Shared Group command, Request To Send (RTS) command, Clear To Send command	5.3.10.1, 5.3.10.2, 5.3.10.3, 5.3.10.4, 5.3.10.5, 5.3.10.6	FD1:M FD2:O FD3:O FD4:O FD5:O FD6:O			
MLF16.3	LL Channel Access	5.1.1.4.4	MLF16:M			
MLF16.4	LL Superframe structure	5.1.1.6	MLF16:M			
MLF16.5	LL Transmission Modes	5.1.7	MLF16:M			
...						
MLF17	DSME Capability	6.2, Table 8c	O			
MLF17.1	DSME MAC management service	6.2, 6.2.21	MLF17:M			
MLF17.1.1	MLME DSME START primitive	6.2.21.2	FD1:M FD2:O			
MLF17.2	DSME command	5.3.11	MLF17:M			
MLF17.2.1	DSME Link status report command	5.3.11.11	FD1:M FD2:O			
MLF17.3	DSME Multi superframe structure	5.1.10.1	MLF17:M			
...						
MLF18	EBR (Enhanced Beacon Request) capability	5.3.12	O			
MLF18.1	EBR Commands	5.3.7	MLF18:O			
MLF18.1.1	EBR Enhanced Beacon request command	5.3.7.2	FD1:M FD2:O			
...						
MLF19	LE Capability	5.1.1.7, 5.1.11	O			

Table D.5—MAC sublayer functions *(continued)*

Item number	Item description	Reference	Status	Support		
				N/A	Yes	No
MLF19.1	LE specific MAC sublayer service specification	6.4.3.7	MLF19:M			
MLF19.2	Coordinated Sampled Listening (CSL) capability	5.1.11.1	MLF19:O.1			
MLF19.3	Receiver Initiated Transmission (RIT) capability	5.1.11.2	MLF19:O.1			
MLF19.4	LE superframe	5.1.1.7.1 5.1.1.7.2, 5.1.1.7.3	MLF19:O.1			
MLF19.5	LE-multipurpose Wake-up frame	5.2.2.8	MLF19.2:M			
MLF19.6	LE, CSL Information Element	5.2.4.7	MLF19.2:M			
MLF19.7	LE RIT Information Element	5.2.4.8	MLF19.3:O			
MLF19.8	LE-commands	5.3.12	MLF19.3:M			
...						
MLF20	MAC Metrics PIB Attributes	6.4.3.9	O			
...						
MLF21	FastA commands	5.1.3.3	O			
MLF23	Channel Hopping	Table 52f	O			
MLF23.1	Hopping IEs	5.2.4.16, 5.2.4.17	MLF18:M			

Insert after Annex H the following new Annex I:

Annex I

(informative)

MAC behaviors for industrial and commercial application domains

I.1 General

Industrial applications (and some commercial applications) have critical requirements such as low latency, robustness in the harsh industrial RF environment, and determinism that are not adequately addressed by IEEE Std 802.15.4-2011. Additionally, many of these application domains have unique requirements that conflict with the requirements from other application domains. To allow IEEE 802.15.4 devices to support a wide range of industrial and commercial applications, a wide range of MAC behaviors was found to be necessary.

These MAC behaviors facilitate industrial applications [such as addressed by IEC 62591, ISA100.11a, and wireless network for industrial automation—process automation (WIA-PA)], in addition to those enhancements defined by the Chinese wireless personal area network (CWPAN) standard that also were not included in IEEE Std 802.15.4-2011. The CWPAN standard has identified MAC behaviors to improve network reliability and increase network throughput to support higher duty-cycle data communication applications.

Specifically, there are two categories of MAC enhancements, as follows:

— Behaviors to support specific application domains such as process automation, factory automation
— General functional improvements not specifically tied to application domains

The convention used in this document is to group those MAC behavior modes according to the specific application domain requirements they address under the following headings:

— Timetimeslotslotted channel hopping (TSCH), for application domains such as process automation, described in I.2.
— Low latency deterministic networks (LLDN), for application domains such as factory automation, described in I.3.
— Deterministic and synchronous multi-channel extension (DSME), for general industrial and commercial application domains cited by CWPAN includes Channel diversity to increase network robustness, described in I.4.
— Radio frequency identification blink (RFID), e.g., for application domains such as item and people identification, location, and tracking described in I.5.
— Asynchronous multi-channel adaptation (AMCA), for large infrastructure application domains described in I.6.

The additional MAC enhancements not specifically tied to a particular application domain mode are as follows:

- Low-energy (LE) protocol, to allow very low duty cycle devices to send ad hoc data using minimal amounts of energy, described in I.7.
- Information elements (IE) to provide extensible MAC data transfers, described in I.8.
- Enhanced beacons (EB) and enhanced beacon requests (EBR), to allow coordinator devices to send beacons with specifically requested data, described in I.9.
- The MAC multipurpose frame, which provides the scalability and extensibility to allow this standard to address new application needs with minimal MAC changes, as described in I.10.
- MAC performance metrics to provide upper layers with critical information on the quality of the communication links, described in I.11.
- FastA to reduce the time required to associate, described in I.12.

The subsequent subclauses of this annex provide tutorial material for a better understanding of the targeted application domain modes and additional enhancements specified in Clause 5 and Clause 6.

I.2 Timestimeslotlotted Channel Hopping (TSCH)

I.2.1 Typical application domains addressed by the TSCH mode

Typical segments of the application domain of TSCH are process automation facilities for the following:

- Oil and gas industry
- Food and beverage products
- Chemical products
- Pharmaceutical products
- Water/waste water treatments
- Green Energy production
- Climate control

I.2.2 TSCH mode overview

The TSCH mode uses time synchronized communication and channel hopping to provide network robustness through spectral and temporal redundancy. Timetimeslotslotted communication links increase potential throughput by minimizing unwanted collisions that can lead to catastrophic failure and provides deterministic yet flexible bandwidth for a wide variety of applications. Channel hopping extends the effective range of communications by mitigating the effects of multipath fading and interference. TSCH is also topology independent; it can be used to form any topology from a star to a full mesh. TSCH MAC primitives can be used with higher layer networking protocols to form the basis of reliable, scalable, and flexible networks.

I.2.3 MAC behaviors unique to TSCH

The TSCH mode introduces no unique commands specific for this mode; however, IEs are used extensively. TSCH uses EBs containing the TSCH Synchronization payload IE, TSCH-Slotframe and Link payload IE, TSCH Timeslot payload IE, and Channel Hopping payload IE to advertise the presence of a TSCH PAN and allow new devices to synchronize to it. Devices maintain synchronization by exchanging acknowledged frames within defined Timeslot windows and providing timing corrections in the acknowledgment via the ACK/NACK Time Correction IE.

Communication resources are broken up into timeslots that are organized into repeating slotframes. The TSCH-Slotframe structure behavior is cited in 5.1.1.5, synchronization is stated in 5.1.1.4, the IEs are described in 5.2.4.11, 5.2.4.13, 5.2.4.14, 5.2.4.15, and the primitives are declared in 6.2.19.

I.3 Low Latency Deterministic Networks

I.3.1 Typical application domains and requirements addressed by LLDNs

Typical application domains addressed by the LLDN mode are as follows:

- Factory automation (such as for automotive manufacturing)
- Robots
- Overhead cranes
- Portable machine tools
- Milling machines, computer-operated lathes
- Automated dispensers
- Cargo
- Airport logistics
- Automated packaging
- Conveyors

In addition to the already stated industrial requirements, these types of applications have the following additional major requirements:

- Very low latency
  - Transmission of sensor data in 10 ms
  - Low round-trip time
- Many sensors per LLDN PAN coordinator (there may be more than 100 sensors per LLDN PAN coordinator)

I.3.2 LLDN application domain overview

Factory automation is comprised of a large number of LLDN devices observing and controlling the production. As an example, LLDN devices are located on robots, cranes, and portable tools in the automotive industry; collect data on machine tools, such as milling machines and lathes; and control revolving robots. Further application areas control of conveyor belts in cargo and logistics scenarios and special engineering machines. Depending on the specific needs of different factory automation branches, many more examples could be named.

Common to these applications in the context of factory automation is the requirement of low latency and high cyclic determinism. The performance should allow for reading sensor data from 20 LLDN devices within 10 ms.

Cabling industrial sensors is very time-consuming and expensive. Furthermore, cables are a frequent source of failures due to the harsh environment in a factory and may cause additional costs by production outage. Wireless access to LLDN devices eliminates the cabling issue and also provides advantages in case of mobility and retrofit situations.

Wireless technologies that could be applied to the factory automation scenario include IEEE 802.11™ (WLAN), IEEE 802.15.1™ (Bluetooth®), and IEEE 802.15.4. IEEE 802.15.4 is designed for sensor applications and offers the lowest energy consumption as well as the required communication range and capacity.[5] Moreover, four IEEE 802.15.4 channels can be utilized in good coexistence with three non-overlapping WLAN channels. Bluetooth offers good real-time capabilities, but often interferes with existing WLAN installations.

IEEE Std 802.15.4 is a worldwide and successfully applied standard for wireless and low power transmission of sensor data. Different protocols reside above IEEE 802.15.4 (*Wireless*HART® according to IEC 62591, ISA100.11a, or ZigBee®) in the context of process automation and are already in the process of standardization.[6]

Those protocols aim at different requirements, but employ the same physical layer hardware as the proposed solution for factory automation, which indicates potential hardware synergies and cost savings. Thus, a solution for factory automation based on IEEE Std 802.15.4 is appropriate.

It can be shown that CSMA-CA operation cannot provide a bounded media access time duration while the IEEE 802.15.4 beacon-enabled PAN's seven guaranteed timeslottimeslots in an interval of 15 ms does not fulfill the factory automation requirements. Therefore, a modification of the IEEE 802.15.4 MAC for the industrial factory automation application domain, i.e., defining a fine granular deterministic TDMA access, is required.

I.3.3 LLDN assumptions

The proposed system should be operated in a controlled configuration to achieve the required performance. Thus, it is assumed that the system is operated in a controlled RF environment with frequency planning. The TDMA channels are allocated in a way that eliminates interference and coexistence issues.

I.3.4 LLDN mode characteristics

I.3.4.1 General

Allocating a dedicated timeslot for each LLDN device provides a deterministic system. The IEEE 802.15.4 DSSS coding together with the exclusive channel access for each LLDN device ensures high reliability of the system. Keeping the packets and timeslots short lead to superframes with durations as short as 10 ms. The number of slots in a superframe determines the number of LLDN devices that can access each channel. This solution can be extended by operating the LLDN PAN coordinator with multiple transceivers on different channels to support a high number of LLDN devices.

Due to the stringent requirements of these low-latency applications, the LLDN mode uses the star topology with a minimal superframe structure. Short MAC frames with a 1-octet MAC header (shortened frame control) are deployed to accelerate frame processing and to reduce transmission time.

---

[5]The Bluetooth trademark is owned by Bluetooth SIG, Inc. USA. This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of this product. Equivalent products may be used if they can be shown to lead to the same results.

[6]*Wireless*HART is a registered trademark of the HART Communication Foundation. ZigBee is a registered trademark of the ZigBee Alliance. This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of this product. Equivalent products may be used if they can be shown to lead to the same results.

I.3.4.2 Time Division Multiple Access (TDMA)

The medium in the LLDN mode is accessed by a TDMA scheme that is defined by a superframe of fixed length. The superframe is synchronized with a beacon transmitted periodically from the LLDN PAN coordinator. Access within the superframe is divided into timeslots. The superframe can be configured to provide the full range of operation from a completely deterministic access to a shared access. To provide deterministic access each device is assigned to a particular timeslot of fixed length. Shared group timeslots allow multiple access for a group of nodes within a duration enclosing an arbitrary number (up to the whole superframe) of dedicated timeslots.

I.3.4.3 Addressing

The LLDN mode supports two addressing schemes. The first addressing scheme is based on the timeslot assigned to a device for communication, i.e., the timeslot corresponds exactly to a single device, refer to 5.2.1.1.6. The second scheme supports the short address format. Both addressing schemes can be used within a single LLDN.

I.3.4.4 Network topology

LLDN devices are connected to a single LLDN PAN coordinator in a star topology shown in Figure I.1. As an example, the LLDN devices attached to sensors would send the sensor data uplink to the LLDN PAN coordinator while LLDN devices attached to actuators would be configured to exchange data uplink and downlink with the LLDN PAN coordinator.

The selection configuration of nodes, channels, and timeslots for communication can be planned in a higher layer. The LLDN devices are configured over the LLDN PAN coordinator possibly based on planning information received from the higher layer.

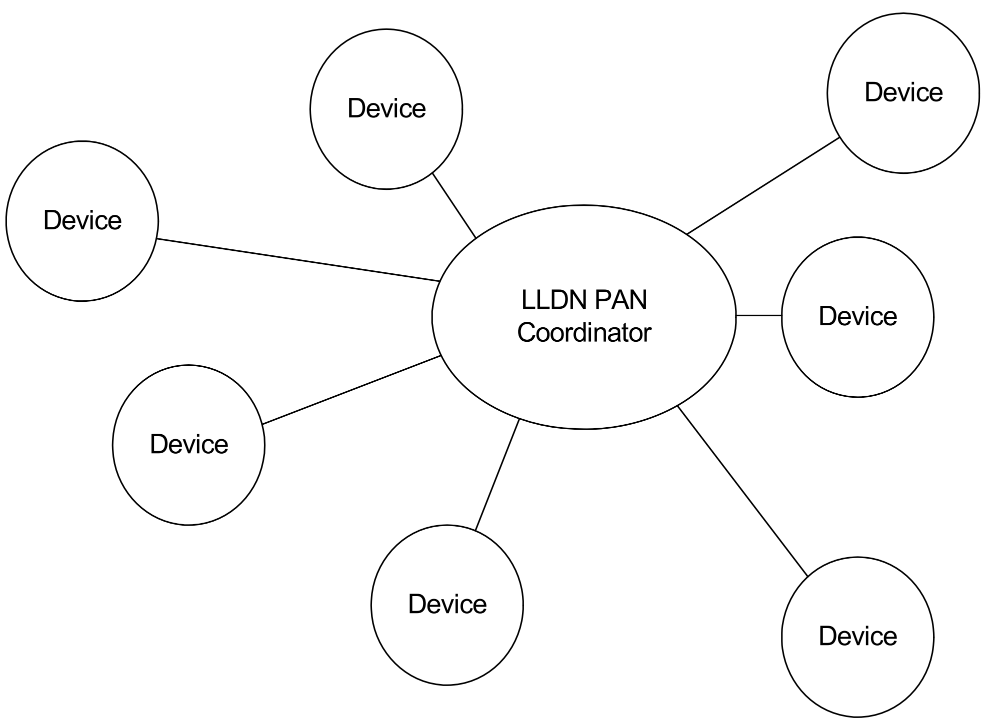

Figure I.1—Star topology LL MAC

I.3.5 MAC behaviors unique to the LLDN mode

The following MAC commands are specific for the LLDN mode: LL-Discover response, LL-Configuration request, LL-Configuration status, LL-CTS shared group, LL-Request To Send (RTS), LL-Clear to send (CTS).

The LLDN mode behavior is cited in 5.1.7, the LLDN commands are stated in 5.3.10.1 through 5.3.11.6, and the primitives are declared in 6.2.20.

I.4 Deterministic and synchronous multi channel extension (DSME)

I.4.1 Typical application domains and requirements

The DSME mode is designed for following types of application domains:

— Industrial applications: process automation, factory automation, smart metering
— Commercial applications: home automation, smart building, entertainment
— Healthcare applications: patient monitoring, telemedicine

These types of application have the following major requirements:

— Deterministic latency
— Flexibility
— High reliability
— Efficiency
— Scalability
— Robustness

I.4.1.1 Features

To satisfy the previous requirements, the DSME mode provides the following features:

— Multi-channel, multi-superframe, mesh extension to GTS for deterministic latency, flexibility, and scalability
— Group acknowledgment option for high reliability and efficiency
— Distributed beacon scheduling and distributed slot allocation for robustness and scalability
— Two channel diversity modes (channel adaptation and channel hopping) as described I.4.4 in for robustness and high reliability even in dynamic channel conditions

It is noted that the GTS described in IEEE Std 802.15.4-2011 can support applications requiring deterministic delay. However, this advantage is limited. First, IEEE Std 802.15.4-2011 supports only up to seven guaranteed timeslots (GTSs), which is not enough to support large scale network applications. Second, the coverage is limited because GTSs are supported only within one hop from the PAN coordinator. Finally, GTSs are restricted to use a single channel.

The DSME mode eliminates the noted GTS limitations to provide time bounded and reliable services for a variety of industrial and commercial applications. For capacity enhancement, the DSME mode enhances IEEE Std 802.15.4-2011 in two important directions: extension of number of GTS timeslots and the number of frequency channels used. To accommodate these enhancements, the DSME mode adopts a versatile frame structure to overcome the seven-slot limitation and single channel operation. In order to save energy, the new frame structure provides CAP reduction. Importantly, using DSME, GTS service can be extended to cover multihop mesh networks with deterministic latency. For reliability, the DSME mode supports two channel diversity methods described in I.4.4 so that designers of mesh networks can choose either channel diversity method to meet the network objectives. The DSME mode is scalable and does not suffer from a single point of failure because beacon scheduling and slot allocation are performed in a distributed manner. In addition, the DSME mode supports a group Acknowledgement option to provide a retransmission opportunity within the same superframe for a data frame that failed in its DSME-GTS transmission. The group Acknowledgement also improves efficiency because the Acknowledgement for multiple data frames is aggregated in a single ACK frame.

I.4.2 Recommended DSME parameter settings for various application types

Recommended DSME mode parameter settings for various application types are presented in Table I.1. For applications such as factory automation require low delay; setting small *macSuperframeOrder* and *macMultiSuperframeOrder* values can reduce the delay. Applications where reliability is critical, such as patient monitoring, require very low probability of data loss. Group ACK can improve the effectiveness of retransmissions. The loss due to time synchronization error can be reduced using Deferred Beacon and Retrieve Synchronization. Applications that have low duty cycles that require low-energy usage such as infrastructure monitoring can use the DSME setting where a device is allowed to sleep during the CFP except for DSME-GTS that are assigned to the device. Note that reducing the CAP increases the duration of CFP and thus increases the duration of sleeping. High throughput applications such as file transfers can be supported by setting relatively large *macSuperframeOrder* value creating a long CAP. For large scale applications such as smart utility networks and process control application domains that have a large number of devices to be supported in a PAN, setting a relatively high value for *macMultiSuperframeOrder* can increase the number of devices supported by DSME-GTSs.

Table I.1—Recommended DSME parameter setting for various application types, where BO = *macBeaconOrder*, SO = *macSuperframeOrder*, MO = *macMultiSuperframeOrder*

Application type	BO	SO	MO	CAP reduction	Group ACK	Deferred beacon	DSME-GTS retrieve synchronization
Delay sensitive	6	0	1	Enabled	Enabled	Enabled	Enabled
Reliability sensitive	8	3		Disabled	Enabled	Enabled	Enabled
Energy critical	14	1	14	Enabled	Disabled	Disabled	Disabled
High throughput	10	5	6	Disabled	Disabled	Disabled	Disabled
Large scale	10	1	8	Enabled	Disabled	Enabled	Enabled

I.4.3 MAC behaviors unique to the DSME mode

The following MAC commands are specific to the DSME mode: DSME-Association request, DSME-Association reply, DSME-GTS request, DSME-GTS reply, DSME-GTS notify, DSME-Information request, DSME-Information reply, DSME-Beacon allocation notification, DSME-Beacon collision notification, DSME-Link status report. The following IE is specific to DSME: DSME PAN Descriptor header IE.

The DSME behavior is cited in 5.1.10, the IEs are described in 5.2.4.9, the DSME commands are stated in 5.3.11, and the primitives are declared in 6.2.21.

I.4.4 Channel diversity

I.4.4.1 Overview

Wireless PANs encounter significant receiver channel quality variations, which degrades the averaged signal reception quality. The main causes of the channel quality variations are multi-path fading and mutual RF interference. As an enhancement to the IEEE 802.15.4 MAC, two types of channel diversity methods, channel adaptation and channel hopping, are provided to overcome these impairments. Channel adaptation is a method whereby the channel is not changed unless the received signal quality drops down lower than a threshold value. In this manner, when channel quality is poor, it switches the channel to another one that is expected to show statistically different reception quality. Channel hopping is a method whereby the channel is periodically switched according to a predefined channel hopping sequence. The channel hopping sequence is defined by a layer above the MAC. The basic idea behind these channel diversity methods is to exploit the large diversity of receiver channel quality over a wide RF channel spectrum. The chances of a channel suffering impairments is statistically much lower than another one suffering deep fading located far apart. Thus, the average reception signal quality is expected to be significantly improved by switching from one channel to other channel frequency significantly different than the first channel.

I.4.4.2 Channel diversity methods

The IEEE 802.15.4 MAC describes two types of PAN operations: beacon-enabled and nonbeacon-enabled. Channel adaptation is implemented over the DSME structure in a beacon enabled PAN, while channel hopping can be implemented in either of the PAN operations. Refer to 5.1.10.2 for more detail. Channel hopping and channel adaptation do not exclude physical layer frequency hopping (PHY-FH) in which the PHY changes the channels independent of the MAC. In this manner, the fundamental difference between the channel hopping done in the MAC versus channel hopping done in the PHY is whether channel switching can occur during the transmission of a PPDU.

Figure I.2 illustrates the hopping methods. In the MAC channel hopping (MAC-CH) scheme, each PPDU is transmitted on different frequency channel [Figure I.2(a)]. In the PHY-FH a PPDU is divided into three fragments and each fragment is transmitted in a different sub timeslot with a different frequency channel [Figure I.2(b)].

It should be noted that the notion of the channel number gets complex and possibly conflicted when a combined channel diversity scheme (i.e., MAC and PHY) is used. To illustrate this point, consider a Hopping Sequence of {1,2,3,4} to be used by the MAC-CH. Before the transmission of the first PPDU, the PHY would configure the physical channel as per information obtained from Hopping Sequence for the MAC-CH while the it would also be configured as per the PHY-FH hopping sequence. Therefore the PHY could have conflicting requirements as to which channel should be used to transmit the frame. To resolve this conflict, the concept of a logical channel number being mapped to a PHY-FH Hopping Sequence is shown in Table I.2. In this example if the PHY-FH employs Hopping Sequences, {1,3,5,7}, {2,4,6,8}, {9,11,13,15}, and {10,12,14,16}, each sequence would be referred to as a logical channel numbers 1 through 4, respectively. Continuing with this example, if the MAC sets the logical channel number to 1, the PHY would use the Hopping Sequence {1,3,5,7} for the transmission of a PPDU. Table I.3 shows an example of logical channel numbers mapped to PHY Hopping Sequences.

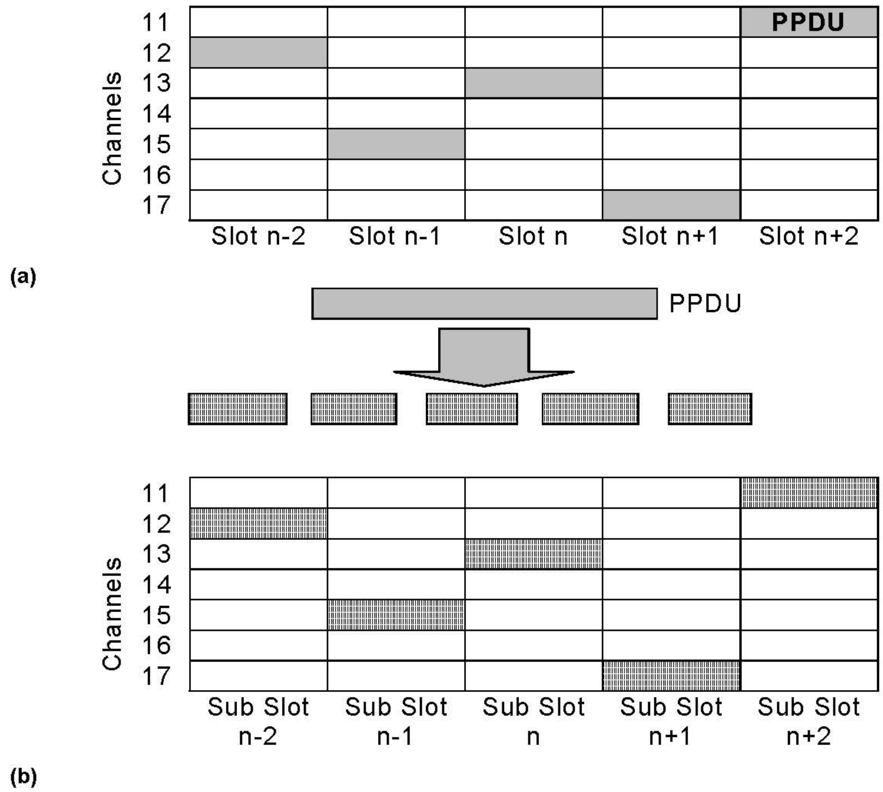

Figure I.2—Illustration of channel hopping in (a) MAC (b) PHY Layer

Table I.2—Logical channel numbering

PHY Hopping Sequence	Logical channel number
{1, 3, 5, 7}	1
{2, 4, 6, 8}	2
{9, 11, 13, 15}	3
{10, 12, 14, 16}	4

Table I.3—PHY Hopping Sequences using the notion of logical channels

MAC Hopping Sequence	PHY Hopping Sequences
{1,2,3,4}	{{1,3,5,7},{2,4,6,8},{9,11,13,15}, {10,12,14,16}}
{2,3,4,1}	{{2,4,6,8},{9,11,13,15}, {10,12,14,16},{1,3,5,7}}
{3,4,1,2}	{{9,11,13,15},{10,12,14,16},{1,3,5,7}, {2,4,6,8}}
{4,1,2,3}	{{10,12,14,16},{1,3,5,7}, {2,4,6,8},{9,11,13,15}}

I.4.4.3 MAC behaviors unique to Channel Diversity

There are no MAC commands specific to Channel Diversity. The following IEs are specific to Channel Diversity: Channel Hopping, Hopping Timing.

Channel Diversity functional behaviors are described in 5.1.10.2, and the IEs are cited in 5.2.4.16 and 5.2.4.17.

I.5 Blink

I.5.1 Blink mode overview

The Blink mode provides a method for a device to communicate its ID (i.e., the EUI-64 Source Address) and/or an alternate ID (in the payload), and optionally additional payload data to other devices, without prior association and without an Acknowledgement. The frame can be used by "transmit only" devices to co-exist within a network, utilizing an Aloha protocol. Any devices that are not interested in this Blink frame can reject the frame at an early stage during frame processing and not burden the whole MAC or higher communication layers with this data traffic.

The Blink mode is based upon the multipurpose frame to yield a minimal frame consisting only of the header fields necessary for its proper operation.

I.5.2 MAC behaviors unique to the Blink mode

There are no MAC commands specific to the Blink mode.

I.6 Asymmetric multi-channel adaptation (AMCA)

I.6.1 AMCA mode overview

There are deployments where a single common channel approach may not be able to connect all devices in a PAN. The variance of channel quality can be large; additionally link asymmetry can occur between two neighboring devices causing one device to not properly receive the other device's transmissions. Such cases are likely to happen in large, geographically diverse networks such as smart utility networks, infrastructure monitoring networks, and process control networks. It is for these cases that the AMCA mode is defined. The AMCA mode is used in a nonbeacon-enabled PAN.

I.6.2 MAC behaviors unique to AMCA

The following MAC commands are specific to the AMCA mode: AMCA-Beacon Request and AMCA-Channel probe.

AMCA functional behaviors are described in 5.1.2, and the AMCA commands are stated in 5.3.13.

I.7 Low energy

I.7.1 LE overview

The low-energy (LE) mechanisms are not specific to any one application domain, rather they are suitable for applications that are willing to trade low latency for low-energy consumption. The LE protocol allows devices to operate down to a fraction of 1% duty cycles while presenting an always-on illusion. The always-on illusion is good for the following:

— Internet protocol (IP) networks, it is what they are used to, i.e., it is a typical assumption
— Manageability, it allows for stateless devices, no prior synchronization of time and state required
— Asynchronous, event-driven, and/or infrequent communications
— Mobility and discovery
— Shifting overhead to transmissions

These behaviors are not exclusive of other low-energy mechanisms, i.e., other low-energy techniques can be layered on top.

Low-energy mechanisms support the conventional superframe structure and they are also applicable in the nonbeacon-enabled PAN as well as in the CAP periods of the beacon mode. It is also possible for the upper layer to temporarily turn off the low-energy mechanisms by operating the radio at 100% duty cycle for emergency messages.

There are two low-energy mechanisms: *coordinated sampled listening* (CSL) and *receiver initiated transmissions* (RIT). CSL is suitable for applications with relatively low latency requirements, e.g., < 1 s. RIT is suitable for applications with a high latency tolerance, e.g., tens of seconds. RIT is also required in cases where the local regulation limits the duration of continuous transmissions to too small a period for CSL to be effective. RIT mode is applicable to low duty cycle, low traffic load type of applications, and especially suitable in the case that consecutive radio emission time is limited by regional or national regulation (e.g., 950 MHz band in Japan).

I.7.2 MAC behaviors unique to LE

The following MAC command is specific to LE: LE-RIT data request. The following IEs are specific to LE: LE CSL, LE RIT, RZ Time.

LE functional behaviors are described in 5.1.1.7, transmission, reception, and acknowledgment in 5.1.11, the IEs are cited in 5.2.4.7, 5.2.4.8, and 5.2.4.10, and the LE commands are stated in 5.3.12.

I.8 Information elements

I.8.1 Overview

The concept of information elements (IEs) has previously been used in IEEE Std 802.11-2007 and IEEE Std 802.15.3-2003. An IE is a well defined, extensible mechanism to exchange data at the MAC sublayer. The IE implementation in this standard leverages the work done in IEEE Std 802.11 and IEEE Std 802.15.3 with alterations to better suit the IEEE 802.15.4 MAC and PHY and the applications served by the standard. It should be noted that the IE implementation allows the MAC to ignore IE types that are not implemented within the MAC.

I.8.2 IE types

There are two IE types: Header IEs and Payload IEs. Header IEs are used by the MAC to process the frame. Header IEs cover security, addressing, etc., and are part of the MHR. Payload IEs are destined for another layer or SAP and are part of the MAC payload.

I.8.3 IE Identification

The ID space is either managed or unmanaged. Managed IDs are defined by IEEE Std 802.15.4-2011. Unmanaged IDs are left for implementers to use with the caveat that these values are implementer specific and could cause misinterpretation in networks consisting of devices from multiple implementers.

I.8.4 IE Construction

An IE consists of four fields: Type Descriptor, Element ID, Length, and Content.

The Type Descriptor field is a single bit that defines the type of IE with zero indicating the IE to be a Header IE and one indicating the IE to be a Payload IE.

The ID field consists of 8 bits. For the allocation of Managed Element IDs refer to Table 4f for Header IEs, and Table 4b for Payload IEs.

The Length field for Header IEs is 7 bits (i.e., 0 to 128 octets) and for Payload IEs is 11 bits (i.e., 0 to 2047 octets).

I.8.5 MAC behaviors unique to IE

IE functional behaviors are described in 5.2.4. Managed Element IDs are defined in 5.2.4.6.

I.9 Enhanced beacons (EB) and enhanced beacon requests (EBR)

I.9.1 EB and EBR overview

The EB is an extension of the IEEE 802.11 beacon frame that provides greater flexibility in content than IEEE 802.15.4 beacons. A device can differentiate the EB from the IEEE 802.15.4 beacon frame by examining the Frame Version field; the EB has the Frame Version field set to 0b10. The EB format is used to construct application specific beacon content, as used in this standard by DSME and TSCH. The different application beacons are constructed by including the relevant IEs. The EB provides a means for application-specific information provided by higher layer protocols to be included in periodic and/or aperiodic beacons. The EB is also used as a query/response mechanism when used with the EBR.

The EBR is an extension of the beacon request MAC command. The EBR format is differentiated from the IEEE 802.15.4 beacon by the Frame Version field being set to 0b10. The content of the EBR is constructed using IEs. The EBR provides a means for the beacon request to specify specific response filters, providing constraints to qualify the beacon response to the request; in this manner, the beacon responses will be limited to the constraints set by the EBR. In this way a device conducts a scan for a PAN coordinator who is allowing new devices to join the PAN, or scans to discover a subset of like neighbors, for example neighbors or PAN coordinators that support a specific set of MAC and/or PHY capabilities. The EBR may also be used as a general query mechanism, specifying a list of requested IEs that a responding device will, if able, include in the beacons sent in response to the request. This can be used, for example, to convey MAC, PHY, or MAC and PHY capabilities between MAC entities, or exchange the performance metrics collected between MAC entities.

I.9.2 MAC behaviors unique to EB and EBR

The EB is defined as beacon frame as per Table 2 with the Version field set to 0b10 as per 5.2.2.1. The EBR is defined as a MAC request command frame according to Table 6 with the frame version set to 0b10.

The EB format is described in 5.2.2.1. Information contained in EB specific to DSME beacons and TSCH beacons is described in the respective functional subclauses.

The EBR format is described in 5.2.2.1. The enhanced active scan provides for selective scanning using response filters and a generalized query/response mechanism as described in 6.2.10. The mechanisms for higher layer interaction with the EBR/EB exchange are described in the MLME-BEACON-REQUEST service and the MLME-BEACON-NOTIFY service.

I.10 Multipurpose frame

I.10.1 Multipurpose frame overview

The multipurpose frame was introduced to provide an extensible, flexible frame format that can address a variety of MAC operations. Using a single frame type value, the multipurpose frame structure can support multiple MAC operations and provides the extensibility to accommodate new MAC requirements while maintaining backward compatibility.

The multipurpose frame flexibility comes from being based upon IEs. Being based upon IEs allows additional functionality without changing the structure or introducing new frame types.

A higher protocol layer can direct the MAC to configure the multipurpose frame to address unique needs of a specific application as evidenced by the RFID Blink described in I.5.

I.10.2 MAC behaviors unique to the multipurpose frame

The multipurpose frame is described in 5.2.2.6. The MCPS-DATA.request primitive described in 6.3.1 includes a parameter unique to the multipurpose frame.

I.11 MAC performance metrics

I.11.1 Requirements of routing (such as in mesh networks)

Many of the deployed networks of IEEE 802.15.4 devices are very large in area of deployment as well as in numbers of devices. These factors stress the networking layer above the MAC, including the mesh networking routing. To allow the networking layer (and other layers) to make optimal decisions, information on the link performances of devices is required.

I.11.2 Requirements of long transmit durations due to large frames or low data rates

The IEEE 802.15.4 MAC was designed for previous PHY layers with upper limits of 127-octet MPDUs. However, to allow direct support of IP packets, PHY amendments have increased the PHY frame length up to a minimum of 1500 octets. Still other PHY amendments have significantly decreased the effective data rates by transmitting fewer symbols per second or by increasing the forward error correction.

The resulting long packets and low rates can yield frame durations of many hundreds of milliseconds while channel impairments and interference bursts may occur in the range of a few milliseconds. Due to this behavior, there is a need to provide a set of metrics to allow higher layers to implement dynamic fragmentation based on channel conditions.

I.11.3 Metrics

Metrics have been added that inform the higher layers on key aspects of data frame transmission and data frame reception. The metrics essentially record the instances of significant link events in counters.

The attributes *macRetryCount, macMultipleRetryCount, macTXFailCount,* and *macTXSuccessCount* relate to data frame transmission.

The attributes *macFCSErrorCount, macSecurityFailure, macDuplicateFrameCount,* and *macRXSuccessCount* relate to data frame reception

If *macMetricsEnabled* is TRUE, the MAC shall collect the metrics listed in Table 52l. MAC PIB attribute *macCounterOctets* defines the size of the counters

I.11.4 MAC behaviors unique to MAC performance metrics

Metrics functional behaviors are described in 5.2.4. Managed element IDs are defined in 5.2.4.2.

I.12 Fast association

I.12.1 Overview

Due to low-energy considerations, the IEEE 802.15.4 association response command is sent using indirect transmission, which introduces a significant delay in the association procedure. There are applications that prioritize reduced association time over energy consumption during that operation. The MAC behavior associated with FastA allows a device to associate in a reduced duration time.

I.12.2 Operation

IEEE 802.15.4 devices will continue to be able to use the "association request command" using indirect transmission, but devices enabled with FastA behavior may configure the "association request command" (e.g., when it associates with an IEEE 802.15.4 device) or "direct association request" for Fast A.

I.12.3 MAC behaviors unique to FastA

FastA functional behaviors are described in 5.1.3.3. There are no unique MAC commands for FastA, the existing commands were enhanced to address fast association. Message sequence charts for FastA can be found in Figure 60a.