IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

Part 15.7: Short-Range Wireless Optical Communication Using Visible Light

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.15.7™-2011

6 September 2011

IEEE Std 802.15.7™-2011

IEEE Standard for Local and metropolitan area networks—

Part 15.7: Short-Range Wireless Optical Communication Using Visible Light

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 16 June 2011

IEEE-SA Standards Board

Abstract: A PHY and a MAC layer for short-range optical wireless communications using visible light in optically transparent media are defined. The visible light spectrum extends from 380 nm to 780 nm in wavelength. The standard is capable of delivering data rates sufficient to support audio and video multimedia services and also considers mobility of the visible link, compatibility with visible-light infrastructures, impairments due to noise and interference from sources like ambient light and a MAC layer that accommodates visible links. The standard adheres to applicable eye safety regulations.

Keywords: IEEE 802.15.7, laser diode, LD, LED, light-emitting diode, short-range optical wireless communications, visible light, visible-light communication, VLC

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

PDF: ISBN 978-0-7381-6665-0 STD97117
Print: ISBN 978-0-7381-6666-7 STDPD97117

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests.For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE.At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE. Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required.

Comments and recommendations on standards, and requests for interpretations should be addressed to:

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

Introduction

This introduction is not part of IEEE Std 802.15.7-2011, IEEE Standard for Local and metropolitan area networks—Part 15.7: Short-Range Wireless Optical Communication using Visible Light.

Visible-light communication (VLC) transmits data by intensity modulating optical sources, such as light-emitting diodes (LEDs) and laser diodes (LDs), faster than the persistence of the human eye. VLC merges lighting and data communications in applications such as area lighting, signboards, streetlights, vehicles, traffic signals, and status indicators. This standard describes the use of VLC for wireless personal area networks (WPAN) and covers topics such as network topologies, addressing, collision avoidance, acknowledgement, performance quality indication, dimming support, visibility support, colored status indication and color-stabilization.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard 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

Updating of IEEE documents

Users of IEEE standards 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 Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

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

Interpretations

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

Patents

Attention is called to the possibility that implementation of this amendment may require use of subject matter covered by patent rights. By publication of this amendment, no position is taken with respect to the existence or validity of any patent rights in connection therewith. 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 amendment 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 for approval, the IEEE P802.15.7 Working Group had the following membership:

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

Eun Tae Won, *IEEE 802.15.7 Chair*
Clint Chaplin, *IEEE 802.15.7 Vice Chair*
Richard D. Roberts, *IEEE 802.15.7 Technical Editor*
Sridhar Rajagopal, *IEEE 802.15.7 Assistant Editor*

Emad Afifi
Gahng-Seop Ahn
Roberto Aiello
Richard Alfvin
Takamasa Asano
Arthur Astrin
Taehan Bae
Michael Bahr
John Barr
Anuj Batra
Tuncer Baykas
Philip Beecher
Ghulam Bhatti
Gary Birk
Mathew Boytim
Peter Bradley
Nancy Bravin
David Britz
Monique Brown
Sverre Brubk
Brian Buchanan
John Buffington
Kiran Bynam
Brent Cain
Edgar Callaway
Chris Calvert
Radhakrishna Canchi
Ruben Cardozo
Russell Chandler
Kuor-Hsin Chang
Soo-Young Chang
Clint Chaplin
Hind Chebbo
Chang-Soon Choi
In-Kyeong Choi
Sangsung Choi
Ciaran Connell
David Cypher
Matthew Dahl
David Davenport
Mark Dawkins
Hendricus De Ruijter
Gang Ding
Paul Dixon
Igor Dotlic
Michael Dow
Dietmar Eggert
David Evans
Charles Farlow
John Farserotu
Kory Fifield
Will Filippo
Jeffrey Fischbeck
Michael Fischer
George Flammer
Ryosuke Fujiwara
Noriyasu Fukatsu
Kiyoshi Fukui
John Geiger
James Gilb
Gregory Gillooly
Tim Godfrey
Paul Gorday
Elad Gottlib
Robert Hall
Shinsuke Hara
Hiroshi Harada
Timothy Harrington
Robert Heile
Rodney Hemminger
Marco Hernandez
Ryoichi Higashi
Garth Hillman
Jin-Meng Ho
Wei Hong
Srinath Hosur
David Howard
Jung-Hwan Hwang
Ichirou Ida
Akio Iso
Adrian Jennings
Wuncheol Jeong
Jorjeta Jetcheva
Steven Jillings
Seong-Soon Joo
Tae-Gyu Kang
Shuzo Kato
Tatsuya Kato
Jeritt Kent
Prithpal Khakuria
Dae Ho Kim
Dong-Sun Kim
Yunjoo Kim
Jeffrey King
Patrick Kinney
Ryuji Kohno
Fumihide Kojima
Bruce Kraemer
Raymond Krasinski
Thomas Kuerner
Masahiro Kuroda
John Lampe
Zhou Lan
Khanh Le
Cheolhyo Lee
Hyungsoo Lee
Myung Lee
Yuro Lee
Daniel Lewis
Huan-Bang Li
Liang Li
Sang-Kyu Lim
Jeremy Link
Lu Liru
Michael Lynch
Robert Mason
Tomokuni Matsumura
Jeff McCullough
Michael McGillan
Michael McInnis
Michael McLaughlin
Charles Millet
Dino Miniutti
Siamak Mirnezami
Rishi Mohindra
Emmanuel Monnerie
Robert Moskowitz
Hamilton Moy
Peter Murray
Theodore Myers
Ken Naganuma
Chiu Ngo
Paul Nikolich
Jong-Ee Oh
David Olson
Okundu Omeni

Laurent Ouvry
James Pace
Hyung-Il Park
Seung-Hoon Park
Taejoon Park
Ranjeet Patro
Albert Petrick
Dalibor Pokrajac
Daniel Popa
Steve Pope
Clinton Powell
Richard Powell
Chang-Woo Pyo
Sridhar Rajagopal
Jayaram Ramasastry
Marc Reed
Ivan Reede
Emmanuel Riou
Richard Roberts
Craig Rodine
June Roh
Benjamin Rolfe
Didier Sagan
Kentaro Sakamoto
Kamran Sayrafian-Pour
Timothy Schmidl
Michael Schmidt
Jean Schwoerer
Cristina Seibert
Kunal Shah
Steve Shearer
Stephen Shellhammer
Jie Shen
Shusaku Shimada
Chang Sub Shin
Cheol Ho Shin
Michael Sim
Jonathan Simon
Jaeseung Son
Paul Stadnik
Rene Struik
Chin-Sean Sum
Hui-Hsia Sung
Gu Sungi
Ronald Tabroff
Kenichi Takizawa
Hirokazu Tanaka
Larry Taylor
James Tomcik
Ichihiko Toyoda
David Tracey
Khanh Tran
Jerry Upton
Jana van Greunen
Hartman Van Wyk
Billy Verso
Bhupender Virk
Khurram Waheed
Joachim Walewski
Junyi Wang
Quan Wang
Xiang Wang
Andy Ward
Scott Weikel
Nicholas West
Mark Wilbur
Ludwig Winkel
Eun Tae Won
Alan Wong
Tao Xing
Wen-Bin Yang
Yang Yang
Kazuyuki Yasukawa
Kamya Yazdandoost
Kaoru Yokoo
Mu Zhao
Bin Zhen

Major technical contributions were received from the following individuals:

Shadi Abu-Surra
Taehan Bae
Michael Bahr
Soo-Young Chang
Clint Chaplin
Praveen Gopalakrishnan
Il Soon Jang
Tae-Gyu Kang
Dae Ho Kim
Doyoung Kim
Sang-Kyu Lim
Eran Pisek
Sridhar Rajagopal
Richard Roberts
Jaeseung Son
Joachim W. Walewski
Euntae Won
Atsuya Yokoi

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

Richard Alfvin
Mark Anderson
Butch Anton
Taehan Bae
Youngkyo Baek
Nancy Bravin
Vern Brethour
Kiran Bynam
William Byrd
Juan Carreon
Soo-young Chang
Youngbin Chang
Clint Chaplin
Yawgeng Chau
Yi-ming Chen
Chihong Cho
Jaeweon Cho
Hokyu Choi
Keith Chow
Charles Cook
Joseph Decuir
Thomas Dineen
Gregory Gillooly
Randall Groves
C. Guy
Robert Heile
Marco Hernandez
Tetsushi Ikegami
Akio Iso
Atsushi Ito
Raj Jain
Oyvind Janbu
Jaehyuk Jang
Junghoon Jee
Suryong Jeong
Tal Kaitz
Shinkyo Kaku
Masahiko Kaneko
Hyunjeong Kang
Noh-gyoung Kang
Tae-gyu Kang
Piotr Karocki
Stuart J. Kerry
Sehoon Kim
Tae Kim
Yongbum Kim
Dongkeon Kong
Bruce Kraemer
Hyukchoon Kwon
Sungjin Lee
Charles Lennon
Michael Lerer
Ying Li
Jan-ray Liao
Chiwoo Lim
Hyoung-kyu Lim
Nae Hyun Lim
Sang-kyu Lim
Daniel Lubar
William Lumpkins
Elvis Maculuba
Apurva Mody
Emmanuel Monnerie
Peter Murray
Michael S. Newman
Satoshi Obara
Chris Osterloh
Satoshi Oyama
Jeongho Park
Jungshin Park
Seung-hoon Park
Sung-eun Park
James Petranovich
Subburajan Ponnuswamy
Clinton Powell
Venkatesha Prasad
Sridhar Rajagopal
Jayaram Ramasastry
Maximilian Riegel
Robert Robinson
Randal Roebuck
Won Il Roh
Bret Rothenberg
Randall Safier
Bartien Sayogo
Cristina Seibert
Jaejeong Shim
Gil Shultz
Jaeseung Son
Jung Je Son
Kapil Sood
Thomas Starai
Rene Struik
Walter Struppler
Mark Sturza
Ichihiko Toyoda
Mark-rene Uchida
Bhupender Virk
George Vlantis
Joachim Walewski
Stanley Wang
Hung-yu Wei
Ludwig Winkel
Andreas Wolf
Eun Tae Won
Hyunkyu Yu
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 16 June 2011, it had the following membership:

Richard H. Hulett, *Chair*
John Kulick, *Vice Chair*
Robert M. Grow, *Past Chair*
Judith Gorman, *Secretary*

Masayuki Ariyoshi
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé
Jim Hughes
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Phil Winston
Howard L. Wolfman
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, NRC Representative
Richard DeBlasio, DOE Representative
Michael Janezic, NIST Representative

Catherine Berger
IEEE Project Editor

Patricia Gerdon
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for
Local and metropolitan area networks—

Part 15.7: Short-Range Wireless Optical Communication Using Visible Light

1. Overview

1.1 Scope

This standard defines a PHY and MAC layer for short-range optical wireless communications using visible light in optically transparent media. The visible light spectrum extends from 380 nm to 780 nm in wavelength. The standard is capable of delivering data rates sufficient to support audio and video multimedia services and also considers mobility of the visible link, compatibility with visible-light infrastructures, impairments due to noise and interference from sources like ambient light and a MAC layer that accommodates visible links. The standard adheres to applicable eye safety regulations.

1.2 Purpose

The purpose of this standard is to provide a global standard for short-range optical wireless communication using visible light. The standard provides (i) access to several hundred THz of unlicensed spectrum; (ii) immunity to electromagnetic interference and noninterference with Radio Frequency (RF) systems; (iii) additional security by allowing the user to see the communication channel; and (iv) communication augmenting and complementing existing services (such as illumination, display, indication, decoration, etc.) from visible-light infrastructures.

2. Normative references

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

ANSI/INCITS 373: Fiber Channel Framing and Signaling Interface (FC-FS).[1]

IEEE Std 802.15.4™-2006, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).[2, 3]

ITU-T I.432.1, Series I: Integrated Services Digital Network, ISDN user-network interfaces—Layer 1RecommendationsB-ISDN user-network interface—Physical layer specification: General characteristics, http://www.itu.int/rec/T-REC-I.432.1-199902-I/en.[4]

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be consulted for terms not defined in this clause.[5]

color function: A function that provides information, such as device status and channel quality, to the human eye via color.

color-shift keying (CSK): A modulation scheme for visible-light communication involving multiple light sources, which keeps the average emitted optical color and the total optical power constant during communication.

color stabilization: A control loop for the stabilization of the color emitted by color-shift-keying transmitters.

color visibility dimming (CVD) frame: A frame used for color, visibility and dimming support. The color visibility dimming frame visually provides information such as communication status and channel quality to the user via various colors. The color visibility dimming frame may also be sent during idle or receive modes of operation for continuous visibility and dimming support. During the color visibility dimming frame transmission, the device is still emitting light while not communicating, and it is thus able to fulfill its lighting function. The payload of the frame consists of visibility patterns of appropriate intensity and color.

compensation time: The idle time inserted in the idle pattern or in the data frame, where the light is turned "ON" or "OFF" with the appropriate ratio to meet dimming requirements.

---

[1]ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

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

[3]The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[4]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

[5]*The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

dimming: Reducing the radiant power of a transmitter while preserving the color of the transmitted light.

field of view (FOV): The angular extent of coverage for the optical transmitter or receiver.

idle pattern: A pattern whose duty cycle variation results in a change of brightness for dimming support and may be transmitted during idle or receive mode.

macro cell: An aggregate cell formed using all the cells available at the optical media and is used for device discovery and association.

modulation-domain spectrum: The spectrum observed at the output of the receiver's photodetector; typically measured at the output of the trans impedance amplifier

on-off keying (OOK): A simple modulation technique that represents digital data as the presence ('ON') or absence ('OFF') of a signal. Note that 'ON' and 'OFF' are simply two logic levels or two distinct amplitude levels for the purposes of communication and does not necessarily require that the light source be turned OFF completely.

optical clock rate: The frequency at which the data is clocked out to the optical source.

photodetector: A photodetector captures optical power and translates it into an output signal. Most photodetectors convert optical power into an electrical current or an electrical voltage.

PHY switch: A switch at the transmission interface between the PHY and the optical SAP, used to send and receive data to and from a single or multiple optical sources and photodetectors in a selective manner.

point-and-shoot: The alignment of devices by the transmission of a color visibility dimming frame for the purpose of illuminating the target receiving device.

switching level: A distinct amplitude level that defines 'ON' and 'OFF' of the light source for the purpose of communications and does not necessarily require that the light source be turned off completely.

variable pulse-position modulation (VPPM): A modulation scheme for visible-light communication that allows pulse-width control for light dimming support, mitigating intra-frame flicker.

visibility pattern: An in-band idle pattern used in the payload of a color visibility dimming frame.

3.2 Acronyms and abbreviations

A/D	analog-to-digital converter
ACK	acknowledgment
AES	advanced encryption standard
AR	acknowledgment request
BE	backoff exponent
BI	beacon interval
BO	beacon order
BSN	beacon-sequence number
CAP	contention access period
CC	convolutional code
CCA	clear channel assessment
CDR	clock and data recovery
CFP	contention-free period
CIE	Commission Internationale de l'Eclairage (International Commission on Illumination)

CRC	cyclic redundancy check
CSK	color-shift keying
CSMA/CA	carrier sense multiple access with collision avoidance
CVD	color visibility dimming
D/A	digital-to-analog converter
D/L	downlink
DC	direct current
DME	device management entity
DSN	data-sequence number
ED	energy detection
ENC	encryption mode
FCS	frame check sequence
FDM	frequency division multiplexing
FEC	forward error correction
FER	frame-error ratio
FLP	fast locking pattern
FLR	fast link recovery
FOV	field of view
GF	Galois field
GTS	guaranteed time slot
HCS	header-check sequence
HP	hopping pattern
IFS	interframe space
ID	identifier
IE	information element
LD	laser diode
LED	light-emitting diode
LIFS	long interframe space
LLC	logical link control
LPDU	logical link control protocol data unit
LOS	line of sight
MAC	medium access control
MCPS	medium-access-control common-part sublayer
MCS	modulation and coding scheme
MD	mobile device
MFR	medium-access-control footer
MFTP	maximum flickering-time period
MHR	medium-access-control header
MIC	message-integrity code
MLME	medium-access-control link-management entity
MPDU	medium-access-control protocol-data unit
MSDU	medium-access-control service-data unit
NB	number of backoffs
OOK	on-off keying
PAN	personal-area network
PD	physical-layer data
PHR	physical-layer header
PHY	physical layer
PIB	physical-layer personal-area-network information base
PID	personal-area-network identifier
PLME	physical-layer management entity
PPDU	physical-layer data unit
PSDU	PHY service data unit
PWM	pulse-width modulation

P2MP	point-to-multipoint
P2P	peer-to-peer
QoS	quality of service
RIFS	reduced interframe space
RLL	run-length limited
RS	Reed-Solomon
RX	receiver
SAP	service access point
SHR	synchronization header
SIFS	short interframe space
SPDU	session-protocol data unit
SO	superframe order
SSCS	service-specific convergence sublayer
TDP	topology dependent pattern
TRX	transceiver
TX	transmitter
U/L	uplink
VPAN	visible-light communication personal area network
VLC	visible-light communication
VPPM	variable pulse-position modulation
WPAN	wireless personal area network
WQI	wavelength quality indication

4. General description

4.1 Introduction

Visible-light communication (VLC) transmits data by intensity modulating optical sources, such as light-emitting diodes (LEDs) and laser diodes (LDs), faster than the persistence of the human eye. VLC merges lighting and data communications in applications such as area lighting, signboards, streetlights, vehicles, and traffic signals. This standard describes the use of VLC for wireless personal area networks (WPAN). Some of the characteristics found in this standard are as follows:

a) Star, peer-to-peer, or broadcast operation
b) 16-bit short or 64-bit extended addresses
c) Scheduled or slotted random access with collision avoidance transmission
d) Fully acknowledged protocol for transfer reliability
e) Wavelength quality indication (WQI)
f) Dimming support
g) Visibility support
h) Color function support
i) Color-stabilization support

4.2 Network topologies

As shown in Table 1, three classes of devices are considered for VLC: infrastructure, mobile, and vehicle.

The IEEE 802.15.7 visible-light communication personal area network (VPAN) standard maps the intended applications to three topologies: peer-to-peer, star, and broadcast, as shown in Figure 1.

Table 1—Device classification

	Infrastructure	Mobile	Vehicle
Fixed coordinator	Yes	No	No
Power supply	Ample	Limited	Moderate
Form factor	Unconstrained	Constrained	Unconstrained
Light source	Intense	Weak	Intense
Physical mobility	No	Yes	Yes
Range	Short/long	Short	Long
Data rates	High/low	High	Low

In the star topology, the communication is established between devices and a single central controller, called the coordinator. In the peer-to-peer topology, one of the two devices in an association takes on the role of the coordinator.

Each device or coordinator has a unique 64-bit address. When a device associates with a coordinator it is allowed to be allocated a shortened 16-bit address. Either address is allowed to be used for communication within the VPAN managed by the coordinator. The coordinator might often be mains powered, while the devices will often be battery powered.

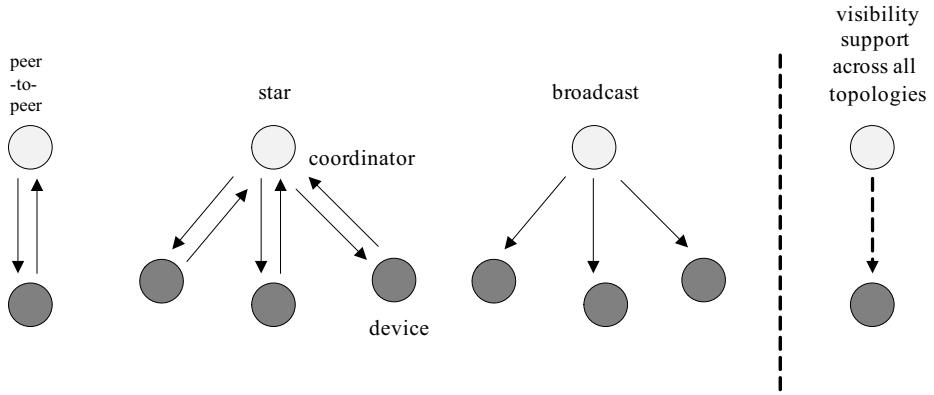


Figure 1—Supported MAC topologies

Each independent VPAN has an identifier, as defined in 5.2.1.3 and 5.2.1.5. This VPAN identifier allows communication between devices within a network using short addresses. The mechanism by which VPAN identifiers are chosen is outside the scope of this standard.

The network formation is performed by the higher layer, which is not part of this standard. Apart from the peer-to-peer and star topologies, IEEE 802.15.7 devices are also allowed to operate in a broadcast only topology without being part of a network, i.e., without being associated to any device or having any devices associated to them. A brief overview on how each supported topology may be formed is provided in 4.2.1, 4.2.2, and 4.2.3.

Visibility support is also provided across all topologies to maintain the illumination function in the absence of communication or in the idle or receive modes of operation. The purpose of this mode is to maintain illumination and mitigate flicker.

4.2.1 Peer-to-peer topology

The basic structure of a peer-to-peer topology is illustrated in Figure 1. In a peer-to-peer topology, each device is capable of communicating with any other device within its coverage area. In a peer-to-peer topology, one of the peers acts as a coordinator. One peer defaults as the coordinator, for instance, by virtue of being the first device to communicate on the channel.

4.2.2 Star topology

The basic structure of a star topology is illustrated in Figure 1. All star networks operate independently from all other star networks currently in operation. This is achieved by choosing a VPAN identifier that is not currently used by any other network within the coverage area. Once the VPAN identifier is chosen, the coordinator allows other devices to join its network. The higher layer is allowed to use the procedures described in 5.1.2 and 5.1.4 to form a star network.

4.2.3 Broadcast topology

The basic structure of a broadcast topology is illustrated in Figure 1. The device in a broadcast mode can transmit a signal to other devices without forming a network. The communication is uni-directional and the destination address is not required.

4.3 Modulation-domain spectrum

Figure 2 illustrates the concept of the modulation-domain spectrum. In Figure 2, the visible light source is "always on"; hence, the output of the photodetector can be observed for performing clear channel assessment (CCA). Prior to time $t = T_1$, the spectrum is all at DC. After $t = T_1$, the spectrum is split between DC and the modulating signal.

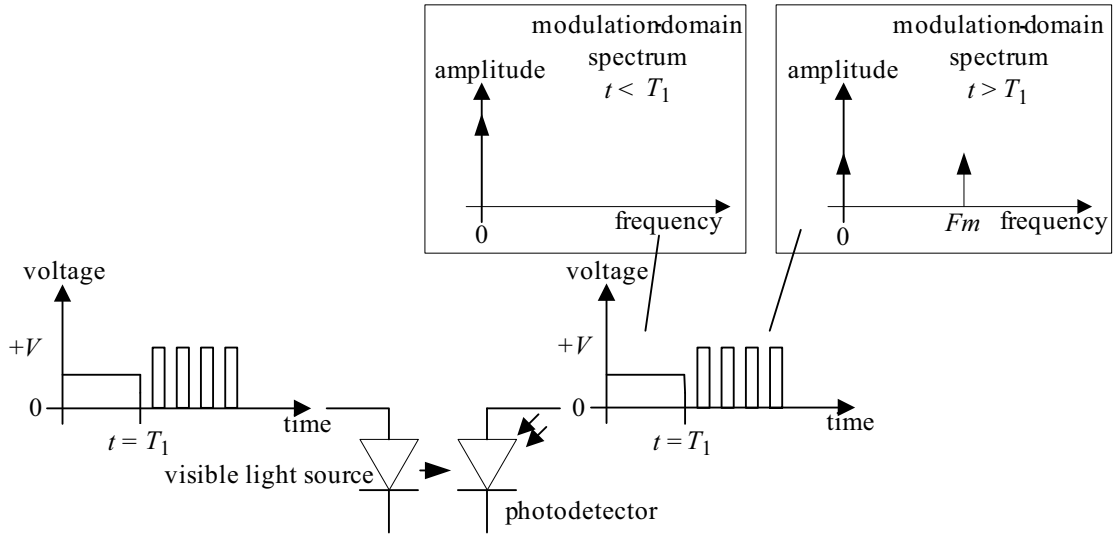


Figure 2—Illustration of modulation-domain spectrum

4.4 Architecture

The IEEE 802.15.7 architecture is defined in terms of a number of layers and sublayers in order to simplify the standard. Each layer is responsible for one part of the standard and offers services to the higher layers. The interface between the layers serve to define the logical links that are described in this standard.

A VPAN device comprises of a PHY layer, which contains the light transceiver along with its low-level control mechanism, and a medium access control (MAC) sublayer that provides access to the physical channel for all types of transfers. Figure 3 shows these layers in a graphical representation, which are described in more detail in 4.4.1 and 4.4.2.

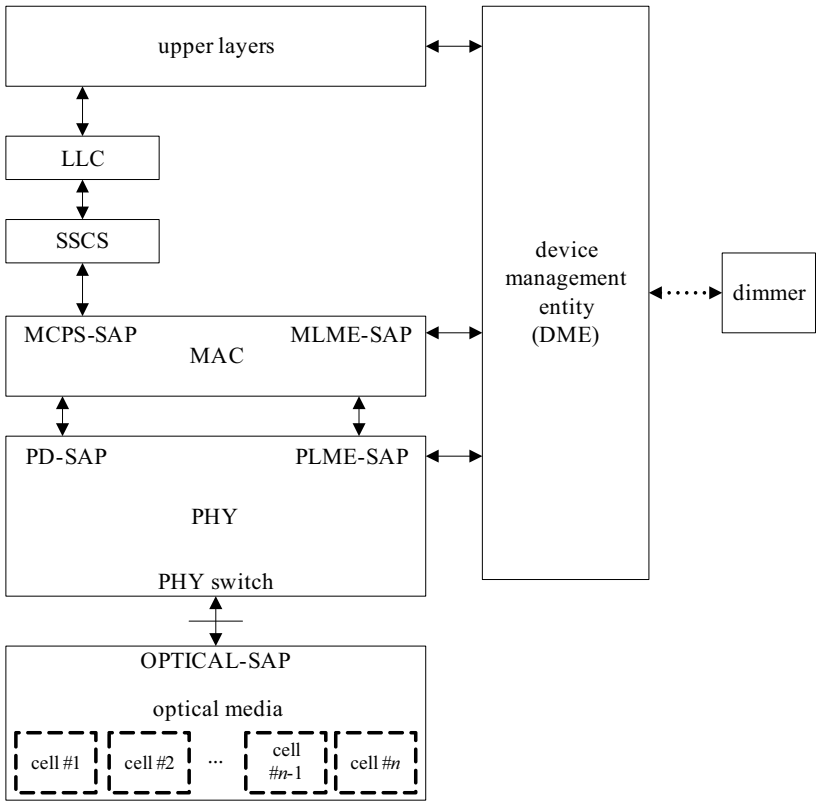


Figure 3—VPAN device architecture

The upper layers, shown in Figure 3, consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of the device. The definition of these upper layers is outside the scope of this standard. A logical link control (LLC) layer can access the MAC sublayer through the service-specific convergence sublayer (SSCS), defined in Annex B.

A device management entity (DME) is also supported in the architecture. The DME can talk to the PLME and MLME for the purposes of interfacing the MAC and PHY with a dimmer. The DME can access certain dimmer related attributes from the MLME and PLME in order to provide dimming information to the MAC and PHY. The DME can also control the PHY switch using the PLME for selection of the optical sources and photodetectors. The details of the DME are outside the scope of this standard. The PHY switch interfaces to the optical SAP and connects to the optical media, which may consist of a single or multiple optical sources and photodetectors. Multiple optical sources and photodetectors are supported in the

standard for PHY III as well for VLC cell mobility. The PLME controls the PHY switch in order to select a cell. The line going to the optical SAP from the PHY switch is a vector. The number of lines comprising the optical SAP has the dimension of $n \times m$, where 'n' is the number of cells and 'm' is the number of distinct data streams from the PHY. The value of 'm' is three for PHY III.

4.4.1 PHY layer

The PHY layer supports multiple PHY types.

a) PHY I: This PHY type is intended for outdoor usage with low data rate applications. This mode uses on-off keying (OOK) and variable pulse position modulation (VPPM) with data rates in the tens to hundreds of kb/s, as defined in Table 73.
b) PHY II: This PHY type is intended for indoor usage with moderate data rate applications. This mode uses OOK and VPPM with data rates in the tens of Mb/s, as defined in Table 74.
c) PHY III: This PHY type is intended for applications using color-shift keying (CSK) that have multiple light sources and detectors. This mode uses CSK with data rates in the tens of Mb/s, as defined in Table 75.

4.4.1.1 PHY frame structure

The MAC protocol data unit (MPDU) at the output of the MAC sublayer passes through the PHY layer and becomes the PHY service data unit (PSDU) at the output of the PHY layer after being processed via the various PHY blocks such as channel coding and line coding. The PSDU is prefixed with a synchronization header (SHR), containing the preamble sequence field; and a PHY header (PHR), which, among other things, contains the length of the PSDU in octets. The preamble sequence enables the receiver to achieve synchronization. The SHR, PHR, and PSDU together form the PHY frame or PHY layer data unit (PPDU). The format of the PHY frame is shown in Figure 118.

4.4.1.2 Interoperability and coexistence between PHY types

The PHY types coexist but do not interoperate. PHY I and PHY II occupy different spectral regions in the modulation-domain spectrum, which enables frequency division multiplexing (FDM) as a coexistence mechanism, as shown in Figure 4. PHY I and PHY III also occupy different spectral regions in the modulation-domain spectrum, with different data rates and different optical rate support, providing coexistence. However, the optical clock frequencies used for PHY II and PHY III overlap, causing significant overlap in the frequency domain spectrum. In addition, not all devices support multiple optical frequency bands needed for PHY III. Hence, all PHY III devices use a PHY II device for device discovery to support coexistence with PHY II.

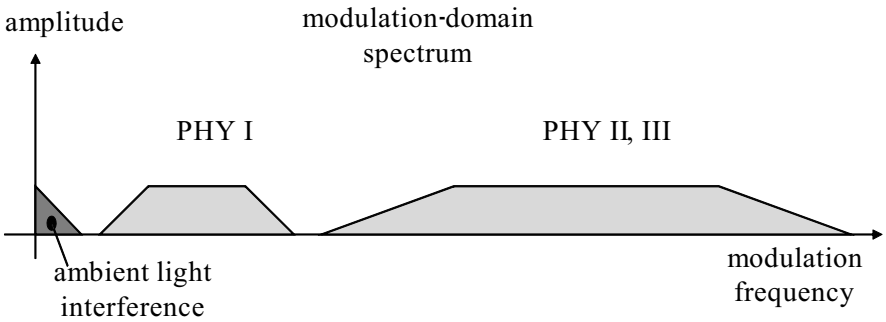


Figure 4—FDM separation of the PHY types in the modulation domain

4.4.2 MAC sublayer

The MAC sublayer provides two services accessed through two service access points (SAPs). MAC data is accessed through the MAC common part sublayer SAP (MCPS-SAP) while MAC management is accessed through the MAC sublayer management entity SAP (MLME-SAP). The MAC data service enables the transmission and reception of MPDUs across the PHY data service.

The features of the MAC sublayer are beacon management, channel access, guaranteed time slot (GTS) management, frame validation, acknowledged frame delivery, association, and disassociation. The MAC sublayer provides hooks for implementing application-appropriate security mechanisms. The MAC sublayer also provides color function, visibility, color-stabilization, and dimming support.

Clause 5 contains the specifications for the MAC sublayer.

4.4.3 Dimming and flicker-mitigation support

This subclause outlines the methods for dimming and flicker-mitigation support. An idle pattern can be transmitted during MAC idle or RX states for infrastructure light sources for dimming support. This is important since it is desired to maintain visibility and flicker-free operation during idle or RX periods at the infrastructure. The idle pattern has the same duty cycle that is used during the active data communication so that there is no flicker seen during idle periods. This idle pattern and its dependence on the dimmer setting is shown in Figure 5. The transition of active operation and idle/RX operation can be in large time scale (block active/idle/RX) or in a small time scale (within a communication session). In the large time scale block session activity, when the VLC activity is “ON”, there can be small time scaled transition of active mode and idle/RX mode. Dimmer setting for high brightness (a) in Figure 5 illustrates a higher duty cycle for higher brightness. Dimmer setting for low brightness (b) illustrates a lower duty cycle for lower brightness. The data and the idle pattern should have the same duty cycle in order to minimize flicker.

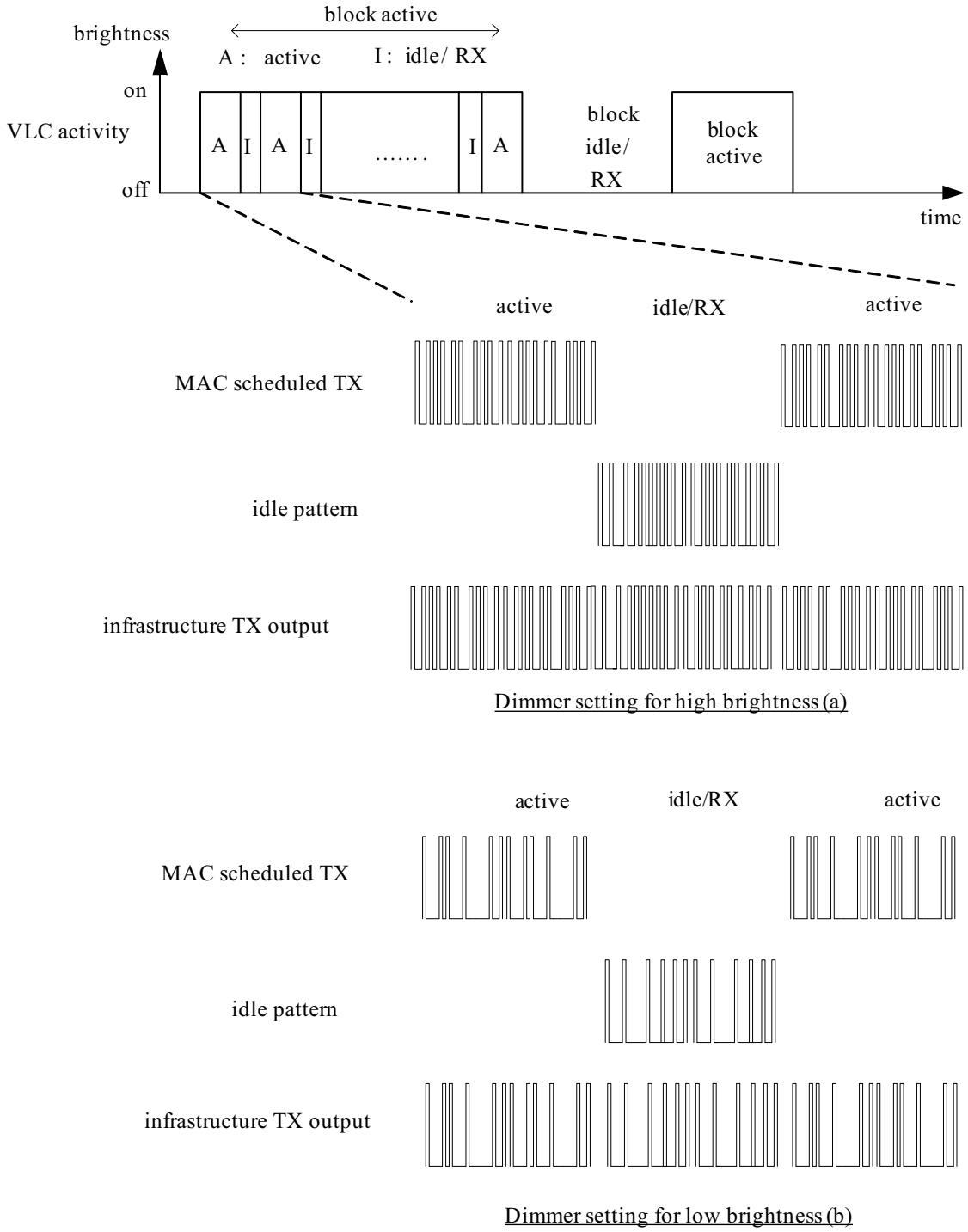


Figure 5—Adapting dimmer pattern and data duty cycle depending on dimmer setting

4.4.3.1 Light dimming

Light dimming is defined as controlling the perceived brightness of the light source according to the user's requirement and is a cross layer function between the PHY and MAC. The details on the light dimming function of MAC sublayer are discussed in 5.3.10.

4.4.3.1.1 Idle pattern and compensation time dimming

The standard allows an idle pattern to be inserted between the data frames for light dimming, as shown in Figure 6. The duty cycle of the idle pattern can be varied to provide brightness variation. The idle pattern selection is not specified in this standard. An idle pattern can either be in-band or out-of-band as defined by the modulation-domain spectrum and both types of idle patterns are supported in this standard. An in-band idle pattern does not require any change in the clock and can be seen by the receiver. An out-of-band idle pattern is typically sent at a much lower optical clock rate (including the option of maintaining visibility via a DC bias only) and is not seen by the receiver (i.e., it does not lie in the receiver's modulation-domain bandpass). The standard also allows a compensation time ("ON" or "OFF" time of a light source) to be inserted into either the idle pattern or into the data frame to reduce or increase the average brightness of a light source.

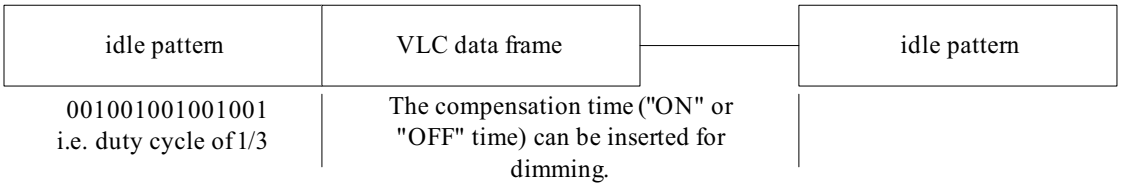


Figure 6—Example of idle pattern and compensation time dimming

4.4.3.1.2 Visibility pattern dimming

Visibility patterns are in-band idle patterns that are used in the payload of a CVD frame. The visibility patterns are used for supporting features such as flicker mitigation, continuous visibility, device discovery, and color stabilization. The visibility patterns are not encoded in the PHY layer and do not have a frame check sequence (FCS) associated with them. In order to generate high resolution visibility patterns from 0% to 100% in steps of 0.1%, there are certain constraints that need to be used in the design criteria for visibility patterns.

a) The number of transitions between ones and zeros can be maximized to provide high-frequency switching in order to avoid flicker and to help the clock and data recovery (CDR) circuit at receiver for synchronization purposes, if used.
b) Visibility pattern generation can be made in a simple manner. Designing a thousand patterns to support low resolutions (as low as 0.1% resolution) is not practical and makes visibility pattern generation and use very complex.
c) Since visibility patterns are transmitted without changing the clock frequency (in-band), the patterns avoiding conflicts with existing RLL code words are recommended.

The generation of the visibility patterns and their usage is defined in 8.5.1.2.

4.4.3.1.3 Color-shift keying (CSK) dimming

CSK supports VLC using multi-color light sources and photo detectors. CSK has the following advantages:

a) Information is provided by the color coordinates: CSK channels are defined by mixed colors that are allocated in the color coordinate plane; therefore, the connectivity is facilitated by the color coding.
b) Total average power is constant: The total average power of all CSK light sources is constant; therefore, the envelope of the sum of all light signals is constant.
c) Variable bit rate: CSK enables variable bit rate due to higher order modulation support; that is, the raw bit rate equals the optical clock rate times the bits per CSK symbol.

CSK dimming employs amplitude dimming and controls the brightness by changing the current driving the light source. However, a color shift of the light source may arise from improper control of driving current for amplitude dimming. For a given dimmer setting, the average optical power from the light sources is constant. This implies that the center color of the color constellation is constant.

4.4.3.1.4 OOK dimming

Since OOK modulation is always sent with a symmetric Manchester symbol, compensation time may need to be inserted into the data frame to adjust the average intensity of the perceived source. The structure for the OOK dimming frame is as shown in Figure 7. This process breaks the frame into subframes and each subframe can be preceded by a resync field that aids in readjusting the data clock after the compensation time. The data frame is fragmented into subframes of the appropriate length after the FCS has been calculated and the forward error correction (FEC) has been applied. An example of OOK dimming to increase brightness by adding compensation symbols is as shown in Figure 8.

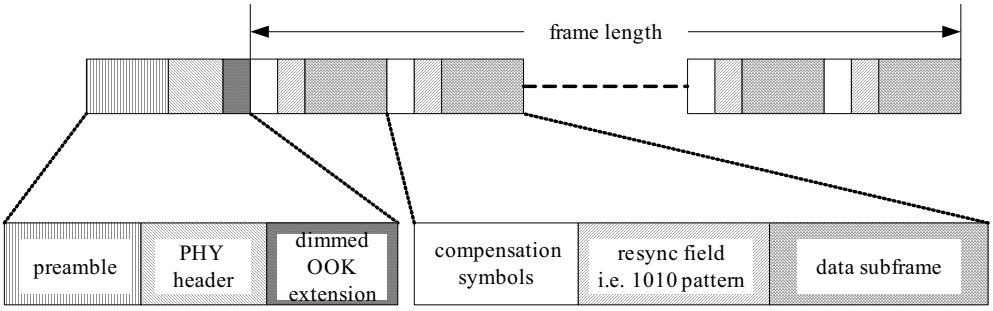


Figure 7—OOK dimming structure

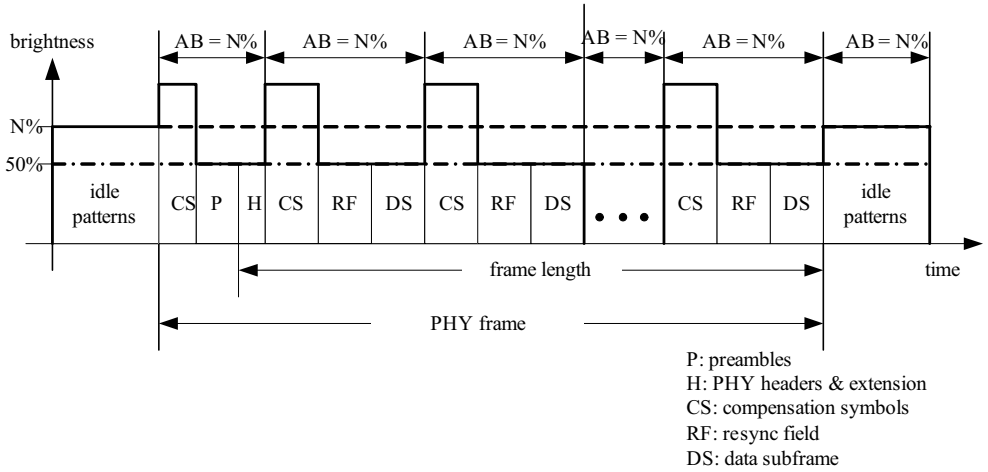


Figure 8—Example of OOK dimming to increase brightness

4.4.3.1.5 VPPM dimming

VPPM is a modulation scheme adapted for pulse width based light dimming and offers protection from intra-frame flicker. It does not create a color-shift in the light source that can arise from amplitude dimming because the pulse amplitude in VPPM is always constant and the dimming control is performed by the pulse width, not the amplitude.

VPPM makes use of the characteristics of 2-PPM (pulse position modulation) for non-flicker and PWM (pulse-width modulation) for dimming control and full brightness. Bits “1” and “0” in VPPM are distinguished by the pulse position within a unit period and have the same pulse width within their respective unit periods. The non-flicker characteristic in VPPM is obtained from the property that the average brightness on bits “1” and “0” is constant.

Dimming and full brightness in VPPM is achieved by controlling the “ON” time pulse width. Figure 9 describes the dimming control mechanism by VPPM. It is possible to adjust the pulse width for VPPM based on the dimming requirements. Therefore, a user can achieve the full brightness that can be provided by the light source.

As shown in Figure 10, the light intensity for the payload can be adjusted by adapting the pulse width of VPPM symbols. The light intensity for the preamble and header can be adjusted by inserting compensation symbols of the appropriate length and intensity before the frame. The details on high resolution dimming using VPPM are described in 8.5.2.3.

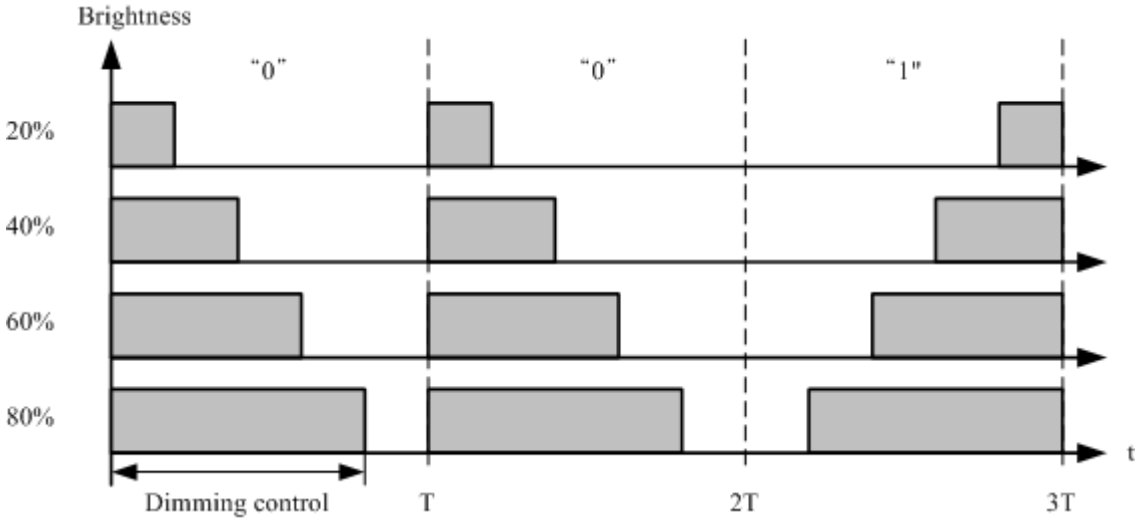


Figure 9—Schematic mechanism for VPPM dimming

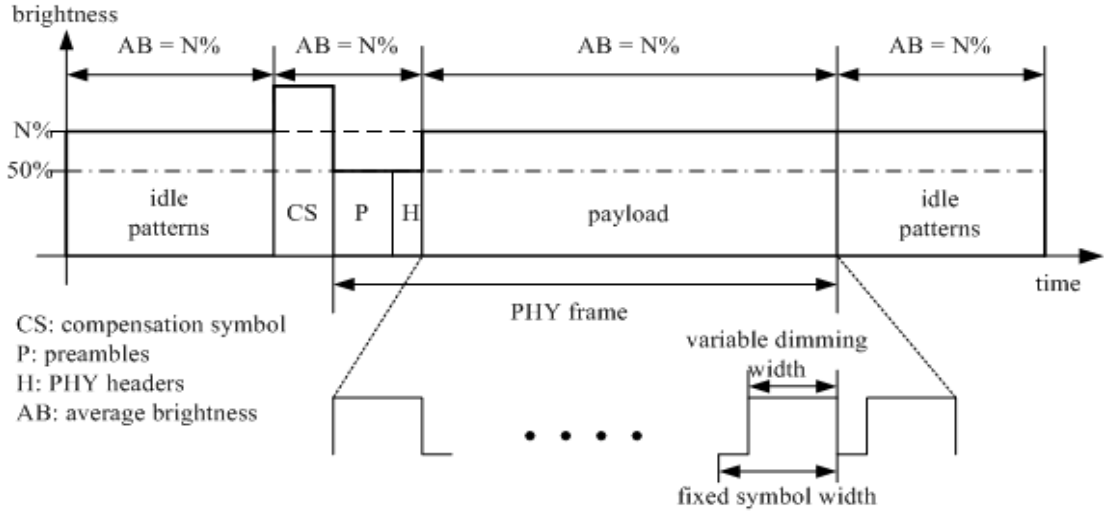


Figure 10—Example of VPPM dimming

4.4.3.2 Flicker mitigation

Flicker is defined as the fluctuation of the brightness of light that can cause noticeable physiological changes in humans. This standard strives for the mitigation of flicker that may be caused due to modulation of the light sources for communication. The maximum flickering time period (MFTP) is defined as the maximum time period over which the light intensity can be changing, but for which the resulting flicker is not perceivable by the human eye (Berman, et al. [B13]). To avoid flicker, the brightness changes over periods longer than MFTP must be avoided.

The flicker in VLC is classified into two categories according to its generation mechanism: intra-frame flicker and interframe flicker. Intra-frame flicker is defined as the perceivable brightness fluctuation within a frame. Interframe flicker is defined as the perceivable brightness fluctuation between adjacent frame transmissions.

4.4.3.2.1 Intra-frame flicker mitigation

Intra-frame flicker mitigation is accomplished by either the use of run length limiting coding, modulation scheme, or both. Specifically, these schemes are manchester encoding as specified in 10.5.2, 4B6B encoding as specified in 10.5.1, 8B10B encoding as specified in 11.3, or VPPM as specified in 10.6.

4.4.3.2.2 Interframe flicker mitigation

The scheme used for interframe flicker mitigation is the transmission of an idle pattern between data frames whose average brightness is equal to that of the data frames, as defined in 8.5.1.1.

4.5 Functional overview

This clause provides a brief overview of the general functions of a VPAN MAC sublayer and includes information on the superframe structure, data transfer model, data frame structure, acknowledgments, and security.

4.5.1 Superframe structure

This standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. The superframe is bounded by beacons sent by the coordinator, as shown in Figure 11a, and is divided into equally sized slots. Optionally, the superframe can have an active and an inactive portion, as shown in Figure 11b.The beacon frame is transmitted in the first slot of each superframe. If a coordinator does not wish to use a superframe structure, it will turn off the beacon transmissions. The beacons are used to synchronize the attached devices, to identify the VPAN, and to describe the structure of the superframes. Any device wishing to communicate during the contention access period (CAP) between two beacons competes with other devices via slotted random access.The standard defines four random access methods: unslotted random access, slotted random access, unslotted CSMA/CA, and slotted CSMA/CA. These methods are described in 5.1.1.3.

All transactions are completed by the time of the next beacon.

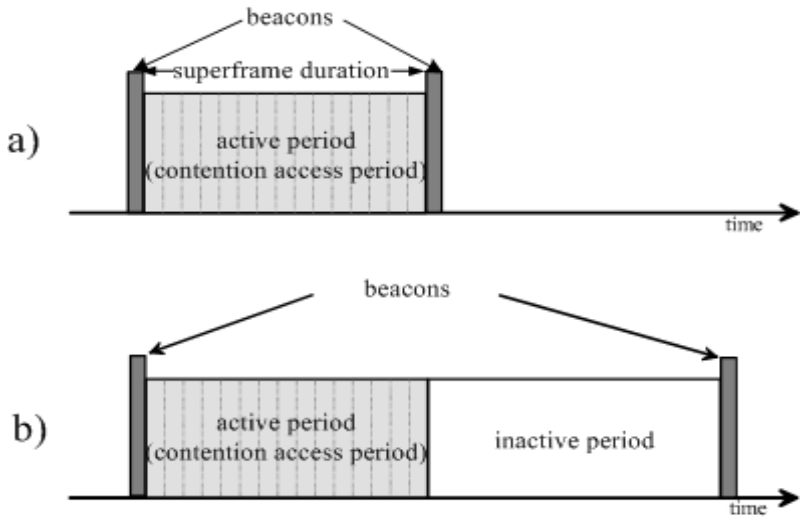


Figure 11—Superframe structure without GTSs

For low-latency applications or applications requiring specific data bandwidth, the coordinator is allowed to dedicate portions of the active superframe to that application. These portions are called guaranteed time slots (GTSs). The GTSs form the contention-free period (CFP), which always appears at the end of the active superframe starting at a slot boundary immediately following the CAP. More information on the GTS structure is provided in Figure 13. The coordinator is allowed to allocate a number of these GTSs, and a GTS may occupy more than one slot period. More information on the GTS slots and the maximum number available can be found in 5.2.2.1. All contention-based transactions are completed before the CFP begins. Also each device transmitting in a GTS ensures that its transaction is complete before the time of the next GTS or the end of the CFP. More information on the superframe structure as described in 5.1.1.1.

4.5.2 Data transfer model

Three types of data transfer transactions exist:

a) The first transaction type is the data transfer to a coordinator in which a device transmits the data.
b) The second transaction type is the data transfer from a coordinator in which the device receives the data.
c) The third transaction type is the data transfer between two peer devices.

In a star or broadcast topology, only the first two transaction types are used, while in a peer-to-peer topology, all three transaction types are allowed.

The mechanisms for each transfer type depend on whether the network supports the transmission of beacons. A beacon-enabled VPAN is used in networks that either require synchronization or support for low-latency devices. If the network does not need synchronization or support for low latency devices, it can elect not to use the beacon for normal transfers. However, the beacon is still required for network discovery. The structure of the frames used for data transfer is specified in 5.2.

4.5.2.1 Data transfer to a coordinator

When a device wishes to transfer data to a coordinator in a beacon-enabled VPAN, it first listens for the beacon. When the beacon is found, the device synchronizes to the superframe structure. At the appropriate

time, the device transmits its data frame, using slotted random access, to the coordinator. The coordinator is allowed to acknowledge the successful reception of the data by transmitting an optional acknowledgment frame.

When a device wishes to transfer data in a non beacon-enabled VPAN, it simply transmits its data frame, using unslotted random access, to the coordinator. The coordinator acknowledges the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete.

4.5.2.2 Data transfer from a coordinator

When the coordinator wishes to transfer data to a device in a beacon-enabled VPAN, it indicates in the beacon that the data message is pending. The device periodically listens to the beacon and, if a message is pending, transmits a MAC command requesting the data, using slotted random access. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. The pending data frame is then sent using slotted random access, or; if possible, immediately after the acknowledgment as described in 5.1.7.3. The device is allowed to acknowledge the successful reception of the data by transmitting an optional acknowledgment frame. The transaction is now complete. Upon successful completion of the data transaction, the message is removed from the list of pending messages in the beacon.

When a coordinator wishes to transfer data to a device in a non beacon-enabled VPAN, it stores the data and waits for the appropriate device to make contact and request the data. A device is allowed to make contact by transmitting a MAC command requesting the data, using unslotted random access, to its coordinator. The coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. If a data frame is pending, the coordinator transmits the data frame, using unslotted random access to the device. If a data frame is not pending, the coordinator indicates this fact either in the acknowledgment frame following the data request or in a data frame with a zero-length payload as described in 5.1.7.3. If requested, the device acknowledges the successful reception of the data frame by transmitting an acknowledgment frame.

4.5.2.3 Peer-to-peer data transfers

In a peer-to-peer VPAN, every device is allowed to communicate with every other device in its coverage area. In order to do this effectively, the devices wishing to communicate will need to either receive constantly or synchronize with each other. In the former case, the device can simply transmit its data using unslotted random access. In the latter case, other measures need to be taken in order to achieve synchronization. Such measures are beyond the scope of this standard.

4.5.3 Clock-rate selection

The standard supports multiple optical clock rates in order to accommodate a wide variety of optical sources and receivers. The standard also supports the use of asymmetric clock rates between two devices since the transmitter and receiver in a device are independent and may support different clock-rate ranges. As an example, the infrastructure transmitter may be unable to switch rapidly but may be able to transmit with high power and require lower error correction while the mobile device transmitter may be able to switch rapidly but may require higher error correction support due to its lower transmit power. The optical clock rate for communication is established using the MAC and can be communicated to the receiver prior to data transfer. The clock-rate selection and negotiation procedure is described in 6.5.

4.5.4 Frame structure

The frame structures have been designed to keep the complexity to a minimum while providing for error protection for transmission on a noisy channel. Each successive protocol layer adds to the structure with layer-specific headers and footers.

a) A beacon frame, used by a coordinator to transmit beacons.
b) A data frame, used for all transfers of data.
c) An acknowledgment frame, used for confirming successful frame reception.
d) A MAC command frame, used for handling all MAC peer entity control transfer.
e) A CVD frame, used to maintain the proper light intensity between data frames, support dimming and for visually providing information such as communication status and channel quality to the user.

4.5.5 Improving probability of successful delivery

The IEEE 802.15.7 VPAN employs various mechanisms to improve the probability of successful data transmission. These mechanisms are random access, frame acknowledgment, and data verification.

4.5.5.1 Random access mechanism

The IEEE 802.15.7 VPAN uses four types of channel access mechanism, depending on the network configuration. Non-beacon-enabled VPANs use an unslotted random channel access mechanism, with or without CSMA/CA, as described in 5.1.1.3. Each time a device wishes to transmit data frames or MAC commands, it waits for a random back off period. Following the random back off, the device transmits its frame of data. If the optional carrier sense mechanism is active and the channel is found to be busy following the random back off, the device waits for another random period before trying to access the channel again. Acknowledgment frames are sent without using a random access mechanism (i.e., scheduled).

Beacon-enabled VPANs use a slotted random channel access mechanism, with or without CSMA/CA, where the back off slots are aligned with the start of the beacon transmission. Each time a device wishes to transmit data frames during the CAP, it locates the boundary of the next back off slot and then waits for a random number of back off slots. If the optional collision avoidance mechanism is active, and the channel is busy, following this random back off the device waits for another random number of back off slots before trying to access the channel again. If the channel is idle or the optional carrier sense mechanism is not active, the device begins transmitting on the next available back off slot boundary. Acknowledgment and beacon frames are sent without using a random access mechanism (i.e., scheduled).

4.5.5.2 Frame acknowledgment

A successful reception and validation of a data or MAC command frame can be optionally confirmed with an acknowledgment, as described in 5.1.7.4. If the receiving device is unable to handle the received data frame for any reason, the message is not acknowledged.

If the originator does not receive an acknowledgment after some period, it assumes that the transmission was unsuccessful. When the acknowledgment is not required, the originator assumes the transmission was successful.

4.5.5.3 Data verification

A cyclic redundancy check is included in the MAC frame and the PHY header, as defined in Annex C, to verify the validity of the received data.

4.6 Security

From a security perspective, IEEE 802.15.7 VPAN is slightly different from other wireless networks, due to directionality and visibility because of the choice of the visible optical spectrum. Because of directionality and visibility, if an unauthorized receiver is in the path of the communication signal, it can be recognized. Also, the signal will not travel across medium such as walls, unlike other radio frequency based wireless

networks. However, security algorithms are still provided in the standard for features such as data confidentiality, authentication and replay protection.

Devices can be low-cost and have limited capabilities in terms of computing power, available storage, and power drain; and it cannot always be assumed they have a trusted computing base nor a high-quality random number generator aboard. Communications cannot rely on the online availability of a fixed infrastructure and might involve short-term relationships between devices that may never have previously communicated. These constraints limit the choice of cryptographic algorithms and protocols and influence the design of the security architecture because the establishment and maintenance of trust relationships between devices need to be addressed with care. In addition, battery lifetime and cost constraints can put severe limits on the security overhead these networks can tolerate, something that is of far less concern with higher bandwidth networks. Most of these security architectural elements can be implemented at higher layers and may, therefore, be considered to be outside the scope of this standard.

The cryptographic mechanism in this standard is based on symmetric-key cryptography and uses keys that are provided by higher layer processes. The establishment and maintenance of these keys is outside the scope of this standard. The mechanism assumes a secure implementation of cryptographic operations and secure and authentic storage of keying material.

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

a) *Data confidentiality:* Assurance that transmitted information is only disclosed to parties for whom it is intended.
b) *Data authenticity:* Assurance of the source of transmitted information (and, thereby, that information was not modified in transit).
c) *Replay protection:* Assurance that duplicate information is detected.

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

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

The cryptographic security mechanisms used for protected MAC frames is described in Clause 7.

4.7 Concept of primitives

This subclause provides a brief overview of the concept of service primitives. The services of a layer are the capabilities it offers to the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 12, showing the service hierarchy and the relationship of the two correspondent users and their associated layer (or sublayer) peer protocol entities.

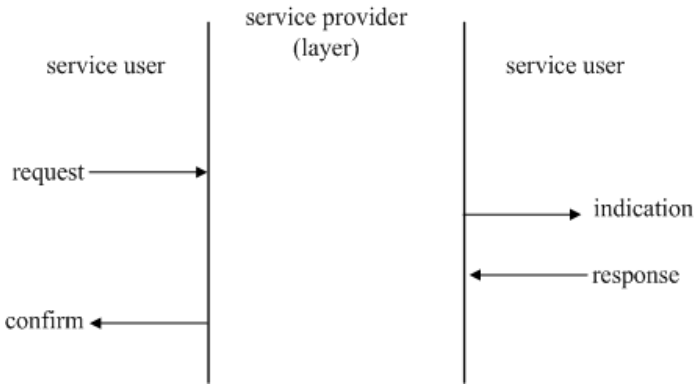


Figure 12—Service primitives

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

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

A primitive can be one of the following four generic types:

a) *Request:* The request primitive is used to request a service to be initiated.
b) *Confirm:* The confirm primitive is used to convey the results of one or more associated previous service requests.
c) *Indication:* The indication primitive is used to indicate the next higher layer of an internal event.
d) *Response:* The response primitive is used to complete a procedure previously invoked by an indication primitive.

5. MAC protocol specification

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

a) Generating network beacons if the device is a coordinator
b) Synchronizing to network beacons
c) Supporting VPAN association and disassociation
d) Supporting color function
e) Supporting visibility
f) Supporting dimming
g) Flicker-mitigation scheme
h) Supporting visual indication of device status and channel quality
i) Supporting device security
j) Providing a reliable link between two peer MAC entities
k) Supporting mobility

Peer-to-peer, star and broadcasting capabilities, as shown in Figure 1, are provided with a single MAC frame structure. All of these diverse modes are supported via a single low complexity integrated frame structure.

Constants and attributes that are specified and maintained by the MAC sublayer are written in the text of this clause in italics. Constants have a general prefix of “a”, e.g., *aBaseSlotDuration*, and are listed in Table 59 (see 6.4.1). Attributes have a general prefix of “mac”, e.g., *macAckWaitDuration*, and are listed in Table 60 (see 6.4.2), while the security attributes are listed in Table 66 (see 7.5.1).

5.1 MAC functional description

This subclause provides a detailed description of the MAC functionality. Subclause 5.1.1 describes the following two mechanisms for channel access: contention based and contention free. Contention-based access allows devices to access the channel in a distributed fashion using an unslotted random access backoff algorithm. Contention-free access is controlled entirely by the coordinator through the use of GTSs.

The mechanisms used for starting and maintaining a VPAN are respectively described in 5.1.2 and 5.1.3. Channel scanning is used by a device to assess the current state of a channel (or channels), locate all beacons within its operating space, or locate a particular beacon with which it has lost synchronization. Before starting a new VPAN, the results of a channel scan can be used to select an appropriate logical channel, as well as a VPAN identifier that is not being used by any other VPAN in the area. Because it is still possible for the operating space of two VPANs with the same VPAN identifier to overlap, a procedure exists to detect and resolve this situation. Following a channel scan and suitable VPAN identifier selection, operation as a coordinator shall commence. Also described in the subclause is a method to allow coordinator beaconing to discover other such devices during normal operations, i.e., when not scanning.

The mechanisms to allow devices to join or leave a VPAN are defined in 5.1.4. The association procedure describes the conditions under which a device may join a VPAN and the conditions necessary for a coordinator to permit devices to join. Also described is the disassociation procedure, which can be initiated by the associated device or its coordinator.

The mechanisms to allow devices to acquire and maintain synchronization with a coordinator are described in 5.1.5. Synchronization on a beacon-enabled VPAN is described after first explaining how a coordinator generates beacon frames. Following this explanation, synchronization on a nonbeacon-enabled VPAN is described. Also described is a procedure to reestablish communication between a device and its coordinator, as it is possible that a device may lose synchronization in the case of either a beacon-enabled or a nonbeacon-enabled VPAN.

This standard has been designed so that application data transfers can be controlled by the devices on a VPAN rather than by the coordinator. The procedures the coordinator uses to handle multiple transactions while preserving this requirement are described in 5.1.6.

The mechanisms for transmitting, receiving, and acknowledging frames, including frames sent using indirect transmission, are described in 5.1.7. In addition, methods for retransmitting frames are also described.

The mechanisms for allocating and deallocating a GTS are described in 5.1.8. The deallocation process may result in the fragmentation of the GTS space, i.e., an unused slot or slots. The subclause describes a mechanism to resolve fragmentation.

The MAC sublayer uses the mechanisms described in Clause 7 for all incoming and outgoing frames.

Throughout this subclause, the receipt of a frame is defined as the successful receipt of the frame by the PHY and the successful verification of the FCS by the MAC sublayer, as described in 5.2.1.9.

The mechanisms to allow devices to recover quickly in case of temporary interference using a fast link recovery process are defined in 5.1.9. The fast link recovery process also enables devices to save power by letting the infrastructure initiate the link recovery.

The mechanisms to allow devices to use multiple channels in case of limited time resources or interference are defined in 5.1.10. Multiple channel resource assignment uses information about multiple channel support and band hopping in order to support more users or improve performance.

The mechanisms to support mobility of the device under an infrastructure that supports multiple optical elements over a wide coverage area are defined in 5.1.11. The concept of a cell is introduced and the support for mobility across multiple cells supported by the infrastructure is presented.

The mechanisms to visually indicate to the user the various states using various colors are defined in 5.1.12. The various states such as device discovery (scan, association, disassociation), file transfer status, wavelength quality indication and acknowledgments can be visually indicated to the user to help with device alignment for communication.

The mechanisms to stabilize the optical color emitted by the transmitter are defined in 5.1.13. The CVD frames are used to estimate the change in color and this information can be provided as feedback to the transmitter to stabilize its color.

The mechanisms for using the visibility and dimming information in the MAC are defined in 5.1.14. Features such as an extended preamble mode for providing visibility with improved synchronization performance, dimming overrides, adjusting the MAC layer transmission schedule to accommodate dimming, association and link adaptation in the presence of dimming are provided.

5.1.1 Channel access

This subclause describes the mechanisms for accessing the physical optical channel. The standard provides a single VLC MAC frame structure that can be configured for multiple modes. The frame is composed of a variable number of slots. A slot can be defined as the minimum time needed to communicate to send the smallest data to a device and is fixed.

5.1.1.1 Superframe structure

A coordinator on a VPAN can optionally bound its channel time using a superframe structure. A superframe is bounded by the transmission of a beacon frame and can have an active portion and an inactive portion. The coordinator may enter a low-power (sleep) mode during the inactive portion.

The structure of this superframe is described by the values of *macBeaconOrder* and *macSuperframeOrder*. The MAC PIB attribute *macBeaconOrder*, describes the interval at which the coordinator shall transmit its beacon frames. The value of *macBeaconOrder*, *BO*, and the beacon interval, *BI*, are related as follows: for $0 \leq BO \leq 14$, $BI = aBaseSuperframeDuration \times 2^{BO}$ optical clocks. If $BO = 15$, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command. The value of *macSuperframeOrder* shall be ignored if $BO = 15$.

The MAC PIB attribute *macSuperframeOrder* describes the length of the active portion of the superframe, which includes the beacon frame. The value of *macSuperframeOrder*, *SO*, and the superframe duration, *SD*, are related as follows: for $0 \leq SO \leq BO \leq 14$, $SD = aBaseSuperframeDuration \times 2^{SO}$ optical clocks. If $SO = 15$, the superframe shall not remain active after the beacon. If $BO = 15$, the superframe shall not exist (the value of *macSuperframeOrder* shall be ignored), and *macRxOnWhenIdle* shall define whether the receiver is enabled during periods of transceiver inactivity.

The active portion of each superframe shall be divided into *aNumSuperframeSlots* equally spaced slots of duration $2^{SO} \times aBaseSlotDuration$ and is composed of three parts: a beacon, a CAP and a CFP. The beacon shall be transmitted, without the use of any random access, at the start of slot 0, and the CAP shall commence immediately following the beacon. The start of slot 0 is defined as the point at which the first bit of the beacon PPDU is transmitted. The CFP, if present, follows immediately after the CAP and extends to the end of the active portion of the superframe. Any allocated GTSs shall be located within the CFP.

The MAC sublayer shall ensure that the integrity of the superframe timing is maintained, e.g., compensating for clock drift error.

VPANs that wish to use the superframe structure (referred to as a beacon-enabled VPAN) shall set *macBeaconOrder* to a value between 0 and 14, both inclusive, and *macSuperframeOrder* to a value between 0 and the value of *macBeaconOrder*, both inclusive.

VPANs that do not wish to use the superframe structure (referred to as a nonbeacon-enabled VPAN) shall set both *macBeaconOrder* and *macSuperframeOrder* to 15. In this case, a coordinator shall not transmit beacons, except upon receipt of a beacon request command; all transmissions, with the exception of acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command, see 5.1.7.3, shall use an unslotted random access mechanism to access the channel. In addition, GTSs shall not be permitted.

An example of a superframe structure is shown in Figure 13. In this case, the beacon interval, *BI*, is twice as long as the active superframe duration, *SD*, and the CFP contains two GTSs.

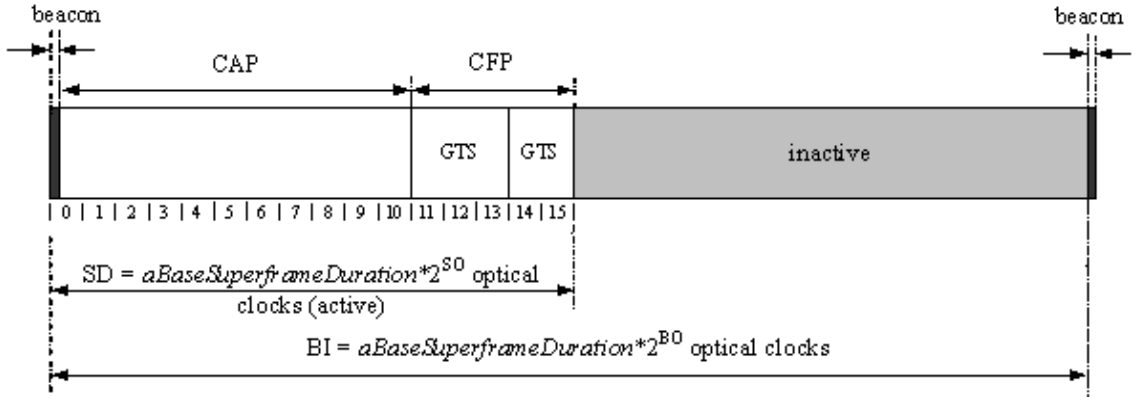


Figure 13—An example of the superframe structure

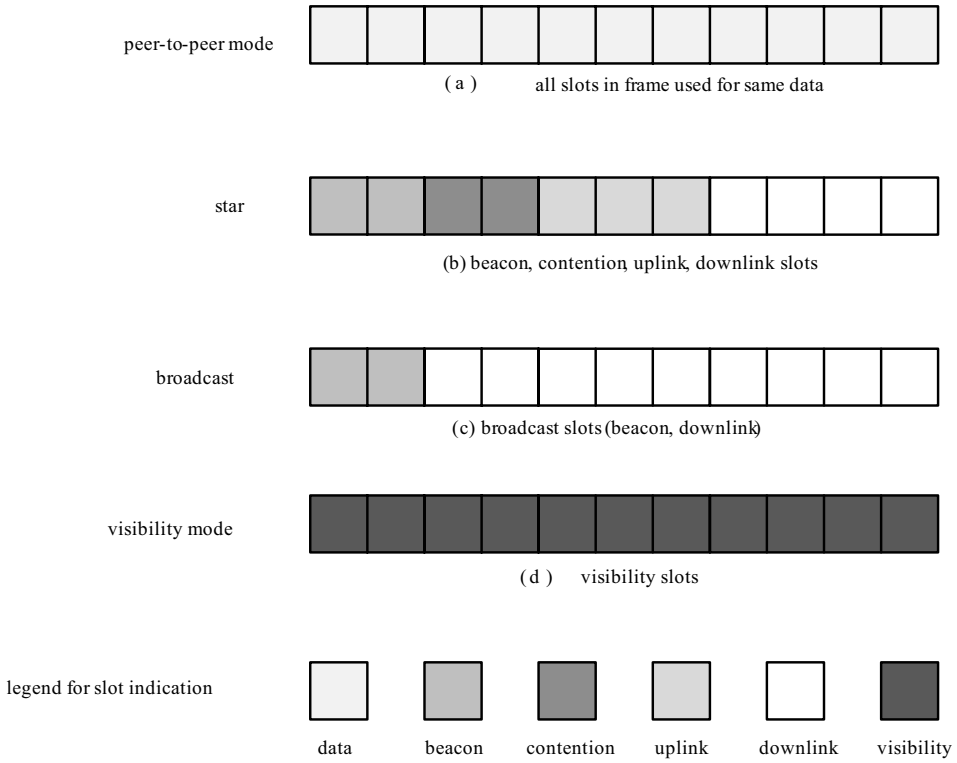


Figure 14—Example usage of frame structure for multiple topologies

Figure 14 provides an example usage of frame structure configuration for multiple topologies such as peer-to-peer, star, broadcast and visibility modes. The beacon slots are used for the beacons and the contention slots are used in the CAP period. The uplink and downlink GTS slots are used in the CFP periods. Visibility or idle patterns can be sent in the visibility slots during idle or RX modes of the infrastructure to ensure continuous output and mitigate flicker and are also used for point-and-shoot mode to ensure visibility.

5.1.1.1.1 Contention access period (CAP)

The CAP shall start immediately following the beacon and complete before the beginning of the CFP on a superframe slot boundary. If the CFP is zero length, the CAP shall complete at the end of the active portion of the superframe. The CAP shall be at least *aMinCAPLength* optical clocks, unless additional space is needed to temporarily accommodate the increase in the beacon frame length needed to perform GTS maintenance (see 5.2.2.1.3) and shall shrink or grow dynamically to accommodate the size of the CFP.

All frames, except acknowledgment frames and any data frame that quickly follows the acknowledgment of a data request command (see 5.1.7.3) transmitted in the CAP shall use a slotted random access mechanism to access the channel. A device transmitting within the CAP shall ensure that its transaction is complete (i.e., including the reception of any acknowledgment) one interframe space (IFS) period (see 5.1.1.2) before the end of the CAP. If this is not possible, the device shall defer its transmission until the CAP of the following superframe.

5.1.1.1.2 Contention-free period (CFP)

The CFP shall start on a slot boundary immediately following the CAP and it shall complete before the end of the active portion of the superframe. If any GTSs have been allocated by the coordinator, they shall be located within the CFP and occupy contiguous slots. The CFP shall therefore grow or shrink depending on the total length of all of the combined GTSs. Communication between devices can take a variable number of slots. A single device or user can have access to more than a single slot for sustained data transfer in the frame, if there are slots available.

No transmissions within the CFP shall use a unslotted random access mechanism to access the channel. A device transmitting in the CFP shall ensure that its transmissions are complete one IFS period (see 5.1.1.2) before the end of its GTS.

5.1.1.1.3 Visibility support during channel access

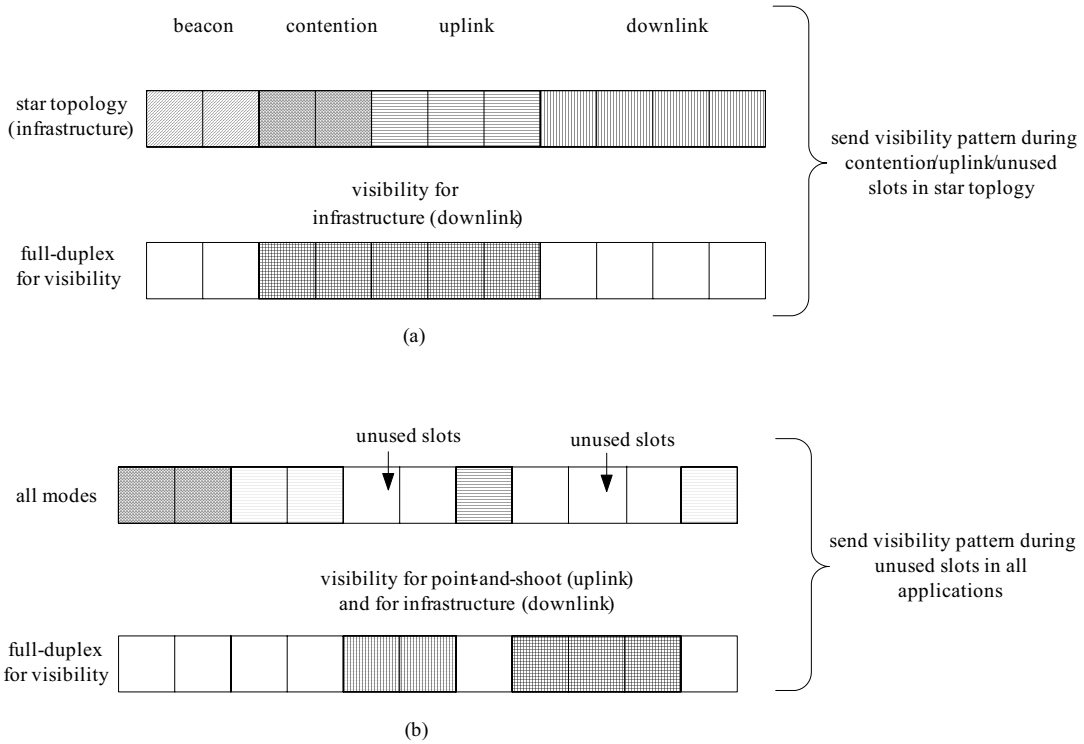


Figure 15—Usage of CVD frames during idle or RX modes of operation

The visibility slots can be used during contention and uplink slots in star topology mode and unused slots in all modes to maintain visibility, reduce flicker and keep the transmitter always “ON” for the infrastructure. This is shown in Figure 15. Visibility support is a very important distinguishing feature for VLC. One may need to transmit idle patterns during receive and idle modes. This can be done by simultaneous reception of data and the transmission of visibility or idle patterns. This is possible due to spatial separation of the light source and the receiving circuitry. As shown in Figure 15, idle patterns are sent during contention, uplink slots and unused downlink slots by the infrastructure to maintain visibility. Idle patterns are also sent during unused slots by the mobile device to help with pointing and alignment for optimal data transfer.

If the continuous visibility bit is set in the capabilities field shown in Table 16, then infrastructure devices shall provide continuous visibility.

5.1.1.2 Interframe spacing (IFS)

The MAC sublayer needs a finite amount of time to process data received by the PHY. To allow for this, two successive frames transmitted from a device shall be separated by at least an IFS period; if the first transmission requires an acknowledgment, the separation between the acknowledgment frame and the second transmission shall be at least an IFS period. The length of the IFS period is dependent on the size of the frame that has just been transmitted. Frames (i.e., MPDUs) of up to *aMaxSIFSFrameSize* octets in length shall be followed by a SIFS period of a duration of at least *macMinSIFSPeriod* optical clocks. Frames (i.e., MPDUs) with lengths greater than *aMaxSIFSFrameSize* octets shall be followed by a long interframe space (LIFS) period of a duration of at least *macMinLIFSPeriod* optical clocks. Burst frames shall have an RIFS of exactly *macMinRIFSPeriod*. The IFS for the different modes are defined in 8.3.4 and the concepts are illustrated in Figure 16.

The slotted random access algorithm shall take this requirement into account for transmissions in the CAP.

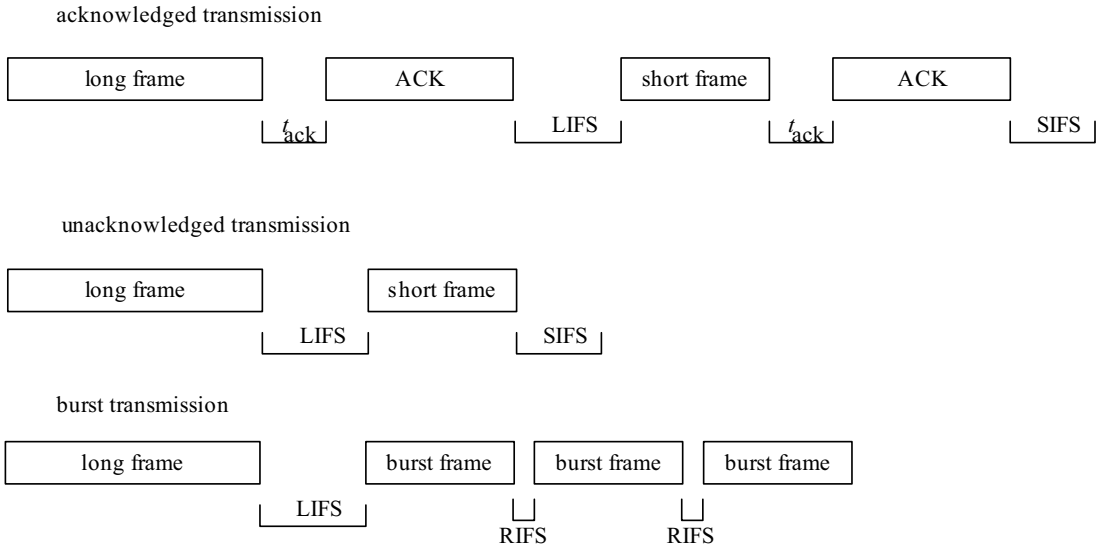


Figure 16—Interframe spacing

5.1.1.3 Random access algorithm

The slotted random access algorithm shall be used before the transmission of data or MAC command frames transmitted within the CAP, unless the frame can be quickly transmitted following the acknowledgment of a data request command (as defined in 5.1.7.3 for timing requirements). None of the random access algorithms shall be used for the transmission of beacon frames in a beacon-enabled VPAN, acknowledgment frames, or data frames transmitted in the CFP.

If periodic beacons are being used in the VPAN, the MAC sublayer shall employ the slotted version of the random access algorithm for transmissions in the CAP of the superframe. Conversely, if periodic beacons are not being used in the VPAN or if a beacon could not be located in a beacon-enabled VPAN, the MAC sublayer shall transmit using the unslotted version of the random access algorithm. In both cases, the algorithm is implemented using units of time called backoff periods, where one backoff period shall be equal to *aUnitBackoffPeriod* optical clocks.

In slotted random access, the backoff period boundaries of every device in the VPAN shall be aligned with the superframe slot boundaries of the coordinator, i.e., the start of the first backoff period of each device is aligned with the start of the beacon transmission. In slotted random access, the MAC sublayer shall ensure that the PHY commences all of its transmissions on the boundary of a backoff period. In unslotted random access, the backoff periods of one device are not related in time to the backoff periods of any other device in the VPAN.

Each device shall maintain two variables for each transmission attempt: *NB* and *BE*. *NB* is the number of times the access algorithm was required to backoff while attempting the current transmission; this value shall be initialized to zero before each new transmission attempt. The variable *BE* is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to access/assess a channel. *BE* shall be initialized to the value of *macMinBE*.

Figure 17 illustrates the steps of the access algorithm. The MAC sublayer shall first initialize NB and BE for slotted random access then locate the boundary of the next backoff period. The MAC sublayer shall delay for a random number of complete backoff periods in the range 0 to $2^{BE} - 1$ [step (2)] and then request that

the PHY perform a transmission or optionally a CCA. In a slotted random access system, the transmission, or CCA if active, shall start on a backoff period boundary. In an unslotted system, the transmission, or CCA if active, shall start immediately.

In a slotted random access system, the MAC sublayer shall ensure that, after the random backoff, the remaining slotted random access operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. Note that any bit padding used by the supported PHY shall be considered in making this determination. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer shall pause the backoff countdown at the end of the CAP and resume it at the start of the CAP in the next superframe. If the number of backoff periods is less than or equal to the remaining number of backoff periods in the CAP, the MAC sublayer shall apply its backoff delay and then evaluate whether it can proceed. The MAC sublayer shall proceed if the remaining unslotted random access algorithm steps, the frame transmission, and any acknowledgment can be completed before the end of the CAP. If the MAC sublayer can proceed and CCA is active, it shall request that the PHY perform the CCA in the current superframe. If the MAC sublayer cannot proceed, it shall wait until the start of the CAP in the next superframe and apply a further random backoff delay before evaluating whether it can proceed again.

If CCA is active and the channel is assessed to be busy, the MAC sublayer shall increment both *NB* and *BE* by one, ensuring that *BE* shall be no more than *macMaxBE*. If the value of *NB* is less than or equal to *macMaxRABackoffs*, the access algorithm shall return to perform a random backoff as shown in Figure 17. If the value of *NB* is greater than *macMaxRABackoffs*, the access algorithm shall terminate with a channel access failure status.

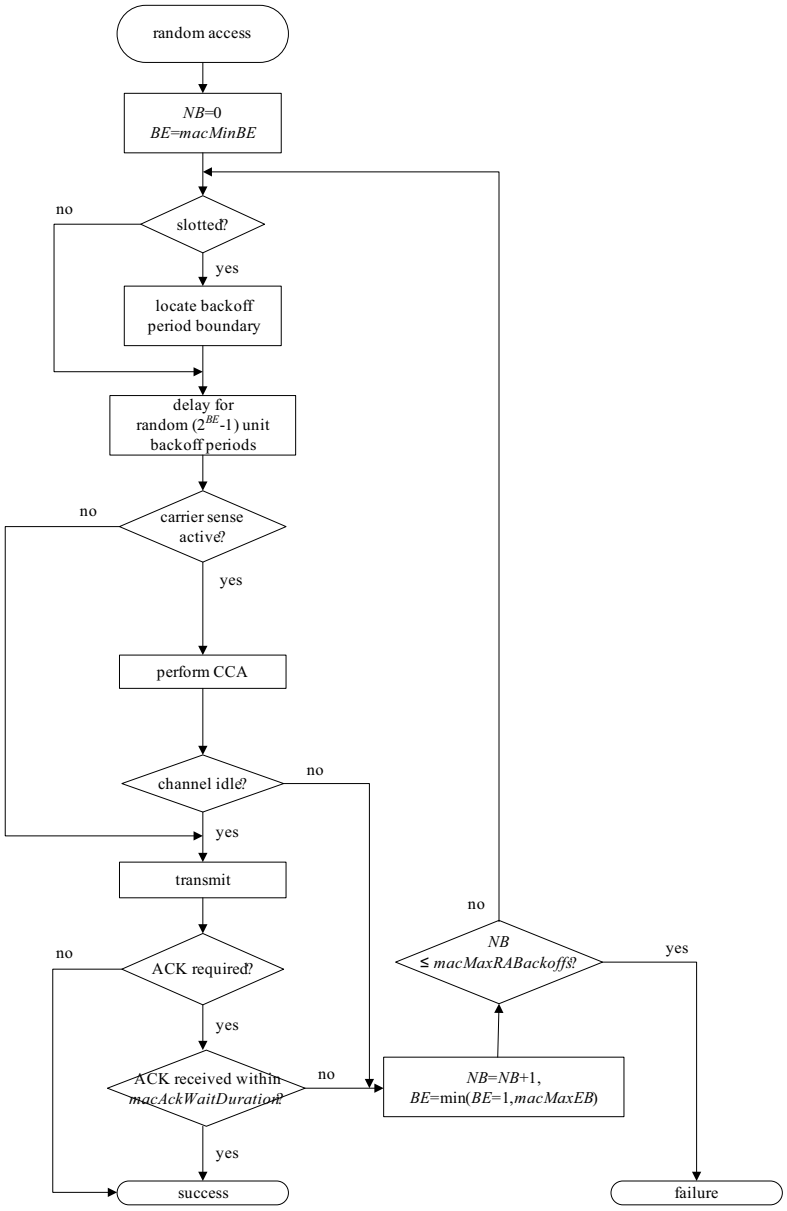


Figure 17—Random access flowchart

5.1.2 Starting a VPAN

5.1.2.1 Scanning through channels

All devices shall be capable of performing passive scans across a specified list of channels. In addition, a coordinator shall be able to perform active scans. A device is instructed to begin a channel scan through the MLME-SCAN.request primitive. For the duration of the scan, the device shall suspend beacon transmissions, if applicable, and shall only accept frames received over the PHY data service that are relevant to the scan being performed. Upon the conclusion of the scan, the device shall recommence beacon transmissions. The results of the scan shall be returned via the MLME-SCAN.confirm primitive.

5.1.2.1.1 Active channel scan

It is anticipated that the active channel scan is used with the peer-to-peer topology.

An active scan allows a device to locate any coordinator transmitting beacon frames within its coverage area. This could be used by a prospective VPAN coordinator to select a VPAN identifier prior to starting a new VPAN, or it could be used by a device prior to association.

During an active scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames. If a beacon frame is received that contains the address of the scanning device in its list of pending addresses, the scanning device shall not attempt to extract the pending data.

Before commencing an active scan, the MAC sublayer shall store the value of *macVPANId* and then set it to 0xffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current VPAN, see 5.1.7.2. On completion of the scan, the MAC sublayer shall restore the value of *macVPANId* to the value stored before the scan began. An active scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate an active scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and send a beacon request command, see 5.3.6. Upon successful transmission of the beacon request command, the device shall enable its receiver for at most [*aBaseSuperframeDuration* × (2*n* + 1)] optical clocks, 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 VPAN descriptor structure, see 6.3.3.1, including the channel information and the preamble code. If a beacon frame is received when *macAutoRequest* is set to TRUE, the list of VPAN descriptor structures shall be stored by the MAC sublayer until the scan is complete; at this time, the list shall be sent to the next higher layer in the VPANDescriptorList parameter of the MLME-SCAN.confirm primitive. A device shall be able to store between one and an implementation-specified maximum number of VPAN descriptors. A beacon frame shall be assumed to be unique if it contains both a VPAN identifier and a source address that has not been seen before during the scan of the current channel. If a beacon frame is received when *macAutoRequest* is set to FALSE, each recorded VPAN descriptor is sent to the next higher layer in a separate MLME-BEACON-NOTIFY. indication primitive. A received beacon frame containing one or more octets of payload shall also cause the VPAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY.indication primitive. If a protected beacon frame is received, i.e., the Security Enabled subfield in the frame control field is set to one, the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.2.3. The security-related elements of the VPAN descriptor corresponding to the beacon, see 6.3.3.1, shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the VPAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise. The information from the unsecured frame shall be recorded in the VPAN descriptor even if the status from the unsecuring process indicated an error. If a coordinator of a beacon-enabled VPAN receives the beacon request command, it shall ignore the command and continue transmitting its periodic beacons as usual. If a coordinator of a nonbeacon-enabled VPAN receives this command, it shall transmit a single beacon frame using unslotted random access or unslotted CSMA-CA.

If *macAutoRequest* is set to TRUE, the active scan on a particular channel shall terminate when the number of beacons found equals the implementation-specified limit or the channel has been scanned for the full time, as specified in 5.1.2.1.1. If *macAutoRequest* is set to FALSE, the active scan on a particular channel shall terminate when the channel has been scanned for the full time. If a channel was not scanned for the full time, it shall be considered to be unscanned.

If *macAutoRequest* is set to TRUE, the entire scan procedure shall terminate when the number of VPAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned. If *macAutoRequest* is set to FALSE, the entire scan procedure shall only terminate when every channel in the set of available channels has been scanned.

5.1.2.1.2 Passive channel scan

It is anticipated that the passive channel scan is used with the star or broadcast topology.

A passive scan, like an active scan, allows a device to locate any coordinator transmitting beacon frames within its coverage area. The beacon request command, however, is not transmitted. This type of scan could be used by a device prior to association. During a passive scan, the MAC sublayer shall discard all frames received over the PHY data service that are not beacon frames. If a beacon frame is received that contains the address of the scanning device in its list of pending addresses, the scanning device shall not attempt to extract the pending data.

Before commencing a passive scan, the MAC sublayer shall store the value of *macVPANId* and then set it to 0xffff for the duration of the scan. This enables the receive filter to accept all beacons rather than just the beacons from its current VPAN, see 5.1.7.2. On completion of the scan, the MAC sublayer shall restore the value of *macVPANId* to the value stored before the scan began. A passive scan over a specified set of logical channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to indicate a passive scan. For each logical channel, the device shall first switch to the channel, by setting *phyCurrentChannel* accordingly, and then enable its receiver for at most [*aBaseSuperframeDuration* × (2*n* + 1)] optical clocks, 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 VPAN descriptor structure, see 6.3.3.1. If a beacon frame is received when *macAutoRequest* is set to TRUE, the list of VPAN descriptor structures shall be stored by the MAC sublayer until the scan is complete; at this time, the list shall be sent to the next higher layer in the VPANDescriptorList parameter of the MLME-SCAN.confirm primitive. A device shall be able to store between one and an implementation- specified maximum number of VPAN descriptors. A beacon frame shall be assumed to be unique if it contains both a VPAN identifier and a source address that has not been seen before during the scan of the current channel. If a beacon frame is received when *macAutoRequest* is set to FALSE, each recorded VPAN descriptor is sent to the next higher layer in a separate MLME-BEACON-NOTIFY. indication primitive. Once the scan is complete, the MLME-SCAN.confirm shall be issued to the next higher layer with a null VPANDescriptorList. A received beacon frame containing one or more octets of payload shall also cause the VPAN descriptor to be sent to the next higher layer via the MLME-BEACON-NOTIFY. indication primitive.

If a protected beacon frame is received (i.e., the Security Enabled subfield in the frame control field is set to one), the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.2.3.

The security-related elements of the VPAN descriptor corresponding to the beacon, as shown in 6.3.3.1, shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the VPAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

The information from the unsecured frame shall be recorded in the VPAN descriptor even if the status from the unsecuring process indicated an error.

If *macAutoRequest* is set to TRUE, the passive scan on a particular channel shall terminate when the number of beacons found equals the implementation specified limit or the channel has been scanned for the full time. If *macAutoRequest* is set to FALSE, the passive scan on a particular channel shall terminate when the channel has been scanned for the full time. If a channel was not scanned for the full time, it shall be considered to be unscanned.

If *macAutoRequest* is set to TRUE, the entire scan procedure shall terminate when the number of VPAN descriptors stored equals the implementation-specified maximum or every channel in the set of available channels has been scanned. If *macAutoRequest* is set to FALSE, the entire scan procedure shall terminate only when every channel in the set of available channels has been scanned.

5.1.2.2 VPAN initiation

The broadcast mode does not have any requirements for starting a VPAN. Capability exchange should occur for all bi-directional communication during device discovery. If a device supports multiple transmit color channels, it can exchange the WQI metrics for channel selection. There is no channel selection process requirement if the device supports only a single color channel. For a star topology, the coordinator establishes the VPAN by sending beacon frames. For peer-to-peer topology, a device can either send an association or active scan command to initiate communication with the peer device.

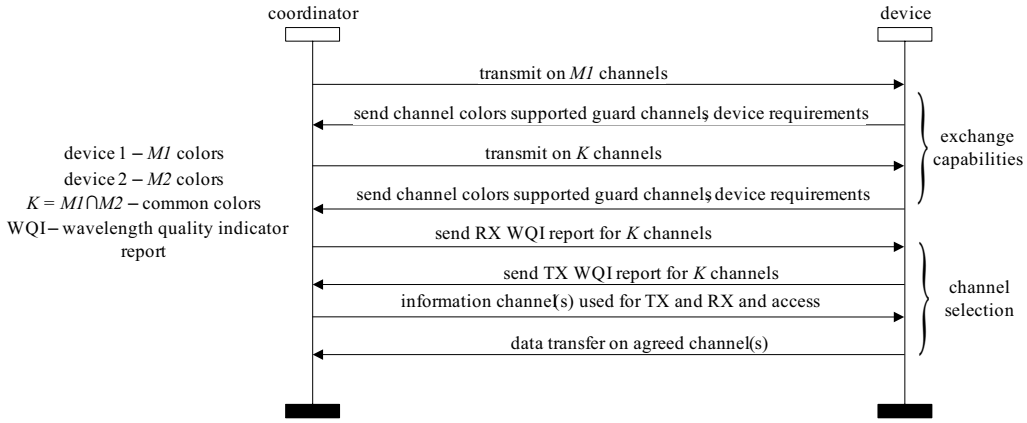


Figure 18—VPAN initiation

Let device 1 support *M1* color channels and let device 2 support *M2* color channels. Let *K* be the number of channels shared between device 1 and device 2, where $K \geq 1$ for communication. For a peer-to-peer network, the first device, which may be the device or coordinator, initiates the communications and transmits on all supported *M1* channels. If there is independent hardware for each color at the transmitter and receiver, parallel transmissions are possible as long as guard color channels are not used for any particular color choice. Each device communicates the capabilities of each device and application requirements via the MAC and PHY capabilities information element (IE) provided. The MAC also reports the number of supported aggregated channels and the associated guard colors for each channel. Next, the other device attempts to receive and synchronize on all *K* channels shared between the devices. However, it may be able to receive on only '*x*' channels, where $1 \leq x \leq K$, due to interference with other light sources. The second device shall receive on at least one channel in order to communicate. The *K* channels and device capabilities are obtained from the mentioned information. Based on the interference energy from ambient light and the energy received during transmission, a WQI is calculated for all *K* channels. The second device then transmits on all *K* common channels to the first device. The second device also provides its supported channels, guard channels and application requirements as part of its capabilities information exchange. Next, the first device attempts to receive and synchronize on all *K* channels. It may receive on only '*y*' channels, where $1 \leq y \leq K$, due to interference. Since VLC is very directional, it is possible that '*x*' and '*y*' may be different. For example, if first device is closer to a window, it may receive more ambient light interference than the second device. The first device calculates its RX WQI for all *K* channels as well and transmits the WQI report back to the second device

Simultaneously, the second device calculates the WQI metrics based on the received information from the first device. Channels where reception is not possible or where other piconets are known to operate by the second device will be tagged unusable with a reception WQI of 0. The second device then reports this RX WQI for all *K* channels back to the first device

The initiating device collects the information for the transmission such as the transmission and reception capabilities of the two devices, the WQI reports, the selected guard color channels for each channel and the requirements of the application. Based on this information, the first device determines a single or multiple channels for communication. The first device then reports the communication channels to the second device. Thus, at the end of this exchange, both devices have an estimate of the WQI for their transmissions that is most suitable for reception at the other end. From that point, both devices can communicate on the agreed channel or channels.

The support for WQI (wavelength quality indication) is provided in the PHY and shall be passed to the MAC via the PD-SAP interface as shown in Table 97.

For a star topology network, the coordinator acts as the initiator for device discovery and association and uses the CAP for association requests and the beacon/management frames to broadcast its association grants.

Starting a VPAN is only applicable to bi-directional communication modes and not for broadcasting.

5.1.2.3 Beacon generation

A device shall be permitted to transmit beacon frames only if *macShortAddress* is not equal to 0xffff.

A coordinator shall use the MLME-START.request primitive to begin transmitting beacons only if the BeaconOrder parameter is less than 15. The coordinator may begin beacon transmission either as the coordinator of a new VPAN or as a device on a previously established VPAN, depending upon the setting of the VPANCoordinator parameter, as shown in 6.3.11.1. The coordinator shall begin beacon transmission on a previously established VPAN only once it has successfully associated with that VPAN.

For the coordinator (i.e., the VPANCoordinator parameter is set to TRUE), the MAC sublayer shall ignore the StartTime parameter and begin beacon transmissions immediately. Setting the StartTime parameter to zero shall also cause the MAC sublayer to begin beacon transmissions immediately. If not acting as the coordinator and the StartTime parameter is nonzero, the time to begin beacon transmissions shall be calculated using the following method. The StartTime parameter, which is rounded to a backoff slot boundary, shall be added to the time, obtained from the local clock, when the MAC sublayer receives the beacon of the coordinator through which it is associated. The MAC sublayer shall then begin beacon transmissions when the current time, obtained from the local clock, equals the number of calculated optical clocks. In order for the beacon transmission time to be calculated by the MAC sublayer, the MAC sublayer shall first track the beacon of the coordinator through which it is associated. If the MLME-START.request primitive is issued with a nonzero StartTime parameter and the MAC sublayer is not currently tracking the beacon of its coordinator, the MLME shall not begin beacon transmissions but shall instead issue the MLME-START.confirm primitive with a status of TRACKING_OFF.

If a device misses between one and (*aMaxLostBeacons*–1) consecutive beacon frames from its coordinator, the device shall continue to transmit its own beacons based on both *macBeaconOrder* (see 5.1.3.5) and its local clock. If the device then receives a beacon frame from its coordinator and, therefore, does not lose synchronization, the device shall resume transmitting its own beacons based on the StartTime parameter and the incoming beacon. If a device does lose synchronization with its coordinator, the MLME of the device shall issue the MLME-SYNC-LOSS.indication primitive to the next higher layer and immediately stop transmitting its own beacons. The next higher layer may, at any time following the reception of the MLME-SYNC-LOSS.indication primitive, resume beacon transmissions by issuing a new MLME-START.request primitive.

On receipt of the MLME-START.request primitive, the MAC sublayer shall set the VPAN identifier in *macVPANId* and use this value in the Source VPAN Identifier field of the beacon frame. The address used in

the Source Address field of the beacon frame shall contain the value of *aExtendedAddress* if *macShortAddress* is equal to 0xfffe or *macShortAddress* otherwise.

The time of transmission of the most recent beacon shall be recorded in *macBeaconTxTime* and shall be computed so that its value is taken at the same position in each beacon frame, the location of which is implementation specific. The position, which is specified by the *macTimeStampOffset* attribute, is the same as that used in the timestamp of the incoming beacon frame, as described in 5.1.5.1.

All beacon frames shall be transmitted at the beginning of each superframe at an interval equal to *aBase-SuperframeDuration* × 2^n optical clocks, where *n* is the value of *macBeaconOrder* (the construction of the beacon frame is specified in 5.2.2.1).

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

5.1.2.4 Device discovery

The coordinator indicates its presence on a VPAN to other devices by transmitting beacon frames. This allows the other devices to perform device discovery.

Device discovery shall be performed at 11.67 Kbps with a 200 KHz optical clock for PHY I and at 1.25 Mbps with a 3.75 MHz optical clock for PHY II. PHY III does not provide device discovery support and shall rely on device discovery using PHY II before operating in that mode. The dimmed OOK mode can be used to support dimming in the device discovery process. This mode is indicated using the MAC PIB attribute, *macUseDimmedOOKmode*, as defined in Table 60. The MAC and PHY capabilities are exchanged in the device discovery process. The clock rate support capabilities are also exchanged. Once the capabilities are exchanged, regular data transmission mode resumes for all three PHY types. Device discovery requires bi-directional communication and is not applicable for broadcasting.

5.1.2.5 Guard and aggregation color channels

The bandplan provides support for seven logical channels in the MAC. However, in order to support association without knowledge of receiver capabilities and to support unidirectional broadcasting, the VLC receiver shall support reception on the entire visible light spectrum with any type of optical light source.

Channel aggregation is used to indicate optical sources that span multiple (>1) bands in the proposed bandplan and are intentionally transmitting on multiple bands due to the choice of optical light source. Guard channels are used to indicate optical sources that unintentionally leak into other bands, whose information can be discarded at the receiver for better performance.

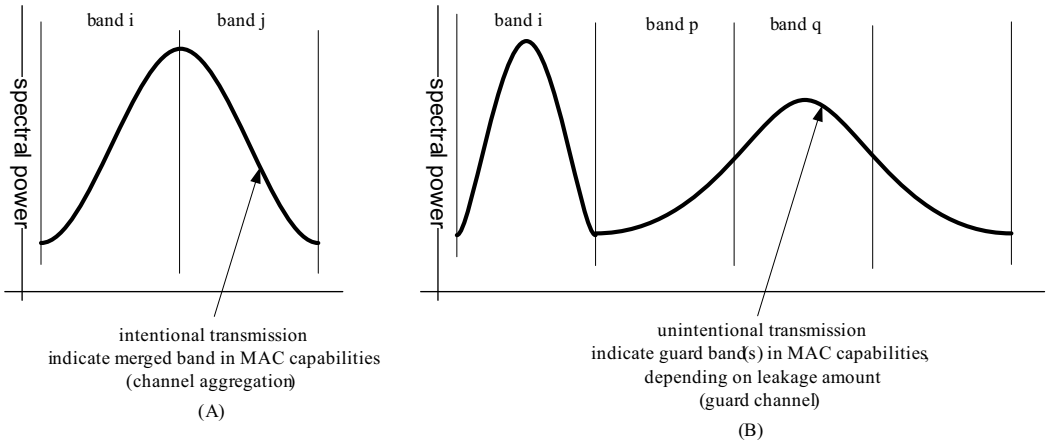


Figure 19—Concept of aggregation channel and guard channel

If multiple bands are aggregated or multiple optical sources are transmitting simultaneously, the same data shall be sent on all optical sources during the preamble and header during device discovery since it is not known what the receiver capabilities are. The details on channel aggregation and guard channel support are provided in the PHY capabilities information element of the MAC. The criterion used for defining a guard color channel or aggregated channel is based on out-of-band leakage exceeding 20 dB over maximum in-channel value. The transmitting device shall indicate channel aggregation and guard channel support using the PHY capabilities during device discovery and association for bi-directional communication modes.

5.1.3 Maintaining VPANs

In some instances a situation could occur in which two VPANs exist in the same operating space with the same VPAN identifier. If this conflict happens, the coordinator and its devices shall perform the VPAN identifier conflict resolution procedure.

This procedure is optional for a device.

5.1.3.1 Detection

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

— A beacon frame is received by the VPAN coordinator with the VPAN Coordinator subfield, see 5.2.2.1.2, set to one and the VPAN identifier equal to *macVPANId*.
— A VPAN ID conflict notification command, see 5.3.5, is received by the VPAN coordinator from an associated device on its VPAN.

A device that is associated through the VPAN coordinator (i.e., *macAssociatedVPANCoord* is set to TRUE) shall conclude that a VPAN identifier conflict is present if the following applies:

— A beacon frame is received by the device with the VPAN Coordinator subfield set to one, the VPAN identifier equal to *macVPANId*, and an address that is equal to neither *macCoordShortAddress* nor *macCoordExtendedAddress*.

5.1.3.2 Resolution

On the detection of a VPAN identifier conflict by a device, it shall generate the VPAN ID conflict notification command, defined in 5.3.5, and send it to its coordinator. Since the VPAN ID conflict notification command contains an acknowledgment request (see 5.3.3.1), the coordinator shall confirm its receipt by sending an acknowledgment frame. Once the device has received the acknowledgment frame from the coordinator, the MLME shall issue an MLME-SYNC-LOSS.indication primitive with the LossReason parameter set to VPAN_ID_CONFLICT. If the device does not receive an acknowledgment frame, the MLME shall not inform the next higher layer of the VPAN identifier conflict.

On the detection of a VPAN identifier conflict by the coordinator, the MLME shall issue an MLME-SYNC-LOSS.indication to the next higher layer with the LossReason parameter set to VPAN_ID_CONFLICT. The next higher layer of the coordinator may then perform an active scan and, using the information from the scan, select a new VPAN identifier. The algorithm for selecting a suitable VPAN identifier is out of the scope of this standard. If the next higher layer does select a new VPAN identifier, it may then issue an MLME-START.request with the CoordRealignment parameter set to TRUE in order to realign the VPAN, as described in 5.1.3.3.

5.1.3.3 Realigning a VPAN

If a coordinator receives the MLME-START.request primitive (see 6.3.11.1) with the CoordRealignment parameter set to TRUE, the coordinator shall attempt to transmit a coordinator realignment command containing the new parameters for VPANId, LogicalChannel.

When the coordinator is already transmitting beacons and the CoordRealignment parameter is set to TRUE, the next scheduled beacon shall be transmitted on the current channel using the current superframe configuration, with the frame pending subfield of the frame control field set to one. Immediately following the transmission of the beacon, the coordinator realignment command shall also be transmitted on the current channel using unslotted random access.

When the coordinator is not already transmitting beacons and the CoordRealignment parameter is set to TRUE, the coordinator realignment command shall be transmitted immediately on the current channel using unslotted random access.

If the transmission of the coordinator realignment command fails due to a channel access failure, the MLME shall notify the next higher layer by issuing the MLME-START.confirm primitive with a status of CHANNEL_ACCESS_FAILURE. The next higher layer may then choose to issue the MLME-START.request primitive again.

Upon successful transmission of the coordinator realignment command, the new superframe configuration and channel parameters shall be put into operation as described in 5.1.3.5 at the subsequent scheduled beacon, or immediately if the coordinator is not already transmitting beacons, and the MAC sublayer shall issue the MLME-START.confirm primitive with a status of SUCCESS.

5.1.3.4 Realignment in a VPAN

If a device has received the coordinator realignment command (see 5.3.7) from the coordinator through which it is associated, the MLME shall issue the MLME-SYNC-LOSS.indication primitive with the LossReason parameter set to REALIGNMENT and the VPANId, LogicalChannel, and the security-related parameters set to the respective fields in the coordinator realignment command. The next higher layer of a coordinator may then issue an MLME-START.request primitive with the CoordRealignment parameter set to TRUE. The next higher layer of a device that is not a coordinator may instead change the superframe configuration or channel parameters through use of the MLME-SET.request primitive.

5.1.3.5 Updating superframe configuration and channel PIB attributes

To update the superframe configuration and channel attributes, the MLME shall assign values from the MLME-START.request primitive parameters to the appropriate PIB attributes. The MLME shall set *macBeaconOrder* to the value of the BeaconOrder parameter. If *macBeaconOrder* is equal to 15, the MLME will also set *macSuperframeOrder* to 15. In this case, this primitive configures a nonbeacon-enabled VPAN. If *macBeaconOrder* is less than 15, the MAC sublayer will set *macSuperframeOrder* to the value of the SuperframeOrder parameter. The MAC sublayer shall also update *macVPANId* with the value of the VPANId parameter and update *phyCurrentChannel* with the values of the LogicalChannel parameters by issuing the PLME-SET.request primitive.

5.1.4 Association and disassociation

This subclause specifies the procedures for association and disassociation.

5.1.4.1 Association

A device shall attempt to associate only after having first performed a MAC sublayer reset, by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE, and then having completed either an active channel scan, see 5.1.2.1.1, or a passive channel scan as shown in 5.1.2.1.2. The results of the channel scan would have then been used to choose a suitable VPAN. The algorithm for selecting a suitable VPAN with which to associate from the list of VPAN descriptors returned from the channel scan procedure is out of the scope of this standard.

Following the selection of a VPAN with which to associate, the next higher layers shall request through the MLME-ASSOCIATE.request primitive that the MLME configures the following PHY and MAC PIB attributes to the values necessary for association:

— *phyCurrentChannel* shall be set equal to the LogicalChannel parameter of the MLME-ASSOCIATE.request primitive.
— *macVPANId* shall be set equal to the CoordVPANId parameter of the MLME-ASSOCIATE.request primitive.
— *macCoordExtendedAddress* or *macCoordShortAddress*, depending on which is known from the beacon frame from the coordinator through which it wishes to associate, shall be set equal to the CoordAddress parameter of the MLME-ASSOCIATE.request primitive.

A coordinator shall allow association only if *macAssociationPermit* is set to TRUE. Similarly, a device should attempt to associate only with a VPAN through a coordinator that is currently allowing association, as indicated in the results of the scanning procedure. If a coordinator with *macAssociationPermit* set to FALSE receives an association request command from a device, the command shall be ignored.

In order to optimize the association procedure on a beacon-enabled VPAN, a device may begin tracking the beacon of the coordinator through which it wishes to associate. This is achieved by the next higher layer issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE.

A device that is instructed to associate with a VPAN, through the MLME-ASSOCIATE.request primitive, shall try to associate only with an existing VPAN and shall not attempt to start its own VPAN.

The MAC sublayer of an unassociated device shall initiate the association procedure by sending an association request command, see 5.3.1, to the coordinator of an existing VPAN; if the association request command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer. Because the association request command contains an acknowledgment request (see 5.3.1.1), the coordinator shall confirm its receipt by sending an acknowledgment frame.

The acknowledgment to an association request command does not mean that the device has associated. The next higher layer of the coordinator needs time to determine whether the current resources available on the VPAN are sufficient to allow another device to associate. The next higher layer should make this decision within *macResponseWaitTime* optical clocks. If the next higher layer of the coordinator finds that the device was previously associated on its VPAN, all previously obtained device-specific information should be removed. If sufficient resources are available, the next higher layer should allocate a 16-bit short address to the device, and the MAC sublayer shall generate an association response command, see 5.3.2, containing the new address and a status indicating a successful association. If sufficient resources are not available, the next higher layer of the coordinator should inform the MAC sublayer, and the MLME shall generate an association response command containing a status indicating a failure as shown in Table 11. The association response command shall be sent to the device requesting association using indirect transmission, i.e., the association response command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 5.1.7.3.

If the allocate address subfield of the capability information field (see 5.3.19.1.1) of the association request command is set to one, the next higher layer of the coordinator shall allocate a 16-bit address with a range depending on the addressing mode supported by the coordinator, as described in Table 2. If the Allocate Address subfield of the association request command is set to zero, the 16-bit short address shall be equal to 0xfffe. A short address of 0xfffe is a special case that indicates that the device has associated, but has not been allocated a short address by the coordinator. In this case, the device shall use only its 64-bit extended address to operate on the network.

On receipt of the acknowledgment to the association request command, the device shall wait for at most *macResponseWaitTime* optical clocks for the coordinator to make its association decision; the PIB attribute *macResponseWaitTime* is a network-topology-dependent parameter and may be set to match the specific requirements of the network that a device is trying to join. If the device is tracking the beacon, it shall attempt to extract the association response command from the coordinator whenever it is indicated in the beacon frame. If the device is not tracking the beacon, it shall attempt to extract the association response command from the coordinator after *macResponseWaitTime* optical clocks. If the device does not extract an association response command frame from the coordinator within *macResponseWaitTime* optical clocks, the MLME shall issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA, and the association attempt shall be deemed a failure. In this case, the next higher layer shall terminate any tracking of the beacon. This is achieved by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to FALSE.

The MLME-ASSOCIATE.response and the subsequent Association response (see 5.3.2) also contain information about what capabilities the device and the coordinator will and will not use during future communication.

Because the association response command contains an acknowledgment request (see 5.3.2.1), the device requesting association shall confirm its receipt by sending an acknowledgment frame. If the Association Status field of the command indicates that the association was successful, the device shall store the address contained in the 16-bit Short Address field of the command in *macShortAddress*; communication on the VPAN using this short address shall depend on its range, as described in Table 2. If the original beacon selected for association following a scan contained the short address of the coordinator, the extended address of the coordinator, contained in the MHR of the association response command frame, shall be stored in *macCoordExtendedAddress.*

If the Association Status field of the command indicates that the association was unsuccessful, the device shall set *macVPANId* to the default value (0xffff).

Table 2—Usage of the 16-bit short address

Value of *macShortAddress*	Description
0x0000–0xfffd	If a source address is included, the device shall use short source addressing mode for beacon and data frames and the appropriate source addressing mode specified in 5.3 for MAC command frames.
0xfffe	If a source address is included, the device shall use extended source addressing mode for beacon and data frames and the appropriate source addressing mode specified in 5.3 for MAC command frames.
0xffff	The device is not associated and, therefore, shall not perform any data frame communication. The device shall use the appropriate source addressing mode specified in 5.3 for MAC command frames.

5.1.4.2 Disassociation

The disassociation procedure is initiated by the next higher layer by issuing the MLME-DISASSOCIATE.request primitive to the MLME.

When a coordinator wants one of its associated devices to leave the VPAN, the MLME of the coordinator shall send the disassociation notification command in the manner specified by the TxIndirect parameter of the MLME-DISASSOCIATE.request primitive previously sent by the next higher layer. If TxIndirect is TRUE, the MLME of the coordinator shall send the disassociation notification command to the device using indirect transmission, i.e., the disassociation notification command frame shall be added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 5.1.7.3. If the command frame is not successfully extracted by the device, the coordinator should consider the device disassociated. Otherwise, the MLME shall send the disassociation notification command to the device directly. In this case, if the disassociation notification command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer.

Because the disassociation command contains an acknowledgment request (see 5.3.3.1), the receiving device shall confirm its receipt by sending an acknowledgment frame. If the direct or indirect transmission fails, the coordinator should consider the device disassociated.

If an associated device wants to leave the VPAN, the MLME of the device shall send a disassociation notification command to its coordinator. If the disassociation notification command cannot be sent due to a channel access failure, the MAC sublayer shall notify the next higher layer. Because the disassociation command contains an acknowledgment request (see 5.3.3.1), the coordinator shall confirm its receipt by sending an acknowledgment frame. However, even if the acknowledgment is not received, the device should consider itself disassociated.

If the source address contained in the disassociation notification command is equal to *macCoord-ExtendedAddress*, the device should consider itself disassociated. If the command is received by a coordinator and the source is not equal to *macCoordExtendedAddress*, it shall verify that the source address corresponds to one of its associated devices; if so, the coordinator should consider the device disassociated. If none of the conditions for disassociation are satisfied, the command shall be ignored.

An associated device shall disassociate itself by removing all references to the VPAN; the MLME shall set *macVPANId*, *macShortAddress*, *macAssociatedVPANCoord*, *macCoordShortAddress* and *macCoordExtendedAddress* to the default values. The next higher layer of a coordinator should disassociate a device by removing all references to that device.

The next higher layer of the requesting device shall be notified of the result of the disassociation procedure through the MLME-DISASSOCIATE.confirm primitive.

5.1.5 Synchronization

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

5.1.5.1 Synchronization with beacons

All devices operating on a beacon-enabled VPAN (i.e., *macBeaconOrder* < 15) shall be able to acquire beacon synchronization in order to detect any pending messages or to track the beacon. Devices shall be permitted to acquire beacon synchronization only with beacons containing the VPAN identifier specified in *macVPANId*. If *macVPANId* specifies the broadcast VPAN identifier (0xffff), a device shall not attempt to acquire beacon synchronization.

A device is instructed to attempt to acquire the beacon through the MLME-SYNC.request primitive. If tracking is specified in the MLME-SYNC.request primitive, the device shall attempt to acquire the beacon and keep track of it by regular and timely activation of its receiver. If tracking is not specified, the device shall either attempt to acquire the beacon only once or terminate the tracking after the next beacon if tracking was enabled through a previous request.

To acquire beacon synchronization, a device shall enable its receiver and search for at most [*aBaseSuperframeDuration* × (2^n + 1)] optical clocks, where *n* is the value of *macBeaconOrder*. If a beacon frame containing the current VPAN identifier of the device is not received, the MLME shall repeat this search. Once the number of missed beacons reaches *aMaxLostBeacons*, the MLME shall notify the next higher layer by issuing the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOSS.

The MLME shall timestamp each received beacon frame at the same symbol boundary within each frame, the location of which is described by the *macTimeStampOffset* attribute. The position shall be the same as that used in the timestamp of the outgoing beacon frame, stored in *macBeaconTxTime*. The timestamp value shall be that of the local clock of the device at this position. The timestamp is intended to be a relative time measurement that may or may not be made absolute, at the discretion of the implementer.

If a protected beacon frame is received (i.e., the Security Enabled subfield in the frame control field is set to one), the device shall attempt to unsecure the beacon frame using the unsecuring process described in 7.2.3.

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

The security-related elements of the VPAN descriptor corresponding to the beacon (see Table 38) shall be set to the corresponding parameters returned by the unsecuring process. The SecurityFailure element of the VPAN descriptor shall be set to SUCCESS if the status from the unsecuring process is SUCCESS and set to one of the other status codes indicating an error in the security processing otherwise.

If a beacon frame is received, the MLME shall discard the beacon frame if the Source Address and the Source VPAN Identifier fields of the MHR of the beacon frame do not match the coordinator source address (*macCoordShortAddress* or *macCoordExtendedAddress*, depending on the addressing mode) and the identifier of the device (*macVPANId*).

If a valid beacon frame is received and *macAutoRequest* is set to FALSE, the MLME shall indicate the beacon parameters to the next higher layer by issuing the MLME-BEACON-NOTIFY.indication primitive. If a beacon frame is received and *macAutoRequest* is set to TRUE, the MLME shall first issue the MLME-

BEACON-NOTIFY.indication primitive if the beacon contains any payload. The MLME shall then compare its address with those addresses in the Address List field of the beacon frame. If the Address List field contains the 16-bit short or 64-bit extended address of the device and the source VPAN identifier matches *macVPANId*, the MLME shall follow the procedure for extracting pending data from the coordinator as shown in 5.1.7.3.

If beacon tracking is activated, the MLME shall enable its receiver at a time prior to the next expected beacon frame transmission, i.e., just before the known start of the next superframe. If the number of consecutive beacons missed by the MLME reaches *aMaxLostBeacons*, the MLME shall respond with the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

5.1.5.2 Synchronization without beacons

All devices operating on a nonbeacon-enabled VPAN (*macBeaconOrder* = 15) shall be able to poll the coordinator for data at the discretion of the next higher layer.

A device is instructed to poll the coordinator when the MLME receives the MLME-POLL.request primitive. On receipt of this primitive, the MLME shall follow the procedure for extracting pending data from the coordinator as shown in 5.1.7.3.

5.1.6 Transaction handling

Transactions can be instigated from the devices themselves rather than from the coordinator. In other words, either the coordinator needs to indicate in its beacon when messages are pending for devices or the devices themselves need to poll the coordinator to determine whether they have any messages pending. Such transfers are called indirect transmissions.

The coordinator shall begin handling a transaction on receipt of an indirect transmission request either via the MCPS-DATA.request primitive or via a request from the MLME to send a MAC command instigated by a primitive from the next higher layer, such as the MLME-ASSOCIATE.response primitive as shown in 6.3.1.3. On completion of the transaction, the MAC sublayer shall indicate a status value to the next higher layer. If a request primitive instigated the indirect transmission, the corresponding confirm primitive shall be used to convey the appropriate status value. Conversely, if a response primitive instigated the indirect transmission, the MLME-COMM-STATUS.indication primitive shall be used to convey the appropriate status value. The MLME-COMM-STATUS.indication primitive can be related to its corresponding response primitive by examining the Destination Address field.

The information contained in the indirect transmission request forms a transaction, and the coordinator shall be capable of storing at least one transaction. On receipt of an indirect transmission request, if there is no capacity to store another transaction, the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_OVERFLOW in the appropriate corresponding primitive.

If the coordinator is capable of storing more than one transaction, it shall ensure that all the transactions for the same device are sent in the order in which they arrived at the MAC sublayer. For each transaction sent, if another exists for the same device, the MAC sublayer shall set its frame pending subfield to one, indicating the additional pending data.

Each transaction shall persist in the coordinator for at most *macTransactionPersistenceTime*. If the transaction is not successfully extracted by the appropriate device within this time, the transaction information shall be discarded and the MAC sublayer shall indicate to the next higher layer a status of TRANSACTION_EXPIRED in the appropriate corresponding primitive. In order to be successfully extracted, an acknowledgment shall be received if one was requested.

If the transaction was successful, the transaction information shall be discarded, and the MAC sublayer shall indicate to the next higher layer a status of SUCCESS in the appropriate corresponding primitive.

If the coordinator transmits beacons, it shall list the addresses of the devices to which each transaction is associated in the Address List field and indicate the number of addresses in the Pending Address Specification field of the beacon frame. If the coordinator is able to store more than seven pending transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses. For transactions requiring a GTS, the coordinator shall not add the address of the recipient to its list of pending addresses in the beacon frame. Instead it shall transmit the transaction in the GTS allocated for the device as shown in 5.1.8.3.

On a beacon-enabled VPAN, if there is a transaction pending for the broadcast address, the frame pending subfield of the frame control field in the beacon frame shall be set to one, and the pending message shall be transmitted immediately following the beacon using the unslotted random access algorithm. If there is a second message pending for the broadcast address, its transmission shall be delayed until the following superframe. Only one broadcast message shall be allowed to be sent indirectly per superframe.

On a beacon-enabled VPAN, a device that receives a beacon containing its address in the list of pending addresses shall attempt to extract the data from the coordinator. On a nonbeacon-enabled VPAN, a device shall attempt to extract the data from the coordinator on receipt of the MLME-POLL.request primitive. The procedure for extracting pending data from the coordinator is described in 5.1.7.3. If a device receives a beacon with the frame pending subfield set to one, it shall leave its receiver enabled for up to *macMaxFrameTotalWaitTime* optical clocks to receive the broadcast data frame from the coordinator.

5.1.7 Transmission, reception, and acknowledgment

This subclause describes the fundamental procedures for transmission, reception, and acknowledgment.

5.1.7.1 Transmission

Each device shall store its current data-sequence number (DSN) value in the MAC PIB attribute *macDSN* and initialize it to a random value; the algorithm for choosing a random number is out of the scope of this standard. Each time a data or a MAC command frame is generated, the MAC sublayer shall copy the value of *macDSN* into the Sequence Number field of the MHR of the outgoing frame and then increment it by one. Each device shall generate exactly one DSN regardless of the number of unique devices with which it wishes to communicate. The value of *macDSN* shall be permitted to roll over.

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

It should be noted that both the DSN and BSN are 8-bit values and, therefore, have limited use to the next higher layer (e.g., in the case of the DSN, in detecting retransmitted frames).

The Source Address field, if present, shall contain the address of the device sending the frame. When a device has associated and has been allocated a 16-bit short address (i.e., *macShortAddress* is not equal to 0xfffe or 0xffff), it shall use that address in preference to its 64-bit extended address (i.e., *aExtendedAddress*) wherever possible. When a device has not yet associated to a VPAN or *macShortAddress* is equal to 0xffff, it shall use its 64-bit extended address in all communications requiring the Source Address field. If the Source Address field is not present, the originator of the frame shall be assumed to be the coordinator, and the Destination Address field shall contain the address of the recipient.

The Destination Address field, if present, shall contain the address of the intended recipient of the frame, which may be either a 16-bit short address or a 64-bit extended address. If the Destination Address field is not present, the recipient of the frame shall be assumed to be the coordinator, and the Source Address field shall contain the address of the originator.

If both destination and source addressing information is present, the MAC sublayer shall compare the destination and source VPAN identifiers. If the VPAN identifiers are identical, the VPAN ID Compression subfield of the frame control field shall be set to one, and the source VPAN identifier shall be omitted from the transmitted frame. If the VPAN identifiers are different, the VPAN ID Compression subfield of the frame control field shall be set to zero, and both Destination VPAN Identifier and Source VPAN Identifier fields shall be included in the transmitted frame. If only either the destination or the source addressing information is present, the VPAN ID Compression subfield of the frame control field shall be set to zero, and the VPAN identifier field of the single address shall be included in the transmitted frame.

If the frame is to be transmitted on a beacon-enabled VPAN, the transmitting device shall attempt to find the beacon before transmitting. If the beacon is not being tracked, as shown in 5.1.5.1, and hence the device does not know where the beacon will appear, it shall enable its receiver and search for at most [*aBaseSuperframeDuration* × (2^n + 1)] optical clocks, where *n* is the value of *macBeaconOrder,* in order to find the beacon. If the beacon is not found after this time, the device shall transmit the frame following the successful application of the unslotted version of the random access algorithm as shown in 5.1.1.3. Once the beacon has been found, either after a search or due to its being tracked, the frame shall be transmitted in the appropriate portion of the superframe. Transmissions in the CAP shall follow a successful application of the slotted version of the random access algorithm, see 5.1.1.3, and transmissions in a GTS shall not use any random access.

If the frame is to be transmitted on a nonbeacon-enabled VPAN, the frame shall be transmitted following the successful application of the unslotted version of the random access algorithm as shown in 5.1.1.3.

For either a beacon-enabled VPAN or a nonbeacon-enabled VPAN, if the transmission is direct and originates due to a primitive issued by the next higher layer and the access algorithm fails, the next higher layer shall be notified. If the transmission is indirect and the access algorithm fails, the frame shall remain in the transaction queue until it is requested again and successfully transmitted or until the transaction expires.

The device shall process the frame using the outgoing frame security procedure described in 7.2.1.

If the status from the outgoing frame security procedure is not SUCCESS, the MLME shall issue the corresponding confirm or MLME-COMM-STATUS.indication primitive with the status parameter set to the status from the outgoing frame security procedure, indicating the error.

To transmit the frame, the MAC sublayer shall first enable the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of TX_ON to the PHY. On receipt of the PLME-SET-TRX-STATE.confirm primitive with a status of either SUCCESS or TX_ON, the constructed frame shall then be transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MAC sublayer shall disable the transmitter by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON or TRX_OFF to the PHY, depending on whether the receiver is to be enabled following the transmission. In the case where the Acknowledgment Request subfield of the frame control field is set to one, the MAC sublayer shall enable the receiver immediately following the transmission of the frame by issuing the PLME-SET-TRX-STATE.request primitive with a state of RX_ON to the PHY.

5.1.7.2 Reception and rejection

Each device may choose whether the MAC sublayer is to enable its receiver during idle periods. During these idle periods, the MAC sublayer shall still service transceiver task requests from the next higher layer.

A transceiver task shall be defined as a transmission request with acknowledgment reception, if required, or a reception request. On completion of each transceiver task, the MAC sublayer shall request that the PHY enables or disables its receiver, depending on the values of *macBeaconOrder* and *macRxOnWhenIdle.* If *macBeaconOrder* is less than 15, the value of *macRxOnWhenIdle* shall be considered relevant only during idle periods of the CAP of the incoming superframe. If *macBeaconOrder* is equal to 15, the value of *macRxOnWhenIdle* shall be considered relevant at all times.

A device with its receiver enabled will be able to receive and decode transmissions from all devices complying with this standard that are currently operating on the same channel and are in its operating space, along with interference from other sources. The MAC sublayer shall, therefore, be able to filter incoming frames and present only the frames that are of interest to the upper layers.

The MAC sublayer shall discard all received frames that do not contain a correct value in their FCS field in the MFR (see 5.2.1.9). The FCS field shall be verified on reception by recalculating the purported FCS over the MHR and MSDU of the received frame and by subsequently comparing this value with the received FCS field. The FCS field of the received frame shall be considered to be correct if these values are the same and incorrect otherwise.

The MAC sublayer shall accept only frames that satisfy all of the following filtering requirements:

— The Frame Type subfield shall not contain a reserved frame type.
— If a destination VPAN identifier is included in the frame, it shall match *macVPANId* or shall be the broadcast VPAN identifier (0xffff).
— If a short destination address is included in the frame, it shall match either *macShortAddress* or the broadcast address (0xffff). Otherwise, if an extended destination address is included in the frame, it shall match *aExtendedAddress*.
— If the frame type indicates that the frame is a beacon frame, the source VPAN identifier shall match *macVPANId* unless *macVPANId* is equal to 0xffff, in which case the beacon frame shall be accepted regardless of the source VPAN identifier.
— If only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is the coordinator and the source VPAN identifier matches *macVPANId.*

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

If the VPAN ID Compression subfield of the frame control field is set to one and both destination and source addressing information is included in the frame, the MAC sublayer shall assume that the omitted Source VPAN Identifier field is identical to the Destination VPAN Identifier field.

The device shall process the frame using the incoming frame security procedure described in 7.2.3.

If the status from the incoming frame security procedure is not SUCCESS, the MLME shall issue the corresponding confirm or MLME-COMM-STATUS.indication primitive with the status parameter set to the status from the incoming frame security procedure, indicating the error, and with the security-related parameters set to the corresponding parameters returned by the unsecuring process.

If the valid frame is a data frame, the MAC sublayer shall pass the frame to the next higher layer. This is achieved by issuing the MCPS-DATA.indication primitive containing the frame information. The security-related parameters of the MCPS-DATA.indication primitive shall be set to the corresponding parameters returned by the unsecuring process.

If the valid frame is a MAC command or beacon frame, it shall be processed by the MAC sublayer accordingly, and a corresponding confirm or indication primitive may be sent to the next higher layer. The security-related parameters of the corresponding confirm or indication primitive shall be set to the corresponding parameters returned by the unsecuring process.

5.1.7.3 Extracting pending data from a coordinator

A device on a beacon-enabled VPAN can determine whether any frames are pending for it by examining the contents of the received beacon frame, as described in 5.1.5.1. If the address of the device is contained in the Address List field of the beacon frame and *macAutoRequest* is TRUE, the MLME of the device shall send a data request command, see 5.3.4, to the coordinator during the CAP with the Acknowledgment Request subfield of the frame control field set to one; the only exception to this is if the beacon frame is received while performing an active or passive scan as shown in 5.1.3.1. There are two other cases for which the MLME shall send a data request command to the coordinator. The first case is when the MLME receives the MLME-POLL.request primitive. In the second case, a device may send a data request command *macResponseWaitTime* optical clocks after the acknowledgment to a request command frame, such as during the association procedure. If the data request is intended for the coordinator, the destination address information may be omitted.

If the data request command originated from an MLME-POLL.request primitive, the MLME shall perform the security process on the data request command based on the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters of the MLME-POLL.request primitive, according to 7.2.1. Otherwise, the MLME shall perform the security process on the data request command based on the *macAutoRequestSecurityLevel*, *macAutoRequestKeyIdMode*, *macAutoRequestKeySource,* and *macAutoRequestKeyIndex* PIB attributes, according to 7.2.1.

On successfully receiving a data request command, the coordinator shall send an acknowledgment frame, thus confirming its receipt. If the coordinator has enough time to determine whether the device has a frame pending before sending the acknowledgment frame (see 5.1.7.4.2), it shall set the frame pending subfield of the frame control field of the acknowledgment frame accordingly to indicate whether a frame is actually pending for the device. If this is not possible, the coordinator shall set the frame pending subfield of the acknowledgment frame to one.

On receipt of the acknowledgment frame with the frame pending subfield set to zero, the device shall conclude that there are no data pending at the coordinator.

On receipt of the acknowledgment frame with the frame pending subfield set to one, a device shall enable its receiver for at most *macMaxFrameTotalWaitTime* CAP optical clocks in a beacon-enabled VPAN, or in a nonbeacon-enabled VPAN, to receive the corresponding data frame from the coordinator. If there is an actual data frame pending within the coordinator for the requesting device, the coordinator shall send the frame to the device using one of the mechanisms described in this subclause. If there is no data frame pending for the requesting device, the coordinator shall send a data frame without requesting acknowledgment to the device containing a zero length payload, indicating that no data are present, using one of the mechanisms described in this subclause.

The data frame following the acknowledgment of the data request command shall be transmitted using one of the following mechanisms:

— Without using slotted random access, if the MAC sublayer can commence transmission of the data frame between *aTurnaroundTime-RX-TX* and (*aTurnaroundTime-RX-TX* + *aUnitBackoffPeriod*) optical clocks, on a backoff slot boundary, and there is time remaining in the CAP for the message, appropriate IFS, and acknowledgment as defined in 9.5.1. If a requested acknowledgment frame is not received following this data frame, the process shall begin anew following the receipt of a new data request command.

— Using slotted random access, otherwise.

If the requesting device does not receive a data frame from the coordinator within *macMaxFrameTotalWaitTime* CAP optical clocks in a beacon-enabled VPAN, or in a nonbeacon-enabled VPAN, or if the requesting device receives a data frame from the coordinator with a zero length payload, it shall conclude that there are no data pending at the coordinator. If the requesting device does receive a data frame from the coordinator, it shall send an acknowledgment frame, if requested, thus confirming receipt.

If the frame pending subfield of the frame control field of the data frame received from the coordinator is set to one, the device still has more data pending with the coordinator. In this case it may extract the data by sending a new data request command to the coordinator.

5.1.7.4 Use of acknowledgments and retransmissions

A data or MAC command frame shall be sent with the Acknowledgment Request subfield of its frame control field set appropriately for the frame. A beacon or acknowledgment frame shall always be sent with the Acknowledgment Request subfield set to zero. Similarly, any frame that is broadcast shall be sent with its Acknowledgment Request subfield set to zero.

5.1.7.4.1 No acknowledgment

A frame transmitted with its Acknowledgment Request subfield set to zero shall not be acknowledged by its intended recipient. The originating device shall assume that the transmission of the frame was successful.

The message sequence chart in Figure 20 shows the scenario for transmitting a single frame of data from an originator to a recipient without requiring an acknowledgment. In this case, the originator transmits the data frame with the Acknowledgment Request (AR) subfield of the frame control field equal to zero.

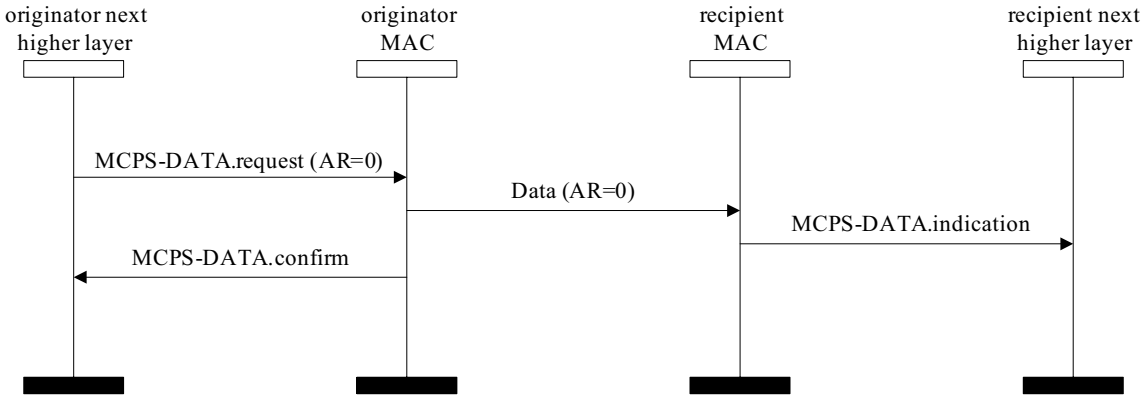


Figure 20—Successful data transmission without an acknowledgment

5.1.7.4.2 Acknowledgment

A frame transmitted with the Acknowledgment Request subfield of its frame control field set to one shall be acknowledged by the recipient. If the intended recipient correctly receives the frame, it shall generate and

send an acknowledgment frame containing the same DSN from the data or MAC command frame that is being acknowledged.

The transmission of an acknowledgment frame in a nonbeacon-enabled VPAN or in the CFP shall commence *aTurnaroundTime-RX-TX*optical clocks after the last optical clock of the data or MAC command frame. The transmission of an acknowledgment frame in the CAP shall commence either *aTurnaroundTime-RX-TX* optical clocks after the reception of the last optical clock of the data or MAC command frame or at a backoff slot boundary. In the latter case, the transmission of an acknowledgment frame shall commence between *aTurnaroundTime-RX-TX* and (*aTurnaroundTime-RX-TX* + *aUnitBackoffPeriod*) optical clocks after the reception of the last optical clock of the data or MAC command frame. The constants *aTurnaroundTime-RX-TX* and *aTurnaroundTime-TX-RX* are defined in Table 99.

The message sequence chart in Figure 21 shows the scenario for transmitting a single frame of data from an originator to a recipient with an acknowledgment. In this case, the originator indicates to the recipient that it requires an acknowledgment by transmitting the data frame with the Acknowledgment Request (AR) subfield of the frame control field set to one.

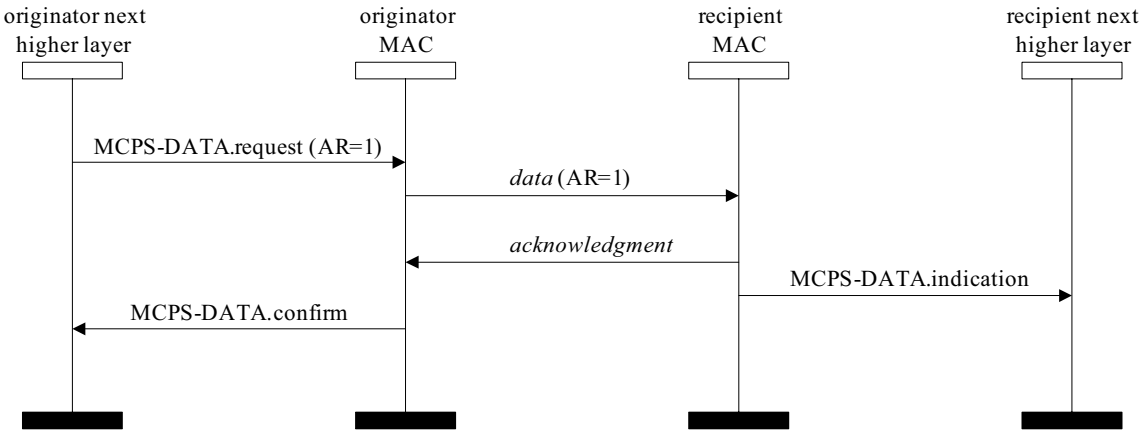


Figure 21—Successful data transmission with an acknowledgment

5.1.7.4.3 Retransmissions

A device that sends a frame with the Acknowledgment Request subfield of its frame control field set to zero shall assume that the transmission was successfully received and shall hence not perform the retransmission procedure.

A device that sends a data or MAC command frame with its Acknowledgment Request subfield set to one shall wait for at most *macAckWaitDuration* optical clocks for the corresponding acknowledgment frame to be received. If an acknowledgment frame is received within *macAckWaitDuration* optical clocks 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* optical clocks or an acknowledgment is received containing a DSN that was not the same as the original transmission, the device shall conclude that the single transmission attempt has failed.

If a single transmission attempt has failed and the transmission was indirect, the coordinator shall not retransmit the data or MAC command frame. Instead, the frame shall remain in the transaction queue of the coordinator and can only be extracted following the reception of a new data request command. If a new data request command is received, the originating device shall transmit the frame using the same DSN as was used in the original transmission.

If a single transmission attempt has failed and the transmission was direct, the device shall repeat the process of transmitting the data or MAC command frame and waiting for the acknowledgment, up to a maximum of *macMaxFrameRetries* times. The retransmitted frame shall contain the same DSN as was used in the original transmission. Each retransmission shall only be attempted if it can be completed within the same portion of the superframe, i.e., the CAP or a GTS in which the original transmission was attempted. If this timing is not possible, the retransmission shall be deferred until the same portion in the next superframe. If an acknowledgment is still not received after *macMaxFrameRetries* retransmissions, the MAC sublayer shall assume the transmission has failed and notify the next higher layer of the failure.

5.1.7.5 Transmission scenarios

Due to the imperfect nature of the wireless medium, a transmitted frame does not always reach its intended destination. Figure 22 illustrates three different data transmission scenarios:

— *Successful data transmission.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* optical clocks. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the data frame to the next higher layer. The originator MAC sublayer receives the acknowledgment from the recipient before its timer expires and then disables and resets the timer. The data transfer is now complete, and the originator MAC sublayer issues a success confirmation to the next higher layer.

— *Lost data frame.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* optical clocks. The recipient MAC sublayer does not receive the data frame and so does not respond with an acknowledgment. The timer of the originator MAC sublayer expires before an acknowledgment is received; therefore, the data transfer has failed. If the transmission was direct, the originator retransmits the data, and this entire sequence may be repeated up to a maximum of *macMaxFrameRetries* times; if a data transfer attempt fails a total of (1 + *macMaxFrameRetries*) times, the originator MAC sublayer will issue a failure confirmation to the next higher layer. If the transmission was indirect, the data frame will remain in the transaction queue until either another request for the data is received and correctly acknowledged or until *macTransactionPersistenceTime* is reached. If *macTransactionPersistenceTime* is reached, the transaction information will be discarded, and the MAC sublayer will issue a failure confirmation to the next higher layer.

— *Lost acknowledgment frame.* The originator MAC sublayer transmits the data frame to the recipient via the PHY data service. In waiting for an acknowledgment, the originator MAC sublayer starts a timer that will expire after *macAckWaitDuration* optical clocks. The recipient MAC sublayer receives the data frame, sends an acknowledgment back to the originator, and passes the data frame to the next higher layer. The originator MAC sublayer does not receive the acknowledgment frame, and its timer expires. Therefore, the data transfer has failed. If the transmission was direct, the originator retransmits the data, and this entire sequence may be repeated up to a maximum of *macMaxFrameRetries* times. If a data transfer attempt fails a total of (1 + *macMaxFrameRetries*) times, the originator MAC sublayer will issue a failure confirmation to the next higher layer. If the transmission was indirect, the data frame will remain in the transaction queue either until another request for the data is received and correctly acknowledged or until *macTransactionPersistenceTime* is reached. If *macTransactionPersistenceTime* is reached, the transaction information will be discarded, and the MAC sublayer will issue a failure confirmation to the next higher layer.

5.1.8 GTS allocation and management

A GTS allows a device to operate on the channel within a portion of the superframe that is dedicated (on the VPAN) exclusively to that device. A GTS shall be allocated only by the coordinator, and it shall be used only for communications between the coordinator and a device associated with the VPAN through the

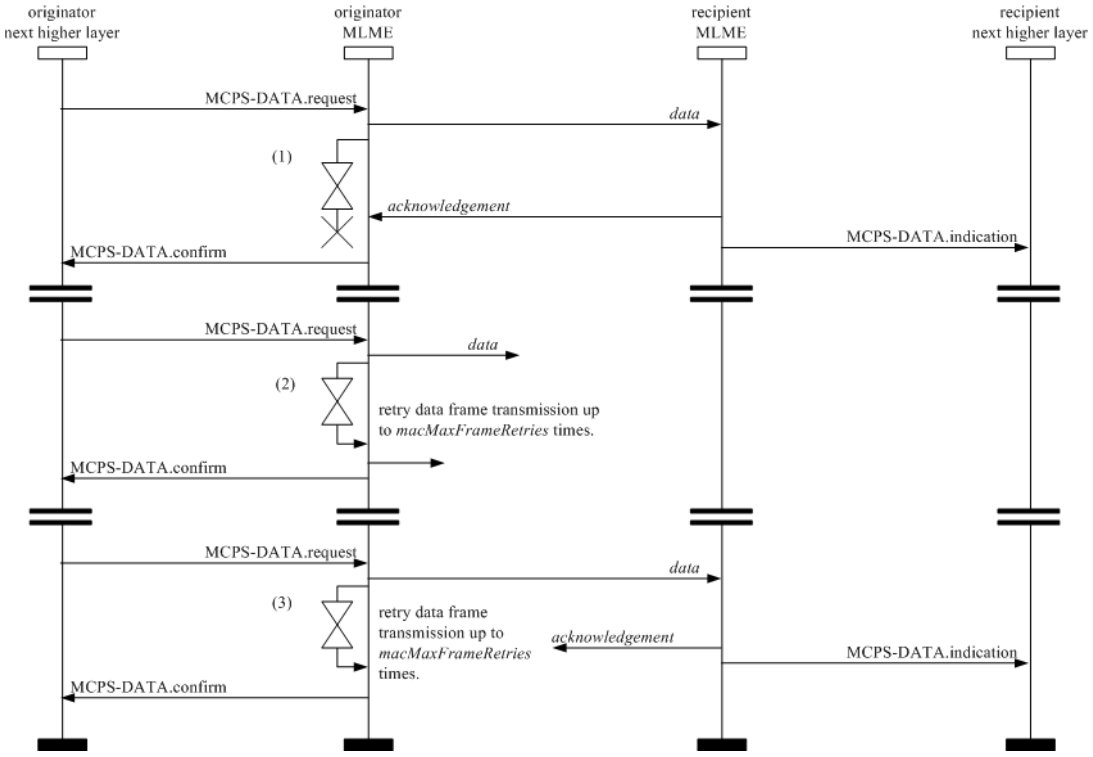


Figure 22—Transmission scenarios, using direct transmission, for frame reliability

coordinator. A single GTS may extend over one or more superframe slots. The coordinator may allocate a number of GTSs at the same time, provided there is sufficient capacity in the superframe.

A GTS shall be allocated before use, with the coordinator deciding whether to allocate a GTS based on the requirements of the GTS request and the current available capacity in the superframe. GTSs shall be allocated on a first-come-first-served basis, and all GTSs shall be placed contiguously at the end of the superframe and after the CAP. Each GTS shall be deallocated when the GTS is no longer required, and a GTS can be deallocated at any time at the discretion of the coordinator or by the device that originally requested the GTS. A device that has been allocated a GTS may also operate in the CAP.

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

The management of GTSs shall be undertaken by the coordinator only. To facilitate GTS management, the coordinator shall be able to store all the information necessary to manage seven GTSs. For each GTS, the coordinator shall be able to store its starting slot, length, direction, and associated device address.

The GTS direction, which is relative to the data flow from the device that owns the GTS, is specified as either transmit or receive. The device address and direction shall, therefore, uniquely identify each GTS. Each device may request one transmit GTS and/or one receive GTS. For each allocated GTS, the device shall be able to store its starting slot, length, and direction. If a device has been allocated a receive GTS, it shall enable its receiver for the entirety of the GTS. In the same way, the coordinator shall enable its receiver for the entirety of the GTS if a device has been allocated a transmit GTS. If a data frame is received during a receive GTS and an acknowledgment is requested, the device shall transmit the acknowledgment frame as usual. Similarly, a device shall be able to receive an acknowledgment frame during a transmit GTS.

A device shall attempt to allocate and use a GTS only if it is currently tracking the beacons. The MLME is instructed to track beacons by issuing the MLME-SYNC.request primitive with the TrackBeacon parameter set to TRUE. If a device loses synchronization with the coordinator, all its GTS allocations shall be lost.

The use of GTSs is optional.

5.1.8.1 CAP maintenance

The coordinator shall preserve the minimum CAP length of *aMinCAPLength* and take preventative action if the minimum CAP is not satisfied. However, an exception shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance. If preventative action becomes necessary, the action chosen is left up to the implementation, but may include one or more of the following:

— Limiting the number of pending addresses included in the beacon.
— Not including a payload field in the beacon frame.
— Deallocating one or more of the GTSs.

5.1.8.2 GTS allocation

A device is instructed to request the allocation of a new GTS through the MLME-GTS.request primitive, with GTS characteristics set according to the requirements of the intended application.

To request the allocation of a new GTS, the MLME shall send the GTS request command, see 5.3.13, to the coordinator. The Characteristics Type subfield of the GTS Characteristics field of the request shall be set to one (GTS allocation), and the length and direction subfields shall be set according to the desired characteristics of the required GTS. Because the GTS request command contains an acknowledgment request (see 5.3.3.1), the coordinator shall confirm its receipt by sending an acknowledgment frame.

On receipt of a GTS request command indicating a GTS allocation request, the coordinator shall first check if there is available capacity in the current superframe, based on the remaining length of the CAP and the desired length of the requested GTS. The superframe shall have available capacity if the maximum number of GTSs has not been reached and allocating a GTS of the desired length would not reduce the length of the CAP to less than *aMinCAPLength*. GTSs shall be allocated on a first-come-first-served basis by the coordinator provided there is sufficient bandwidth available. The coordinator shall make this decision within *aGTSDescPersistenceTime* superframes.

On receipt of the acknowledgment to the GTS request command, the device shall continue to track beacons and wait for at most *aGTSDescPersistenceTime* superframes. If no GTS descriptor for the device appears in the beacon within this time, the MLME of the device shall notify the next higher layer of the failure. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive, see 6.3.5.3, with a status of NO_DATA.

When the coordinator determines whether capacity is available for the requested GTS, it shall generate a GTS descriptor with the requested specifications and the 16-bit short address of the requesting device. If the GTS was allocated successfully, the coordinator shall set the start slot in the GTS descriptor to the superframe slot at which the GTS begins and the length in the GTS descriptor to the length of the GTS. In addition, the coordinator shall notify the next higher layer of the new GTS. This notification is achieved when the MLME of the coordinator issues the MLME-GTS.indication primitive, see 6.3.5.2, with the characteristics of the allocated GTS. If there was not sufficient capacity to allocate the requested GTS, the start slot shall be set to zero and the length to the largest GTS length that can currently be supported. The coordinator shall then include this GTS descriptor in its beacon and update the GTS Specification field of the beacon frame accordingly. The coordinator shall also update the Final CAP Slot subfield of the

Superframe Specification field of the beacon frame, indicating the final superframe slot utilized by the decreased CAP. The GTS descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes, after which it shall be removed automatically. The coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress*, the device shall process the descriptor. The MLME of the device shall then notify the next higher layer of whether the GTS allocation request was successful. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive with a status of SUCCESS (if the start slot in the GTS descriptor was greater than zero) or DENIED (if the start slot was equal to zero or if the length did not match the requested length).

5.1.8.3 GTS usage

When the MAC sublayer of a device that is not the coordinator receives an MCPS-DATA.request primitive, see 6.2.1, with the TxOptions parameter indicating a GTS transmission, it shall determine whether it has a valid transmit GTS. If a valid GTS is found, the MAC sublayer shall transmit the data during the GTS, i.e., between its starting slot and its starting slot plus its length. At this time, the MAC sublayer shall transmit the MPDU immediately without using any random access, provided the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer shall defer the transmission until the specified GTS in the next superframe. Note that the MAC shall allow for the PHY overhead in making this determination.

If the device has any receive GTSs, the MAC sublayer of the device shall ensure that the receiver is enabled at a time prior to the start of the GTS and for the duration of the GTS, as indicated by its starting slot and its length.

When the MAC sublayer of the coordinator receives an MCPS-DATA.request primitive with the TxOptions parameter indicating a GTS transmission, it shall determine whether it has a valid receive GTS corresponding to the device with the requested destination address. If a valid GTS is found, the coordinator shall defer the transmission until the start of the receive GTS. In this case, the address of the device with the message requiring a GTS transmission shall not be added to the list of pending addresses in the beacon frame as shown in 5.1.6. At the start of the receive GTS, the MAC sublayer shall transmit the data without using any random access, provided the requested transaction can be completed before the end of the GTS. If the requested transaction cannot be completed before the end of the current GTS, the MAC sublayer shall defer the transmission until the specified GTS in the next superframe.

For all allocated transmit GTSs (relative to the device), the MAC sublayer of the coordinator shall ensure that its receiver is enabled at a time prior to the start and for the duration of each GTS.

Before commencing transmission in a GTS, each device shall ensure that the data transmission, the acknowledgment, if requested, and the IFS, suitable to the size of the data frame, can be completed before the end of the GTS.

If a device misses the beacon at the beginning of a superframe, it shall not use its GTSs until it receives a subsequent beacon correctly. If a loss of synchronization occurs due to the loss of the beacon, the device shall consider all of its GTSs deallocated.

5.1.8.4 GTS deallocation

A device is instructed to request the deallocation of an existing GTS through the MLME-GTS.request primitive specified in 6.3.5.1, using the characteristics of the GTS it wishes to deallocate. From this point onward, the GTS to be deallocated shall not be used by the device, and its stored characteristics shall be reset.

To request the deallocation of an existing GTS, the MLME shall send the GTS request command, specified in 5.3.13, to the coordinator. The Characteristics Type subfield of the GTS Characteristics field of the request shall be set to zero (i.e., GTS deallocation), and the length and direction subfields shall be set according to the characteristics of the GTS to deallocate. Because the GTS request command contains an acknowledgment request, specified in 5.3.3.1, the coordinator shall confirm its receipt by sending an acknowledgment frame. On receipt of the acknowledgment to the GTS request command, the MLME shall notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.confirm primitive, see 6.3.5.3, with a status of SUCCESS and a GTSCharacteristics parameter with its Characteristics Type subfield set to zero. If the GTS request command is not received correctly by the coordinator, it shall determine that the device has stopped using its GTS by the procedure described in 5.1.8.6.

On receipt of a GTS request command with the Characteristics Type subfield of the GTS Characteristics field set to zero (GTS deallocation), the coordinator shall attempt to deallocate the GTS. If the GTS characteristics contained in the GTS request command do not match the characteristics of a known GTS, the coordinator shall ignore the request. If the GTS characteristics contained in the GTS request command match the characteristics of a known GTS, the MLME of the coordinator shall deallocate the specified GTS and notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive, see 6.3.5.2, with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero. The coordinator shall also update the Final CAP Slot subfield of the Superframe Specification field of the beacon frame, indicating the final superframe slot utilized by the increased CAP. It shall not add a descriptor to the beacon frame to describe the deallocation.

GTS deallocation may be initiated by the coordinator due to a deallocation request from the next higher layer, the expiration of the GTS (see 5.1.8.6), or maintenance required to maintain the minimum CAP length, *aMinCAPLength* (see 5.1.8.1).

When a GTS deallocation is initiated by the next higher layer of the coordinator, the MLME shall receive the MLME-GTS.request primitive with the GTS Characteristics field of the request set to zero (i.e. GTS deallocation) and the length and direction subfields set according to the characteristics of the GTS to deallocate.

When a GTS deallocation is initiated by the coordinator either due to the GTS expiring or due to CAP maintenance, the MLME shall notify the next higher layer of the change. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero.

In the case of any deallocation initiated by coordinator, the coordinator shall deallocate the GTS and add a GTS descriptor into its beacon frame corresponding to the deallocated GTS, but with its starting slot set to zero. The descriptor shall remain in the beacon frame for *aGTSDescPersistenceTime* superframes. The coordinator shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a start slot equal to zero, the device shall immediately stop using the GTS. The MLME of the device shall then notify the next higher layer of the deallocation. This notification is achieved when the MLME issues the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS and a Characteristics Type subfield set to zero.

5.1.8.5 GTS reallocation

The deallocation of a GTS may result in the superframe becoming fragmented. For example, Figure 23 shows three stages of a superframe with allocated GTSs. In stage 1, three GTSs are allocated starting at slots

14, 10, and 8, respectively. If GTS 2 is now deallocated (stage 2), there will be a gap in the superframe during which nothing can happen. To solve this, GTS 3 will have to be shifted to fill the gap, thus increasing the size of the CAP (stage 3).

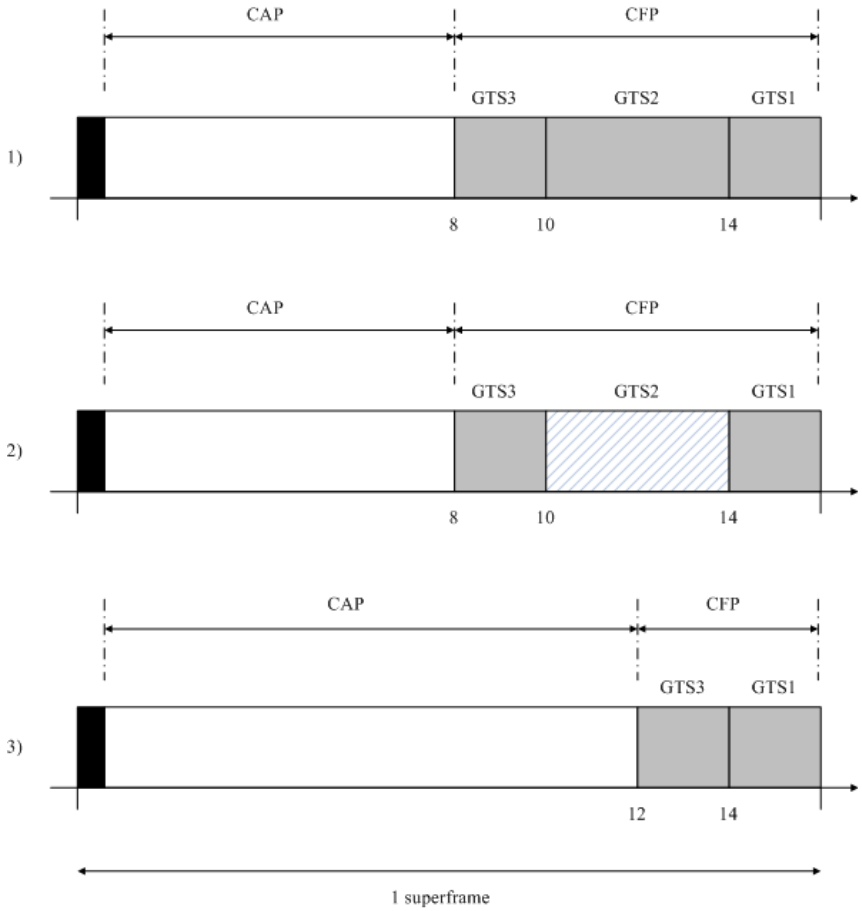


Figure 23—CFP defragmentation on GTS deallocations

The coordinator shall ensure that any gaps occurring in the CFP, appearing due to the deallocation of a GTS, are removed to maximize the length of the CAP.

When a GTS is deallocated by the coordinator, it shall add a GTS descriptor into its beacon frame indicating that the GTS has been deallocated. If the deallocation is initiated by a device, the coordinator shall not add a GTS descriptor into its beacon frame to indicate the deallocation. For each device with an allocated GTS having a starting slot lower than the GTS being deallocated, the coordinator shall update the GTS with the new starting slot and add a GTS descriptor to its beacon corresponding to this adjusted GTS. The new starting slot is computed so that no space is left between this GTS and either the end of the CFP, if the GTS appears at the end of the CFP, or the start of the next GTS in the CFP.

In situations where multiple reallocations occur at the same time, the coordinator may choose to perform the reallocation in stages. The coordinator shall keep each GTS descriptor in its beacon for *aGTSDescPersistenceTime* superframes.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a direction and length corresponding to one of its GTSs, the device shall adjust the starting slot of the GTS corresponding to the GTS descriptor and start using it immediately.

In cases where it is necessary for the coordinator to include a GTS descriptor in its beacon, it shall be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length. After *aGTSDescPersistenceTime* superframes, the coordinator shall remove the GTS descriptor from the beacon.

5.1.8.6 GTS expiration

The MLME of the coordinator shall attempt to detect when a device has stopped using a GTS using the following rules:

— For a transmit GTS, the MLME of the coordinator shall assume that a device is no longer using its GTS if a data frame is not received from the device in the GTS at least every 2*n* superframes, where *n* is defined below.
— For receive GTSs, the MLME of the coordinator shall assume that a device is no longer using its GTS if an acknowledgment frame is not received from the device at least every 2*n* superframes, where *n* is defined below. If the data frames sent in the GTS do not require acknowledgment frames, the MLME of the coordinator will not be able to detect whether a device is using its receive GTS. However, the coordinator is capable of deallocating the GTS at any time.

The value of *n* is defined as follows:

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

$n = 1$ $\quad 9 \le macBeaconOrder \le 14$

5.1.9 Fast link recovery

In the star topology, a fast link recovery process may be triggered at the device end during communication. The trigger may be initiated when the device does not receive ACKs for a number of times given by the MAC PIB attribute *macNumAcks,* as defined in Table 60. In the fast link recovery process, the device may decide on its own to stop sending data. The device may also send the fast link recovery (FLR) signal repeatedly (within the allocated resource) to the coordinator if the device is connected to mains power. Upon receiving the FLR signal, the coordinator shall send a FLR response to the device. The communication resumes after the device receives the response. If there is bi-directional data transfer during communication, the device may wait after stopping sending data. If the device does not receive any FLR response signal within a timer given by the MAC PIB attribute *macLinkTimeOut*, the device may assume the link is broken and may disassociate.

The FLR signal and response are defined in 5.3.11. The FLR signal and response shall be sent at the lowest data rate corresponding to the currently negotiated optical rate.

Figure 24 shows an example of the process of device stopping sending data based on the retransmission count.

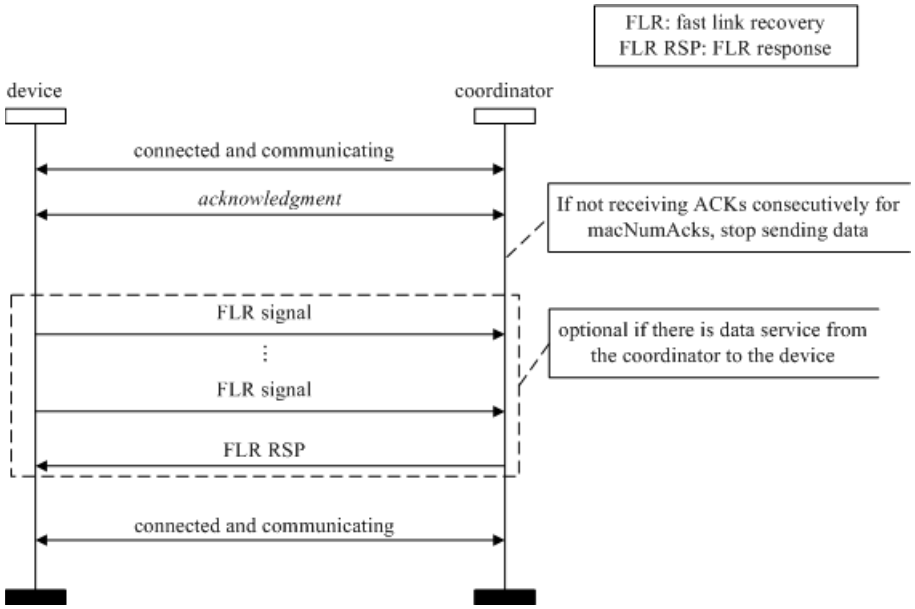


Figure 24—An example of the process of device stopping data transmission based on the retransmission count, and triggering FLR

In the star topology, a fast link recovery may also be triggered by the coordinator. The trigger may be initiated when the coordinator does not receive contiguous ACKs for a number of times given by the MAC PIB attribute *macNumAcks*. In the fast link recovery process, the coordinator may stop sending data to the device. The coordinator then sends fast link recovery (FLR) signal repeatedly to the device. The coordinator may hold the uplink grant allocated to the device. Upon receiving FLR signal, the device shall send a FLR response to the coordinator. The communication resumes after the device receives the response.

Figure 25 shows an example of the process of the coordinator stopping sending data based on the retransmission count.

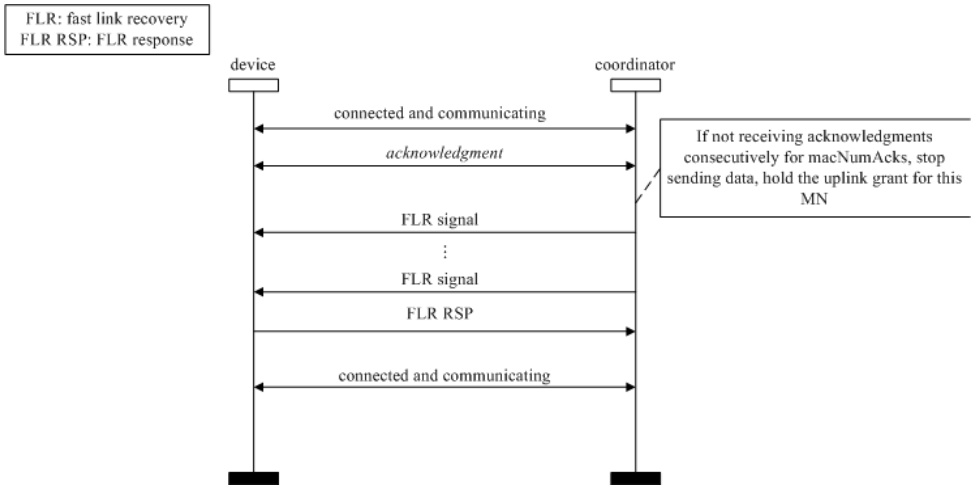


Figure 25—An example of the process of the coordinator stopping sending data based on the retransmission count, and triggering FLR

In peer-to-peer VLC, the devices may let each other know their battery life. If the conditions to trigger the fast link recovery process are satisfied, the device may further compare its own battery life with the battery life of its peer (the one it is communicating). If the battery life of the device is shorter than its peer's, then the device stops sending data, and waits. If the battery life of the device is longer than its peer's, then the device stops sending data and initiates the fast link recovery process.

When the fast link recovery is triggered, and if the device has spare wavelength bands, some or all of the spare bands may also start sending fast link recovery signals to recover the link. The device then shall choose a band that gets the fast link recovery response to continue the communication.

The address field of MHR in FLR signal and response may include the address or the identifier of the color bands.

Figure 26 shows a flowchart of the process for color band assisted fast link recovery.

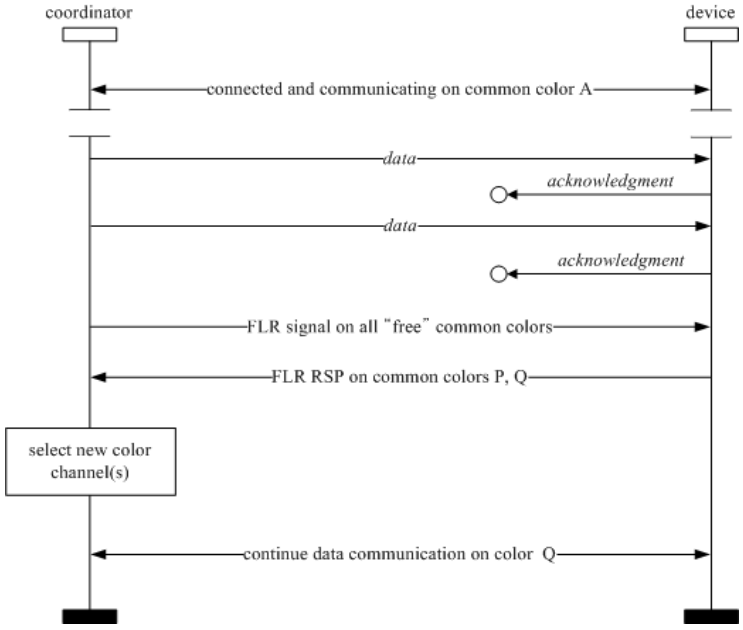


Figure 26—Flowchart of process for color band assisted fast link recovery

When the fast link recovery is triggered, if the device has other communication directions/angles (e.g., a light with multiple LEDs with different angles) some or all of the other angles may also start sending fast link recovery signaling to recover the link. The device then shall choose an angle that gets the fast link recovery response to continue the communication. The process of fast link recovery on other directions/ angles is done successively (i.e., one direction after another). The direction is indicated in the link recovery mechanism provided by the command frame structure.

The address field of MHR in FLR signal and response may include the address or the identifier of the angles or directions.

Figure 27 shows a flowchart of the process for multiple angle assisted fast link recovery.

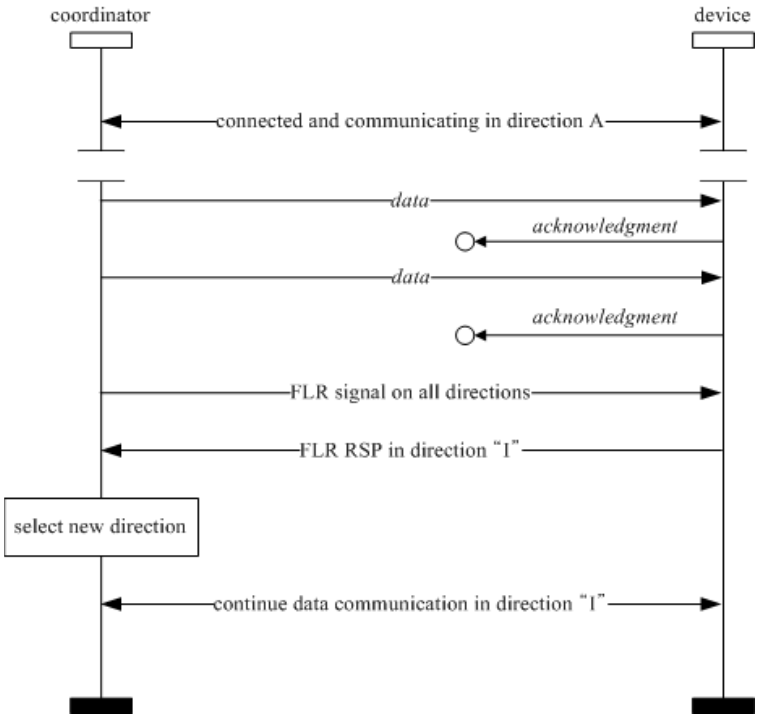


Figure 27—Flowchart showing process of multiple angles assisted fast link recovery

5.1.10 Multiple channel resource assignment

5.1.10.1 Multiple channel information

When the coordinator does not have time slot resources to assign for new user, the coordinator should extend the resource by using multiple bands. Figure 28 shows one example of multiple band usage.

Figure 29 describes the procedure of multiple band usage when the multiple band function is needed. When device 2 tries to initially access the coordinator for communication and no time slot is available but other bands are available for device 2, the coordinator can assign another band except the default band. Capability exchange should occur for all bi-directional communication during device discovery (see 5.1.2.4). If multiple bands are used, the coordinator should transmit to the device the "Src_multi_info" in the MAC command payload field which is defined in Table 3 to the device. Then the device 2 shall respond to the coordinator using the "Des_multi_info," which is defined in Table 3, informing the device of available multiple bands of the device.

If the coordinator does not support multiple bands, because the coordinator has a single band light source, or does not want to use multiple bands, the coordinator should transmit Src_multi_info set with code '0000000' as shown in Annex D.

If the device also cannot support multiple bands due to hardware limitations, such as a single band light source or an interference situation, or does not want to use multiple bands, the device should respond with Des_multi_info set to code '0000000' as shown in Annex D.

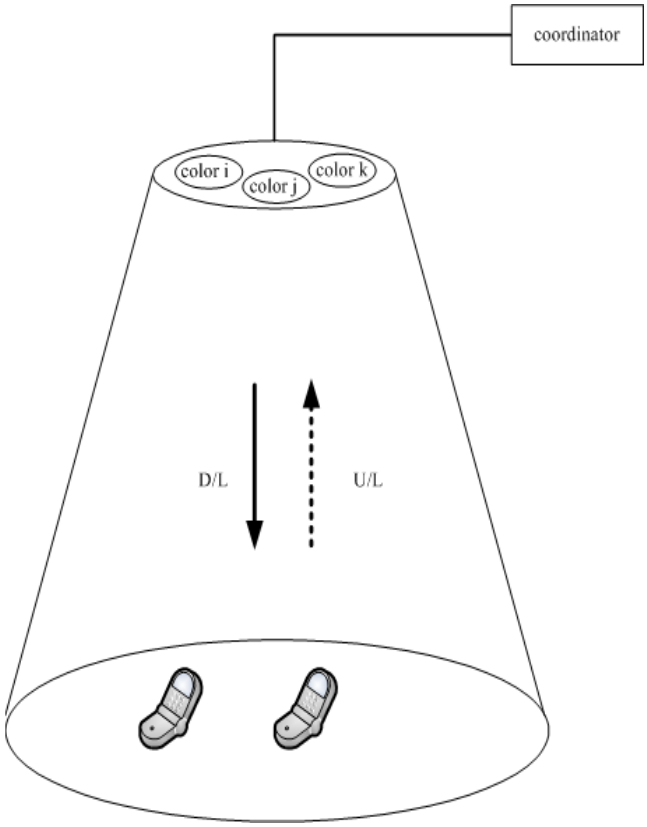


Figure 28—Example of multiple channel usage

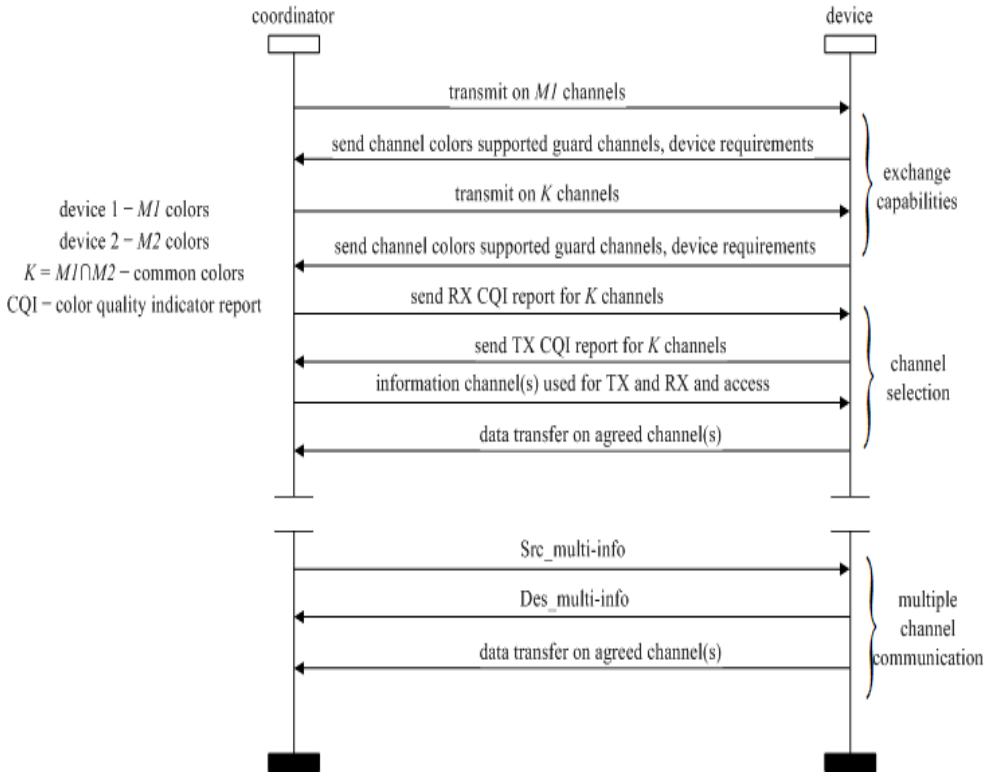


Figure 29—MSC for multi-band information

Table 3—Command frame payload for multiple bands

MAC command frame payload	Bits	Usage/Description	Down/Up Link
Src_multi_info	b0...b6	Bit map that indicates the available channels to the coordinator ex: 0000000: No multiple channel mode ex: 0000001: using channel “Band 7” ex: 0000101: using channel “Band 5” and “Band 7”	D/L
Des_multi_info	b0...b6	Bit map that indicates the available channels to the mobile device ex: 0000000: No multiple channel mode ex: 0000001: using channel “Band 7” ex: 0000101: using channel “Band 5” and “Band 7”	U/L

5.1.10.2 Band hopping for interference avoidance

A single coordinator can service multiple cells.

If interference is being experienced from an adjacent light then hopping can be used to mitigate the interference. When spatial reuse due to direction optics is not present, and when the VLC communications system uses the same time slot between the adjacent light sources or cells with multiple band communication, and when multiple bands are supported by the PHY, band hopping can be used. In order to avoid interference and increase system capacity, pre-assigned hopping patterns (HPs) should be adopted.

The hopping pattern should be assigned to the device and then the device should operate and hop based on the assigned hopping pattern. The coordinator shall transmit to the device the 'H_pattern' using the MAC command frame payload that is defined in Table 4.

If the VLC system does not use multiple bands (Src_multi_info is set to code '0000000'), then the hopping function is not supported. The hopping patterns shall be structured so as not to change the visual perception of the light. For example, the patterns could hop between RGB in the proper time averaged portion so as to appear white.

Table 4—Command frame payload for channel hopping

MAC command frame payload	Bit	Usage/Description	Down/Up Link
H_pattern	b0, b1, b2, b3, b4	Band hopping information	D/L

5.1.11 VLC cell design and mobility support

There may be a need to support link switching due to physical movement or interference. Mobility can be of two types: physical and logical. Physical mobility occurs when the VLC device M1 changes its position due to the movement within the coverage area of infrastructure I1 while logical mobility occurs when the device M1 changes its communication link from a link with infrastructure I2 to one with infrastructure I3 due to interference or deliberate channel switching, as shown in Figure 30.

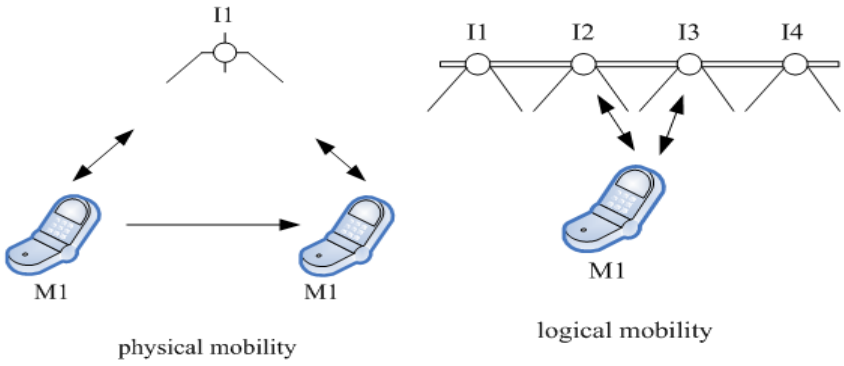


Figure 30—Physical and logical mobility

A coordinator DME can separate the optical media into multiple cells for supporting applications such as location-based services.

5.1.11.1 Mobility using boundary information

A single coordinator can support mobility of the device through multiple cells using the PHY switch, controlled by the DME, as shown in Figure 31. Each optical element in a cell is denoted by *cell_ID(i,j),* where *j* is the index of the element in the i^{th} cell. The size and the position of the cell in the optical media shown in Figure 3 can be variable and can be programmed by the DME. The actual size and position determination for the cell by the coordinator DME is not defined in the standard. If device 1 moves to the next cell, for example, from *cell_ID(i,j)* to *cell_ID(i+1,j)*, the coordinator can detect the mobility of the device using the uplink signal (i.e., acknowledgment frame).

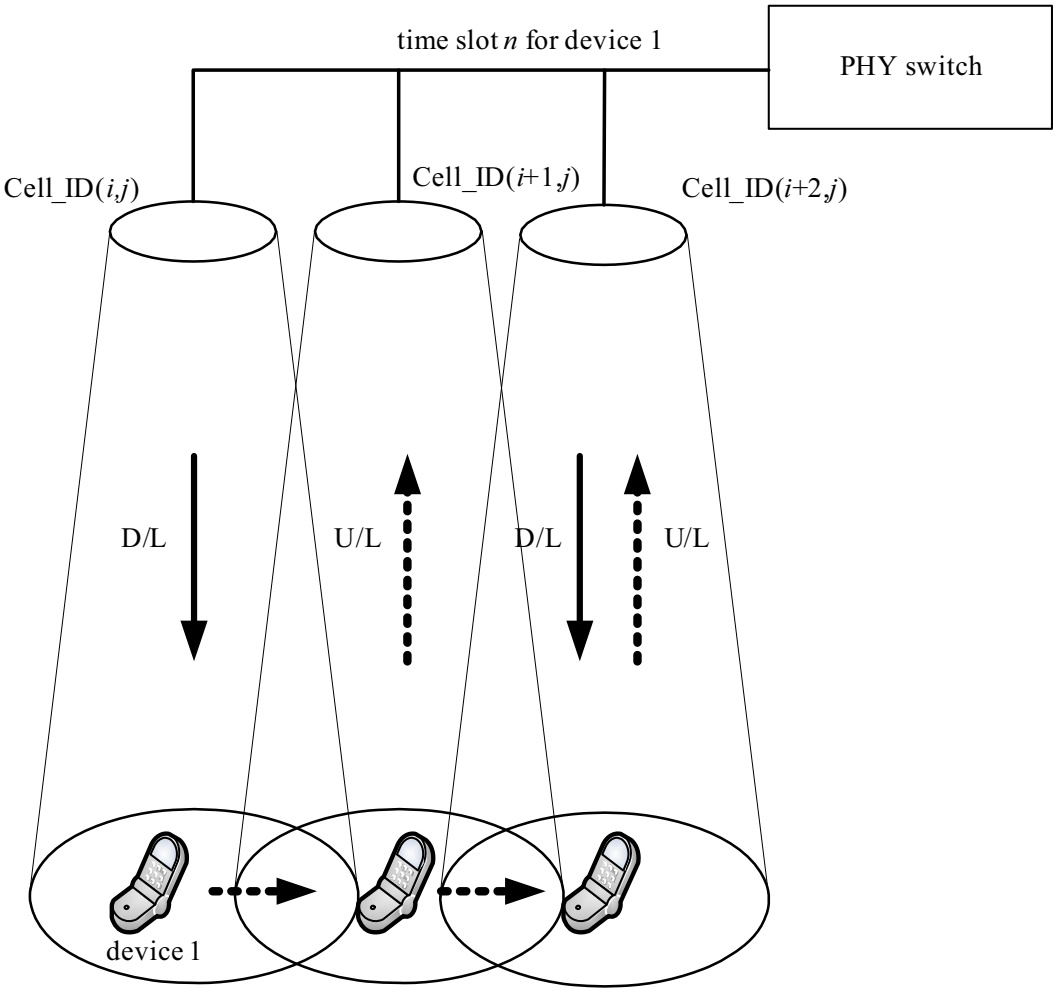


Figure 31—Cell configuration for VLC mobility

Figure 32 shows the mobility support for a device through multiple cells. When device 1 moves out from *Cell_ID(i,j)* to *Cell_ID(i+1,j)*, the coordinator may not receive the uplink transmission (for example, acknowledgment frame or CVD frame) from *Cell_ID(i,j)*. The coordinator may then search for the device through the adjacent cells such as *Cell_ID(i+1,j)* and *Cell_ID(i-1,j)* during the same time slots assigned to device 1 in the superframe. The other devices in *cell_ID(i,j)* will continue communication in the same cell. The coordinator may also expand the cell size in order to provide coverage for mobility of the device. The coordinator can decide on the new cell selection for the device on receiving the uplink transmission from device 1. Thus, if the coordinator can resume communication with the device in *cell_ID(i+1,j)*, the

coordinator DME may set the PHY switch to use *cell_ID(i+1,j)* for device 1 during the time slots allocated for device 1 and then switch back to *cell_ID(i,j)* to service any existing devices in *cell_ID(i,j)* in the remaining time slots. The searching process can be terminated if the device is not found within the link timeout period, defined in MAC PIB attribute *macLinkTimeOut* in Table 60, and the device can then be considered to be disassociated from the coordinator.

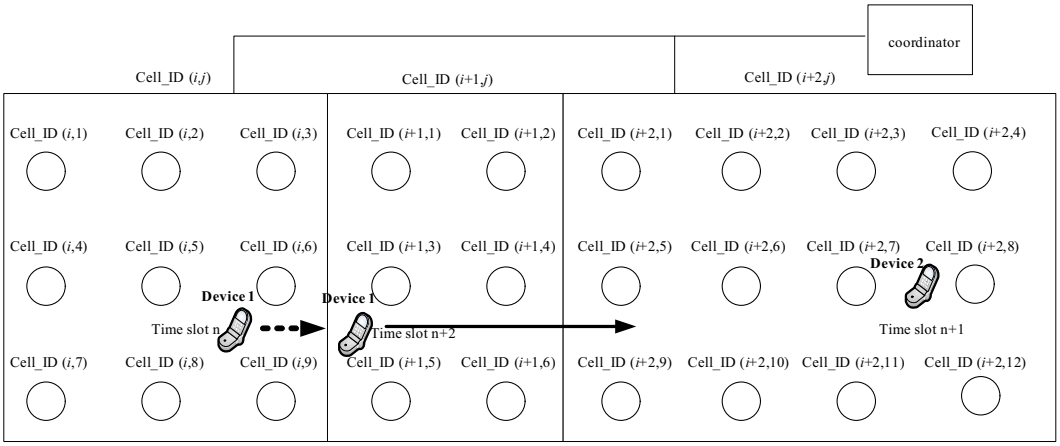


Figure 32—Mobility support for a device through multiple cells

5.1.11.2 Cell configuration during superframe

In order to support access for new devices through the entire superframe, the entire optical media shall be configured to a single macro cell during the beacon and CAP periods. Once devices are discovered and associated, the cell sizes and positions can be determined and the cell structure can be applied to the individual device(s) for communication, as shown in Figure 33.

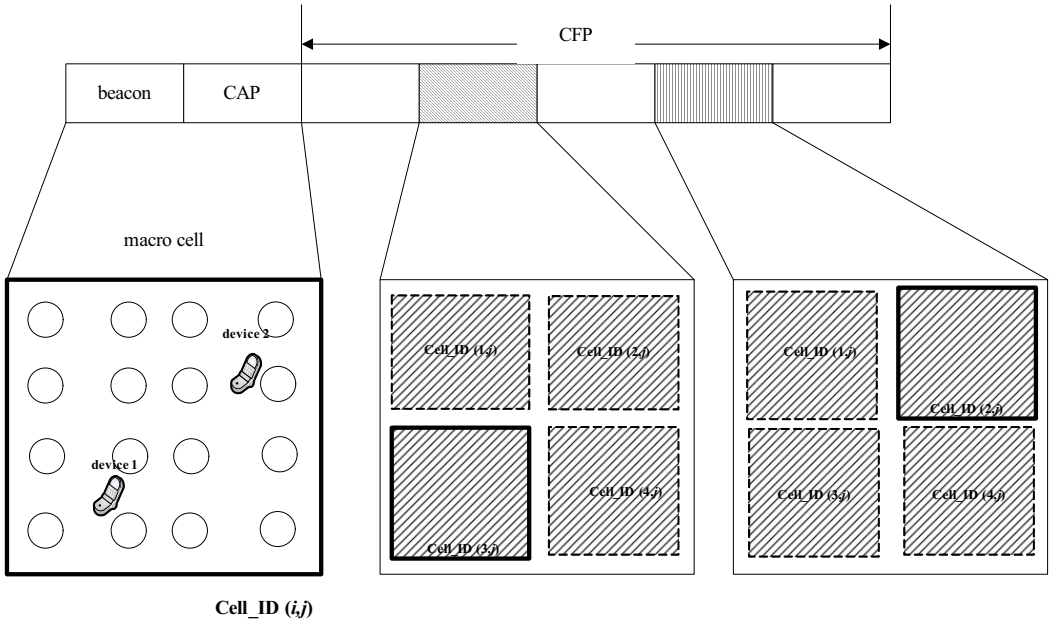


Figure 33—Superframe configuration for mobility support

5.1.11.3 Cell size and location search procedure

Once a device is associated with a coordinator using the beacon and CAP, the coordinator may establish the size and location of the cell in order to service the new device in the CFP with a smaller cell size. In order to determine the size and location of the cell, the coordinator first sets the *cellSearchEn* bit in the superFrame specification field of the beacon frame as defined in Figure 49. If the *cellSearchEn* bit is set, the *cellSearchLength* is transmitted as an additional field in the beacon frame, as shown in Figure 46. If the *cellSearchEn* bit is set, the coordinator readjusts its superframe GTS allocation to ensure the first *cellSearchLength* slots of the CFP are allocated for cell size and location search.

The first *cellSearchLength* slots are used as visibility slots by the coordinator and the devices. During the first *cellSearchLength* slots, the coordinator sequentially cycles through the *cellSearchLength* cells and transmits CVD frames in all the cells. Figure 34 shows an example of the sequential search for 4 cells. CS1 to CS4 are the 4 cell search slots that are made available for searching via setting the *cellSearchLength* to 4 and setting the *cellSearchEn* bit in the beacon frame.

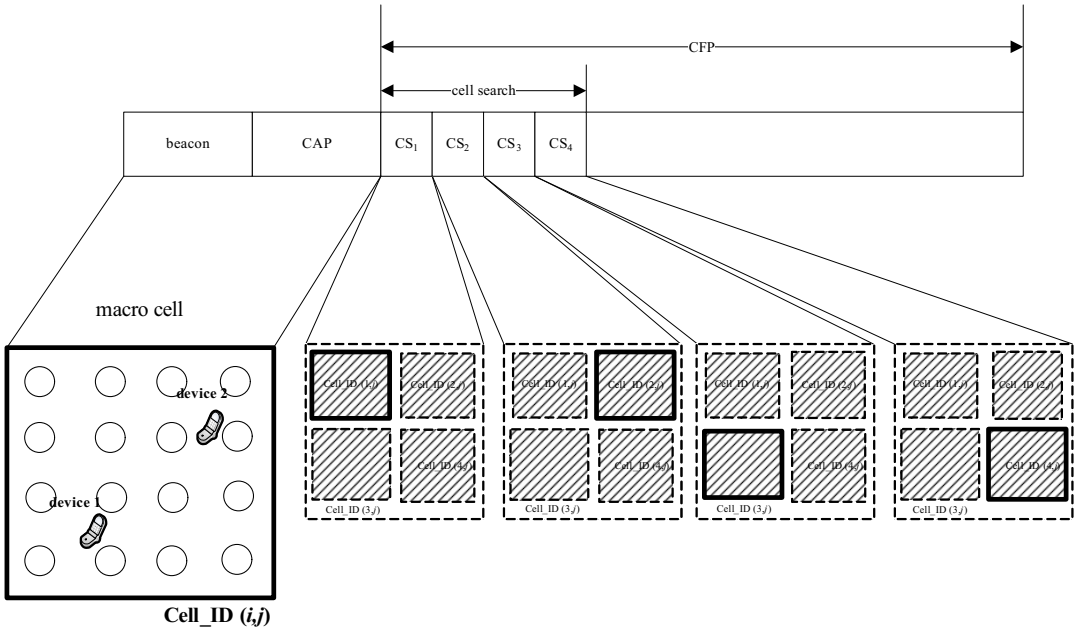


Figure 34—Cell size and location search procedure

If a device receives a beacon with the *cellSearchEn* bit set to 1, the device shall also continuously transmit CVD frames during the *cellSearchEn* slots while also monitoring the CVD frame reception from the coordinator. The device shall note the WQI during each of the *cellSearchLength* slots and shall report this information back to the coordinator using the mobility notification command frame, as described in 5.3.12.

The coordinator makes the determination of the cell sizes and location based on the information from the mobility notification command and its own reception of the CVD frames from the device during the cell search slots.

5.1.12 Color function support

The CVD frame, using various colors, can be used to display various statuses of a device. The colors mapped for each status of the devices are based on the *phyColorFunction* (see Table 100). The colors chosen for different statuses are left to the discretion of the implementer. Multiple statuses may choose the same color, depending on the number of colors supported by the device. The use of color function through the CVD frame has the potential to change the color of the emitted light.

5.1.12.1 CVD frame usage for MAC state indication

The CVD frames are used between state changes to provide visual information to the user regarding the communication status. The MLME primitives for association (see 6.3.1.1), scan (see 6.3.8.1), and disassociation (see 6.3.2.1) are used to support this functionality. The corresponding colors, as described in Table 5 can be used to display various states of a device. The MAC PIB attributes, *macDuringASSOCColor*, *macDuringDISASSOCColor*, and *macDuringSCANColor* as shown in Table 60, are used for the color assignment of the CVD frame when the CVD frame is sent to indicate the MAC state during the association, disassociation, or scan process.

Table 5—Color table for MAC state indication

State	Color choice	Color resolution range
scan	Color “A”	0–255
association	Color “B”	0–255
disassociation	Color “C”	0–255

For example, the device sends an association request to the coordinator (see Figure 35) and indicates this to the user with a chosen color. This information about the color choice is communicated using the MLME-ASSOCIATE.request primitive as in 6.3.1.1.

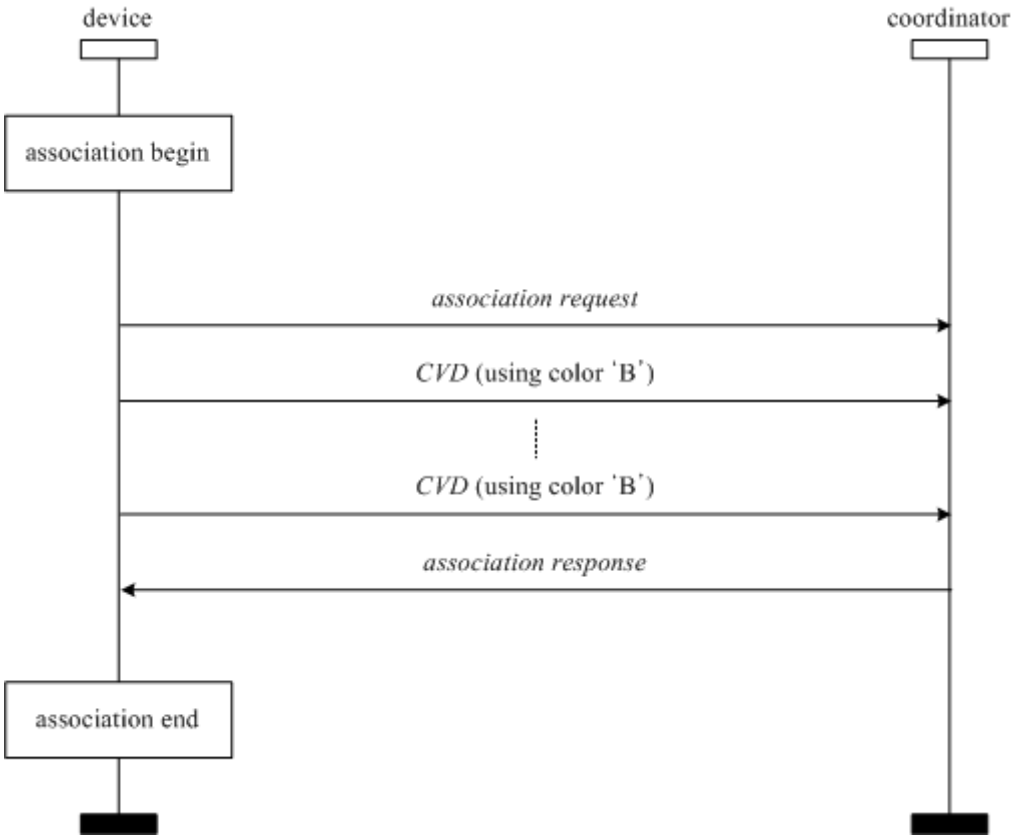


Figure 35—MSC when color function for association indication is invoked

5.1.12.2 CVD frame usage for acknowledgment indication

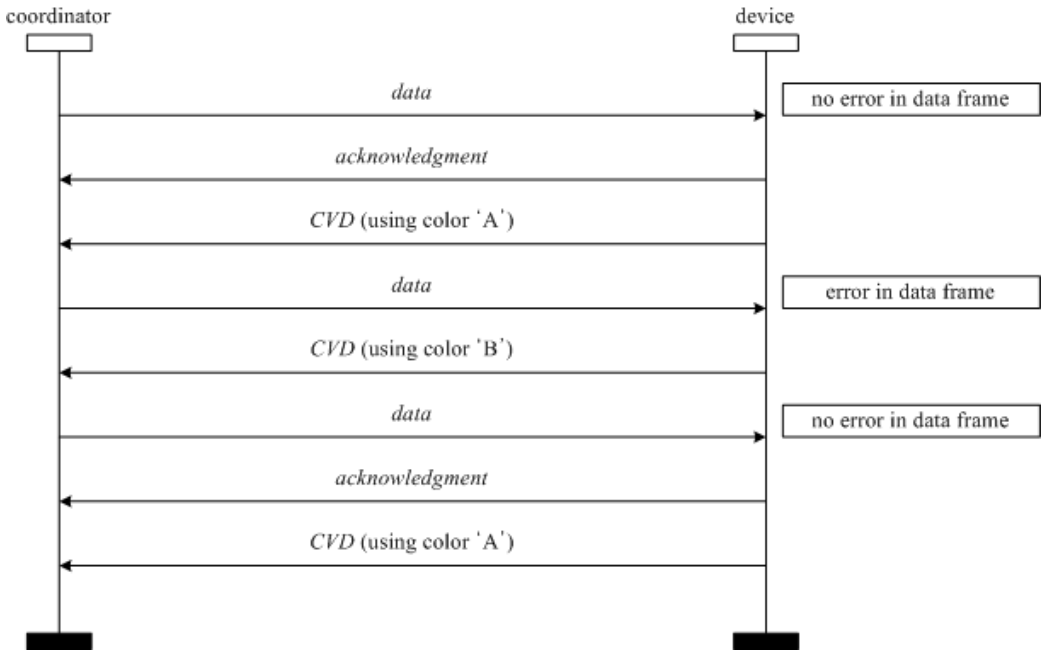


Figure 36—CVD frame usage for acknowledgment indication

Figure 36 shows an example of how the user can infer whether a receiver successfully receives data or not. According to this figure, the device sends a CVD frame after the acknowledgment (ACK) frame has been sent. The CVD frame can indicate that the received data has errors or is error-free, based on the choice of colors. The MAC PIB attribute, *macColorReceived* as shown in Table 60, is used for the color assignment of the CVD frame when the ACK frame is sent and the color function for the ACK state indication is achieved by the CVD frame. The MAC PIB attribute, *macColorNotReceived* as shown in Table 60, is used for the color assignment of the CVD frame when the ACK frame is not sent but the color function for the non-ACK state indication is achieved by the CVD frame.

5.1.12.3 CVD frame usage for channel quality indication

Table 6 describes how the user can infer the quality of the data transmission or the communication quality through the CVD frame. The communication quality may be obtained by various metrics. For example, frame-error ratio (FER) statistics can be averaged over multiple frames. The *ppduLinkQuality* of 9.3.3 (PD-DATA.indication) can also be used for this purpose. This information can help provide misalignment indication to the user. Different colors can be used to indicate different states of misalignment. The choice of the colors and the FER range is left to the implementer and is out of scope of the standard.

Table 6—Color table for channel quality indication

Color of CVD frame	Channel quality
Color “A”	Current FER < FER #1
Color “B”	FER #1 ≤ FER < FER #2
Color “C”	Current FER ≥ FER #2

5.1.12.4 CVD frame usage for file-transfer status indication

Figure 37 shows an example of how the user can infer the remaining or transferred file size through the color of the CVD frame. As shown in the example of Figure 37, the coordinator transfers files to the device. Different stages of the file transfer process can be represented with different choices of colors. In order to use this indication, the device needs to know the total file size to be transmitted. The remaining file size can be obtained by subtracting the transferred file size from the total file size. The MAC PIB attribute, *macCFAppColor* as shown in Table 60, is used for the color assignment of the CVD frame when the CVD frame is sent to indicate the application-dependent information, such as the file-transfer status.

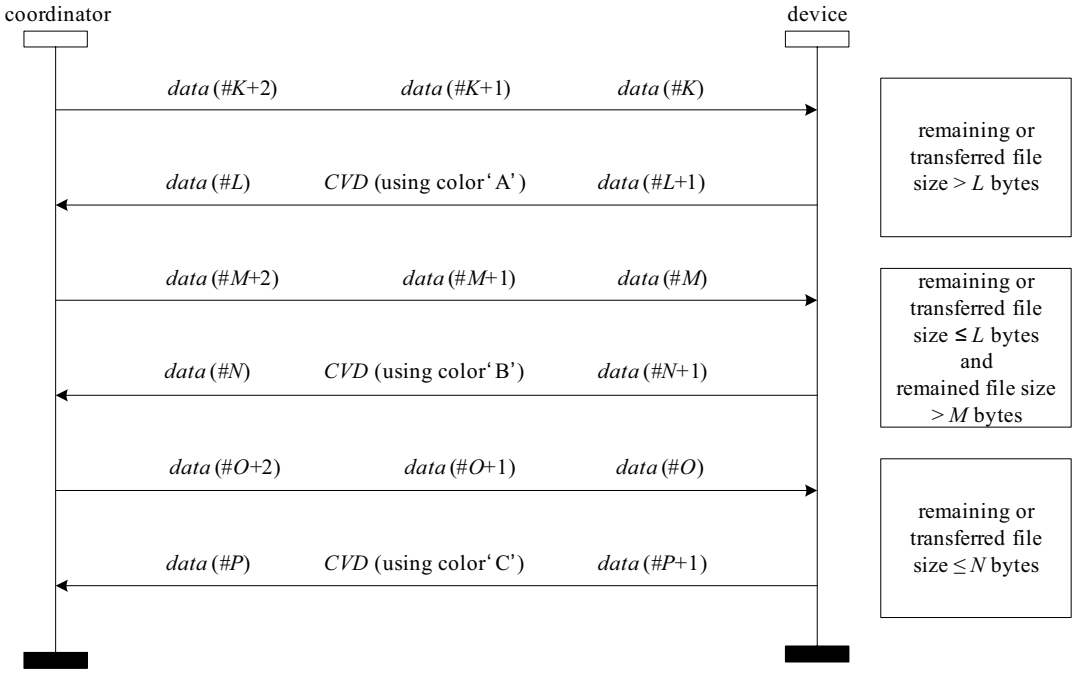


Figure 37—Example of MSC for CVD frame usage for file-transfer status indication

5.1.12.5 Generic color assignment mechanism

The color function can be used beyond the applications as described from 5.1.12.1 to 5.1.12.4. The colors to support the various color functions shall be chosen from the *phyColorFunction* PHY PIB attribute as shown in Table 100, using the MLME-SET.request and PLME-SET.request primitives available to the DME shown in Figure 3.

5.1.13 Color stabilization

When a device joins a network (administrated by a coordinator), it advertises its capability of color stabilization in CSK links as shown in Table 16. It is assumed that at least one link is functioning as a CSK bidirectional link. Otherwise, no color-stabilization functionality is invoked in the network. Also, for the sake of simplicity, it is assumed that only the device will be requested to send color-stabilization updates.

The device and the coordinator go through the steps of association as in 5.1.4. Upon the issuance of a MLME-ASSOCIATE.request the device sends an Association request, among other things advertising its capability for Rx-side CSK-color stabilization.[6] Upon reception of this request, the coordinator MLME creates an MLME-ASSOCIATE.indication to the next higher layer in the coordinator. There, a decision is made whether and where color stabilization will be invoked. If the link to be established is a duplex CSK link, the coordinator can also choose to stabilize the color of the device Tx. (As already mentioned, we are describing the case of color stabilization of the coordinator, but the other possible cases can be inferred from the description in a straight-forward manner). After this decision has been made, the pertinent capability negotiation response field in the MLME-ASSOCIATE.response is set according to Table 32 and the pertinent information is then translated by the coordinator MLME into the MAC association response message. Upon reception of this message, the device MLME creates the MLME-ASSOCIATE.confirm and sends it to the next higher layer in the device for further processing.

When the coordinator starts sending CVD frames to the device (identified by the pertinent MAC header as shown in 5.1.12), the device sends color stabilization information back to the coordinator. The MAC command frame used for this can be found in 5.3.17. After a time set in the variable *macColorStabilizationTimer*, as shown in Table 60, the current information is sent again from the device to the coordinator. If the coordinator wants to change the time between two such updates, it can send a color-stabilization-timer notification command (see 5.3.16) to the device, upon which the device MLME sets the pertinent timer, which is not further described in this standard.

Upon dissociation, the *macColorStabilization* variable is set back to its default value '00'.

5.1.14 Visibility and dimming support

The standard supports visibility for the following purposes:

a) Alignment (device discovery, negotiation, connection)
b) Visible guiding for user alignment
c) Infrastructure continuous light output
d) Blinking for unexpected interference, disconnection warnings

5.1.14.1 Visibility pattern

The MAC passes the visibility pattern requirement to the PHY layer via the PLME interface using the *phyDim* PIB attribute as shown in Table 100. Sending an idle pattern is a mandatory requirement for infrastructure during idle or receive operation to ensure continuous illumination. Sending an idle pattern is optional for the mobile device.

5.1.14.2 Extended preamble mode for visibility

The MAC provides an extended preamble mode for visibility. The advantage of this mode is to provide additional time for synchronization while simultaneously providing visibility.

[6]If not otherwise mentioned, an acknowledgment shall be sent after the reception of each message.

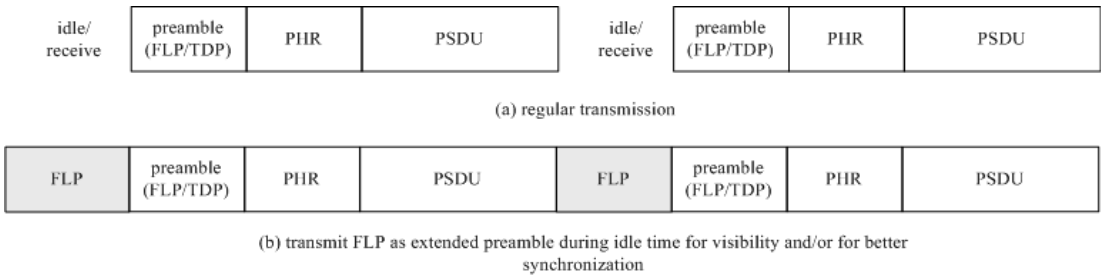


Figure 38—Extended preamble mode provided by the MAC

The MAC uses the knowledge of the idle time and may increase the number of preamble repetitions during the frame transmission to cover the idle time period. The extended preamble is made continuous to the existing preamble of the next frame transmission. There is a possibility that the idle time may not be an integral multiple of the preamble length. In such cases, it is acceptable to transmit a fraction of the preamble (the latter part) in order to maintain visibility. This fraction of the preamble can be called a truncated preamble.

The MAC can choose to either transmit a idle pattern or an extended preamble in the idle mode during regular operation. The choice is made by the DME and is indicated to the PHY via PLME access to the PHY PIB attribute *phyUseExtendedMode* (see Table 100).

The fast locking pattern (FLP) part of the preamble sequence (1010...) shall be used in the extended preamble mode, as shown in Figure 39. Since idle time is not an integral multiple of the preamble, only a fraction of the preamble pattern such as '010' can be sent to complete the idle time.

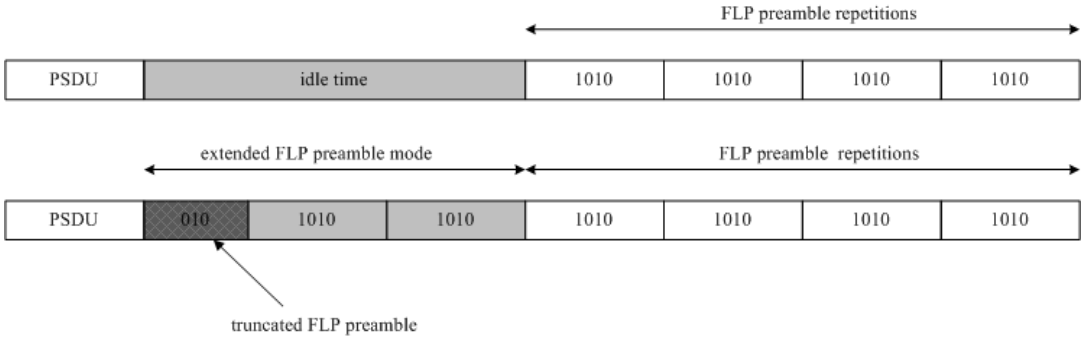


Figure 39—Truncated preamble in extended preamble mode for utilizing idle time for visibility

NOTE—the dimming requirements shall be met, even during the preambles. The implementer needs to be mindful of this when doing repetitions of the preamble.[7]

5.1.14.3 Transmitting visibility pattern during uplink for star topology mode

For the star topology mode, assuming the visibility pattern is sent “in-band” as described in the modulation domain (see 4.3), the point-and-shoot visibility signal from the mobile device cannot be transmitted

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

continuously since multiple users could be pointing to the infrastructure fixed coordinator. This makes the visibility signal difficult to attain due to the low duty cycle. Hence, the knowledge of idle periods (unused slots) is transmitted by the beacons and the mobile device uses the idle periods for transmitting the visibility pattern to the fixed coordinator. All mobile devices talking to a coordinator can share the empty slots for the CVD frame transmission during uplink.

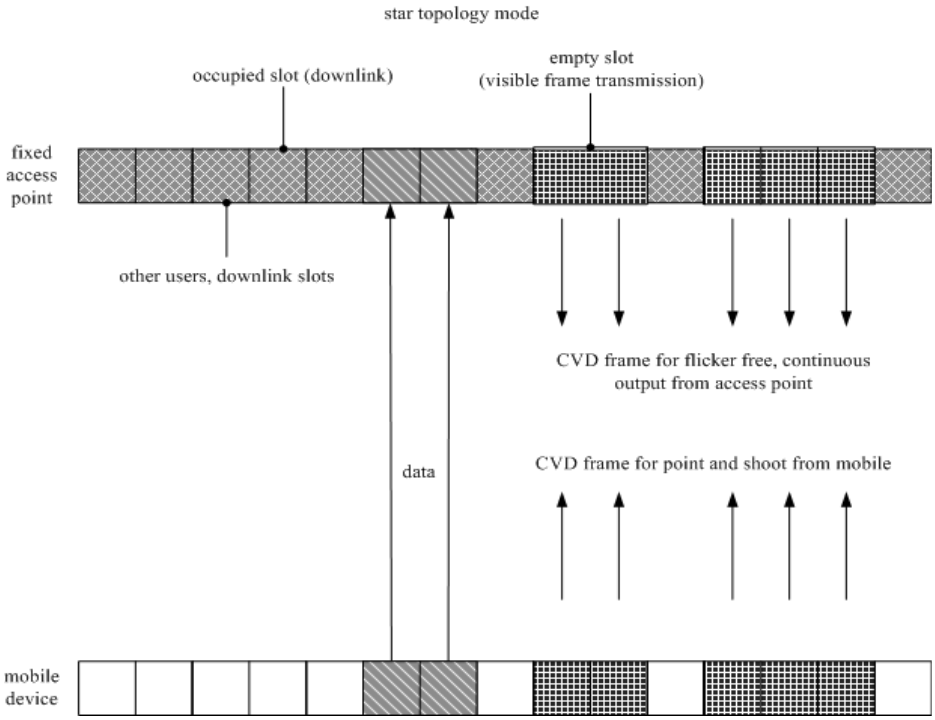


Figure 40—Usage of CVD frames during star topology operation

5.1.14.4 Dimming override capability

This standard supports bypassing the dimmer functionality during VLC operation. The dimmer control can be set to maximum brightness to facilitate VLC communication. As soon as the VLC communication is completed, the dimmer regains control of the optical source driver and resumes normal operation.

A dimmer override capability request signal is added to the MLME SAP and provided to the external dimmer interface, using the MAC PIB attribute, *macDimOverrideRequest*, as shown in Table 60. This dimmer override request attribute shall be set to ‘1’ during VLC operation and shall be set to ‘0’ after the communication has been completed. The dimmer circuit can decide whether to accept or reject this request. The response to this dimmer override request signal by the external dimmer circuit is out-of-scope of this standard. The MLME-GET (see 6.3.4) and MLME-SET(see 6.3.10) primitives are used to read and write PIB attributes for dimming.

5.1.14.5 PWM signal override

A PWM signal override request signal is added to the MLME SAP, using the MAC PIB attribute, *macDimPWMOverrideRequest*, as defined in Table 60 and provided to the external dimmer interface. This PWM override request attribute shall be set to ‘1’ to inform the dimmer circuit that the VLC PHY will be responsible for dimming and to disable any PWM circuit present in the dimmer. The duty cycle for dimming

is then driven by modulation mode provided by the VLC PHY (such as VPPM). The response to this PWM override request signal by the external dimmer circuit is out-of-scope of this standard. The MLME-GET (see 6.3.4) and MLME-SET (see 6.3.10) primitives are used to read and write PIB attributes for dimming.

5.1.14.6 MAC layer transmission adjustment for dimming

Referring to Figure 41, the infrastructure MAC adjusts the data transmission to match the duty cycle requirements from the dimmer.

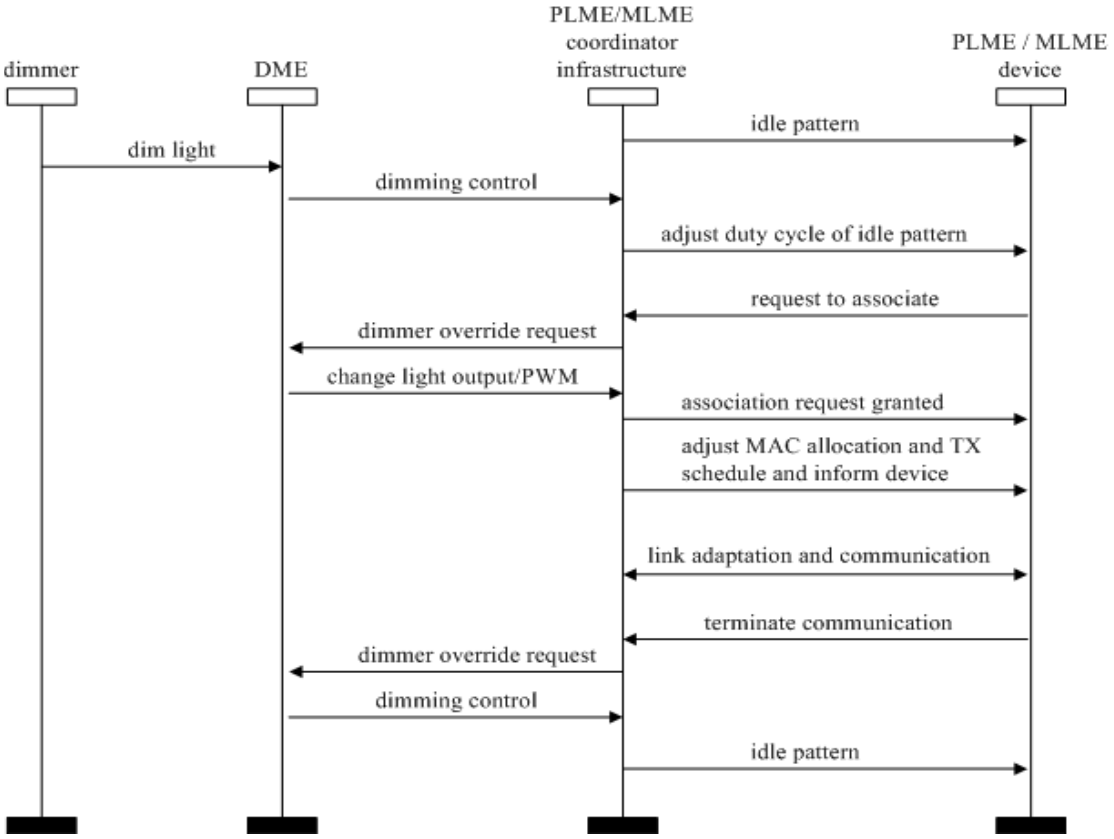


Figure 41—MSC for dimming

5.1.14.7 Device discovery and association in the presence of dimming and visibility

The visibility pattern can help with device discovery when the idle pattern or the data has been modified because of the PHY and MAC layer modulation changes to support dimming. Based on the dimming pattern change and duty cycle, the VLC device may choose to associate with a different coordinator that is currently not being dimmed or has a higher duty cycle (more illumination). The visibility pattern is uncoded as shown in Figure 59. The header for the CVD frame is sent at the lowest data rate corresponding to the currently

negotiated clock rate. Figure 42 shows an example of using the visibility pattern as a signal to establish the best connectivity to an infrastructure device.

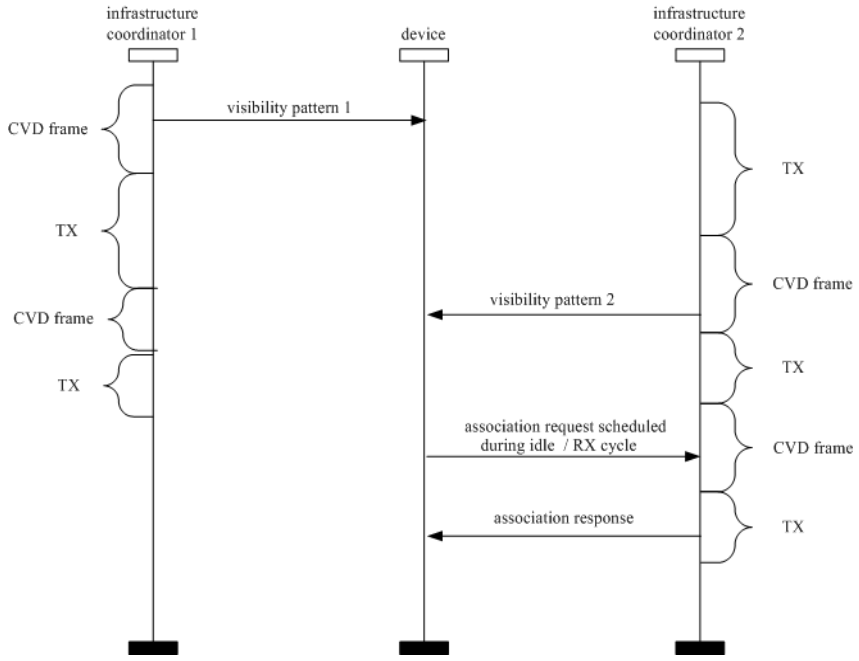


Figure 42—Example of using the visibility pattern to establish best connectivity to an infrastructure device

5.1.14.8 Link adaptation for dimming support

Dimming requirements of the infrastructure should be notified to VLC RX device, so that the VLC receiver can adapt to the dimming pattern of the data when VPPM is used. The infrastructure coordinator may receive an external dimming request. A dimming adaptation timer is used that delays the time between the dimming request and the actual dimming of the light source. With this knowledge of an incoming dimming, the link between the devices can be adapted to work at a new (lower) data rate (if dimmed) without requiring the link to be interrupted or possible link failure. This dimming adaptation is indicated and supported by the MAC dimming notification command frame in 5.3.10. Figure 43 shows an example of delay dimming and adapt resources for uninterrupted link.

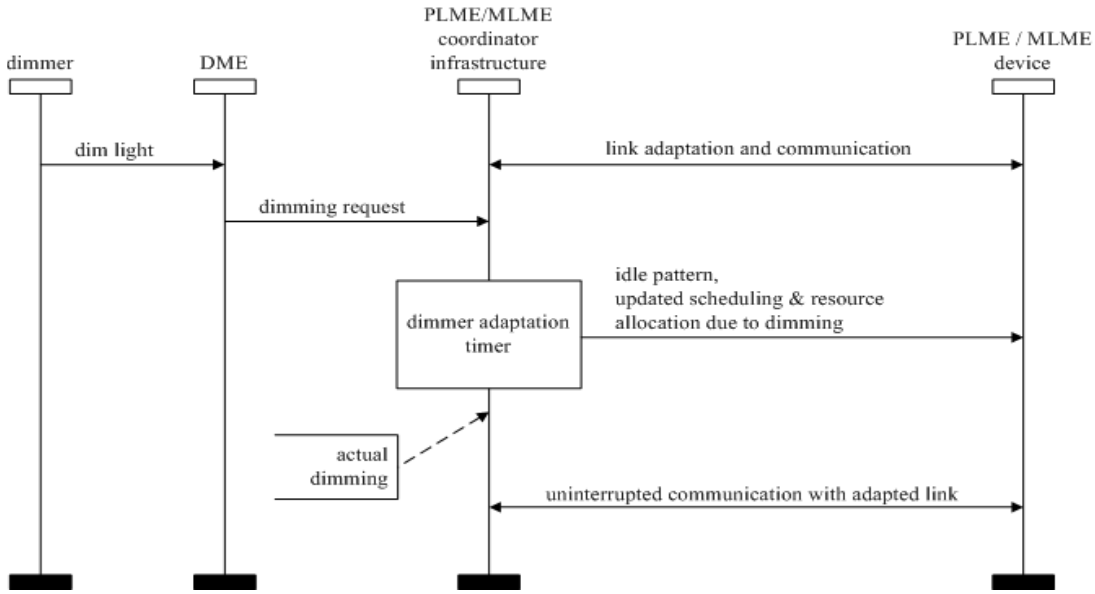


Figure 43—Usage of MAC layer to delay dimming and adapt resources for uninterrupted link

5.2 MAC frame formats

This subclause specifies the format of the MAC frame (MPDU). Each MAC frame consists of the following basic components:

a) A MHR, which comprises frame control, sequence number, address information, and security-related information.
b) A MSDU, of variable length, which contains information specific to the frame type. Acknowledgment frames do not contain a payload.
c) A MFR, which contains a FCS.

The frames in the MAC sublayer are described as a sequence of fields in a specific order. All frame formats in this subclause are depicted in the order in which they are transmitted by the PHY, from left to right, where the left most bit is transmitted first in time. Bits within each field are numbered from 0 (left most and least significant) to $k - 1$ (right most and most significant), where the length of the field is k bits. Fields that are longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits.

For every MAC frame, all reserved bits shall be ignored upon receipt.

5.2.1 General MAC frame format

The MAC frame format is composed of a MHR, a MSDU, and a MFR. The fields of the MHR appear in a fixed order; however, the addressing fields may not be included in all frames. The general MAC frame shall be formatted as illustrated in Figure 44.

<table>
<tr><td>Octets: 2</td><td>1</td><td>0/2</td><td>0/2/8</td><td>0/2</td><td>0/2/8</td><td>0/5/6/10/ 14</td><td>variable</td><td>2</td></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destina-tion VPAN Identifier</td><td>Destination Address</td><td>Source VPAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td rowspan="2">Frame Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="5">Addressing fields</td></tr>
<tr><td colspan="7">MHR</td><td>MSDU</td><td>MFR</td></tr>
</table>

Figure 44—General MAC frame format

5.2.1.1 Frame control field

The frame control field is 2 octets in length and contains information defining the frame type, addressing fields, and other control flags. The frame control field shall be formatted as illustrated in Figure 45. Reserved bits are set to zero on transmission and ignored on reception.

Bits: 0–1	**2–5**	**6–8**	**9**	**10**	**11**	**12–13**	**14–15**
Frame Version	Reserved	Frame Type	Security Enabled	Frame Pending	Ack Request	Dest Addressing Mode	Source Addressing Mode

Figure 45—Format of the frame control field

5.2.1.1.1 Frame Version subfield

The Frame Version subfield specifies the version number corresponding to the frame. This subfield shall be set to 0b00 to indicate a frame compatible with IEEE Std 802.15.7. All other subfield values shall be reserved for future use.

5.2.1.1.2 Frame type subfield

The Frame Type subfield shall be set to one of the nonreserved values listed in Table 7.

Table 7—Values of the Frame Type subfield

Frame type value $b_2\ b_1\ b_0$	**Description**
000	Beacon
001	Data
010	Acknowledgment
011	Command

Table 7—Values of the Frame Type subfield *(continued)*

Frame type value b_2 b_1 b_0	Description
100	CVD
101–111	Reserved

5.2.1.1.3 Security Enabled subfield

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

5.2.1.1.4 Frame Pending subfield

The Frame Pending subfield is 1 bit in length and shall be set to one if the device sending the frame has more data for the recipient. This subfield shall be set to zero otherwise (see 5.1.7.3).

The Frame Pending subfield shall be used only in beacon frames or frames transmitted either during the CAP by devices operating on a beacon-enabled VPAN or at any time by devices operating on a nonbeacon-enabled VPAN.

At all other times, it shall be set to zero on transmission and ignored on reception.

5.2.1.1.5 Acknowledgment Request subfield

The Acknowledgment Request subfield is 1 bit in length and specifies whether an acknowledgment is required from the recipient device on receipt of a data or MAC command frame. If this subfield is set to one, the recipient device shall send an acknowledgment frame only if, upon reception, the frame passes the third level of filtering as shown in 5.1.7.2. If this subfield is set to zero, the recipient device shall not send an acknowledgment frame.

5.2.1.1.6 Destination Addressing Mode subfield

The Destination Addressing Mode subfield shall be set to one of the nonreserved values listed in Table 8.

If this subfield is equal to zero and the Frame Type subfield does not specify that this frame is an acknowledgment or beacon frame, the Source Addressing Mode subfield shall be nonzero, implying that the frame is directed to the VLC coordinator with the VPAN identifier as specified in the Source VPAN Identifier field. If this subfield is equal to 01, the Source Addressing Mode subfield shall be equal to 01, implying that the frame is a broadcast frame, and no source or destination address fields are present in the frame.

5.2.1.1.7 Source Addressing Mode subfield

The Source Addressing Mode subfield shall be set to one of the nonreserved values listed in Table 8.

If this subfield is equal to zero and the Frame Type subfield does not specify that this frame is an acknowledgment frame, the Destination Addressing Mode subfield shall be nonzero, implying that the frame has originated from the coordinator with the VPAN identifier as specified in the Destination VPAN Identifier field.

Table 8—Possible values of the Destination Addressing Mode and Source Addressing Mode subfields

Addressing mode value b_1 b_0	Description
00	VPAN identifier and address fields are not present.
01	No address field (broadcast only mode with no address fields present). Addresses with all ones of 16 bits or 64 bits are defined as broadcast.
10	Address field contains a 16-bit short address.
11	Address field contains a 64-bit extended address.

If this subfield is equal to 01, the Source Addressing Mode subfield shall be equal to 01, implying that the frame is a broadcast frame, and no source or destination address fields are present in the frame.

5.2.1.2 Sequence Number field

The Sequence Number field is 1 octet in length and specifies the sequence identifier for the frame.

For a beacon frame, the Sequence Number field shall specify a BSN. For a data, acknowledgment, or MAC command frame, the Sequence Number field shall specify a DSN that is used to match an acknowledgment frame to the data or MAC command frame.

5.2.1.3 Destination VPAN Identifier field

The Destination VPAN Identifier field, when present, is 2 octets in length and specifies the unique VPAN identifier of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast VPAN identifier, which shall be accepted as a valid VPAN identifier by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode subfield of the frame control field is 10 or 11.

5.2.1.4 Destination Address field

The Destination Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Destination Addressing Mode subfield of the frame control field, see 5.2.1.1.6, and specifies the address of the intended recipient of the frame. A 16-bit value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid 16-bit short address by all devices currently listening to the channel.

This field shall be included in the MAC frame only if the Destination Addressing Mode subfield of the frame control field is nonzero.

5.2.1.5 Source VPAN Identifier field

The Source VPAN Identifier field, when present, is 2 octets in length and specifies the unique VPAN identifier of the originator of the frame. This field shall be included in the MAC frame only if the Source Addressing Mode and VPAN ID Compression subfields of the frame control field are nonzero and equal to zero, respectively.

The VPAN identifier of a device is initially determined during association on a VPAN, but may change following a VPAN identifier conflict resolution as discussed in 5.1.3.

5.2.1.6 Source Address field

The Source Address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the Source Addressing Mode subfield of the frame control field, as shown in 5.2.1.1.7, and specifies the address of the originator of the frame. This field shall be included in the MAC frame only if the Source Addressing Mode subfield of the frame control field is 10 or 11.

5.2.1.7 Auxiliary Security Header field

The Auxiliary Security Header field has a variable length and specifies information required for security processing, including how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (see 7.5.1). This field shall be present only if the Security Enabled subfield is set to one. For details on formatting, see 7.4.

5.2.1.8 Frame Payload field

The Frame Payload field has a variable length and contains information specific to individual frame types. If the Security Enabled subfield is set to one in the frame control field, the frame payload is protected as defined by the security suite selected for that frame.

5.2.1.9 FCS field

The FCS field is 2 octets in length and is explained in Annex C. The FCS is calculated over the MHR and MSDU parts of the frame. The FCS shall be only generated for payloads greater than zero bytes.

5.2.2 Format of individual frame types

Five frame types are defined: beacon, data, acknowledgment, command, and CVD. These frame types are discussed in 5.2.2.1 through 5.2.2.4.3.

5.2.2.1 Beacon frame format

The beacon frame shall be formatted as illustrated in Figure 46.

Octets: 2	**1**	**4/10**	**0/5/6/10/14**	**2**	**variable**	**variable**	**0/1**	**variable**	**2**
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Superframe Spec	GTS fields (Figure 47)	Pending address fields (Figure 48)	cellSearch Length	Beacon Payload	FCS
MHR				MSDU					MFR

Figure 46—Beacon frame format

The GTS fields shall be formatted as illustrated in Figure 47, and the pending address fields shall be formatted as illustrated in Figure 48.

Octets: 1	0/1	variable
GTS Specification	GTS Directions	GTS List

Figure 47—Format of the GTS information fields

Octets: 1	variable
Pending Address Specification	Address List

Figure 48—Format of the pending address information fields

The order of the fields of the beacon frame shall conform to the order of the general MAC frame as illustrated in Figure 44.

5.2.2.1.1 Beacon frame MHR fields

The MHR for a beacon frame shall contain the frame control field, the Sequence Number field, the Source VPAN Identifier field, and the Source Address field.

In the frame control field, the Frame Type subfield shall contain the value that indicates a beacon frame, as shown in Table 7, and the Source Addressing Mode subfield shall be set as appropriate for the address of the coordinator transmitting the beacon frame. If protection is used for the beacon, the Security Enabled subfield shall be set to one. If a broadcast data or command frame is pending, the frame pending subfield shall be set to one. All other subfields shall be set to zero by the sender and ignored on reception.

The Sequence Number field shall contain the current value of *macBSN*.

The addressing fields shall comprise only the source address fields. The Source VPAN Identifier and Source Address fields shall contain the VPAN 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.

5.2.2.1.2 Superframe Specification field

The Superframe Specification field shall be formatted as illustrated in Figure 49.

Bits: 0–3	4-7	8-11	12	13	14	15
Beacon Order	Superframe Order	Final CAP Slot	Reserved	VPAN Coordinator	Association Permit	cellSearchEn

Figure 49—Format of the Superframe Specification field

The Beacon Order subfield shall specify the transmission interval of the beacon. Refer to 5.1.1.1 for an explanation of the relationship between the beacon order and the beacon interval.

The Superframe Order subfield shall specify the length of time during which the superframe is active (i.e., receiver enabled), including the beacon frame transmission time. Refer to 5.1.1.1 for an explanation of the relationship between the superframe order and the superframe duration.

The Final CAP Slot subfield specifies the final superframe slot utilized by the CAP. The duration of the CAP, as implied by this subfield, shall be greater than or equal to the value specified by *aMinCAPLength*. However, an exception is allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance, as in 5.2.2.1.3.

The VPAN Coordinator subfield shall be set to one if the beacon frame is being transmitted by the coordinator. Otherwise, the VPAN Coordinator subfield shall be set to zero.

The Association Permit subfield shall be set to one if *macAssociationPermit* is set to TRUE (i.e., the coordinator is accepting association to the VPAN). The association permit bit shall be set to zero if the coordinator is currently not accepting association requests on its network.

If the cellSearchEn bit is set, the cellSearchLength is transmitted as an additional field in the beacon frame, as shown in Figure 46.

5.2.2.1.3 GTS Specification field

The GTS Specification field shall be formatted as illustrated in Figure 50.

Bits: 0–2	3–6	7
GTS Descriptor Count	Reserved	GTS Permit

Figure 50—Format of the GTS Specification field

The GTS Descriptor Count subfield specifies the number of 3-octet GTS descriptors contained in the GTS List field of the beacon frame. If the value of this subfield is greater than zero, the size of the CAP shall be allowed to dip below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length caused by the inclusion of the subfield. If the value of this subfield is zero, the GTS Directions field and GTS List field of the beacon frame are not present.

The GTS Permit subfield shall be set to one if *macGTSPermit* is equal to TRUE (i.e., the coordinator is accepting GTS requests). Otherwise, the GTS Permit field shall be set to zero.

5.2.2.1.4 GTS Directions field

The GTS Directions field shall be formatted as illustrated in Figure 51.

Bits: 0–6	7
GTS Directions Mask	Reserved

Figure 51—Format of the GTS Directions field

The GTS Directions Mask subfield contains a mask identifying the directions of the GTSs in the superframe. The lowest bit in the mask corresponds to the direction of the first GTS contained in the GTS List field of the beacon frame, with the remainder appearing in the order that they appear in the list. Each bit shall be set to one if the GTS is a receive-only GTS or to zero if the GTS is a transmit-only GTS. GTS direction is defined relative to the direction of the data frame transmission by the device.

5.2.2.1.5 GTS List field

The size of the GTS List field is defined by the values specified in the GTS Specification field of the beacon frame and contains the list of GTS descriptors that represents the GTSs that are being maintained. The maximum number of GTS descriptors shall be limited to seven.

Each GTS descriptor shall be formatted as illustrated in Figure 52.

Bits: 0–15	16–19	20–23
Device Short Address	GTS Starting Slot	GTS Length

Figure 52—Format of the GTS descriptor

The Device Short Address subfield shall contain the short address of the device for which the GTS descriptor is intended.

The GTS Starting Slot subfield contains the superframe slot at which the GTS is to begin.

The GTS Length subfield contains the number of contiguous superframe slots over which the GTS is active.

5.2.2.1.6 Pending Address Specification field

The Pending Address Specification field shall be formatted as illustrated in Figure 53.

Bits: 0–2	3	4–6	7
Number of Short Addresses Pending	Reserved	Number of Extended Addresses Pending	Reserved

Figure 53—Format of the Pending Address Specification field

The Number of Short Addresses Pending subfield indicates the number of 16-bit short addresses contained in the Address List field of the beacon frame.

The Number of Extended Addresses Pending subfield indicates the number of 64-bit extended addresses contained in the Address List field of the beacon frame.

5.2.2.1.7 Address List field

The size of the Address List field is determined by the values specified in the Pending Address Specification field of the beacon frame and contains the list of addresses of the devices that currently have messages pending with the coordinator. The address list shall not contain the broadcast short address 0xffff.

The maximum number of addresses pending shall be limited to seven and may comprise both short and extended addresses. All pending short addresses shall appear first in the list followed by any extended addresses. If the coordinator is able to store more than seven transactions, it shall indicate them in its beacon on a first-come-first-served basis, ensuring that the beacon frame contains at most seven addresses.

5.2.2.1.8 Beacon Payload field

The Beacon Payload field is an optional sequence of up to *aMaxBeaconPayloadLength* octets specified to be transmitted in the beacon frame by the next higher layer. The set of octets contained in *macBeaconPayload* shall be copied into this field.

5.2.2.2 Data frame format

The data frame shall be formatted as illustrated in Figure 54.

Octets: 2	**1**	**(As defined in 5.2.2.2.1)**	**0/5/6/10/14**	**variable**	**2**
frame control	Sequence Number	Addressing fields	Auxiliary Security Header	Data Payload	FCS
MHR				MSDU	MFR

Figure 54—Data frame format

The order of the fields of the data frame shall conform to the order of the general MAC frame as illustrated in Figure 44.

5.2.2.2.1 Data frame MHR fields

The MHR for a data frame shall contain the frame control field, the Sequence Number field, the destination VPAN identifier/address fields, and/or the source VPAN identifier/address fields.

In the frame control field, the Frame Type subfield shall contain the value that indicates a data frame, as shown in Table 7. If protection is used for the data, the Security Enabled subfield shall be set to one. All other subfields shall be set appropriately according to the intended use of the data frame. All reserved subfields shall be set to zero by the sender and ignored on reception.

The Sequence Number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the frame control field.

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

5.2.2.2.2 Data Payload field

The payload of a data frame shall contain the sequence of octets that the next higher layer has requested the MAC sublayer to transmit. The data type field is 1 byte and is explained in Table 9.

Table 9—Data Payload field

Bits 0–1	Bits 2–7	variable
00—Single 01—Packed 10—Burst 11—Reserved	Number of PPDUs per data frame	Data payload

The data type field mentions the format used for sending the data—single, packed, or burst. It also mentions the number of PPDUs that are associated for this data frame.

The payload of a data frame shall contain the sequence of octets that the next higher layer has requested the MAC sub layer to transmit.

5.2.2.3 Acknowledgment frame format

The acknowledgment frame shall be formatted as illustrated in Figure 55.

Octets: 2	1	variable	2
frame control	sequence number	B-ACK frame payload (optional)	FCS
MHR		MSDU	MFR

Figure 55—Acknowledgment frame format

The order of the fields of the acknowledgment frame shall conform to the order of the general MAC frame as illustrated in Figure 44. The sequence number is defined in 5.2.2.1.1.

In B-ACK frames, the DestAddr field is set to the SrcAddr of the frame that requested the B-ACK. The B-ACK frame acknowledges correct or incorrect receipt of the previous sequence of frames and provides information for the transmission of the next sequence of frames as described in 5.2.2.3. The B-ACK frame payload is defined in Figure 56.

Octets: 2	1	1	2	0 – n
buffer size	frame count	reserved	sequence control	frame bitmap

Figure 56—B-ACK frame payload

The Buffer Size field specifies the maximum number of octets in the sum of the frame payloads of all frames in the next B-ACK sequence. The Frame Count field specifies the maximum number of frames in the next B-ACK sequence. The Sequence Control and frame bitmap fields together specify an acknowledgment window of MSDU fragments and their reception status. The Sequence Control field specifies the Sequence Number and Fragment Number that start the acknowledgment window.

Bits: b15–b14	b13–b3	b2–b0
reserved	sequence number	fragment number

Figure 57—B-ACK frame bitmap

The frame bitmap field varies in length. A zero-length frame bitmap field indicates an acknowledgment window of length zero. Otherwise, the least-significant octet of the frame bitmap field corresponds to the MSDU indicated by the Sequence Control field, and each bit of the octet corresponds to a fragment of that MSDU. The least-significant bit in each octet corresponds to the first fragment and successive bits correspond to successive fragments. Successive octets present in the frame bitmap field correspond to successive MSDUs, and each bit corresponds to a fragment of the MSDU. The acknowledgment window ends at fragment seven of the MSDU that corresponds to the most-significant octet in the frame bitmap. For all bits within the frame bitmap, a value of ONE indicates that the corresponding fragment was received in either the current sequence or an earlier one. A value of ZERO indicates that the corresponding fragment was not received in the current sequence (although it may have been received in an earlier one). Bits of the least-significant octet of the frame bitmap field corresponding to fragments prior to the start of the acknowledgment window are undefined. Frames with a Sequence Number earlier than the Sequence Number indicated in the Sequence Control field were not received in the last B-ACK sequence. Such frames were previously received or are no longer expected.

The block ACK is applicable to the packed data type. The bitmap and sequence number is repeated for every frame in the burst mode (multiple frames)

The order of the fields of the acknowledgment frame shall conform to the order of the general MAC frame as illustrated.

The MHR for an acknowledgment frame shall contain only the frame control field and the Sequence Number field.

In the frame control field, the Frame Type subfield shall contain the value that indicates an acknowledgment frame, as shown in Table 7. If the acknowledgment frame is being sent in response to a received data request command, the device sending the acknowledgment frame shall determine whether it has data pending for the recipient. If the device can determine this before sending the acknowledgment frame (see 5.1.7.4.2), it shall set the frame pending subfield according to whether there is pending data. Otherwise, the frame pending subfield shall be set to one. If the acknowledgment frame is being sent in response to either a data frame or another type of MAC command frame, the device shall set the frame pending subfield to zero. All other subfields, except the security enabled subfield, shall be set to zero by the sender and ignored on reception.

The Sequence Number field shall contain the value of the sequence number received in the frame for which the acknowledgment is to be sent.

5.2.2.4 Command frame format

The command frame shall be formatted as illustrated in Figure 58.

The order of the fields of the MAC command frame shall conform to the order of the general MAC frame as illustrated in Figure 44.

5.2.2.4.1 MAC command frame MHR fields

The MHR for a MAC command frame shall contain the frame control field, the Sequence Number field, the destination VPAN identifier/address fields and/or the source VPAN identifier/address fields.

Octets: 2	**1**	**(As defined in 5.2.2.4.1)**	**0/5/6/10/14**	**1**	**variable**	**2**
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Command Frame Identifier	Command Payload	FCS
MHR				MSDU		MFR

Figure 58—Command frame format

In the frame control field, the Frame Type subfield shall contain the value that indicates a MAC command frame, as shown in Table 7. If the frame is to be secured, the Security Enabled subfield of the frame control field shall be set to one and the frame secured according to the process described in 7.5.4. Otherwise the Security Enabled subfield of the frame control field shall be set to zero. All other subfields shall be set appropriately according to the intended use of the MAC command frame. All reserved subfields shall be set to zero by the sender and ignored on reception.

The Sequence Number field shall contain the current value of *macDSN*.

The addressing fields shall comprise the destination address fields and/or the source address fields, dependent on the settings in the frame control field.

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

5.2.2.4.2 Command Frame Identifier field

The Command Frame Identifier field identifies the MAC command being used. This field shall be set to one of the nonreserved values listed in Table 10.

5.2.2.4.3 Command Payload field

The Command Payload field contains the MAC command itself. The formats of the individual commands are described in 5.3.

5.2.2.5 CVD frame format

Octets: 2	**Octet: 2**	**variable**
Frame control	FCS	Visibility pattern
MHR	MFR	

Figure 59—CVD frame

The structure of the CVD frame is as shown in Figure 59. The CVD frame is used to visually provide information on the communication status, such as misalignment between the two devices, transmission direction, or sending data status; the data transmission quality; and the transferred file size and remaining file size. The visibility pattern has no error protection. The length of the visibility pattern shall be set in the PHY header and the FCS shall not include the visibility pattern of the CVD frame. The FCS is only

calculated over the frame control field (MHR) using the cyclic redundancy check (CRC) described in Annex C. The visibility pattern will be generated based on the dimming level requirements and is described in 8.5.1.2. The CVD frame is used by the infrastructure to maintain visibility at all times and by the mobile device for point-and-shoot. The CVD frame can also be used for color stabilization for PHY III as explained in 8.5.4. It should be noted that the CVD frame is not used for communicating the dimming level; rather, the dimming notification command is used for this function as described in 5.3.10.

The CVD frame is sent at the currently negotiated optical clock.

5.3 MAC command frames

The command frames defined by the MAC sublayer are listed in Table 10. A coordinator shall be capable of transmitting and receiving all command frame types, with the exception of the GTS request command, while the requirements for a device are indicated by an "X" in Table 10. A P2P device functioning as a coordinator shall be capable of transmitting and receiving all supported command frames in a device. MAC commands shall only be transmitted in the CAP for beacon-enabled VPANs or at any time for nonbeacon-enabled VPANs.

How the MLME shall construct the individual commands for transmission is detailed in 5.3.1 through 5.3.18. MAC command reception shall abide by the procedure described in 5.1.7.2.

Table 10—Command frames

Command frame identifier	Command name	Device		P2P coordinator		Subclause
		Tx	Rx	Tx	Rx	
0x01	Association request	X		X	X	5.3.1
0x02	Association response		X	X	X	5.3.2
0x03	Disassociation notification	X	X	X	X	5.3.3
0x04	Data request	X		X	X	5.3.4
0x05	VPAN ID conflict notification	X		X	X	5.3.5
0x06	Beacon request					5.3.6
0x07	Coordinator realignment		X	X	X	5.3.7
0x08	GTS request					5.3.8
0x09	Blinking notification					5.3.9
0x0a	Dimming notification		X	X	X	5.3.10
0x0b	Fast link recovery					5.3.11
0x0c	Mobility notification					5.3.12
0x0d	GTS Response					5.3.13
0x0e	Clock rate change notification		X	X	X	5.3.14
0x0f	Multiple channel assignment					5.3.15
0x10	Band hopping					5.1.10.2

Table 10—Command frames *(continued)*

Command frame identifier	Command name	Device		P2P coordinator		Subclause
		Tx	Rx	Tx	Rx	
0x11	Color stabilization timer notification	X	X			5.3.16
0x12	Color stabilization information	X	X			5.3.17
0x13	CVD disable					5.3.18
0x14	Information element	X	X	X	X	5.3.19
0x15–0xff	Reserved					—

5.3.1 Association request command

The association request command allows a device to request association with a VPAN through the coordinator. This command shall only be sent by an unassociated device that wishes to associate with a VPAN. A device shall only associate with a VPAN through the coordinator as determined through the scan procedure.

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

The association request command shall be formatted as illustrated in Figure 60.

Octets: (see 5.2.2.4)	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	Capability Information

Figure 60—Association request command format

5.3.1.1 MHR fields

The Source Addressing Mode subfield of the frame control field shall be set to three (64-bit extended addressing). The Destination Addressing Mode subfield shall be set to the same mode as indicated in the beacon frame to which the association request command refers.

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The Destination VPAN Identifier field shall contain the identifier of the VPAN 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 VPAN Identifier field shall contain the broadcast VPAN identifier (i.e., 0xffff). The Source Address field shall contain the value of *aExtendedAddress*.

5.3.2 Association response command

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

This command shall only be sent by the coordinator or a coordinator to a device that is currently trying to associate.

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

The association response command shall be formatted as illustrated in Figure 61.

Octets: (see 5.2.2.4)	**1**	**2**	**1**	**1**
MHR fields	Command Frame Identifier (as defined in Table 10)	Short Address	Association Status	Capability-negotiation response

Figure 61—Association response command format

The capability-negotiation response is the same as that of the color-stabilization scheme in Table 20.

5.3.2.1 MHR fields

The Destination Addressing Mode and Source Addressing Mode subfields of the frame control field shall each be set to three (i.e., 64-bit extended addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The VPAN ID Compression subfield of the frame control field shall be set to one. In accordance with this value of the VPAN ID Compression subfield, the Destination VPAN Identifier field shall contain the value of *macVPANId*, while the Source VPAN 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*.

5.3.2.2 Short Address field

If the coordinator was not able to associate this device to its VPAN, 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 VPAN, this field shall contain the short address that the device may use in its communications on the VPAN until it is disassociated.

A Short Address field value equal to 0xfffe shall indicate that the device has been successfully associated with a VPAN, but has not been allocated a short address. In this case, the device shall communicate on the VPAN using only its 64-bit extended address.

5.3.2.3 Association Status field

The Association Status field shall contain one of the nonreserved values listed in Table 11.

Table 11—Valid values of the Association Status field

Association status	Description
0x00	Association successful.
0x01	VPAN at capacity.
0x02	VPAN access denied.
0x03–0x7f	Reserved.
0x80–0xff	Reserved for MAC primitive enumeration values.

5.3.2.4 Capability negotiation response field

The capability negotiation response field describes if and where (device and/or coordinator) color stabilization is performed. All allowed settings are shown in Table 12.

Table 12—Capability negotiation response field

Bits: 0–1	Bits: 2–7
00: No color stabilization 01: Color-stabilization information to be sent from device to coordinator upon reception of CVD frames 10: Color-stabilization information to be sent from coordinator to device upon reception of CVD frames 11: Color-stabilization information to be sent from device to coordinator and from coordinator to device when either receives CVD frames	Reserved

5.3.3 Disassociation notification command

The VLC coordinator or an associated device may send the disassociate notification command.

All devices shall implement this command.

The disassociation notification command shall be formatted as illustrated in Figure 62.

Octets: (see 5.2.2.4)	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	Disassociation reason

Figure 62—Disassociation notification command format

5.3.3.1 MHR fields

The Destination Addressing Mode subfield of the frame control field shall be set according to the addressing mode specified by the corresponding primitive. The Source Addressing Mode subfield shall be set to three (i.e., 64-bit extended addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The VPAN ID Compression subfield of the frame control field shall be set to one. In accordance with this value of the VPAN ID Compression subfield, the Destination VPAN Identifier field shall contain the value of *macVPANId*, while the Source VPAN Identifier field shall be omitted. If the coordinator wants an associated device to leave the VPAN, then the Destination Address field shall contain the address of the device being removed from the VPAN. If an associated device wants to leave the VPAN, then the Destination Address field shall contain the value of either *macCoordShortAddress*, if the Destination Addressing Mode subfield is equal to two, or *macCoordExtendedAddress*, if the Destination Addressing Mode subfield is equal to three. The Source Address field shall contain the value of *aExtendedAddress*.

5.3.3.2 Disassociation Reason field

The Disassociation Reason field shall contain one of the nonreserved values listed in Table 13.

Table 13—Valid disassociation reason codes

Disassociate reason	Description
0x00	*Reserved.*
0x01	The coordinator wishes the device to leave the VPAN.
0x02	The device wishes to leave the VPAN.
0x03	Device cannot support communications for the requested dimming value.
0x04f–0x7f	*Reserved.*
0x80–0xff	Reserved for MAC primitive enumeration values.

5.3.4 Data request command

The data request command is sent by a device to request data from the coordinator.

There are three cases for which this command is sent. On a beacon-enabled VPAN, this command shall be sent by a device when *macAutoRequest* is equal to TRUE and a beacon frame indicating that data are pending for that device is received from its coordinator. The coordinator indicates pending data in its beacon frame by adding the address of the recipient of the data to the Address List field. This command shall also be sent when instructed to do so by the next higher layer on reception of the MLME-POLL.request primitive. In addition, a device may send this command to the coordinator *macResponseWaitTime* optical clocks after the acknowledgment to an association request command.

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

The data request command shall be formatted as illustrated in Figure 63.

Octets: (see 5.2.2.4)	1
MHR fields	Command Frame Identifier (as defined in Table 10)

Figure 63—Data request command format

If the data request command is being sent in response to the receipt of a beacon frame indicating that data are pending for that device, the Destination Addressing Mode subfield of the frame control field may be set to zero (i.e., destination addressing information not present) if the beacon frame indicated in its Superframe Specification field (see 5.2.2.1.2) that it originated from the coordinator (see 5.2.1.1.6) or set otherwise according to the coordinator to which the data request command is directed. If the destination addressing information is to be included, the Destination Addressing Mode subfield shall be set according to the value of *macCoordShortAddress*. If *macCoordShortAddress* is equal to 0xfffe, extended addressing shall be used: the Destination Addressing Mode subfield shall be set to three, and the Destination Address field shall contain the value of *macCoordExtendedAddress*. Otherwise, short addressing shall be used: the Destination Addressing Mode subfield shall be set to two, and the Destination Address field shall contain the value of *macCoordShortAddress*.

If the data request command is being sent in response to the receipt of a beacon frame indicating that data are pending for that device, the Source Addressing Mode subfield shall be set according to the addressing mode used for the pending address. If the Source Addressing Mode subfield is set to two, short addressing shall be used: the Source Address field shall contain the value of *macShortAddress*. Otherwise, extended addressing shall be used: the Source Addressing Mode subfield shall be set to three, and the Source Address field shall contain the value of *aExtendedAddress*.

If the data request command is triggered by the reception of an MLME-POLL.request primitive from the next higher layer, then the destination addressing information shall be the same as that contained in the primitive. The Source Addressing Mode subfield shall be set according to the value of *macShortAddress*. If *macShortAddress* is less than 0xfffe, short addressing shall be used. Extended addressing shall be used otherwise.

If the data request command is being sent following the acknowledgment to an association request command frame, the Destination Addressing Mode subfield of the frame control field shall be set according to the coordinator to which the data request command is directed. If *macCoordShortAddress* is equal to 0xfffe, extended addressing shall be used. Short addressing shall be used otherwise. The Source Addressing Mode subfield shall be set to use extended addressing.

If the Destination Addressing Mode subfield is set to zero (i.e., destination addressing information not present), the VPAN ID Compression subfield of the frame control field shall be set to zero and the source VPAN identifier shall contain the value of *macVPANId*. Otherwise, the VPAN ID Compression subfield shall be set to one. In this case and in accordance with the VPAN ID Compression subfield, the Destination VPAN Identifier field shall contain the value of *macVPANId*, while the Source VPAN Identifier field shall be omitted.

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

5.3.5 VPAN ID conflict notification command

The VPAN ID conflict notification command is sent by a device to the coordinator when a VPAN identifier conflict is detected.

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

The VPAN ID conflict notification command shall be formatted as illustrated in Figure 64.

Octets: (see 5.2.2.4)	1
MHR fields	Command Frame Identifier (as defined in Table 10)

Figure 64—VPAN ID conflict notification command format

The Destination Addressing Mode and Source Addressing Mode subfields of the frame control field shall both be set to three (i.e., 64-bit extended addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The VPAN ID Compression subfield of the frame control field shall be set to one. In accordance with this value of the VPAN ID Compression subfield, the Destination VPAN Identifier field shall contain the value of *macVPANId*, while the Source VPAN Identifier field shall be omitted. The Destination Address field shall contain the value of *macCoordExtendedAddress*. The Source Address field shall contain the value of *aExtendedAddress*.

5.3.6 Beacon request command

The beacon request command is used by a device to locate all coordinators within its operating space during an active scan.

This command is optional for a device.

The beacon request command shall be formatted as illustrated in Figure 65.

Octets: 7	1
MHR fields	Command Frame Identifier (as defined in Table 10)

Figure 65—Beacon request command format

The Destination Addressing Mode subfield of the frame control field shall be set to two (i.e., 16-bit short addressing), and the Source Addressing Mode subfield shall be set to zero (i.e., source addressing information not present).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception. The Acknowledgment Request subfield and Security Enabled subfield shall also be set to zero.

The Destination VPAN Identifier field shall contain the broadcast VPAN identifier (i.e., 0xffff). The Destination Address field shall contain the broadcast short address (i.e., 0xffff).

5.3.7 Coordinator realignment command

The coordinator realignment command is sent by the coordinator or a coordinator when any of its VPAN configuration attributes change due to the receipt of an MLME-START.request primitive.

If this command is sent when any VPAN configuration attributes (i.e., VPAN identifier, short address, or logical channel) change, it is broadcast to the VPAN.

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

The coordinator realignment command shall be formatted as illustrated in Figure 66.

Octets: 17/18/23/24	1	2	2	1	2
MHR fields	Command Frame Identifier (as defined in Table 10)	VPAN Identifier	Coordinator Short Address	Logical Channel	Short Address

Figure 66—Coordinator realignment command format

5.3.7.1 MHR fields

The Destination Addressing Mode subfield of the frame control field shall be set to two (e.g., 16-bit short addressing) if it is to be broadcast to the VPAN. The Source Addressing Mode subfield of the frame control field shall be set to three (e.g., 64-bit extended addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception.

The Acknowledgment Request subfield of the frame control field shall be set to zero if the command is to be broadcast to the VPAN.

The Destination VPAN Identifier field shall contain the broadcast VPAN identifier (e.g., 0xffff). The Destination Address field shall contain the broadcast short address (e.g., 0xffff). The Source VPAN Identifier field shall contain the value of *macVPANId*, and the Source Address field shall contain the value of *aExtendedAddress*.

5.3.7.2 VPAN Identifier field

The VPAN Identifier field shall contain the VPAN identifier that the coordinator intends to use for all future communications.

5.3.7.3 Coordinator Short Address field

The Coordinator Short Address field shall contain the value of *macShortAddress*.

5.3.7.4 Logical Channel field

The Logical Channel field shall contain the logical channel that the coordinator intends to use for all future communications.

5.3.7.5 Short Address field

If the coordinator realignment command is broadcast to the VPAN, the Short Address field shall be set to 0xffff and ignored on reception.

5.3.8 GTS request command

The GTS request command is used by an associated device that is requesting the allocation of a new GTS or the deallocation of an existing GTS from the coordinator. Only devices that have a 16-bit short address less than 0xfffe shall send this command.

This command is optional.

The GTS request command shall be formatted as illustrated in Figure 67.

Octets: 7	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	GTS Characteristics

Figure 67—GTS request command format

5.3.8.1 MHR fields

The Destination Addressing Mode subfield of the frame control field shall be set to zero (e.g., destination addressing information not present), and the Source Addressing Mode subfield shall be set to two (e.g., 16-bit short addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The Source VPAN Identifier field shall contain the value of *macVPANId*, and the Source Address field shall contain the value of *macShortAddress*.

5.3.8.2 GTS Characteristics field

The GTS Characteristics field shall be formatted as illustrated in Figure 68.

Bits: 0–3	4	5	6–7
GTS Length	GTS Direction	Characteristics Type	Reserved

Figure 68—GTS Characteristics field format

The GTS Length subfield shall contain the number of superframe slots being requested for the GTS.

The GTS Direction subfield shall be set to one if the GTS is to be a receive-only GTS. Conversely, this subfield shall be set to zero if the GTS is to be a transmit-only GTS. GTS direction is defined relative to the direction of data frame transmissions by the device.

The Characteristics Type subfield shall be set to one if the characteristics refers to a GTS allocation or zero if the characteristics refers to a GTS deallocation.

5.3.9 Blinking notification command

The blinking notification command (see Figure 69) is sent by a coordinator when the device is no longer responding. A reason for this might be the misalignment between the device TX and the coordinator RX (limited FOV of receiver, low device TX power, mobility of the device, etc.). In such cases, the device can change the visibility indication from continuous visibility for point-and-shoot to blinking indication. The device can then change from point-and-shoot mode to blinking mode in order to indicate to the user that the uplink to the coordinator is disconnected. This indication can be applied to both P2MP and P2P modes of operation.

Octets: 7	**1**	**1**
MHR fields	Command Frame Identifier (see Table 10)	Blinking frequency

Figure 69—Blinking notification command

The blinking notification bit shall be set when the MAC PIB attribute, *macUseBlinkingNotification* and *macBlinkingNotificationFrequency*, as defined in Table 60 indicates the blinking notification usage.

To support the blinking notification, the frequency shall be chosen from the *phyBlinkingNotificationFrequency* PHY PIB attribute as shown in Table 100, using the MLME-SET.request and PLME-SET.request primitives.

This feature can help to align the link and is only intended for mobile devices.

5.3.9.1 Blinking frequency

The frequency subfield shall contain the frequency for blinking (see Figure 70).

Bits: 0–3	**4–7**
Frequency	Reserved

Figure 70—Blinking frequency field format

5.3.10 Dimming notification command

The DME indicates the dimming level to the MAC using the MAC PIB attribute, *macDim*, as defined in Table 60. The dimming notification command is used to communicate the dimming level set by the *macDim* PIB attribute to the receiver. The dimming notification command (see Figure 71) shall be sent at the lowest data rate corresponding to the currently negotiated optical clock rate. The symbol shape information for VPPM is derived using the algorithm of Figure 114 after the dimming level is obtained.

Octets: 7	1	2	2
MHR fields	Command frame identifier (see Table 10)	Dimming level	Dimmer adaptation timer (see 5.1.14.8)

Figure 71—Dimming notification command

The dimming level is two bytes long and contains a value between 0 and 1000, where 0 represents 0% visibility and 1000 represents 100% visibility. The dimming levels are defined with a resolution of 0.1%. The dimmer adaptation timer provides a resolution of 0–16383 MAC clock cycles. The dimming notification command transmits the dimmer level from the TX to the RX along with the dimmer adaptation timer information. VPPM by default uses only 50% duty cycle, so if dimming is supported using VPPM as in 8.5.2.3, the VPPM pulse shape is obtained using the dimmer notification command in conjunction with the algorithm shown in Figure 114. Before dimming is supported using VPPM, the dimming notification command needs to be sent by the MAC to the receiver.

5.3.11 Fast link recovery command

Fast link recovery command is used for the device or coordinator to send the fast link recovery (FLR) signal and the fast link recovery response (FLR RSP), to help the link recovery.

Fast link recovery signal and response use the fast link recovery command format. The fast link recovery command shall be formatted as illustrated in Figure 72.

Octets: (as defined in 5.2.2.4)	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	FLR field
		FLR field explanation Bit 0: ‘0’ indicating it is FLR signal, ‘1’ indicating it is FLR RSP Bits 1–3: index of FLR signal direction, if bit 0 is ‘0’. received FLR signal direction index if bit 0 is ‘1’. Bits 4–7: reserved

Figure 72—Fast link recovery command

The FLR signal and the FLR response (RSP) are differentiated by the first bit (bit 0) of the FLR field in the fast link recovery command frame. The device can indicate the index of FLR signal direction by using bits 1 to 3 of the FLR field in the command frame. If the device receives the FLR signal and needs to send FLR RSP, it repeats the received FLR signal direction index by using bits 1 to 3 of the FLR field in the command frame. If the device is uni-direction, it uses ‘000’ as the index of the direction.

The usage of the FLR is presented in 5.1.9.

5.3.12 Mobility notification command

The mobility notification command is shown in Figure 73. The concept of VLC cell mobility is defined in 5.1.11.

Octets: 7	1	variable
MHR fields	Command Frame Identifier (Table 10)	cellSearchQuality (see 5.1.11.3)

Figure 73—Mobility notification command

The results from the cell search are provided in the mobility notification command as shown in Figure 73. The WQI values (in octets) obtained for the current channel during the cell search procedure defined in 5.1.11.3 shall be included in the command frame. The number of octets sent shall be equal to cellSearchLength, as defined in 5.1.11.3.

5.3.13 GTS response command

The optional GTS.response primitive is generated in response to a GTS.request primitive. When used, the GTS response command shall be formatted as illustrated in Figure 74.

Octets: 7	1	1	1
MHR fields	Command Frame Identifier as defined in Table 10)	GTS characteristics	GTS Starting Slot (see 5.2.2.1.5)

Figure 74—GTS response command format

5.3.13.1 MHR fields

The Destination Addressing Mode subfield of the frame control field shall be set to zero (e.g., destination addressing information not present), and the Source Addressing Mode subfield shall be set to two (e.g., 16-bit short addressing).

The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the Acknowledgment Request subfield shall be set to one.

The Source VPAN Identifier field shall contain the value of *macVPANId*, and the Source Address field shall contain the value of *macShortAddress*.

5.3.13.2 GTS Characteristics field

The GTS Characteristics field shall be formatted as illustrated in Figure 75.

The GTS Length subfield shall contain the number of superframe slots being requested for the GTS.

The GTS Direction subfield shall be set to one if the GTS is to be a receive-only GTS. Conversely, this subfield shall be set to zero if the GTS is to be a transmit-only GTS. GTS direction is defined relative to the direction of data frame transmissions by the device.

Bits: 0–3	4	5	6–7
GTS Length	GTS Direction	Characteristics Type	Reserved

Figure 75—GTS Characteristics field format

The Characteristics Type subfield shall be set to one if the characteristics refers to a GTS allocation or zero if the characteristics refers to a GTS deallocation.

5.3.14 Clock rate change notification command

The command format for the clock rate change notification is as shown in Figure 76. This clock rate change notification is sent at the current clock rate negotiated between the devices. All future transmissions from the current device to the receiving device will occur at this new clock rate.

Octets: 7	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	New clock rate for future TX

Figure 76—Clock rate change notification format

The modulation and coding scheme (MCS) ID from Table 83 shall be used to indicate the optical clock rate. Any MCS ID corresponding to the chosen future clock rate can be used. The 6 LSBs shall be set to the MCS ID corresponding to the future clock rate. The other bits are set to 0 and reserved for future use.

5.3.15 Multiple channel assignment command

Octets: 7	1	1
MHR fields	Command Frame Identifier (as defined in Table 10)	Multiple Channels

Figure 77—Multiple channel assignment command format

Multiple channels should be used in the VLC system when time slot resources are not enough to cover all the currents users. These channels should be assigned based on the band-plan in Table 76. Refer to Table 3 for the contents of the Multiple Channels field.

5.3.16 Color stabilization timer notification command

The color stabilization timer notification command shall be formatted as illustrated in Figure 78. This command is used to inform a device or coordinator about the minimum time between two color-stabilization updates (upon reception of CVD frames).

The color stabilization timer field has the same format as the *macColorStabilizationTimer* (see Table 60).

Octets: 7	1	2	4
MHR fields	Command frame identifier (see Table 10)	Short address	Color stabilization timer

Figure 78—Color stabilization timer notification command format

5.3.17 Color stabilization information command

The color stabilization information command shall be formatted as illustrated in Figure 79. This command is used for relaying the color-stabilization updates (upon reception of CVD frames) back to the pertinent CSK transmitter (see Figure 116).

Octets: 7	1	2	Color stabilization information		
			2	2	2
MHR fields	Command frame identifier (see Table 10)	Short address	band *i*	band *j*	band *k*

Figure 79—Color stabilization information command format

The color stabilization information per band is 2 octets long and is used by the color-stabilization module (see Figure 117). It consists of the received signal levels in each of the three CSK bands. Two octets are used for each of the bands. A linear scale is used for each band, where the highest value corresponds to the maximum receive signal and the lowest value to the minimum receive signal. These fields are sent LSB first.

5.3.18 CVD disable command

Octets: 7	1	1
MHR fields	Command frame identifier (see Table 10)	CVD disable

Figure 80—CVD disable command format

The CVD frame can be transmitted depending on bi-directional, multicasting, and broadcasting capabilities. A device shall not transmit a CVD frame after the device has received a frame from an associated device that has the “CVD usage option” bit set to ‘0’ as defined in Table 14. A device may resume sending CVD frames after it has received a frame from associated devices that have the “CVD usage option” bit set to ‘1’. When the coordinator transmits and receives data with a device, if another device transmits an in-band CVD frame, interference may occur in the link between the coordinator and device. Out-of-band idle patterns may be used to maintain visibility when interference is seen in devices due to use of the CVD frame. In this case, the coordinator may indicate the transmission of “CVD usage option” with the CVD usage option bit set to ‘0’. The CVD frame should be used prudently so as to cause minimal interference and prolong battery life. In many cases, a light source is used for illumination, which takes precedence over the use for communication.

The CVD usage option subfield is 1 bit in length and shall be set to ‘1’ if the device is sending a CVD frame. This subfield shall be set to ‘0’ otherwise.

Table 14—CVD disable field

Command frame payload	Bit	Usage/Description
CVD usage option	b0	logic 1 indicates that the device shall transmit the CVD frame logic 0 indicates that the device shall not transmit the CVD frame and may use out-of-band idle patterns if visibility needs to be maintained
	b1–b7	Reserved

5.3.19 Information element command

The format of an individual information element (IE) is shown in Figure 81. The first octet is the Element ID and the second octet is the Length (Ln) of the payload of the IE in octets. The following Ln octets are the payload for the IE. Unless otherwise specified, these elements may appear in any order in the frames that are allowed to include more than one of these elements.

Octets: 1	1	Ln
Element ID	Length (=Ln)	IE payload

Figure 81—Information element format

The information elements defined in this standard are listed in Table 15.

Table 15—Information elements

Element ID hex value	Element	Subclause
0x01	Capabilities	5.3.19.1
0x02	Wavelength quality indication	5.3.19.2

When the information elements are used, they shall be added at the end of command frame format. Multiple information elements can be part of a single command frame. IEs can be added to any command frame.

5.3.19.1 Capabilities IE

The capabilities IE is used to convey device MAC and PHY capabilities to peer devices. The capabilities IE, as shown in Figure 82, consists of two fields: the capability information field, refer to Table 16, which indicates general capabilities of the device; and the aggregation bitmap field, which is specified in Figure 82.

Octets: 8	variable: 8*n*	variable: 8*n*
Capability Information Field	Aggregation bitmap field	Guard bitmap field

Figure 82—Capabilities IE

5.3.19.1.1 Capability information field

The capability information field is illustrated in Table 16.

Table 16—Capability information field

	Bit position	Function
MAC layer capabilities	0	Power source
	1–2	Battery information
	3	Security capability
	4	Coordinator capability
	5	Traffic support
	6–8	Topology support
	9–10	Device type
	11	Beacon support
	12	Dimming support
	13	Continuous visibility transmission (for infrastructure)
	14	CVD support
	15–23	Reserved
PHY layer capabilities	24	PHY I support
	25	PHY II support
	26	PHY III support
	27–28	Color stabilization capability
	29–31	Max supported TX clock
	32–34	Max supported RX clock
	35	Explicit clock notification request
	36	CCA support
	37–39	Reserved

Table 16—Capability information field *(continued)*

	Bit position	Function
Physical device capabilities	40–42	Number of optical sources
	43–45	Multiple direction support
	46–55	Number of cells supported (n)
Band capabilities	56–63	Bands used for PHY III (any 3 bits of the bits set to 1 can be used)

The power source subfield is 1 bit in length and shall be set to one if the device is receiving power from the alternating current mains. Otherwise, the power source subfield shall be set to zero.

The battery information subfield, shown in Table 17, is set to reserved (11) if the power source is set to 1.

Table 17—Battery Indication

Bits (b2 b1)	Battery indication
00	Unknown
11	< 50% (low battery)
10	≥ 50% (sufficient battery)
11	Reserved

The security capability subfield is 1 bit in length and shall be set to one if the device is capable of sending and receiving cryptographically protected MAC frames; otherwise, it shall be set to zero.

The coordinator capability subfield is 1 bit in length and shall be set to 1 if the device is capable of functioning as a coordinator; otherwise, it shall be set to zero.

The traffic support capability subfield is 1 bit in length. It shall be set to 0 if the device is only capable of broadcasting (unidirectional) communication. Otherwise, it shall be set to 1.

The topology support capability subfield can support multiple topologies via the bit maps of Table 18.

Table 18—Topology support capability

Bits (b8 b7 b6)	Topology indication
b8	P2MP
b7	P2P
b6	Broadcast

The device-type capability subfield is set according to Table 19. This information is provided to assist upper layers.

Table 19—Device-type capability

Bits (b10 b9)	Device capability
00	Infrastructure
01	Mobile
10	Vehicle
11	Unknown/reserved

The beacon support capability subfield is 1 bit in length. It shall be set to 1 if the device is capable of sending beacons. Otherwise, it shall be set to 0.

The dimming support in MAC capability subfield is 1 bit in length. It shall be set to 1 if the device is capable of supporting dimming in the MAC using duty cycling and idle patterns. Otherwise, it shall be set to 0. A device shall honor all dimming requests. If the dimming support bit is not set then the device shall not attempt to communicate when a dimming request is received and shall comply with the dimming request even if the device must disassociate from the network as discussed in 5.1.4.2. Even if the device supports dimming but is unable to communicate during dimming, it shall set the *macDimDataFailureIndication* MAC PIB attribute as mentioned in Table 60, but shall still comply with the dimming request at the expense of loss of communication.

The continuous visibility transmission subfield is one bit in length. It shall be set to 1 if the device will be continuously transmitting to maintain illumination. Otherwise, it shall be to 0.

The CVD support subfield is 1 bit in length. It shall be set to 1 if the device is capable of transmitting various colors; otherwise, it shall be set to 0.

The PHY I support subfield is 1 bit in length. It shall be set to 1 if the device supports PHY I.

The PHY II support subfield is 1 bit in length. It shall be set to 1 if the device supports PHY II.

The PHY III support subfield is 1 bit in length. It shall be set to 1 if the device supports PHY III.

The color-stabilization capability subfield describes if and where (device and/or coordinator) color stabilization is performed. All allowed settings are shown in Table 20.

The max supported TX clock subfield and max RX clock subfields follow the usage as indicated in Table 21. Support for 200 kHz is mandatory for PHY I and support of 3.75 MHz is mandatory for PHY II. Support for 12 MHz is mandatory for PHY III and shall be indicated using bits ‘100’ as in Table 21.

The explicit clock notification subfield is 1 bit in length. The subfield shall be set to 1 if the receiving device needs an explicit clock change notification from the transmitter before any change of clock frequency.

If CCA is supported, then the CCA Support bit is set to 1; otherwise, the bit is set to 0.

The number of optical sources subfield indicates the number of optical sources in the transmitter of the device that have distinct frequency responses.

Table 20—Color-stabilization capability

Bits (b28 b27)	Color-stabilization scheme
00	No color stabilization
01	Color-stabilization information to be sent from device to coordinator upon reception of CVD frames
10	Color-stabilization information to be sent from coordinator to device upon reception of CVD frames
11	Color-stabilization information to be sent from device to coordinator and from coordinator to device when either receives CVD frames

Table 21—Maximum supported optical clock frequency

Bits (b31 b30 b29)	Description
000	200 kHz
001	≤ 400 kHz
010	≤ 3.75 MHz
011	≤ 7.5 MHz
100	≤ 15 MHz
101	≤ 30 MHz
110	≤ 60 MHz
111	≤ 120 MHz

The multiple direction support subfield indicates the number of distinct directions supported by the device transmitter supported by the multiple optical sources. This is used for fast link recovery as defined in 5.3.11.

The number of cells *n* indicates the maximum number of cells supported in the device. The number of cells supported shall not be more than 1023.

In regards to the bands used for PHY III, bit 7 is reserved and bits 0–6 map to the bits corresponding to the bandplan. Only 3 bits shall be set to indicate PHY III usage. If the device supports more colors and wants to change the PHY III usage, it needs to send the capabilities information again with the new bitmap.

5.3.19.1.2 Aggregation and guard channel

The aggregation and guard channels are used to support any visible light optical source for VLC that may have variable spectral widths and center frequencies. The aggregation and guard bitmap for a single optical source type is as shown in Figure 83. The bit map is variable in length. The length of the aggregation and guard bit maps are ‘n’ octets each, where ‘n’ is the number of optical source types. The aggregation and guard channel bit usage are defined by an 8-bit bitmap for every optical source type supported by the transmitter of the device. The 8-bit bitmap is indexed by the bandplan identification number. The bit position ‘m’ is set to a ‘1’ for band 'm' if that band is used by the optical source. The reserved bit in Figure 83 shall be set to 0.

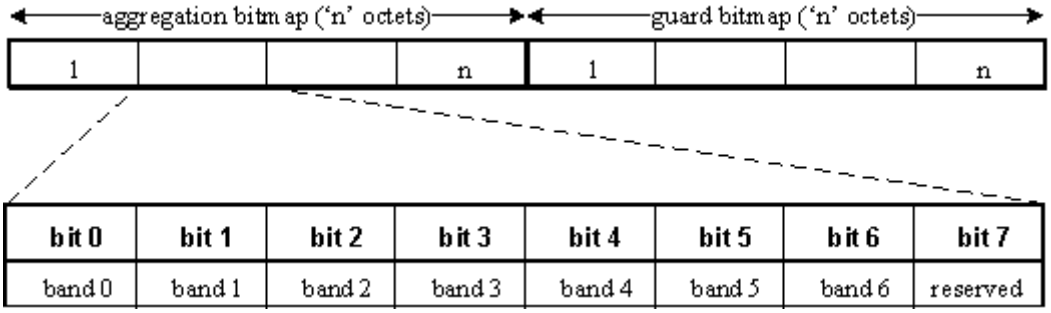


Figure 83—Aggregation and guard bitmap per optical source type

For example,

if band 1 and band 2 need to be aggregated [assuming a blue light-emitting diode (LED)], the aggregation bit-map is indicated as 0110000 and the guard bit-map is indicated as 0000000.

if band 1 is being used but there is leakage in bands 3, 4, 5 (assuming a white LED, which is realized via a blue LED with yellow phosphor), the aggregation bit map is indicated as 0100000 and the guard bit-map is indicated as 0001110.

5.3.19.2 Wavelength quality indication (WQI) IE

WQI is communicated to another device using the WQI Information Element. The WQI value to be sent in the Information Element may be an average value across a number of packets, and WQI value sets for a number of band plan ID's can be reported using the WQI information element as shown in Table 22. The wavelength quality indication IE is 7 octets in length and the WQI information is provided for all band plan IDs. If a band plan ID is not supported, WQI of 0 shall be reported.

Table 22—Wavelength quality indication IE

Band plan code	WQI value
0x00	0x00 to 0xff
0x01	0x00 to 0xff
0x02	0x00 to 0xff
0x03	0x00 to 0xff
0x04	0x00 to 0xff
0x05	0x00 to 0xff
0x06	0x00 to 0xff

6. MAC sublayer service specification

6.1 Overview

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

Figure 84 depicts the components and interfaces of the MAC sublayer.

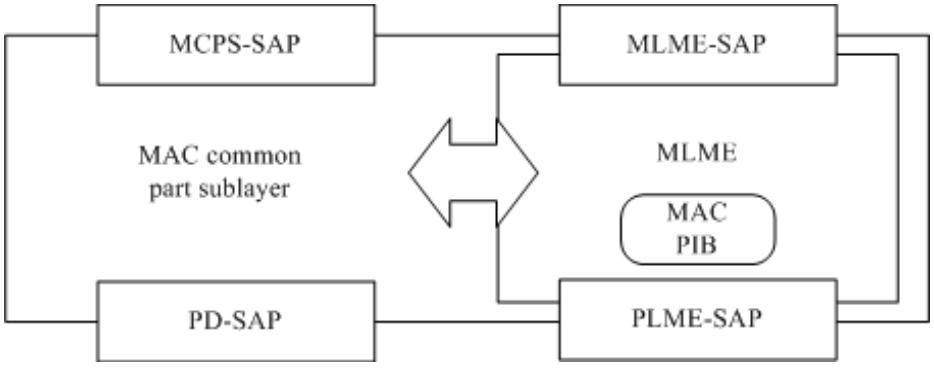


Figure 84—MAC sublayer reference model

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

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

These two services provide the interface between the SSCS and the PHY, via the PD-SAP and PLME-SAP interfaces (see Clause 9). In addition to these external interfaces, an implicit interface also exists between the MLME and the MCPS that allows the MLME to use the MAC data service.

6.2 MAC data service

The MCPS-SAP supports the transport of SSCS protocol data units (SPDUs) between peer SSCS entities. Table 23 lists the primitives supported by the MCPS-SAP. These primitives are discussed in the subclauses referenced in this table.

Table 23—MCPS-SAP primitives

MCPS-SAP primitive	Request	Confirm	Indication
MCPS-DATA	6.2.1	6.2.2	6.2.3
MCPS-PURGE	6.2.4	6.2.5	—

6.2.1 MCPS-DATA.request

The MCPS-DATA.request primitive requests the transfer of a data SPDU (i.e., MSDU) from a local SSCS entity to a single peer SSCS entity. In the packed mode, multiple MSDU are passed via a local SSCS entity to a single peer SSCS entity.

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

```
MCPS-DATA.request (
SrcAddrMode,
DstAddrMode,
DstVPANId,
DstAddr,
MsduLength,
Msdu,
MsduHandle,
TxOptions,
SecurityLevel,
KeyIdMode,
KeySource,
KeyIndex,
DataRate,
BurstMode,
ColorReceived,
ColorNotReceived
)
```

Table 24 specifies the parameters for the MCPS-DATA.request primitive.

Table 24—MCPS-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = No address (addressing fields omitted, as defined in 5.2.1.1.7). 0x01 = Reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = No address (addressing fields omitted, as defined in 5.2.1.1.6). 0x01 = No address field (broadcast only mode with no address fields present). 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstVPANId	Integer	0x0000–0xffff	The 16-bit VPAN identifier of the entity to which the MSDU is being transferred.

Table 24—MCPS-DATA.request parameters *(continued)*

Name	Type	Valid range	Description
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
MsduLength	Integer	≤ *aMaxMACPayloadSize*	The number of octets contained in the MSDU to be transmitted by the MAC sublayer entity.
Msdu	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
MsduHandle	Integer	0x00–0xff	The handle associated with the MSDU to be transmitted by the MAC sublayer entity.
TxOptions	Bitmap	3-bit field	The 3 bits (b_0, b_1, b_2) indicate the transmission options for this MSDU. For b_0, 1 = acknowledged transmission, 0 = unacknowledged transmission. For b_1, 1 = GTS transmission, 0 = CAP transmission for a beacon-enabled VPAN. For b_2, 1 = indirect transmission, 0 = direct transmission. For a non-beacon-enabled VPAN, bit b_1 should always be set to 0.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
DataRate	Enumeration	6-bit field	The data rate of the PHY frame to be transmitted by the PHY entity as shown in Table 83.
BurstMode	Boolean	TRUE or FALSE	The BurstMode bit shall be set TRUE if the burst mode is being used (as discussed in 8.6.1); otherwise, the BurstMode bit shall be set FALSE.
ColorReceived	Boolean	TRUE or FALSE	ColorReceived shall be set as TRUE, when the ACK frame is sent and the color function for the ACK state indication is used by the CVD frame. ColorReceived shall be set as FALSE when the ACK frame is sent but the color function for the ACK state indication is not used by the CVD frame.

Table 24—MCPS-DATA.request parameters *(continued)*

Name	Type	Valid range	Description
ColorNotReceived	Boolean	TRUE or FALSE	ColorNotReceived shall be set as TRUE, when the ACK frame is not sent but the color function for the non-ACK state indication is used by the CVD frame. ColorNotReceived shall be set as FALSE when the ACK frame is not sent and the color function for the non-ACK state indication is not used by the CVD frame.

6.2.1.1 Appropriate usage

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

6.2.1.2 Effect on receipt

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

The MAC sublayer builds an MPDU to transmit from the supplied arguments. The flags in the SrcAddrMode and DstAddrMode parameters correspond to the addressing subfields in the frame control field, as shown in 5.2.1.1, and are used to construct both the frame control and addressing fields of the MHR. If both the SrcAddrMode and the DstAddrMode parameters are set to 0x00 (i.e., addressing fields omitted), the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_ADDRESS.

The TxOptions parameter indicates how the MAC sublayer data service transmits the supplied MSDU. If the TxOptions parameter specifies that an acknowledged transmission is required, the Acknowledgment Request subfield of the frame control field will be set to one (see 5.1.7.4).

If the TxOptions parameter specifies that a GTS transmission is required, the MAC sublayer will determine whether it has a valid GTS (for GTS usage rules, as defined in 5.1.8.3). If a valid GTS could not be found, the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of INVALID_GTS. If a valid GTS was found, the MAC sublayer will defer, if necessary, until the GTS. If the TxOptions parameter specifies that a GTS transmission is not required, the MAC sublayer will transmit the MSDU using either slotted random access in the CAP for a beacon-enabled VPAN or unslotted random access for a nonbeacon-enabled VPAN. Specifying a GTS transmission in the TxOptions parameter overrides an indirect transmission request.

If the TxOptions parameter specifies that an indirect transmission is required and this primitive is received by the MAC sublayer of a coordinator, the data frame is sent using indirect transmission, i.e., the data frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 5.1.7.3. Transactions with a broadcast destination address will be transmitted using the mechanism described in 5.2.1.1.4. Transactions with a unicast destination address can then be extracted at the discretion of each device concerned using the method described in 5.1.7.3. If there is no capacity to store the transaction, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will issue the MCPS-DATA.confirm primitive with a status of TRANSACTION_EXPIRED. The

transaction handling procedure is described in 5.1.6. If the TxOptions parameter specifies that an indirect transmission is required and if the device receiving this primitive is not a coordinator, the destination address is not present, or the TxOptions parameter also specifies a GTS transmission, the indirect transmission option will be ignored.

If the TxOptions parameter specifies that an indirect transmission is not required, the MAC sublayer will transmit the MSDU using slotted random access either in the CAP for a beacon-enabled VPAN or immediately for a nonbeacon-enabled VPAN. If the TxOptions parameter specifies that a direct transmission is required and the MAC sublayer does not receive an acknowledgment from the recipient after *macMaxFrameRetries* retransmissions (see 5.1.7.4), it will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of NO_ACK.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MAC sublayer will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on the DstAddr, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MAC sublayer will discard the frame and issue the MCPS-DATA.confirm primitive with the error status returned by outgoing frame processing.

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

If the transmission attempts a random access (either slotted or unslotted) and the random access algorithm failed due to adverse conditions on the channel, and the TxOptions parameter specifies that a direct transmission is required, the MAC sublayer will discard the MSDU and issue the MCPS-DATA.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MAC sublayer receives the request while transmission is prohibited, it shall delay transmission until transmission is permitted.

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

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

6.2.2 MCPS-DATA.confirm

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

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

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

Table 25 specifies the parameters for the MCPS-DATA.confirm primitive.

Table 25—MCPS-DATA.confirm parameters

Name	Type	Valid range	Description
MsduHandle	Integer	0x00–0xff	The handle associated with the MSDU being confirmed.
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_GTS, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the last MSDU transmission.
Timestamp	Integer	0x000000–0xffffff	Optional. The time, in optical clocks, at which the data were transmitted (see 5.1.5.1). The value of this parameter will be considered valid only if the value of the status parameter is SUCCESS; if the status parameter is not equal to SUCCESS, the value of the Timestamp parameter shall not be used for any other purpose. The boundary is described by *macTimeStampOffset* (as defined in Table 60). The time stamp 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.

6.2.2.1 When generated

The MCPS-DATA.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-DATA.request primitive. The MCPS-DATA.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or the appropriate error code. The status values are fully described in 6.2.1.2 and subclauses referenced by 6.2.1.2.

6.2.2.2 Appropriate usage

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

6.2.3 MCPS-DATA.indication

The MCPS-DATA.indication primitive indicates the transfer of a data SPDU (i.e., MSDU) from the MAC sublayer to the local SSCS entity. In the packed mode, multiple MSDU are passed via a local SSCS entity to a single peer SSCS entity.

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

```
MCPS-DATA.indication (
                     SrcAddrMode,
                     SrcVPANId,
                     SrcAddr,
                     DstAddrMode,
                     DstVPANId,
                     DstAddr,
                     MsduLength,
                     Msdu,
                     MpduLinkQuality,
                     DSN,
                     Timestamp,
                     SecurityLevel,
                     KeyIdMode,
                     KeySource,
                     KeyIndex,
                     DataRate,
                     BurstMode,
                     ColorReceived,
                     ColorNotReceived
                     )
```

Table 26 specifies the parameters for the MCPS-DATA.indication primitive.

Table 26—MCPS-DATA.indication parameters

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

Table 26—MCPS-DATA.indication parameters *(continued)*

Name	Type	Valid range	Description
DstVPANId	Integer	0x0000–0xffff	The 16-bit VPAN identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
MsduLength	Integer	≤*aMaxMacPayloadSize*	The number of octets contained in the MSDU being indicated by the MAC sublayer entity.
Msdu	Set of octets	—	The set of octets forming the MSDU being indicated by the MAC sublayer entity.
MpduLinkQuality	Integer	0x00–0xff	WQI value measured during reception of the MPDU. Lower values represent lower WQI (see 5.3.19.2)
DSN	Integer	0x00–0xff	The DSN of the received data frame.
Timestamp	Integer	0x000000–0xffffff	Optional. The time, in optical clocks, at which the data were received (see 5.1.5.1). The boundary is described by *macTimeStampOffset* (as defined in Table 60). The time stamp is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received data frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
DataRate	Enumeration	6-bit field	The data rate of the PHY frame to be transmitted by the PHY entity as shown in Table 83.
BurstMode	Boolean	TRUE or FALSE	The BurstMode bit shall be set TRUE if the burst mode is being used (as discussed in 8.6.1); otherwise, the BurstMode bit shall be set FALSE.
ColorReceived	Boolean	TRUE or FALSE	ColorReceived shall be set as TRUE, if CVD frame is sent when data frame is successfully received.
ColorNotReceived	Boolean	TRUE or FALSE	ColorNotReceived shall be set as TRUE, if CVD frame is sent when data frame is not received.

6.2.3.1 When generated

The MCPS-DATA.indication primitive is generated by the MAC sublayer and issued to the SSCS on receipt of a data frame at the local MAC sublayer entity that passes the appropriate message filtering operations as described in 5.1.7.2.

6.2.3.2 Appropriate usage

On receipt of the MCPS-DATA.indication primitive, the SSCS is notified of the arrival of data at the device.

6.2.4 MCPS-PURGE.request

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

This primitive is optional for a device.

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

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

Table 27 specifies the parameters for the MCPS-PURGE.request primitive.

Table 27—MCPS-PURGE.request parameters

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

6.2.4.1 Appropriate usage

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

6.2.4.2 Effect on receipt

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

6.2.5 MCPS-PURGE.confirm

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

This primitive is optional for a device.

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

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

Table 28 specifies the parameters for the MCPS-PURGE.confirm primitive.

Table 28—MCPS-PURGE.confirm parameters

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

6.2.5.1 When generated

The MCPS-PURGE.confirm primitive is generated by the MAC sublayer entity in response to an MCPS-PURGE.request primitive. The MCPS-PURGE.confirm primitive returns a status of either SUCCESS, indicating that the purge request was successful, or INVALID_HANDLE, indicating an error. The status values are fully described in 6.2.5.2.

6.2.5.2 Appropriate usage

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

6.2.6 Data service message sequence chart

Figure 85 illustrates a sequence of messages necessary for a successful data transfer between two devices. Figure 106 and Figure 107 also illustrate this, including the steps taken by the PHY.

6.3 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 29 summarizes the primitives supported by the MLME through the MLME-SAP interface. Primitives marked with a diamond (♦) are optional for an RFD. Primitives marked with an asterisk (*) are optional for both device types (i.e., RFD and FFD). The primitives are discussed in the subclauses referenced in this table.

6.3.1 Association primitives

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

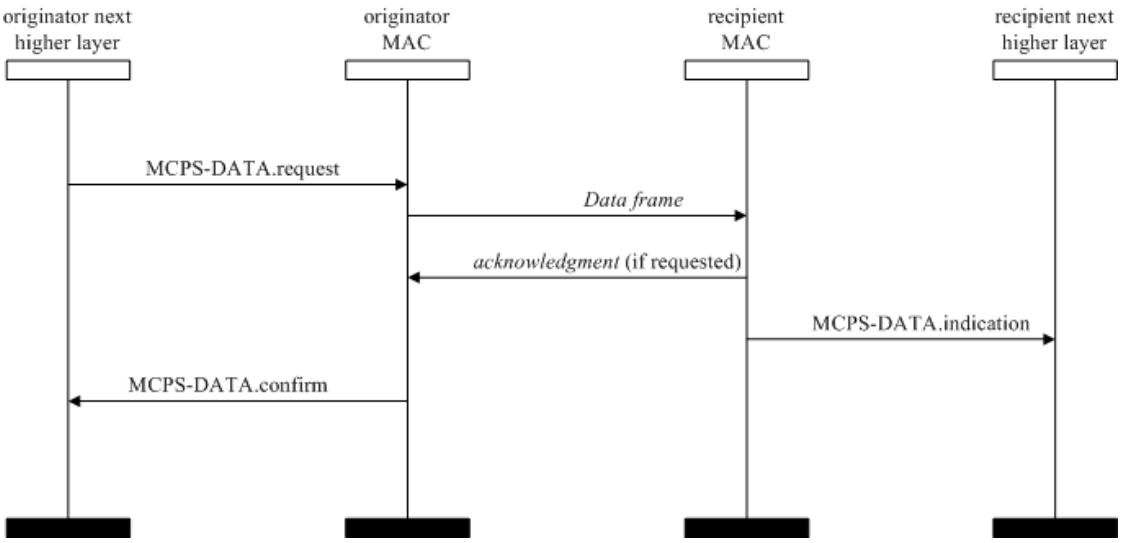


Figure 85—Message sequence chart describing the MAC data service

Table 29—Summary of the primitives accessed through the MLME-SAP

Name	Request	Indication	Response	Confirm
MLME-ASSOCIATE	6.3.1.1	6.3.1.2♦	6.3.1.3♦	6.3.1.4
MLME-DISASSOCIATE	6.3.2.1	6.3.2.2		6.3.2.3
MLME-BEACON-NOTIFY		6.3.3.1		
MLME-GET	6.3.4.1			6.3.4.2
MLME-GTS	6.3.5.1	6.3.5.2		6.3.5.3
MLME-RESET	6.3.6.1			6.3.6.2
MLME-RX-ENABLE	6.3.7.1			6.3.7.2
MLME-SCAN	6.3.8.1			6.3.8.2
MLME-COMM-STATUS		6.3.9.1		
MLME-SET	6.3.10.1			6.3.10.2
MLME-START	6.3.11.1♦			6.3.11.2♦
MLME-SYNC	6.3.12.1*			
MLME-SYNC-LOSS		6.3.13.1		
MLME-POLL	6.3.14.1			6.3.14.2

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

6.3.1.1 MLME-ASSOCIATE.request

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

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

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

Table 30 specifies the parameters for the MLME-ASSOCIATE.request primitive.

Table 30—MLME-ASSOCIATE.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see Table 76)	The logical channel on which to attempt association.
CoordAddrMode	Integer	0x02–0x03	The coordinator addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 2 = 16-bit short address. 3 = 64-bit extended address.
CoordVPANId	Integer	0x0000–0xffff	The VPAN identifier of the coordinator as specified in the received beacon frame.
CoordAddress	Device address	As specified by the CoordAddrMode parameter	The address of the coordinator with which to associate.
CapabilityInformation	Bitmap	As defined in 5.3.19.1	Specifies the operational capabilities of the associating device.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

Table 30—MLME-ASSOCIATE.request parameters *(continued)*

Name	Type	Valid range	Description
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
ColorAssoc	Boolean	TRUE or FALSE	ColorAssoc shall be set as TRUE if the color CVD frame is to be transmitted after the association request command is sent.

6.3.1.1.1 Appropriate usage

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

6.3.1.1.2 Effect on receipt

On receipt of the MLME-ASSOCIATE.request primitive, the MLME of an unassociated device first updates the appropriate PHY and MAC PIB attributes and then generates an association request command as shown in 5.3.1, as dictated by the association procedure described in 5.1.4.1.

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

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on the CoordAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-ASSOCIATE.confirm primitive with the error status returned by outgoing frame processing.

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

If the MLME successfully transmits an association request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_ACK (see 5.1.7.4).

If the MLME of an unassociated device successfully receives an acknowledgment to its association request command, the MLME will wait for a response to the request (see 5.1.4.1). If the MLME of the device does not receive a response, it will issue the MLME-ASSOCIATE.confirm primitive with a status of NO_DATA.

If the MLME of the device extracts an association response command frame from the coordinator, it will then issue the MLME-ASSOCIATE.confirm primitive with a status equal to the contents of the Association Status field in the association response command as shown in 5.3.2.3.

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

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

6.3.1.2 MLME-ASSOCIATE.indication

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

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

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

Table 31 specifies the parameters for the MLME-ASSOCIATE.indication primitive.

Table 31—MLME-ASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting association.
CapabilityInformation	Bitmap	Refer to 5.3.19.1	The operational capabilities of the device requesting association.
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received MAC command frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.1.2.1 When generated

The MLME-ASSOCIATE.indication primitive is generated by the MLME of the coordinator and issued to its next higher layer to indicate the reception of an association request command (as defined in 5.3.1).

6.3.1.2.2 Appropriate usage

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

The association decision and the response should become available at the coordinator within a time of *macResponseWaitTime* (see 5.1.4.1). After this time, the device requesting association attempts to extract the association response command frame from the coordinator, using the method described in 5.1.7.3, in order to determine whether the association was successful.

6.3.1.3 MLME-ASSOCIATE.response

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

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

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

Table 32 specifies the parameters for the MLME-ASSOCIATE.response primitive.

Table 32—MLME-ASSOCIATE.response parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting association.
AssocShortAddress	Integer	0x0000–0xffff	The 16-bit short device address allocated by the coordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful.
status	Enumeration	Refer to 5.3.2.3	The status of the association attempt.
CapabilityNegotiationResponse	Integer	00–11	The coordinator indicates who will send color compensation information (same definitions and usage as the color stabilization scheme subfield in Table 20).

Table 32—MLME-ASSOCIATE.response parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

6.3.1.3.1 Appropriate usage

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

6.3.1.3.2 Effect on receipt

When the MLME of a coordinator receives the MLME-ASSOCIATE.response primitive, it generates an association response command as shown in 5.3.2. The command frame is sent to the device requesting association using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 5.1.7.3.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based the DeviceAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-COMM-STATUS.indication primitive with the error status returned by outgoing frame processing.

Upon receipt of the MLME-ASSOCIATE.response primitive, the coordinator attempts to add the information contained in the primitive to its list of pending transactions. If there is no capacity to store the transaction, the MAC sublayer will discard the frame and issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransactionPersistenceTime*, the transaction information will be discarded and the MAC sublayer will issue the MLME-COMM-STATUS.indication primitive with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 5.1.6.

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

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

6.3.1.4 MLME-ASSOCIATE.confirm

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

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

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

Table 33 specifies the parameters for the MLME-ASSOCIATE.confirm primitive.

Table 33—MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
AssocShortAddress	Integer	0x0000–0xffff	The short device address allocated by the coordinator on successful association. This parameter will be equal to 0xffff if the association attempt was unsuccessful.
status	Enumeration	The value of the Status field of the association response command (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 INVALID_PARAMETER	The status of the association attempt.
CapabilityNegotiationResponse	Integer	00–11	Coordinator indicates who will send (see Table 20).

Table 33—MLME-ASSOCIATE.confirm parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	If the primitive was generated following failed outgoing processing of an association request command: The security level to be used (as defined in Table 64 in 7.4.2.1). If the primitive was generated following receipt of an association response command: The security level purportedly used by the received frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was generated following failed outgoing processing of an association request command: The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00. If the primitive was generated following receipt of an association response command: The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated following failed outgoing processing of an association request command: The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated following receipt of an association response command: The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

Table 33—MLME-ASSOCIATE.confirm parameters *(continued)*

Name	Type	Valid range	Description
KeyIndex	Integer	0x01–0xff	If the primitive was generated following failed outgoing processing of an association request command: The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated following receipt of an association response command: The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.1.4.1 When generated

The MLME-ASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-ASSOCIATE.request primitive. If the request was successful, the status parameter will indicate a successful association, as contained in the Status field of the association response command. Otherwise, the status parameter indicates either an error code from the received association response command or the appropriate error code from Table 33. The status values are fully described in 6.3.1.1.2 and subclauses referenced by 6.3.1.1.2.

6.3.1.4.2 Appropriate usage

On receipt of the MLME-ASSOCIATE.confirm primitive, the next higher layer of the initiating device is notified of the result of its request to associate with a coordinator. If the association attempt was successful, the status parameter will indicate a successful association, as contained in the Status field of the association response command, and the device will be provided with a 16-bit short address as specified in Table 2 in 5.1.4.1. If the association attempt was unsuccessful, the address will be equal to 0xffff, and the status parameter will indicate the error.

6.3.1.5 Association-message sequence charts

Figure 86 illustrates a sequence of messages that may be used by a device that is not tracking the beacon of the coordinator, specified in 5.1.7.3, to successfully associate with a VPAN. Figure 102 and Figure 103, and described in 6.6, illustrate this same scenario, including steps taken by the PHY, for a device associating with a coordinator and for a coordinator allowing association by a device, respectively.

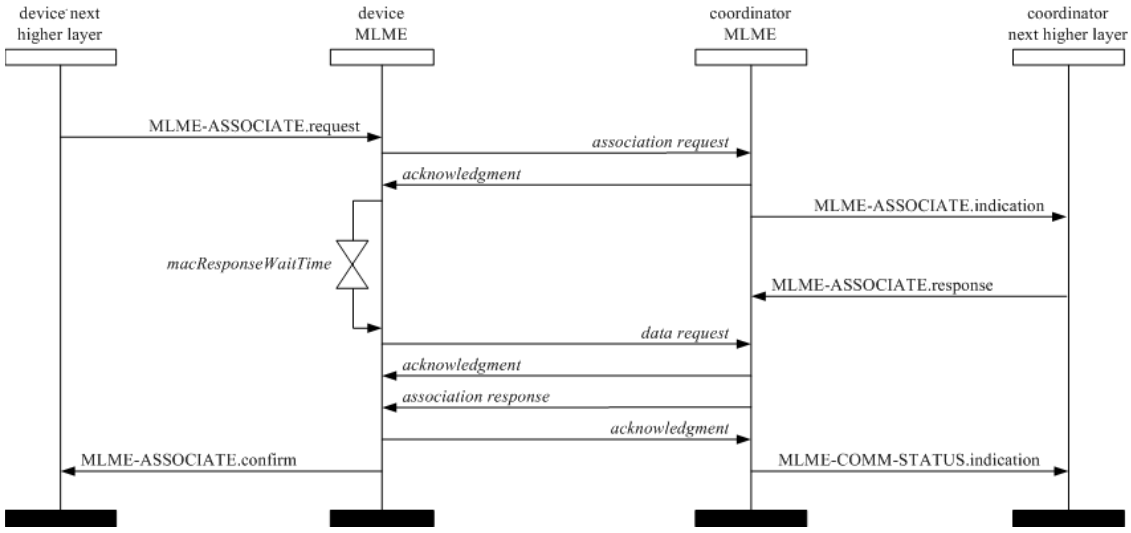


Figure 86—Message sequence chart for association

6.3.2 Disassociation primitives

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

All devices shall provide an interface for these disassociation primitives.

6.3.2.1 MLME-DISASSOCIATE.request

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

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

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

Table 34 specifies the parameters for the MLME-DISASSOCIATE.request primitive.

Table 34—MLME-DISASSOCIATE.request parameters

Name	Type	Valid range	Description
DeviceAddrMode	Integer	0x02–0x03	The addressing mode of the device to which to send the disassociation notification command.
DeviceVPANId	Integer	0x0000–0xffff	The VPAN identifier of the device to which to send the disassociation notification command.
DeviceAddress	Device address	As specified by the DeviceAddrMode parameter.	The address of the device to which to send the disassociation notification command.
DisassociateReason	Integer	0x00–0xff	The reason for the disassociation (as defined in 5.3.3.2).
TxIndirect	Boolean	TRUE or FALSE	TRUE if the disassociation notification command is to be sent indirectly.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
ColorDisAssoc	Boolean	TRUE or FALSE	ColorDisAssoc shall be set as TRUE if the color CVD frame is to be transmitted after the disassociation notification command is sent.

6.3.2.1.1 Appropriate usage

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

6.3.2.1.2 Effect on receipt

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

If the DeviceAddrMode parameter is equal to 0x02 and the DeviceAddress parameter is equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to 0x03 and the DeviceAddress parameter is equal to *macCoordExtendedAddress*, the TxIndirect parameter is ignored, and the MLME sends

a disassociation notification command (see 5.3.3) to its coordinator in the CAP for a beacon-enabled VPAN or immediately for a nonbeacon-enabled VPAN.

If the DeviceAddrMode parameter is equal to 0x02 and the DeviceAddress parameter is not equal to *macCoordShortAddress* or if the DeviceAddrMode parameter is equal to 0x03 and the DeviceAddress parameter is not equal to *macCoordExtendedAddress*, and if this primitive was received by the MLME of a coordinator with the TxIndirect parameter set to TRUE, the disassociation notification command will be sent using indirect transmission, i.e., the command frame is added to the list of pending transactions stored on the coordinator and extracted at the discretion of the device concerned using the method described in 5.1.7.3.

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

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

If the disassociation notification command is to be sent using indirect transmission and there is no capacity to store the transaction, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with a status of TRANSACTION_OVERFLOW. If there is capacity to store the transaction, the coordinator will add the information to the list. If the transaction is not handled within *macTransaction-PersistenceTime*, the transaction information will be discarded, and the MLME will issue the MLME-DISASSOCIATE.confirm with a status of TRANSACTION_EXPIRED. The transaction handling procedure is described in 5.1.6.

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

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on the DeviceAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-DISASSOCIATE.confirm primitive with the error status returned by outgoing frame processing.

If the MLME successfully transmits a disassociation notification command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received and this primitive was received either by the MLME of a coordinator with the TxIndirect parameter set to FALSE or by the MLME of a device, the MLME will issue the MLME-DISASSOCIATE.confirm primitive with a status of NO_ACK (see 5.1.7.4).

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

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

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

6.3.2.2 MLME-DISASSOCIATE.indication

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

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

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

Table 35 specifies the parameters for the MLME-DISASSOCIATE.indication primitive.

Table 35—MLME-DISASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	An extended 64-bit IEEE address	The address of the device requesting disassociation.
DisassociateReason	Integer	0x00–0xff	The reason for the disassociation (as defined in 5.3.3.2).
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received MAC command frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.2.2.1 When generated

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

6.3.2.2.2 Appropriate usage

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

6.3.2.3 MLME-DISASSOCIATE.confirm

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

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

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

Table 36 specifies the parameters for the MLME-DISASSOCIATE.confirm primitive.

Table 36—MLME-DISASSOCIATE.confirm parameters

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

6.3.2.3.1 When generated

The MLME-DISASSOCIATE.confirm primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-DISASSOCIATE.request primitive. This primitive returns a status of either SUCCESS, indicating that the disassociation request was successful, or the appropriate error code. The status values are fully described in 6.3.2.1.2 and subclauses referenced by 6.3.2.1.2.

6.3.2.3.2 Appropriate usage

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

6.3.2.4 Disassociation-message sequence charts

The request to disassociate may originate either from a device or from the coordinator through which the device has associated. Figure 87 illustrates the sequence of messages necessary for a device to successfully disassociate itself from the VPAN.

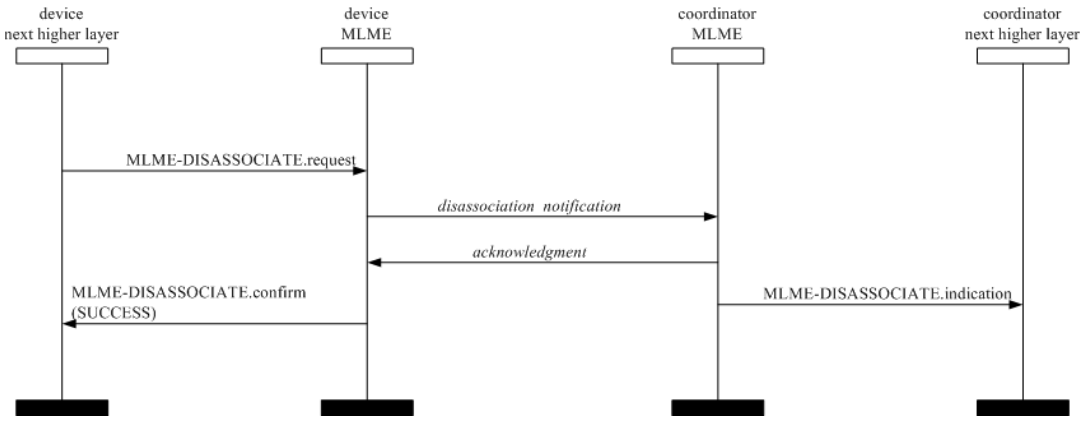


Figure 87—Message sequence chart for disassociation initiated by a device

Figure 88 illustrates the sequence necessary for a coordinator in a beacon-enabled VPAN to successfully disassociate a device from its VPAN using indirect transmission.

6.3.3 Beacon notification primitive

The MLME-SAP beacon notification primitive defines how a device may be notified when a beacon is received during normal operating conditions.

All devices shall provide an interface for the beacon notification primitive.

6.3.3.1 MLME-BEACON-NOTIFY.indication

The MLME-BEACON-NOTIFY.indication primitive is used to send parameters contained within a beacon frame received by the MAC sublayer to the next higher layer. The primitive also sends a measure of the WQI and the time the beacon frame was received.

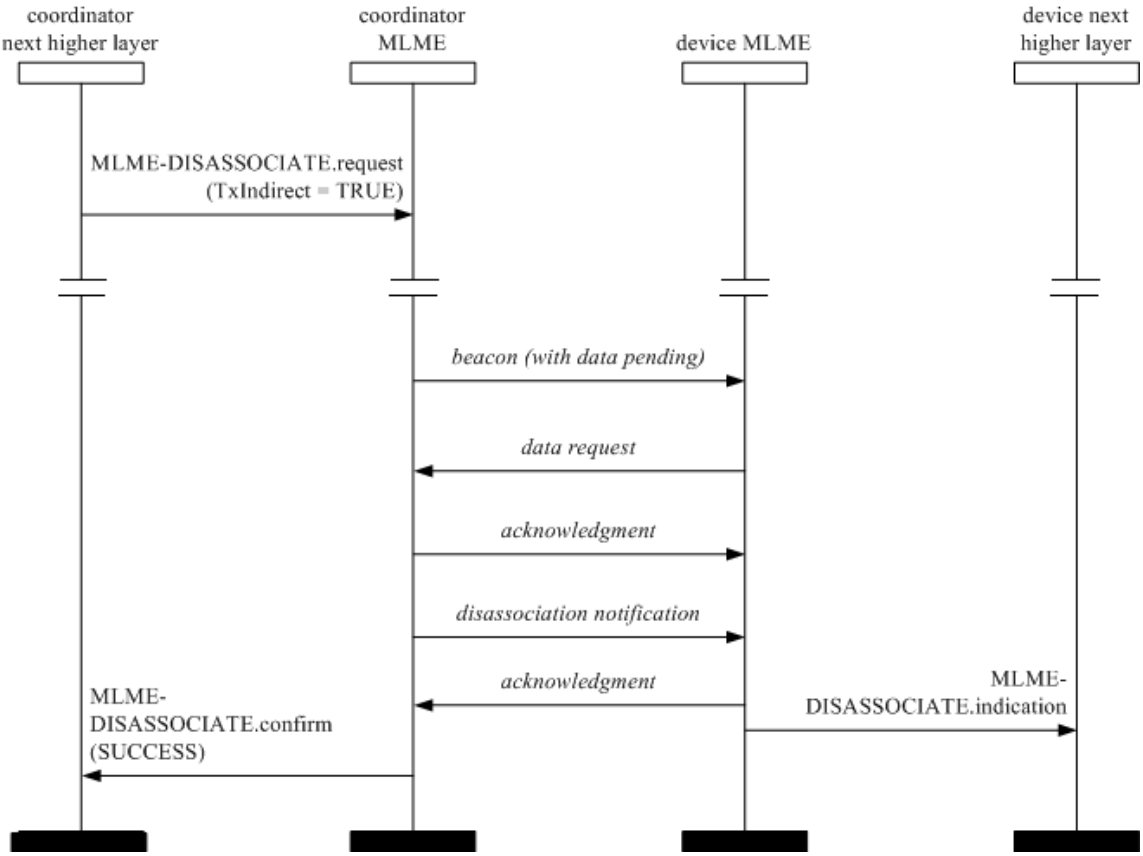


Figure 88—Message sequence chart for disassociation initiated by a coordinator, using indirect transmission, in a beacon-enabled VPAN

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

```
MLME-BEACON-NOTIFY.indication   (
                                BSN,
                                VPANDescriptor,
                                PendAddrSpec,
                                AddrList,
                                sduLength,
                                sdu
                                )
```

Table 37 specifies the parameters for the MLME-BEACON-NOTIFY.indication primitive.

Table 38 describes the elements of the VPANDescriptor type.

6.3.3.1.1 When generated

The MLME-BEACON-NOTIFY.indication primitive is generated by the MLME and issued to its next higher layer upon receipt of a beacon frame either when *macAutoRequest* is set to FALSE or when the beacon frame contains one or more octets of payload.

Table 37—MLME-BEACON-NOTIFY.indication parameters

Name	Type	Valid range	Description
BSN	Integer	0x00–0xff	The beacon sequence number.
VPANDescriptor	VPANDescriptor value	Refer to Table 38	The VPANDescriptor for the received beacon.
PendAddrSpec	Bitmap	Refer to 5.2.2.1.6	The beacon pending address specification.
AddrList	List of device addresses	—	The list of addresses of the devices for which the beacon source has data.
sduLength	Integer	0 – *aMaxBeaconPayloadLength*	The number of octets contained in the beacon payload of the beacon frame received by the MAC sublayer.
sdu	Set of octets	—	The set of octets comprising the beacon payload to be transferred from the MAC sublayer entity to the next higher layer.

Table 38—Elements of VPAN descriptor

Name	Type	Valid range	Description
CoordAddrMode	Integer	0x02–0x03	The coordinator addressing mode corresponding to the received beacon frame. This value can take one of the following values: 2 = 16-bit short address. 3 = 64-bit extended address.
CoordVPANId	Integer	0x0000–0xffff	The VPAN identifier of the coordinator as specified in the received beacon frame.
CoordAddress	Device address	As specified by the CoordAddrMode parameter	The address of the coordinator as specified in the received beacon frame.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see Table 76).	The current logical channel occupied by the network.
SuperframeSpec	Bitmap	Refer to 5.2.2.1.2	The superframe specification as specified in the received beacon frame.
GTSPermit	Boolean	TRUE or FALSE	TRUE if the beacon is from the coordinator that is accepting GTS requests.
LinkQuality	Integer	0x00–0xff	The WQI at which the network beacon was received. Lower values represent lower WQI (see 5.3.19.2).
TimeStamp	Integer	0x000000–0xffffff	The time at which the beacon frame was received, in symbols. This value is equal to the timestamp taken when the beacon frame was received, as described in 5.1.5.1. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.

Table 38—Elements of VPAN descriptor *(continued)*

Name	Type	Valid range	Description
SecurityFailure	Enumer-ation	SUCCESS, COUNTER_ERROR, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY	SUCCESS if there was no error in the security processing of the frame. One of the other status codes indicating an error in the security processing otherwise (see 7.2.3).
SecurityLevel	Integer	0x00–0x07	The security level purportedly used by the received beacon frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.3.1.2 Appropriate usage

On receipt of the MLME-BEACON-NOTIFY.indication primitive, the next higher layer is notified of the arrival of a beacon frame at the MAC sublayer.

6.3.4 Primitives for reading PIB attributes

The MLME-SAP get primitives define how to read values from the PIB.

All devices shall provide an interface for these get primitives.

6.3.4.1 MLME-GET.request

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

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

```
MLME-GET.request        (
                        PIBAttribute,
                        PIBAttributeIndex
                        )
```

Table 39 specifies the parameters for the MLME-GET.request primitive.

Table 39—MLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	Refer to Table 60	The identifier of the PIB attribute to read.
PIBAttributeIndex	Integer	Attribute specific; as defined in Table 60	The index within the table of the specified PIB attribute to read. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.

6.3.4.1.1 Appropriate usage

The MLME-GET.request primitive is generated by the next higher layer and issued to its MLME to obtain information from the PIB.

6.3.4.1.2 Effect on receipt

On receipt of the MLME-GET.request primitive, the MLME checks to see if the PIB attribute is a MAC PIB attribute or PHY PIB attribute. If the requested attribute is a MAC attribute, the MLME attempts to retrieve the requested MAC PIB attribute from its database. If the identifier of the PIB attribute is not found in the database, the MLME will issue the MLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttributeIndex parameter specifies an index for a table that is out of range, the MLME will issue the MLME-GET.confirm primitive with a status of INVALID_INDEX. If the requested MAC PIB attribute is successfully retrieved, the MLME will issue the MLME-GET.confirm primitive with a status of SUCCESS.

If the requested attribute is a PHY PIB attribute, the request is passed to the PHY by issuing the PLME-GET.request primitive. Once the MLME receives the PLME-GET.confirm primitive, it will translate the received status value because the status values used by the PHY are not the same as those used by the MLME (e.g., the status values for SUCCESS are 0x00 and 0x07 in the MAC and PHY enumeration tables, respectively). Following the translation, the MLME will issue the MLME-GET.confirm primitive to the next higher layer with the status parameter resulting from the translation and the PIBAttribute and PIBAttributeValue parameters equal to those returned by the PLME primitive.

6.3.4.2 MLME-GET.confirm

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

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

```
MLME-GET.confirm        (
                        status,
                        PIBAttribute,
                        PIBAttributeIndex,
                        PIBAttributeValue
                        )
```

Table 40 specifies the parameters for the MLME-GET.confirm primitive.

Table 40—MLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE or INVALID_INDEX	The result of the request for PIB attribute information.
PIBAttribute	Integer	Refer to Table 60	The identifier of the PIB attribute that was read.
PIBAttributeIndex	Integer	Attribute specific; as defined in Table 60	The index within the table or array of the specified PIB attribute to read. This parameter is valid only for MAC PIB attributes that are tables or arrays; it is ignored when accessing PHY PIB attributes.
PIBAttributeValue	Various	Attribute specific; as defined in Table 60	The value of the indicated PIB attribute that was read. This parameter has zero length when the status parameter is set to UNSUPPORTED_ATTRIBUTE.

6.3.4.2.1 When generated

The MLME-GET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-GET.request primitive. This primitive returns a status of either SUCCESS, indicating that the request to read a PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. When an error code of UNSUPPORTED_ATTRIBUTE is returned, the PIBAttribute value parameter will be set to length zero. The status values are fully described in 6.3.4.1.2.

6.3.4.2.2 Appropriate usage

On receipt of the MLME-GET.confirm primitive, the next higher layer is notified of the results of its request to read a PIB attribute. If the request to read a PIB attribute was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.3.5 GTS management primitives

The MLME-SAP GTS management primitives define how GTSs are requested and maintained. A device wishing to use these primitives and GTSs in general will already be tracking the beacons of its coordinator.

These GTS management primitives are optional.

6.3.5.1 MLME-GTS.request

The MLME-GTS.request primitive allows a device to send a request to the coordinator to allocate a new GTS or to deallocate an existing GTS. This primitive is also used by the coordinator to initiate a GTS deallocation.

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

```
MLME-GTS.request        (
                        GTSCharacteristics,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 41 specifies the parameters for the MLME-GTS.request primitive.

Table 41—MLME-GTS.request parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	Refer to 5.3.13.2	The characteristics of the GTS request, including whether the request is for the allocation of a new GTS or the deallocation of an existing GTS.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

6.3.5.1.1 Appropriate usage

The MLME-GTS.request primitive is generated by the next higher layer of a device and issued to its MLME to request the allocation of a new GTS or to request the deallocation of an existing GTS. It is also generated by the next higher layer of the coordinator and issued to its MLME to request the deallocation of an existing GTS.

6.3.5.1.2 Effect on receipt

On receipt of the MLME-GTS.request primitive by a device, the MLME of a device attempts to generate a GTS request command, specified in 5.3.13, with the information contained in this primitive and, if successful, sends it to the coordinator.

If *macShortAddress* is equal to 0xfffe or 0xffff, the device is not permitted to request a GTS. In this case, the MLME issues the MLME-GTS.confirm primitive containing a status of NO_SHORT_ADDRESS.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on *macCoordExtendedAddress* and the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-GTS.confirm primitive with the error status returned by outgoing frame processing.

If the GTS request command cannot be sent due to an unslotted random access algorithm failure, the MLME will issue the MLME-GTS.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a GTS request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_ACK (see 5.1.7.4).

If a GTS is being allocated (see 5.1.8.2) and the request has been acknowledged, the device will wait for a confirmation via a GTS descriptor specified in a beacon frame from its coordinator. If the MLME of the coordinator can allocate the requested GTS, it will issue the MLME-GTS.indication primitive with the characteristics of the allocated GTS and generate a GTS descriptor with the characteristics of the allocated GTS and the 16-bit short address of the requesting device. If the MLME of the coordinator cannot allocate the requested GTS, it will generate a GTS descriptor with a start slot of zero and the short address of the requesting device.

If the device receives a beacon frame from its coordinator with a GTS descriptor containing a 16-bit short address that matches *macShortAddress*, the device will process the descriptor. If no descriptor for that device is received, the MLME will issue the MLME-GTS.confirm primitive with a status of NO_DATA.

If a descriptor is received that matches the characteristics requested (indicating that the coordinator has approved the GTS allocation request), the MLME of the device will issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to one, indicating a GTS allocation.

If the descriptor is received with a start slot of zero (indicating that the coordinator has denied the GTS allocation request), the device requesting the GTS issues the MLME-GTS.confirm primitive with a status of DENIED, indicating that the GTSCharacteristics parameter is to be ignored.

If a GTS is being deallocated (see 5.1.8.4) at the request of a device and the request has been acknowledged by the coordinator, the device will issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to zero, indicating a GTS deallocation. On receipt of a GTS request command with a request type indicating a GTS deallocation, the coordinator will acknowledge the frame and deallocates the GTS. The MLME of the coordinator will then issue the MLME-GTS.indication primitive with the appropriate GTS characteristics. If the coordinator does not receive the deallocation request, countermeasures can be applied by the coordinator to ensure consistency is maintained as discussed in 5.1.8.6.

If the MLME of the coordinator receives an MLME-GTS.request primitive indicating deallocation, the coordinator will deallocate the GTS and issue the MLME-GTS.confirm primitive with a status of SUCCESS and a GTSCharacteristics parameter with a characteristics type equal to zero.

If the device receives a beacon frame from its coordinator with a GTS descriptor containing a short address that matches *macShortAddress* and a start slot equal to zero, the device immediately stops using the GTS. The MLME of the device then notifies the next higher layer of the deallocation by issuing the MLME-GTS.indication primitive with a GTSCharacteristics parameter containing the characteristics of the deallocated GTS.

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

6.3.5.2 MLME-GTS.indication

The MLME-GTS.indication primitive indicates that a GTS has been allocated or that a previously allocated GTS has been deallocated.

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

```
MLME-GTS.indication (
                    DeviceAddress,
                    GTSCharacteristics,
                    SecurityLevel,
                    KeyIdMode,
                    KeySource,
                    KeyIndex
                    )
```

Table 42 specifies the parameters for the MLME-GTS.indication primitive.

Table 42—MLME-GTS.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	0x0000–0xfffd	The 16-bit short address of the device that has been allocated or deallocated a GTS.
GTSCharacteristics	GTS characteristics	Refer to 5.3.13.2	The characteristics of the GTS.
SecurityLevel	Integer	0x00–0x07	If the primitive was generated when a GTS deallocation is initiated by the coordinator itself, the security level to be used is set to 0x00. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The security level purportedly used by the received MAC command frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was generated when a GTS deallocation is initiated by the coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.

Table 42—MLME-GTS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated when a GTS deallocation is initiated by the coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	If the primitive was generated when a GTS deallocation is initiated by the coordinator itself, this parameter is ignored. If the primitive was generated whenever a GTS is allocated or deallocated following the reception of a GTS request command: The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.5.2.1 When generated

The MLME-GTS.indication primitive is generated by the MLME of the coordinator to its next higher layer whenever a GTS is allocated or deallocated following the reception of a GTS request command by the MLME as discussed in 5.3.13. The MLME of the coordinator also generates this primitive when a GTS deallocation is initiated by the coordinator itself. The Characteristics Type field in the GTSCharacteristics parameter will be equal to one if a GTS has been allocated or zero if a GTS has been deallocated.

This primitive is generated by the MLME of a device and issued to its next higher layer when the coordinator has deallocated one of its GTSs. In this case, the Characteristics Type field of the GTSCharacteristics parameter is equal to zero.

6.3.5.2.2 Appropriate usage

On receipt of the MLME-GTS.indication primitive the next higher layer is notified of the allocation or de-allocation of a GTS.

6.3.5.3 MLME-GTS.confirm

The MLME-GTS.confirm primitive reports the results of a request to allocate a new GTS or deallocate an existing GTS.

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

```
MLME-GTS.confirm        (
                        GTSCharacteristics,
                        status
                        )
```

Table 43 specifies the parameters for the MLME-GTS.confirm primitive.

Table 43—MLME-GTS.confirm parameters

Name	Type	Valid range	Description
GTSCharacteristics	GTS characteristics	Refer to 5.3.13.2	The characteristics of the GTS.
status	Enumeration	SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER	The status of the GTS request.

6.3.5.3.1 When generated

The MLME-GTS.confirm primitive is generated by the MLME and issued to its next higher layer in response to a previously issued MLME-GTS.request primitive.

If the request to allocate or deallocate a GTS was successful, this primitive will return a status of SUCCESS and the Characteristics Type field of the GTSCharacteristics parameter will have the value of one or zero, respectively. Otherwise, the status parameter will indicate the appropriate error code. The reasons for these status values are fully described in 6.3.5.1.2 and subclauses referenced by 6.3.5.1.2.

6.3.5.3.2 Appropriate usage

On receipt of the MLME-GTS.confirm primitive the next higher layer is notified of the result of its request to allocate or deallocate a GTS. If the request was successful, the status parameter will indicate a successful GTS operation. Otherwise, the status parameter will indicate the error.

6.3.5.4 GTS management message sequence charts

Figure 89 and Figure 90 illustrate the sequence of messages necessary for successful GTS management. The first depicts the message flow for the case in which the device initiates the GTS allocation. The second depicts the message flow for the two cases for which a GTS deallocation occurs, first, by a device (a) and, second, by the coordinator (b).

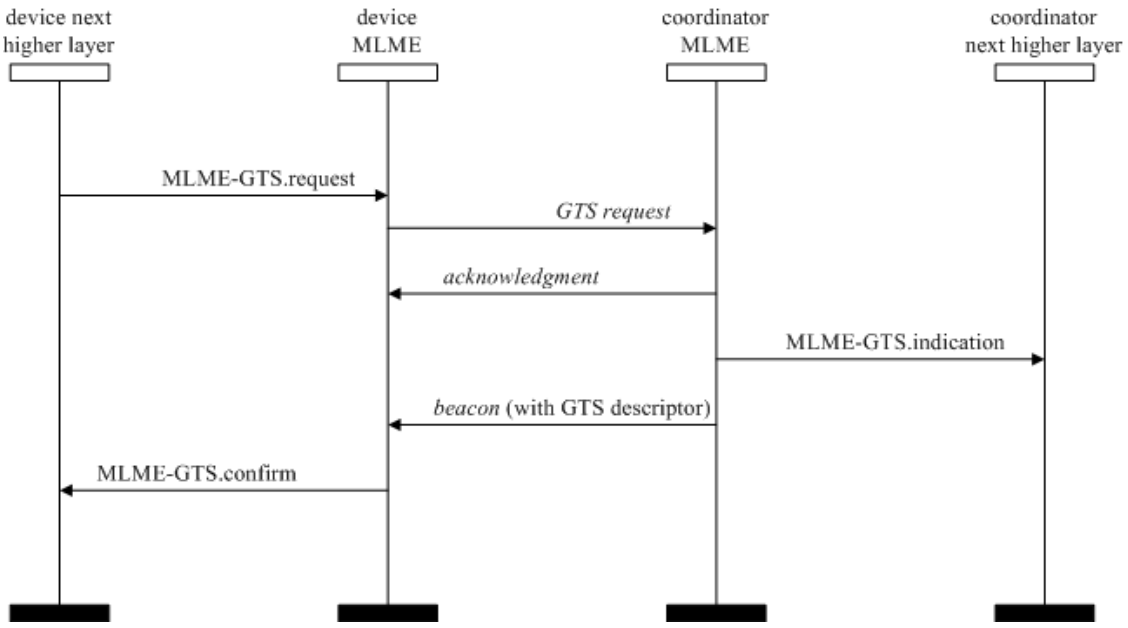


Figure 89—Message sequence chart for GTS allocation initiated by a device

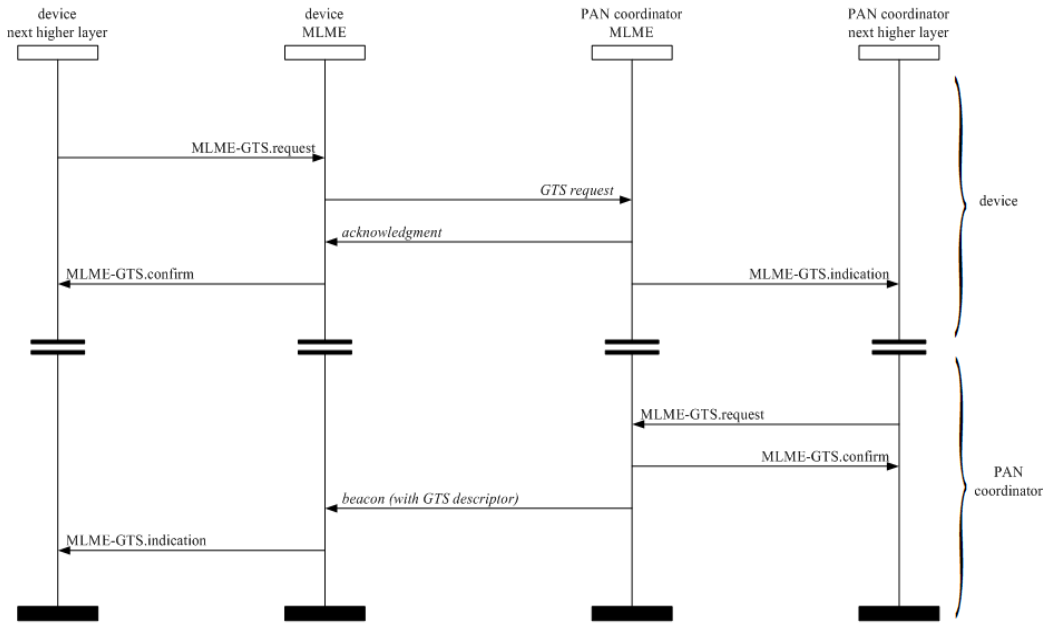


Figure 90—Message sequence chart for GTS deallocation initiated by a device (a) and the PAN coordinator (b)

6.3.6 Primitives for resetting the MAC sublayer

MLME-SAP reset primitives specify how to reset the MAC sublayer to its default values.

All devices shall provide an interface for these reset primitives.

6.3.6.1 MLME-RESET.request

The MLME-RESET.request primitive allows the next higher layer to request that the MLME performs a reset operation.

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

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

Table 44 specifies the parameter for the MLME-RESET.request primitive.

Table 44—MLME-RESET.request parameter

Name	Type	Valid range	Description
SetDefaultPIB	Boolean	TRUE or FALSE	If TRUE, the MAC sublayer is reset, and all MAC PIB attributes are set to their default values. If FALSE, the MAC sublayer is reset, but all MAC PIB attributes retain their values prior to the generation of the MLME-RESET.request primitive.

6.3.6.1.1 Appropriate usage

The MLME-RESET.request primitive is generated by the next higher layer and issued to the MLME to request a reset of the MAC sublayer to its initial conditions. The MLME-RESET.request primitive is issued prior to the use of the MLME-START.request or the MLME-ASSOCIATE.request primitives.

6.3.6.1.2 Effect on receipt

On receipt of the MLME-RESET.request primitive, the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of FORCE_TRX_OFF. On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MAC sublayer is then set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultPIB parameter is set to TRUE, the MAC PIB attributes are set to their default values.

The MLME-RESET.confirm primitive with a status of SUCCESS is issued on completion.

6.3.6.2 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

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

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

Table 45 specifies the parameter for the MLME-RESET.confirm primitive.

Table 45—MLME-RESET.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS	The result of the reset operation.

6.3.6.2.1 When generated

The MLME-RESET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RESET.request primitive and following the receipt of the PLME-SET-TRX-STATE.confirm primitive.

6.3.6.2.2 Appropriate usage

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of its request to reset the MAC sublayer. This primitive returns a status of SUCCESS indicating that the request to reset the MAC sublayer was successful.

6.3.7 Primitives for specifying the receiver enable time

MLME-SAP receiver state primitives define how a device can enable or disable the receiver at a given time.

These receiver state primitives are optional.

6.3.7.1 MLME-RX-ENABLE.request

The MLME-RX-ENABLE.request primitive allows the next higher layer to request that the receiver is either enabled for a finite period of time or disabled.

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

```
MLME-RX-ENABLE.request      (
                            DeferPermit,
                            RxOnTime,
                            RxOnDuration
                            )
```

Table 46 specifies the parameters for the MLME-RX-ENABLE.request primitive.

6.3.7.1.1 Appropriate usage

The MLME-RX-ENABLE.request primitive is generated by the next higher layer and issued to the MLME to enable the receiver for a fixed duration, at a time relative to the start of the current or next superframe on a beacon-enabled VPAN or immediately on a nonbeacon-enabled VPAN. This primitive may also be generated to cancel a previously generated request to enable the receiver. The receiver is enabled or disabled exactly once per primitive request.

Table 46—MLME-RX-ENABLE.request parameters

Name	Type	Valid range	Description
DeferPermit	Boolean	TRUE or FALSE	TRUE if the requested operation can be deferred until the next superframe if the requested time has already passed. FALSE if the requested operation is only to be attempted in the current superframe. This parameter is ignored for nonbeacon-enabled VPANs. If the issuing device is the VPAN coordinator, the term *superframe* refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated.
RxOnTime	Integer	0x000000–0xffffff	The number of optical clocks measured from the start of the superframe before the receiver is to be enabled or disabled. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant. This parameter is ignored for nonbeacon-enabled VPANs. If the issuing device is the VPAN coordinator, the term *superframe* refers to its own superframe. Otherwise, the term refers to the superframe of the coordinator through which the issuing device is associated.
RxOnDuration	Integer	0x000000–0xffffff	The number of optical clocks for which the receiver is to be enabled. If this parameter is equal to 0x000000, the receiver is to be disabled.

6.3.7.1.2 Effect on receipt

The MLME will treat the request to enable or disable the receiver as secondary to other responsibilities of the device (e.g., GTSs, coordinator beacon tracking, beacon transmissions). When the primitive is issued to enable the receiver, the device will enable its receiver until either the device has a conflicting responsibility or the time specified by RxOnDuration has expired. In the case of a conflicting responsibility, the device will interrupt the receive operation. After the completion of the interrupting operation, the RxOnDuration will be checked to determine whether the time has expired. If so, the operation is complete. If not, the receiver is re-enabled until either the device has another conflicting responsibility or the time specified by RxOnDuration has expired. When the primitive is issued to disable the receiver, the device will disable its receiver unless the device has a conflicting responsibility.

On a nonbeacon-enabled VPAN, the MLME ignores the DeferPermit and RxOnTime parameters and requests that the PHY enable or disable the receiver immediately. If the request is to enable the receiver, the receiver will remain enabled until RxOnDuration symbols have elapsed.

Before attempting to enable the receiver on a beacon-enabled VPAN, the MLME first determines whether (RxOnTime + RxOnDuration) is less than the beacon interval, as defined by *macBeaconOrder*. If (RxOnTime + RxOnDuration) is not less than the beacon interval, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of ON_TIME_TOO_LONG.

The MLME then determines whether the receiver can be enabled in the current superframe. The VPAN coordinator issuing this primitive makes the determination based on its own superframe. A device that is not

the VPAN coordinator makes the determination based on the superframe of the coordinator through which it is associated. If the current number of optical clocks measured from the start of the superframe is less than RxOnTime, the MLME attempts to enable the receiver in the current superframe. If the current number of optical clocks measured from the start of the superframe is greater than or equal to RxOnTime and DeferPermit is equal to TRUE, the MLME defers until the next superframe and attempts to enable the receiver in that superframe. Otherwise, if the MLME cannot enable the receiver in the current superframe and is not permitted to defer the receive operation until the next superframe, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of PAST_TIME.

If the RxOnDuration parameter is equal to zero, the MLME requests that the PHY disable its receiver.

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

If the request to enable or disable the receiver was successful, the MLME issues the MLME-RX-ENABLE.confirm primitive with a status of SUCCESS.

6.3.7.2 MLME-RX-ENABLE.confirm

The MLME-RX-ENABLE.confirm primitive reports the results of the attempt to enable or disable the receiver.

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

```
MLME-RX-ENABLE.confirm      (
                            status
                            )
```

Table 47 specifies the parameter for the MLME-RX-ENABLE.confirm primitive.

Table 47—MLME-RX-ENABLE.confirm parameter

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, PAST_TIME, ON_TIME_TOO_LONG, INVALID_PARAMETER	The result of the request to enable or disable the receiver.

6.3.7.2.1 When generated

The MLME-RX-ENABLE.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RX-ENABLE.request primitive.

6.3.7.2.2 Appropriate usage

On receipt of the MLME-RX-ENABLE.confirm primitive, the next higher layer is notified of its request to enable or disable the receiver. This primitive returns a status of either SUCCESS, if the request to enable or disable the receiver was successful, or the appropriate error code. The status values are fully described in 6.3.7.1.2.

6.3.7.3 Message sequence chart for changing the state of the receiver

Figure 91 illustrates the sequence of messages necessary for enabling the receiver for a fixed duration when the device does not have any conflicting responsibilities. Figure 91a) illustrates the case for a beacon-enabled VPAN where it is assumed both that the MLME-RX-ENABLE.request has been received by the MLME without sufficient time available to enable the receiver in the current superframe and that the DeferPermit parameter is TRUE. Figure 91b) illustrates the case for a nonbeacon-enabled VPAN where the receiver is enabled immediately.

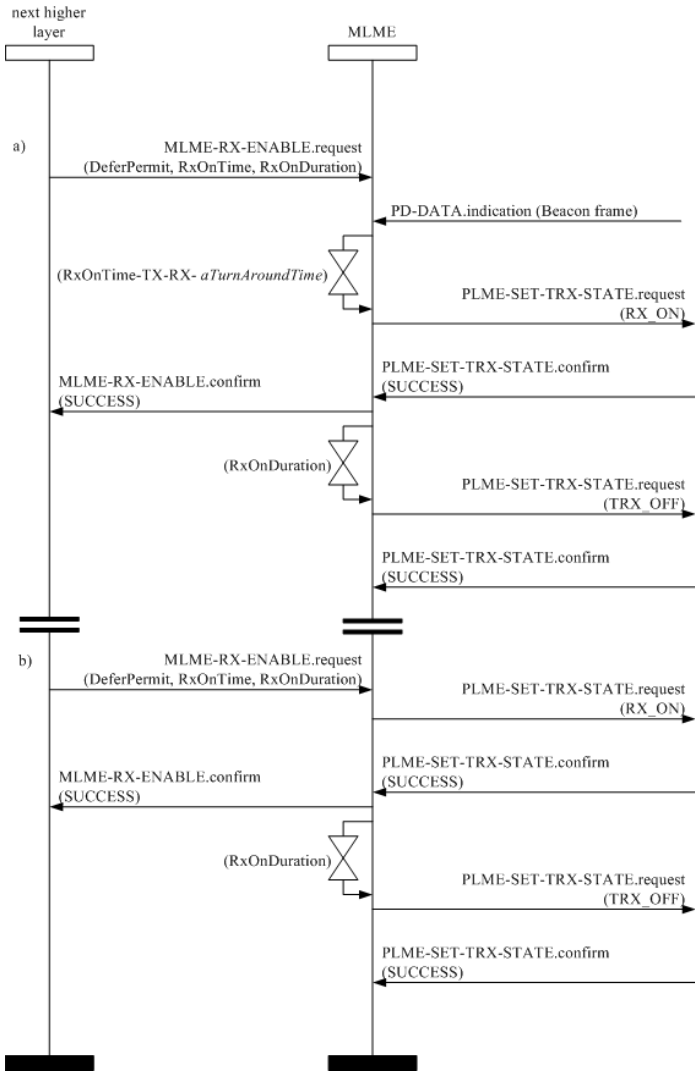


Figure 91—Message sequence chart for changing the state of the receiver

6.3.8 Primitives for channel scanning

MLME-SAP scan primitives define how a device can determine the energy usage or the presence or absence of VPANs in a communications channel.

All devices shall provide an interface for these scan primitives.

6.3.8.1 MLME-SCAN.request

The MLME-SCAN.request primitive is used to initiate a channel scan over a given list of channels. A device can use a channel scan to measure the energy on the channel, search for the coordinator with which it associated, or search for all coordinators transmitting beacon frames within the coverage area of the scanning device.

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

```
MLME-SCAN.request       (
                        ScanType,
                        ScanChannels,
                        ScanDuration,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex,
                        ColorScan
                        )
```

Table 48 specifies the parameters for the MLME-SCAN.request primitive.

Table 48—MLME-SCAN.request parameters

Name	Type	Valid range	Description
ScanType	Integer	0x00–0x01	Indicates the type of scan performed: 0x00 = active scan (optional for a device). 0x01 = passive scan.
ScanChannels	Bitmap	7-bit field	The 7 bits (b_0, b_1,... b_6) indicate which channels are to be scanned (1 = scan, 0 = do not scan).
ScanDuration	Integer	0–14	The time spent scanning each channel is [*aBaseSuperframeDuration* × (2^n + 1)] optical clocks, where *n* is the value of the ScanDuration parameter.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
ColorScan	Boolean	TRUE or FALSE	ColorScan shall be set as TRUE if the color CVD frame is to be transmitted either during passive scan or after the beacon request command is sent (see 5.1.2.1) for an active scan.

6.3.8.1.1 Appropriate usage

The MLME-SCAN.request primitive is generated by the next higher layer and issued to its MLME to initiate a channel scan to search for activity within the coverage area of the device. This primitive can be used to perform an active or passive scan to locate beacon frames containing any VPAN identifier. Refer to 5.1.3.1 for a description of each type of scan in detail.

All devices shall be capable of performing passive scans, while active scans are optional for a device. However, a device may support active scanning to participate in a nonbeacon-enabled network.

6.3.8.1.2 Effect on receipt

If the MLME receives the MLME-SCAN.request primitive while performing a previously initiated scan operation, it issues the MLME-SCAN.confirm primitive with a status of SCAN_IN_PROGRESS. Otherwise, the MLME initiates a scan in all channels specified in the ScanChannels parameter.

The active scan is performed on each channel by the MLME first sending a beacon request command as specified in 5.3.6. The MLME then enables the receiver and records the information contained in each received beacon in a VPAN descriptor structure as shown in Table 38. The active scan on a particular channel terminates when the number of VPAN descriptors stored equals an implementation-specified maximum or when [*aBaseSuperframeDuration* × (2^n + 1)] optical clocks, where *n* is the value of the ScanDuration parameter, have elapsed, whichever comes first. Refer to 5.1.2.1.1 for more detailed information on the active channel scan procedure.

The passive scan is performed on each channel by the MLME enabling its receiver and recording the information contained in each received beacon in a VPAN descriptor structure as specified in Table 38. The passive scan on a particular channel terminates when the number of VPAN descriptors stored equals an implementation-specified maximum or when [*aBaseSuperframeDuration* × (2^n + 1)] optical clocks, 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 passive channel scan procedure.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on *macCoordExtendedAddress*, the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-SCAN.confirm primitive with the error status returned by outgoing frame processing.

The results of an active or passive scan are reported to the next higher layer through the MLME-SCAN.confirm primitive. If the scan is successful and *macAutoRequest* is set to TRUE, the primitive results will include a set of VPAN descriptor values. If the scan is successful and *macAutoRequest* is set to FALSE, the primitive results will contain a null set of VPAN descriptor values; each VPAN descriptor value will be sent individually to the next higher layer using separate MLME-BEACON-NOTIFY (see 6.3.3.1) primitives. In both cases, the MLME-SCAN.confirm primitive will contain a list of unscanned channels and a status of SUCCESS.

If, during an active scan, the MLME is unable to transmit a beacon request command on a channel specified by the ScanChannels parameter due to a channel access failure, the channel will appear in the list of unscanned channels returned by the MLME-SCAN.confirm primitive. If the MLME was able to send a beacon request command on at least one of the channels but no beacons were found, the MLME-SCAN.confirm primitive will contain a null set of VPAN descriptor values, regardless of the value of *macAutoRequest*, and a status of NO_BEACON.

If, during an active or passive scan, the implementation-specified maximum is reached thus terminating the scan procedure, the MAC sublayer will issue the MLME-SCAN.confirm primitive with a status of LIMIT_REACHED.

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

6.3.8.2 MLME-SCAN.confirm

The MLME-SCAN.confirm primitive reports the result of the channel scan request.

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

```
MLME-SCAN.confirm (
                  status,
                  ScanType,
                  UnscannedChannels,
                  ResultListSize,
                  VPANDescriptorList
                  )
```

Table 49 specifies the parameters for the MLME-SCAN.confirm primitive.

Table 49—MLME-SCAN.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, LIMIT_REACHED, NO_BEACON, SCAN_IN_PROGRESS, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the scan request.
ScanType	Integer	0x00–0x01	Indicates the type of scan performed: 0x00 = active scan 0x01 = passive scan
UnscannedChannels	Bitmap	7-bit field	Indicates which channels given in the request were not scanned (1 = not scanned, 0 = scanned or not requested).
ResultListSize	Integer	Implementation specific	The number of elements returned in the appropriate result lists.
VPANDescriptorList	List of VPAN descriptor values	Refer to Table 38	The list of VPAN descriptors, one for each beacon found during an active or passive scan if *macAutoRequest* is set to TRUE. This parameter is null when *macAutoRequest* is set to FALSE during an active or passive scan.

6.3.8.2.1 When generated

The MLME-SCAN.confirm primitive is generated by the MLME and issued to its next higher layer when the channel scan initiated with the MLME-SCAN.request primitive has completed. If the MLME-SCAN.request primitive requested an active or passive scan with *macAutoRequest* set to FALSE, the VPANDescriptorList parameter will be null.

The MLME-SCAN.confirm primitive returns a status of either SUCCESS, indicating that the requested scan was successful, or the appropriate error code. The status values are fully described in 6.3.8.1.2 and subclauses referenced by 6.3.8.1.2.

6.3.8.2.2 Appropriate usage

On receipt of the MLME-SCAN.confirm primitive, the next higher layer is notified of the results of the scan procedure. If the requested scan was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.3.8.3 Channel scan message sequence charts

Figure 104 and Figure 105 illustrate the sequence of messages necessary to perform a passive scan and an active scan. These figures include steps taken by the PHY.

6.3.9 Communication status primitive

The MLME-SAP communication status primitive defines how the MLME communicates to the next higher layer about transmission status, when the transmission was instigated by a response primitive, and about security errors on incoming packets.

All devices shall provide an interface for this communication status primitive.

6.3.9.1 MLME-COMM-STATUS.indication

The MLME-COMM-STATUS.indication primitive allows the MLME to indicate a communications status.

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

```
MLME-COMM-STATUS.indication (
                            VPANId,
                            SrcAddrMode,
                            SrcAddr,
                            DstAddrMode,
                            DstAddr,
                            status,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 50 specifies the parameters for the MLME-COMM-STATUS.indication primitive.

Table 50—MLME-COMM-STATUS.indication parameters

Name	Type	Valid range	Description
VPANId	Integer	0x0000–0xffff	The 16-bit VPAN identifier of the device from which the frame was received or to which the frame was being sent.
SrcAddrMode	Integer	0x00–0x03	The source addressing mode for this primitive. This value can take one of the following values: 0 = no address (addressing fields omitted). 0x01 = no address field (broadcast only mode with no address fields present). 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcAddr	Device address	As specified by the SrcAddrMode parameter	The individual device address of the entity from which the frame causing the error originated.
DstAddrMode	Integer	0x00–0x03	The destination addressing mode for this primitive. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the device for which the frame was intended.
status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, IMPROPER_KEY_TYPE, IMPROPER_SECURITY_LEVEL, SECURITY_ERROR, UNAVAILABLE_KEY, UNSUPPORTED_LEGACY, UNSUPPORTED_SECURITY, INVALID_PARAMETER	The communications status.

Table 50—MLME-COMM-STATUS.indication parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	If the primitive was generated following a transmission instigated through a response primitive: The security level to be used (as defined in Table 64 in 7.4.2.1). If the primitive was generated on receipt of a frame that generates an error in its security processing: The security level purportedly used by the received frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was generated following a transmission instigated through a response primitive: The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was generated following a transmission instigated through a response primitive: The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

Table 50—MLME-COMM-STATUS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeyIndex	Integer	0x01–0xff	If the primitive was generated following a transmission instigated through a response primitive: The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00. If the primitive was generated on receipt of a frame that generates an error in its security processing: The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.9.1.1 When generated

The MLME-COMM-STATUS.indication primitive is generated by the MLME and issued to its next higher layer either following a transmission instigated through a response primitive or on receipt of a frame that generates an error in its security processing (see 7.2.3).

The MLME-COMM-STATUS.indication primitive is generated by the MAC sublayer entity following the MLME-ASSOCIATE.response primitive. This primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, an error code of TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK, or INVALID_PARAMETER (these status values are fully described in 6.3.1.3.2), or an error code resulting from failed security processing (these status values are fully described in 7.2.1 and 7.2.3).

6.3.9.1.2 Appropriate usage

On receipt of the MLME-COMM-STATUS.indication primitive, the next higher layer is notified of the communication status of a transmission or notified of an error that has occurred during the secure processing of incoming frame.

6.3.10 Primitives for writing PIB attributes

MLME-SAP set primitives define how PIB attributes may be written.

All devices shall provide an interface for these set primitives.

6.3.10.1 MLME-SET.request

The MLME-SET.request primitive attempts to write the given value to the indicated PIB attribute.

6.3.10.1.1 Semantics of the primitive

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


```
MLME-SET.request        (
                        PIBAttribute,
                        PIBAttributeIndex,
                        PIBAttributeValue
                        )
```

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

Table 51—MLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Integer	Refer to Table 60 and Table 66	The identifier of the PIB attribute to write.
PIBAttributeIndex	Integer	Attribute specific; as defined in Table 60 and Table 66	The index within the table of the specified PIB attribute to write. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.
PIBAttributeValue	Various	Attribute specific; as defined in Table 60 and Table 66	The value to write to the indicated PIB attribute.

6.3.10.1.2 Appropriate usage

The MLME-SET.request primitive is generated by the next higher layer and issued to its MLME to write the indicated PIB attribute.

6.3.10.1.3 Effect on receipt

On receipt of the MLME-SET.request primitive, the MLME checks to see if the PIB attribute is a MAC PIB attribute or PHY PIB attribute. If the requested attribute is a MAC attribute, the MLME attempts to write the given value to the indicated MAC PIB attribute in its database. If the PIBAttribute parameter specifies an attribute that is a read-only attribute, shown in Table 60, the MLME will issue the MLME-SET.confirm primitive with a status of READ_ONLY. If the PIBAttribute parameter specifies an attribute that is not found in the database, the MLME will issue the MLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttributeIndex parameter specifies an index for a table that is out of range, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_INDEX. If the PIBAttributeValue parameter specifies a value that is out of the valid range for the given attribute, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_PARAMETER. If the requested MAC PIB attribute is successfully written, the MLME will issue the MLME-SET.confirm primitive with a status of SUCCESS.

If the PIBAttribute parameter indicates that *macBeaconPayloadLength* is to be set and the length of the resulting beacon frame exceeds *aMaxPHYFrameSize* (e.g., due to the additional overhead required for security processing), the MAC sublayer shall not update *macBeaconPayloadLength* and will issue the MLME-GET.confirm primitive with a status of INVALID_PARAMETER.

If the requested attribute is a PHY PIB attribute, the request is passed to the PHY by issuing the PLME-SET.request primitive. Once the MLME receives the PLME-SET.confirm primitive, it will translate the received status value because the status values used by the PHY are not the same as those used by the

MLME (e.g., the status values for SUCCESS are 0x00 and 0x07 in the MAC and PHY enumeration tables, respectively). Following the translation, the MLME will issue the MLME-SET.confirm primitive to the next higher layer with the status parameter resulting from the translation and the PIBAttribute parameter equal to that returned by the PLME primitive.

6.3.10.2 MLME-SET.confirm

The MLME-SET.confirm primitive reports the results of an attempt to write a value to a PIB attribute.

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

```
MLME-SET.confirm        (
                        status,
                        PIBAttribute,
                        PIBAttributeIndex
                        )
```

Table 52 specifies the parameters for the MLME-SET.confirm primitive.

Table 52—MLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, READ_ONLY, UNSUPPORTED_ATTRIBUTE, INVALID_INDEX, INVALID_PARAMETER	The result of the request to write the PIB attribute.
PIBAttribute	Integer	Refer to Table 60 and Table 66	The identifier of the PIB attribute that was written.
PIBAttributeIndex	Integer	Attribute specific; as defined in Table 60 and Table 66	The index within the table of the specified PIB attribute to write. This parameter is valid only for MAC PIB attributes that are tables; it is ignored when accessing PHY PIB attributes.

6.3.10.2.1 When generated

The MLME-SET.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-SET.request primitive. The MLME-SET.confirm primitive returns a status of either SUCCESS, indicating that the requested value was written to the indicated PIB attribute, or the appropriate error code. The status values are fully described in 6.3.10.1.3.

6.3.10.2.2 Appropriate usage

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

6.3.11 Primitives for updating the superframe configuration

MLME-SAP start primitives define how a coordinator can request to start using a new superframe configuration in order to initiate a VPAN, begin transmitting beacons on an already existing VPAN, thus facilitating device discovery, or to stop transmitting beacons.

These start primitives are optional for a device.

6.3.11.1 MLME-START.request

The MLME-START.request primitive allows the VLC coordinator to initiate a new VPAN or to begin using a new superframe configuration. This primitive may also be used by a device already associated with an existing VPAN to begin using a new superframe configuration.

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

```
MLME-START.request          (
                            VPANId,
                            LogicalChannel,
                            StartTime,
                            BeaconOrder,
                            SuperframeOrder,
                            VPANCoordinator,
                            CoordRealignment,
                            CoordRealignSecurityLevel,
                            CoordRealignKeyIdMode,
                            CoordRealignKeySource,
                            CoordRealignKeyIndex,
                            BeaconSecurityLevel,
                            BeaconKeyIdMode,
                            BeaconKeySource,
                            BeaconKeyIndex
                            )
```

Table 53 specifies the parameters for the MLME-START.request primitive.

Table 53—MLME-START.request parameters

Name	Type	Valid range	Description
VPANId	Integer	0x0000–0xffff	The VPAN identifier to be used by the device.
LogicalChannel	Integer	Selected from the available logical channels	The logical channel on which to start using the new superframe configuration.

Table 53—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
StartTime	Integer	0x000000–0xffffff	The time at which to begin transmitting beacons. If this parameter is equal to 0x000000, beacon transmissions will begin immediately. Otherwise, the specified time is relative to the received beacon of the coordinator with which the device synchronizes. This parameter is ignored if either the BeaconOrder parameter has a value of 15 or the VPANCoordinator parameter is TRUE. The time is specified in optical clocks and is rounded to a backoff slot boundary. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest 4 bits being the least significant.
BeaconOrder	Integer	0–15	How often the beacon is to be transmitted. A value of 15 indicates that the coordinator will not transmit periodic beacons. Refer to 5.1.1.1 for an explanation of the relationship between the beacon order and the beacon interval.
SuperframeOrder	Integer	0–*BO* or 15	The length of the active portion of the superframe, including the beacon frame. If the BeaconOrder parameter (*BO*) has a value of 15, this parameter is ignored. Refer to 5.1.1.1 for an explanation of the relationship between the superframe order and the superframe duration.
VPANCoordinator	Boolean	TRUE or FALSE	If this value is TRUE, the device will become the coordinator of a new VPAN. If this value is FALSE, the device will begin using a new superframe configuration on the VPAN with which it is associated.
CoordRealignment	Boolean	TRUE or FALSE	TRUE if a coordinator realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.
CoordRealignSecurity-Level	Integer	0x00–0x07	The security level to be used for coordinator realignment command frames (as defined in Table 64 in 7.4.2.1).

Table 53—MLME-START.request parameters *(continued)*

Name	Type	Valid range	Description
CoordRealignKeyId-Mode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the CoordRealignSecurityLevel parameter is set to 0x00.
CoordRealignKey-Source	Set of 0, 4, or 8 octets	As specified by the CoordRealignKeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the CoordRealignKeyId-Mode parameter is ignored or set to 0x00.
CoordRealignKeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the CoordRealignKeyIdMode parameter is ignored or set to 0x00.
BeaconSecurityLevel	Integer	0x00–0x07	The security level to be used for beacon frames (as defined in Table 64 in 7.4.2.1).
BeaconKeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the BeaconSecurityLevel parameter is set to 0x00.
BeaconKeySource	Set of 0, 4, or 8 octets	As specified by the BeaconKeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.
BeaconKeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the BeaconKeyIdMode parameter is ignored or set to 0x00.

6.3.11.1.1 Appropriate usage

The MLME-START.request primitive is generated by the next higher layer and issued to its MLME to request that a device start using a new superframe configuration.

6.3.11.1.2 Effect on receipt

If the MLME-START.request primitive is received when *macShortAddress* is set to 0xffff, the MLME will issue the MLME-START.confirm primitive with a status of NO_SHORT_ADDRESS.

When the CoordRealignment parameter is set to TRUE, the coordinator attempts to transmit a coordinator realignment command frame as described in 5.1.3.3. If the transmission of the coordinator realignment command fails due to a channel access failure, the MLME will not make any changes to the superframe configuration (i.e., no PIB attributes will be changed) and will issue an MLME-START.confirm with a status of CHANNEL_ACCESS_FAILURE. If the coordinator realignment command is successfully transmitted, the MLME updates the appropriate PIB parameters with the values of the BeaconOrder, SuperframeOrder, VPANId, and LogicalChannel parameters, as described in 5.1.3.5, and will issue an MLME-START.confirm with a status of SUCCESS.

When the CoordRealignment parameter is set to FALSE, the MLME updates the appropriate PIB parameters with the values of the BeaconOrder, SuperframeOrder, VPANId, and LogicalChannel parameters, as described in 5.1.3.5.

The address used by the coordinator in its beacon frames is determined by the current value of *macShortAddress*, which is set by the next higher layer before issuing this primitive.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame, as described in 7.2.1. If the CoordRealignment parameter is set to TRUE, the CoordRealignSecurityLevel, CoordRealignKeyIdMode, CoordRealignKeySource, and CoordRealignKeyIndex parameters will be used to process the MAC command frame. If the BeaconOrder parameter indicates a beacon-enabled network, the BeaconSecurityLevel, BeaconKeyIdMode, BeaconKeySource, and BeaconKeyIndex parameters will be used to process the beacon frame. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-START.confirm primitive with the error status returned by outgoing frame processing.

If the length of the beacon frame exceeds *aMaxPHYFrameSize* (e.g., due to the additional overhead required for security processing), the MAC sublayer shall discard the beacon frame and issue the MLME-START.confirm primitive with a status of FRAME_TOO_LONG.

The MLME shall ignore the StartTime parameter if the BeaconOrder parameter is equal to 15 because this indicates a nonbeacon-enabled VPAN. If the BeaconOrder parameter is less than 15, the MLME examines the StartTime parameter to determine the time to begin transmitting beacons; the time is defined in optical clocks and is rounded to a backoff slot boundary. If the VLC coordinator parameter is set to TRUE, the MLME ignores the StartTime parameter and begins beacon transmissions immediately. Setting the StartTime parameter to 0x000000 also causes the MLME to begin beacon transmissions immediately. If the VPANCoordinator parameter is set to FALSE and the StartTime parameter is nonzero, the MLME calculates the beacon transmission time by adding StartTime optical clocks to the time, obtained from the local clock, when the MLME receives the beacon of the coordinator through which it is associated. If the time calculated causes the outgoing superframe to overlap the incoming superframe, the MLME shall not begin beacon transmissions. In this case, the MLME issues the MLME-START.confirm primitive with a status of SUPERFRAME_OVERLAP. Otherwise, the MLME then begins beacon transmissions when the current time, obtained from the local clock, equals the number of calculated optical clocks.

If the StartTime parameter is nonzero and the MLME is not currently tracking the beacon of the coordinator through which it is associated, the MLME will issue the MLME-START.confirm primitive with a status of TRACKING_OFF.

On completion of this procedure, the MLME responds with the MLME-START.confirm primitive. If the attempt to start using a new superframe configuration was successful, the status parameter will be set to SUCCESS. If any parameter is not supported or is out of range, the status parameter will be set to INVALID_PARAMETER.

6.3.11.2 MLME-START.confirm

The MLME-START.confirm primitive reports the results of the attempt to start using a new superframe configuration.

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

```
MLME-START.confirm      (
                        status
                        )
```


Table 54 specifies the parameters for the MLME-START.confirm primitive.

Table 54—MLME-START.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_SHORT_ADDRESS, SUPERFRAME_OVERLAP, TRACKING_OFF, INVALID_PARAMETER, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, CHANNEL_ACCESS_FAILURE	The result of the attempt to start using an updated superframe configuration.

6.3.11.2.1 When generated

The MLME-START.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-START.request primitive. The MLME-START.confirm primitive returns a status of either SUCCESS, indicating that the MAC sublayer has started using the new superframe configuration, or the appropriate error code. The status values are fully described in 6.3.11.1.2 and subclauses referenced by 6.3.11.1.2.

6.3.11.2.2 Appropriate usage

On receipt of the MLME-START.confirm primitive, the next higher layer is notified of the result of its request to start using a new superframe configuration. If the MAC sublayer has been successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.3.11.3 Message sequence chart for updating the superframe configuration

Figure 92 illustrates the sequence of messages necessary for initiating beacon transmissions as a coordinator. Figure 101 illustrates the sequence of messages necessary for the VLC coordinator to start beaconing on a new VPAN; this figure includes steps taken by the PHY.

6.3.12 Primitive for synchronizing with a coordinator

MLME-SAP synchronization primitives define how synchronization with a coordinator may be achieved and how a loss of synchronization is communicated to the next higher layer.

All devices shall provide an interface for the indication primitive. The request primitive is optional.

6.3.12.1 MLME-SYNC.request

The MLME-SYNC.request primitive requests to synchronize with the coordinator by acquiring and, if specified, tracking its beacons.

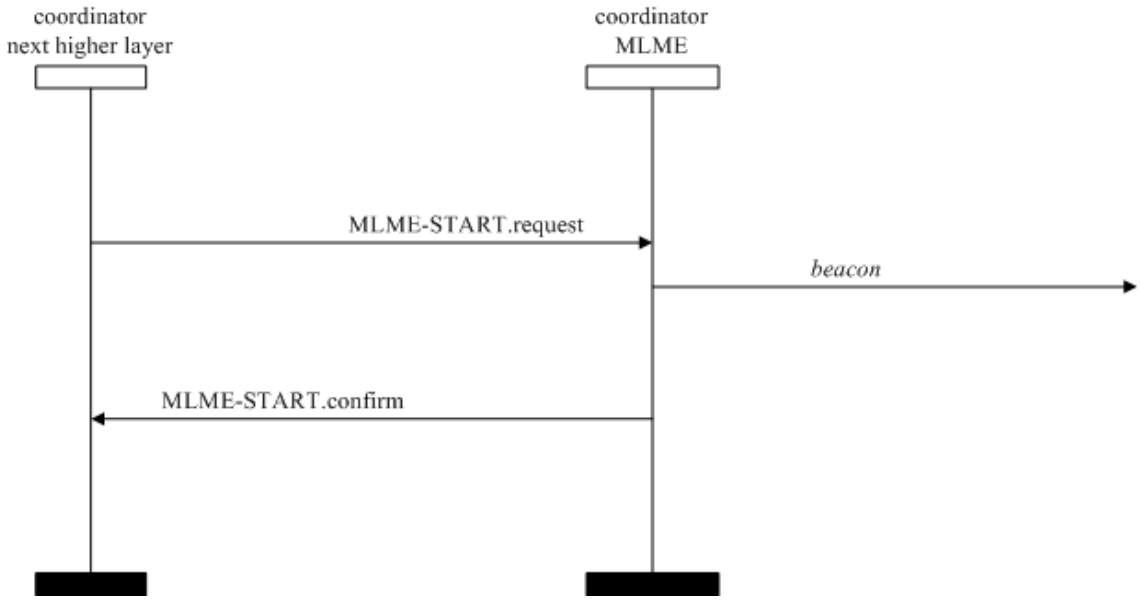


Figure 92—Message sequence chart for updating the superframe configuration

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

```
MLME-SYNC.request        (
                         LogicalChannel,
                         TrackBeacon
                         )
```

Table 55 specifies the parameters for the MLME-SYNC.request primitive.

Table 55—MLME-SYNC.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The logical channel on which to attempt coordinator synchronization.
TrackBeacon	Boolean	TRUE or FALSE	TRUE if the MLME is to synchronize with the next beacon and attempt to track all future beacons. FALSE if the MLME is to synchronize with only the next beacon.

6.3.12.1.1 Appropriate usage

The MLME-SYNC.request primitive is generated by the next higher layer of a device on a beacon-enabled VPAN and issued to its MLME to synchronize with the coordinator.

6.3.12.1.2 Effect on receipt

If the MLME-SYNC.request primitive is received by the MLME on a beacon-enabled VPAN, it will first set *phyCurrentChannel* equal to the values of the LogicalChannel parameters, respectively; both attributes are updated by issuing the PLME-SET.request primitive. If the TrackBeacon parameter is equal to TRUE, the MLME will track the beacon, i.e., enable its receiver just before the expected time of each beacon so that the beacon frame can be processed. If the TrackBeacon parameter is equal to FALSE, the MLME will locate the beacon, but not continue to track it.

If this primitive is received by the MLME while it is currently tracking the beacon, the MLME will not discard the primitive, but rather treat it as a new synchronization request.

If the beacon could not be located either on its initial search or during tracking, the MLME will issue the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

6.3.13 Primitive for synchronization loss with a coordinator

6.3.13.1 MLME-SYNC-LOSS.indication

The MLME-SYNC-LOSS.indication primitive indicates the loss of synchronization with a coordinator.

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

```
MLME-SYNC-LOSS.indication   (
                            LossReason,
                            VPANId,
                            LogicalChannel,
                            SecurityLevel,
                            KeyIdMode,
                            KeySource,
                            KeyIndex
                            )
```

Table 56 specifies the parameters for the MLME-SYNC-LOSS.indication primitive.

Table 56—MLME-SYNC-LOSS.indication parameters

Name	Type	Valid range	Description
LossReason	Enumeration	VPAN_ID_CONFLICT, REALIGNMENT, or BEACON_LOST	The reason that synchronization was lost.
VPANId	Integer	0x0000–0xffff	The VPAN identifier with which the device lost synchronization or to which it was realigned.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see Table 76).	The logical channel on which the device lost synchronization or to which it was realigned.

Table 56—MLME-SYNC-LOSS.indication parameters *(continued)*

Name	Type	Valid range	Description
SecurityLevel	Integer	0x00–0x07	If the primitive was either generated by the device itself following loss of synchronization or generated by the coordinator upon detection of a VPAN ID conflict, the security level is set to 0x00. If the primitive was generated following the reception of either a coordinator realignment command or a VPAN ID conflict notification command: The security level purportedly used by the received MAC frame (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	If the primitive was either generated by the device itself following loss of synchronization or generated by the coordinator upon detection of a VPAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a VPAN ID conflict notification command: The mode used to identify the key purportedly used by the originator of the received frame (as defined in Table 65 in 7.4.2.2). This parameter is invalid if the SecurityLevel parameter is set to 0x00.

Table 56—MLME-SYNC-LOSS.indication parameters *(continued)*

Name	Type	Valid range	Description
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	If the primitive was either generated by the device itself following loss of synchronization or generated by the coordinator upon detection of a VPAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a VPAN ID conflict notification command: The originator of the key purportedly used by the originator of the received frame (see 7.4.4.1). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.
KeyIndex	Integer	0x01–0xff	If the primitive was either generated by the device itself following loss of synchronization or generated by the coordinator upon detection of a VPAN ID conflict, this parameter is ignored. If the primitive was generated following the reception of either a coordinator realignment command or a VPAN ID conflict notification command: The index of the key purportedly used by the originator of the received frame (see 7.4.4.2). This parameter is invalid if the KeyIdMode parameter is invalid or set to 0x00.

6.3.13.2 Message sequence chart for synchronizing with a coordinator

Figure 93 illustrates the sequence of messages necessary for a device to synchronize with a coordinator. In Figure 93a), a single synchronization request is issued. The MLME then searches for a beacon and, if found, determines whether the coordinator has any data pending for the device. If so, the data are requested as described in 5.1.7.3. In Figure 93b), a track synchronization request is issued. The MLME then searches for a beacon and, if found, attempts to keep track of it using a timer that expires just before the expected time of the next beacon.

For both examples Figure 93a) and Figure 93b), the received beacon frames do not contain payload, and *macAutoRequest* is set to TRUE. The MLME also checks for any data pending in the coordinator for the device when a beacon frame is received.

6.3.13.2.1 When generated

The MLME-SYNC-LOSS.indication primitive is generated by the MLME of a device and issued to its next higher layer in the event of a loss of synchronization with the coordinator. It is also generated by the MLME of the VLC coordinator and issued to its next higher layer in the event of a VPAN ID conflict.

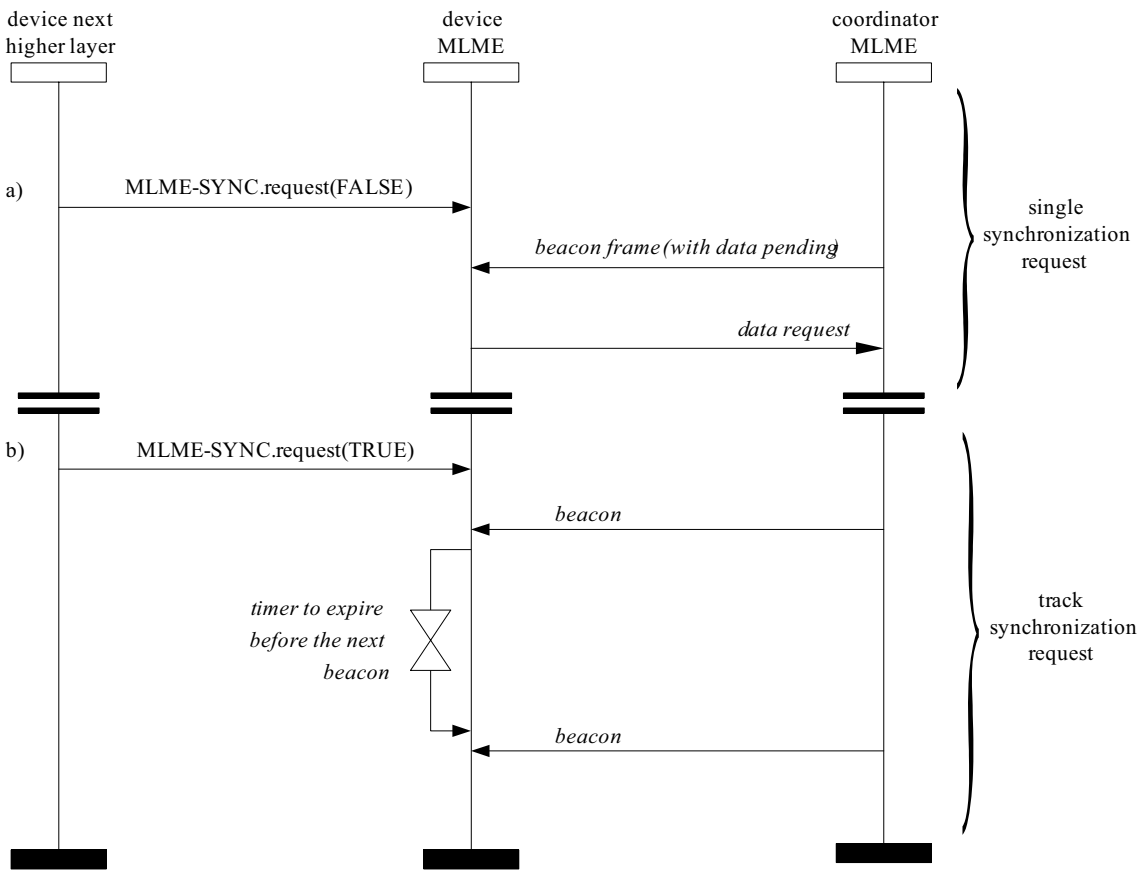


Figure 93—Message sequence chart for synchronizing to a coordinator in a beacon-enabled VPAN

If a device that is associated through the VLC coordinator has detected a VPAN identifier conflict and communicated it to the VLC coordinator, the MLME will issue this primitive with the LossReason parameter set to VPAN_ID_CONFLICT. Similarly, if the VLC coordinator receives a VPAN ID conflict notification command, as specified in 5.3.5, the MLME will issue this primitive with the LossReason parameter set to VPAN_ID_CONFLICT.

If a device has received the coordinator realignment command, specified in 5.3.7, from the coordinator through which it is associated, the MLME will issue this primitive with the LossReason parameter set to REALIGNMENT and the VPANId, LogicalChannel, and security-related parameters set as described in 5.1.3.4.

If a device has not heard the beacon for *aMaxLostBeacons* consecutive superframes following an MLME-SYNC.request primitive, either initially or during tracking, the MLME will issue this primitive with the LossReason parameter set to BEACON_LOST. The VPANId, LogicalChannel parameters shall be set according to the coordinator with which synchronization was lost. The SecurityLevel parameter shall be set to zero and the KeyIdMode, KeySource, and KeyIndex parameters shall be ignored. If the beacon was being tracked, the MLME will not attempt to track the beacon any further.

6.3.13.2.2 Appropriate usage

On receipt of the MLME-SYNC-LOSS.indication primitive, the next higher layer is notified of a loss of synchronization.

6.3.14 Primitives for requesting data from a coordinator

MLME-SAP polling primitives define how to request data from a coordinator.

All devices shall provide an interface for these polling primitives.

6.3.14.1 MLME-POLL.request

The MLME-POLL.request primitive prompts the device to request data from the coordinator.

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

```
MLME-POLL.request       (
                        CoordAddrMode,
                        CoordVPANId,
                        CoordAddress,
                        SecurityLevel,
                        KeyIdMode,
                        KeySource,
                        KeyIndex
                        )
```

Table 57 specifies the parameter for the MLME-POLL.request primitive.

Table 57—MLME-POLL.request parameters

Name	Type	Valid range	Description
CoordAddrMode	Integer	0x02–0x03	The addressing mode of the coordinator to which the poll is intended. This parameter can take one of the following values: 2 = 16-bit short address, 3 = 64-bit extended address.
CoordVPANId	Integer	0x0000–0xfffe	The VPAN identifier of the coordinator to which the poll is intended.
CoordAddress	Device-Address	As specified by the CoordAddrMode parameter	The address of the coordinator to which the poll is intended.
SecurityLevel	Integer	0x00–0x07	The security level to be used (as defined in Table 64 in 7.4.2.1).
KeyIdMode	Integer	0x00–0x03	The mode used to identify the key to be used (as defined in Table 65 in 7.4.2.2). This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of 0, 4, or 8 octets	As specified by the KeyIdMode parameter	The originator of the key to be used (see 7.4.4.1). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.
KeyIndex	Integer	0x01–0xff	The index of the key to be used (see 7.4.4.2). This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00.

6.3.14.1.1 Appropriate usage

The MLME-POLL.request primitive is generated by the next higher layer and issued to its MLME when data are to be requested from a coordinator.

6.3.14.1.2 Effect on receipt

On receipt of the MLME-POLL.request primitive, the MLME generates and sends a data request command, as specified in 5.3.4. If the poll is directed to the coordinator, the data request command may be generated without any destination address information present. Otherwise, the data request command is always generated with the destination address information in the CoordVPANId and CoordAddress parameters.

If the SecurityLevel parameter is set to a valid value other than 0x00, indicating that security is required for this frame, the MLME will set the Security Enabled subfield of the frame control field to one. The MAC sublayer will perform outgoing processing on the frame based on the CoordAddress, SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters, as described in 7.2.1. If any error occurs during outgoing frame processing, the MLME will discard the frame and issue the MLME-POLL.confirm primitive with the error status returned by outgoing frame processing.

If the data request command cannot be sent due to an unslotted random access algorithm failure, the MLME will issue the MLME-POLL.confirm primitive with a status of CHANNEL_ACCESS_FAILURE.

If the MLME successfully transmits a data request command, the MLME will expect an acknowledgment in return. If an acknowledgment is not received, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_ACK (see 5.1.7.4).

If an acknowledgment is received, the MLME will request that the PHY enable its receiver if the frame pending subfield of the acknowledgment frame is set to one. If the frame pending subfield of the acknowledgment frame is set to zero, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

If a frame is received from the coordinator with a zero length payload or if the frame is a MAC command frame, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA. If a frame is received from the coordinator with nonzero length payload, the MLME will issue the MLME-POLL.confirm primitive with a status of SUCCESS. In this case, the actual data are indicated to the next higher layer using the MCPS-DATA.indication primitive as specified in 6.2.3.

If a frame is not received within *macMaxFrameTotalWaitTime* CAP optical clocks in a beacon-enabled VPAN, or optical clocks in a nonbeacon-enabled VPAN, even though the acknowledgment to the data request command has its frame pending subfield set to one, the MLME will issue the MLME-POLL.confirm primitive with a status of NO_DATA.

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

6.3.14.2 MLME-POLL.confirm

The MLME-POLL.confirm primitive reports the results of a request to poll the coordinator for data.

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

```
MLME-POLL.confirm (
status
)
```


Table 58 specifies the parameters for the MLME-POLL.confirm primitive.

Table 58—MLME-POLL.confirm parameters

Name	Type	Valid range	Description
status	Integer	SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY or INVALID_PARAMETER	The status of the data request.

6.3.14.2.1 When generated

The MLME-POLL.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-POLL.request primitive. If the request was successful, the status parameter will be equal to SUCCESS, indicating a successful poll for data. Otherwise, the status parameter indicates the appropriate error code. The status values are fully described in 6.3.14.1.2 and the subclauses referenced by 6.3.14.1.2.

6.3.14.2.2 Appropriate usage

On receipt of the MLME-POLL.confirm primitive, the next higher layer is notified of the status of the procedure to request data from the coordinator.

6.3.14.3 Message sequence chart for requesting data from a coordinator

Figure 94 illustrates the sequence of messages necessary, including the layer behavior of the device and the over-the-air interface, for a device to request data from a coordinator.

In both scenarios Figure 94a) and Figure 94b), a poll request is issued to the MLME, which then sends a data request command to the coordinator. In Figure 94a), the corresponding acknowledgment has the frame pending (FP) subfield set to zero and the MLME issues the poll request confirmation immediately. In Figure 94b), the corresponding acknowledgment has the frame pending subfield set to one and the MLME enables the receiver in anticipation of the data frame from the coordinator. On receipt of this data frame, the MLME issues a poll request confirmation followed by a data indication containing the data of the received frame.

6.4 MAC constants and PIB attributes

This subclause specifies the constants and attributes required by the MAC sublayer.

6.4.1 MAC constants

The constants that define the characteristics of the MAC sublayer are presented in Table 59.

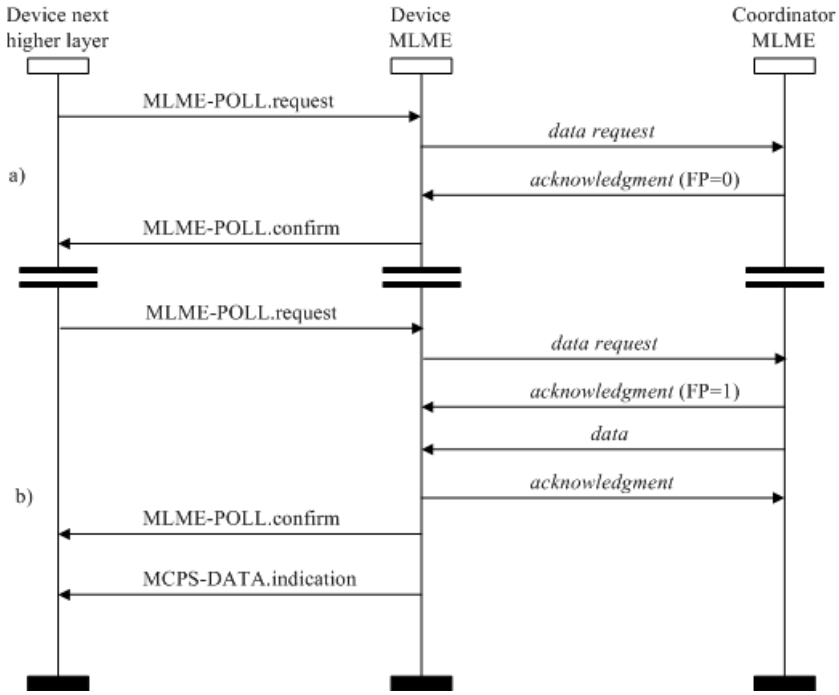


Figure 94—Message sequence chart for requesting data from a coordinator

Table 59—MAC sublayer constants

Constant	Description	Value
aBaseSlotDuration	The number of optical clocks forming a superframe slot when the superframe order is equal to 0 (see 5.1.1.1).	60
aBaseSuperframeDuration	The number of optical clocks forming a superframe when the superframe order is equal to 0.	*aBaseSlotDuration* × *aNumSuperframeSlots*
aExtendedAddress	The 64-bit (IEEE) address assigned to the device.	Device specific
aGTSDescPersistenceTime	The number of superframes in which a GTS descriptor exists in the beacon frame of the coordinator.	4
aMaxBeaconOverhead	The maximum number of octets added by the MAC sublayer to the MSDU of a beacon frame.	75
aMaxBeaconPayloadLength	The maximum size, in octets, of a beacon payload.	*aMaxPHYFrameSize* – *aMaxBeaconOverhead*
aMaxLostBeacons	The number of consecutive lost beacons that will cause the MAC sublayer of a receiving device to declare a loss of synchronization.	4
aMaxMACSafePayloadSize	The maximum number of octets that can be transmitted in the MSDU field of an unsecured MAC frame that will be guaranteed not to exceed *aMaxPHYFrameSize*.	*aMaxPHYFrameSize* – *aMaxMPDUUnsecuredOverhead*
aMaxMACPayloadSize	The maximum number of octets that can be transmitted in the MSDU field.	*aMaxPHYFrameSize* – *aMinMPDUOverhead*

Table 59—MAC sublayer constants *(continued)*

Constant	Description	Value
aMaxMPDUUnsecuredOverhead	The maximum number of octets added by the MAC sublayer to the PSDU without security.	25
aMaxSIFSFrameSize	The maximum size of an MPDU, in octets, that can be followed by a SIFS period.	18
aMinCAPLength	The minimum number of optical clocks forming the CAP. This ensures that MAC commands can still be transferred to devices when GTSs are being used. An exception to this minimum shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance (as defined in 5.2.2.1.3).	440
aMinMPDUOverhead	The minimum number of octets added by the MAC sublayer to the PSDU.	9
aNumSuperframeSlots	The number of slots contained in any superframe.	16
aUnitBackoffPeriod	The number of optical clocks forming the basic time period used by the unslotted random access algorithm.	20

6.4.2 MAC PIB attributes

The MAC PIB comprises the attributes required to manage the MAC sublayer of a device. The attributes contained in the MAC PIB are presented in Table 60 and Table 66. Attributes marked with a dagger (†) are read-only attributes (i.e., attribute can only be set by the MAC sublayer), which can be read by the next higher layer using the MLME-GET.request primitive. All other attributes can be read or written by the next higher layer using the MLME-GET.request or MLME-SET.request primitives, respectively. Higher layers may impose additional constraints on read/write operations, without making devices non-compliant. Attributes marked with a diamond (◆) are optional for a device (i.e., not operating as a coordinator).

The read-only attribute *macAckWaitDuration* is dependent on a combination of constants and PHY PIB attributes. The formula for relating the constants and attributes is shown in Equation (1).

$$AckWaitTime = backoff\ period + aTurnaroundTime\text{-}RX\text{-}TX + clock\ period \times numSymAckFrame \quad (1)$$

where numSymAckFrame is the number of bits in the acknowledgment frame and is equal to 103 for PHY I and II and 111 for PHY III. For B-ACK mode, the AckWaitTime would be larger, depending on the number of acknowledgments in the B-ACK mode as explained in 5.2.2.2. The clock period is obtained via the optical rates specified in Table 73, Table 74, and Table 75.

The attribute *macMaxFrameTotalWaitTime* may be set by the next higher layer and is dependent upon a combination of PHY and MAC PIB attributes and constants. The formula relating the attributes and constants is shown in Equation (2).

$$macMaxFrameTotalWaitTime = \quad (2)$$

$$\left[\left(\sum_{k=0}^{m-1} 2^{macMinBE+k}\right) + (2^{macMaxBE} - 1)\cdot(macMaxCSMABackoffs - m)\right] \bullet aUnitBackoffPeriod + phyMaxFrameDuration$$

where

m is min(*macMaxBE-macMinBE, macMaxCSMABackoffs)*

Table 60—MAC PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macAckWaitDuration†	0x40	Integer	Refer to Equation (1)	The maximum number of optical clocks to wait for an acknowledgment frame to arrive following a transmitted data frame. This value is dependent on the supported PHY, which determines both the selected logical channel. The calculated value is the time to commence transmitting the ACK plus the length of the ACK frame. The commencement time is described in 5.1.7.4.2.	Dependent on currently selected PHY.
macAssociatedVPAN-Coord	0x41	Boolean	TRUE or FALSE	Indication of whether the device is associated to the VPAN through the coordinator. A value of TRUE indicates the device has associated through the coordinator. Otherwise, the value is set to FALSE.	FALSE
macAssociation-Permit◆	0x42	Boolean	TRUE or FALSE	Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.	FALSE
macAutoRequest	0x43	Boolean	TRUE or FALSE	Indication of whether a device automatically sends a data request command if its address is listed in the beacon frame. A value of TRUE indicates that the data request command is automatically sent. This attribute also affects the generation of the MLME-BEACON-NOTIFY.indication primitive (see 6.3.3.1.1).	TRUE
macBeaconPayload◆	0x44	Set of octets	—	The contents of the beacon payload.	NULL
macBeaconPayload-Length◆	0x45	Integer	0 – *aMax-Beacon-PayloadLength*	The length, in octets, of the beacon payload.	0

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macBeaconOrder◆	0x46	Integer	0–15	Specification of how often the coordinator transmits its beacon. If *BO* = 15, the coordinator will not transmit a periodic beacon. Refer to 5.1.1.1 for an explanation of the relationship between the beacon order and the beacon interval.	15
macBeaconTxTime†◆	0x47	Integer	0x000000–0xffffff	The time that the device transmitted its last beacon frame, in symbol periods. The measurement shall be taken at the same symbol boundary within every transmitted beacon frame, the location of which is implementation specific. This is a 24-bit value, and the precision of this value shall be a minimum of 20 bits, with the lowest four bits being the least significant.	0x000000
macBSN◆	0x48	Integer	0x00–0xff	The sequence number added to the transmitted beacon frame.	Random value from within the range
macCoordExtended-Address	0x49	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the coordinator through which the device is associated.	—
macCoordShort-Address	0x4a	Integer	0x0000–0xffff	The 16-bit short address assigned to the coordinator through which the device is associated. A value of 0xfffe indicates that the coordinator is only using its 64-bit extended address. A value of 0xffff indicates that this value is unknown.	0xffff
macDSN	0x4b	Integer	0x00–0xff	The sequence number added to the transmitted data or MAC command frame.	Random value from within the range
macGTSPermit	0x4c	Boolean	TRUE or FALSE	TRUE if the coordinator is to accept GTS requests. FALSE otherwise.	TRUE
macMaxBE	0x4d	Integer	3–15	The maximum value of the backoff exponent, BE, in the unslotted random access algorithm. Refer to 5.1.1.3 for a detailed explanation of the backoff exponent.	5

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macMaxCSMABackoffs	0x4e	Integer	0–5	The maximum number of backoffs the unslotted random access algorithm will attempt before declaring a channel access failure.	4
macMaxFrameTotal-WaitTime	0x4f	Integer	Refer to Equation (2)	The maximum number of optical clocks in a beacon-enabled VPAN, or in a nonbeacon-enabled VPAN, to wait either for a frame intended as a response to a data request frame or for a broadcast frame following a beacon with the frame pending subfield set to one. This attribute, which shall only be set by the next higher layer, is dependent upon *macMinBE*, *macMaxBE*, *macMaxCSMABackoffs* and the number of optical clocks per octet. Refer to 6.4.2 for the formula relating the attributes.	Dependent on currently selected PHY
macMaxFrameRetries	0x50	Integer	0–7	The maximum number of retries allowed after a transmission failure.	3
macMinBE	0x51	Integer	0–*macMaxBE*	The minimum value of the backoff exponent (BE) in the unslotted random access algorithm. Refer to 5.1.1.3 for a detailed explanation of the backoff exponent.	3
macMinLIFSPeriod†	0x52	Integer	As defined in Table 77 in 8.3.4	The minimum number of optical clocks forming a LIFS period.	Dependent on currently selected PHY.
macMinSIFSPeriod†	0x53	Integer	As defined in Table 77 in 8.3.4	The minimum number of optical clocks forming a SIFS period.	Dependent on currently selected PHY.
macVPANId	0x54	Integer	0x0000–0xffff	The 16-bit identifier of the VPAN on which the device is operating. If this value is 0xffff, the device is not associated.	0xffff

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macResponseWaitTime	0x55	Integer	2–64	The maximum time, in multiples of *aBaseSuperframeDuration*, a device shall wait for a response command frame to be available following a request command frame.	32
macRxOnWhenIdle	0x56	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer is to enable its receiver during idle periods. For a beacon-enabled VPAN, this attribute is relevant only during the CAP of the incoming superframe. For a nonbeacon-enabled VPAN, this attribute is relevant at all times.	FALSE
macSecurityEnabled	0x57	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer has security enabled. A value of TRUE indicates that security is enabled, while a value of FALSE indicates that security is disabled.	TRUE
macShortAddress	0x58	Integer	0x0000–0xffff	The 16-bit address that the device uses to communicate in the VPAN. If the device is the coordinator, this value shall be chosen before a VPAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfffe indicates that the device has associated but has not been allocated an address. A value of 0xffff indicates that the device does not have a short address.	0xffff
macSuperframe-Order†◆	0x59	Integer	0–15	The length of the active portion of the outgoing superframe, including the beacon frame. If superframe order, *SO*, = 15, the superframe will not be active following the beacon. Refer to 5.1.1.1 for an explanation of the relationship between the superframe order and the superframe duration.	15

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macTimestamp-Supported†	0x5a	Boolean	TRUE or FALSE	Indication of whether the MAC sublayer supports the optional time stamping feature for incoming and outgoing data frames.	Imple-mentation specific
macTransaction-PersistenceTime◆	0x5b	Integer	0x0000–0xffff	The maximum time (in unit periods) that a transaction is stored by a coordinator and indicated in its beacon. The unit period is governed by *macBeaconOrder*, *BO*, as follows: For $0 \leq BO \leq 14$, the unit period will be *aBase-SuperframeDuration* * 2^{BO}. For *BO = 15*, the unit period will be *aBaseSuperframe-Duration*.	0x01f4
macDim	0x5c	Integer	0–1000	Percentage dimming; 0 is 0% visibility and 1000 is 100% visibility.	0
macNumAcks	0x5d	Integer	0–15	Maximum number of times not receiving ACKs to trigger fast link recovery procedure.	3
macLinkTimeOut	0x5e	Integer	0–255	A timer initiated when the link recovery procedure is triggered. If the timer expires while the device has not received any fast link recovery response (FLR RSP) signal since the fast link recovery procedure is triggered, the device assumes that the link is broken and cannot be recovered. The range for *macLinkTimeOut* is defined in terms of the number of superframes.	63
macDimOverrideRequest	0x5f	Boolean	TRUE or FALSE	shall be set to ‘1’ after VLC device association and shall be set to ‘0’ after the VLC device disassociation	0
macDimPWMOverrideRequest	0x60	Boolean	TRUE or FALSE	shall be set to ‘1’ to inform the dimmer circuit that the VLC device will be responsible for dimming and to disable any PWM circuit present in the dimmer	0
macDimDataFailureIndication	0x61	Boolean	TRUE or FALSE	shall be set to ‘1’ when the device is unable to perform data communication under dimming	0

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macDuringASSOCColor	0x62	Unsigned	0–255	Use *macDuringASSOCColor* for the color assignment of the CVD frame when the color function for the association MAC state indication between MLME-ASSOCIATE.request and MLME-ASSOCIATE.confirm is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0
macDuringDISASSOCColor	0x63	Unsigned	0–255	Use *macDuringDISASSOCColor* for the color assignment of the CVD frame when the color function for the disassociation MAC state indication between MLME-DISASSOCIATE.request and MLME-DISASSOCIATE.confirm is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0
macDuringSCANColor	0x64	Unsigned	0–255	Use *macDuringSCANColor* for the color assignment of the CVD frame when the color function for the scan MAC state indication between MLME-SCAN.request and MLME-SCAN.confirm is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macColorReceived	0x65	Unsigned	0–255	Use *macColorReceived* for the color assignment oft he CVD Frame when the ACK frame is sent and the color function for the ACK state indication is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, phyColorFunction, as shown in Table 100, PHY PIB attributes.	0
macColorNotReceived	0x66	Unsigned	0–255	Use *macColorNotReceived* for the color assignment of the CVD Frame when the ACK frame is not sent but the color function for the non-ACK state indication is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, phyColorFunction, as shown in Table 100, PHY PIB attributes.	0
macCQIColorLFER	0x67	Unsigned	0–255	Use *macCQIColorLFER* for the color assignment of the CVD frame when the color function for the channel quality indication showing the low FER is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0
macCQIColorMFER	0x68	Unsigned	0–255	Use *macCQIColorMFER* for the color assignment of the CVD frame when the color function for the channel quality indication showing the medium FER is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macCQIColorHFER	0x69	Unsigned	0–255	Use *macCQIColorHFER* for the color assignment of the CVD frame when the color function for the channel quality indication showing the high FER is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0
macCFAppColor	0x6a	Unsigned	0–255	Use *macCFAppColor* for the color assignment of the CVD frame when the color function for the indication of application-dependent information is used by the CVD frame. The unsigned integer is the index for the look-up table for the color function table, *phyColorFunction*, as shown in Table 100, PHY PIB attributes.	0
macColorStabilization	0x6b	Binary Integer	00–11	The color stabilization action entailed when receiving CVD frames. The information for setting these two bits is found in Table 20.	0
macColorStabilizationTimer	0x6c	Integer	0x0–0xffffffff	Minimum time between two stabilization measurements (see 8.5.4) that are send back to the corresponding CSK Tx. The time is measured in multiples of aMaxPHYFrameSize frames for color stabilization.	0x00400000
macUseDimmedOOKmode	0x6d	Boolean	TRUE or FALSE	Shall be set to 1 when dimming is to be performed in the dimmed OOK mode in conjunction with OOK.	0
macTimeStampOffset	0x6e	Octet	0x00–0xff	The location of the time stamp after the end of the preamble in optical clocks.	0
macUseBlinkingNotification	0x6f	Boolean	TRUE or FALSE	Shall be set to 0 when blinking notification is to be performed.	1

Table 60—MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macBlinkingNotificationFrequency	0x70	Integer	0–10	The frequency of blinking notification 0: 0.25Hz 1: 0.5Hz 2: 0.75Hz 3: 1Hz 4: 1.25Hz 5: 1.5Hz 6: 1.75Hz 7: 2Hz 8: 2.25Hz 9: 2.5Hz 10: 2.75Hz	0

6.5 Optical-clock-rate selection

The standard supports multiple optical clock rates in order to accommodate a wide variety of optical sources and receivers. The standard also supports the use of asymmetric clock rates between transmitter and receiver since they constitute independent chains and may support different clock-rate ranges. The multiple clocks associated with each PHY type are respectively shown in Table 73, Table 74, and Table 75.

Support for the minimum clock rate for a given PHY type shall be mandatory for all TX and RX devices. All specified clock rates less than the maximum supported clock rate in a given device shall also be supported in that device. If a clock rate is supported, all data rates associated with that clock rate shall be supported. The preamble, headers, and payload in the PHY shall have the same clock rate. The header shall be sent at lowest data rate for the chosen clock rate. The payload can choose any data rate belonging to the chosen clock rate.

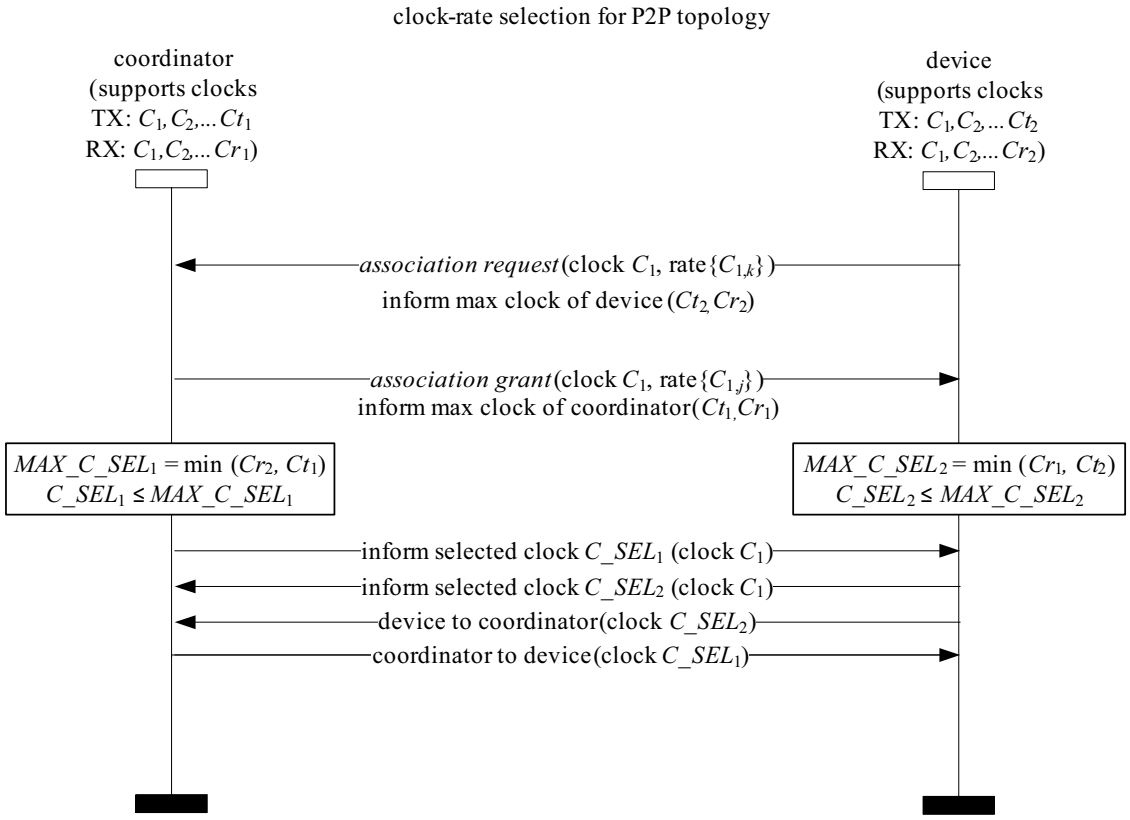
The clock-rate negotiation can be supported with or without explicit clock-rate negotiation, as indicated in the capabilities information field in Table 16. Explicit clock-rate negotiation implies that the devices shall transmit a clock rate change notification command as in 5.3.14 before a new clock rate is used. If explicit clock-rate negotiation is not used, the device shall have the capability to perform synchronization at all supported optical clock rates without any prior knowledge of the clock rate chosen at the transmitter for communication.

6.5.1 Optical-clock-rate selection for P2P topology

Let us assume that Device 1 supports clock rates at the transmitter (C_1, C_2, ... C_{t1}), where C_{t1} is the maximum clock rate supported at the transmitter at of the coordinator. Also, $C_1 < C_2 < ... < C_{t1}$. Within a PHY type, the clock rates are integral multiples of each other to make the clock generation and selection simple at the transmitter (i.e., $C_{i+1}/C_i = m$, which is an integer). The receiver may support more or less clock rates than the transmitter since the receiver optronics is physically independent of the transmitter clock. Let the clocks supported by the receiver of device 1 be C_1, C_2, ... C_{r1}, where C_{r1} is the maximum clock rate supported at the receiver of device 1. Similarly, let C_{t2} and C_{r2} be the maximum clock rates supported by the device 2. Support for the lowest clock rate C_1 is mandatory at both the transmitter and receiver for all devices i.e., t_1, t_2, r_1, $r_2 \geq 1$. For every clock rate, there is an associated set of data rates at the physical layer. This data rate is dependent on the modulation, RLL coding, and FEC used at the physical layer for a given clock rate. Let the data rate be represented by rate$\{C_{i,p}\}$, where C_i is the chosen clock rate and $1 \leq p \leq N(C_i)$, where $N(C_i)$ is the number of physical-layer data rates associated with clock rate C_i.

6.5.1.1 Explicit notification

In Figure 95, a device sends the association request at the lowest clock C_1 at a physical layer data rate rate$\{C_{1,k}\}$. The data rate index k is typically chosen to be the lowest data rate to guarantee maximum range and reliability for the given clock rate. In this association request, the device also informs the coordinator of the maximum clock rate supported by its transmitter and receiver (C_{t2}, C_{r2}). The maximum-clock-rate information is provided by the capabilities IE as shown in Table 16, which shall also be transmitted during this association request. The coordinator receives the association request and compares the received information about the supported clocks at the device and compares it with its supported clocks. In order for it to communicate, it shall select a clock rate C_SEL_1 that is equal to or lower than $MAX\_C\_SEL_1$, which is the minimum of its maximum transmitter clock and the maximum receiver clock supported by the coordinator. The decision to use clock rates lower than $MAX\_C\_SEL_1$ and $MAX\_C\_SEL_2$ at the coordinator and the device for, respectively, the transmission depends on the performance and throughput needs of the coordinator and the devices. The coordinator also sends an association grant back to the device at the same lowest clock rate C_1 supported by all devices. The devices then exchange the selected clock frequencies by using the clock-rate-change notification command for future communication before they switch to the selected clock frequencies. The devices may also decide to change the clock rate anytime in future communication, as long as it is below $MAX\_C\_SEL_1$ and $MAX\_C\_SEL_2$ for transmission at the coordinator and device, respectively.



There is a data rate table rate$\{C_{i,j}\}$ associated with each clock rate C_i where j is the data rate index associated with clock rate C_i

Figure 95—Clock-rate selection for P2P topology (explicit notification)

6.5.1.2 Without explicit notification

It is also possible for the coordinator to send the association grant at the new clock rate C_SEL_2 and not have to explicitly exchange notification information, as shown in Figure 96, if the coordinator has the capability to detect all clock rates less than its maximum receive clock rate. In this case, communication can occur without overhead for explicit notification.

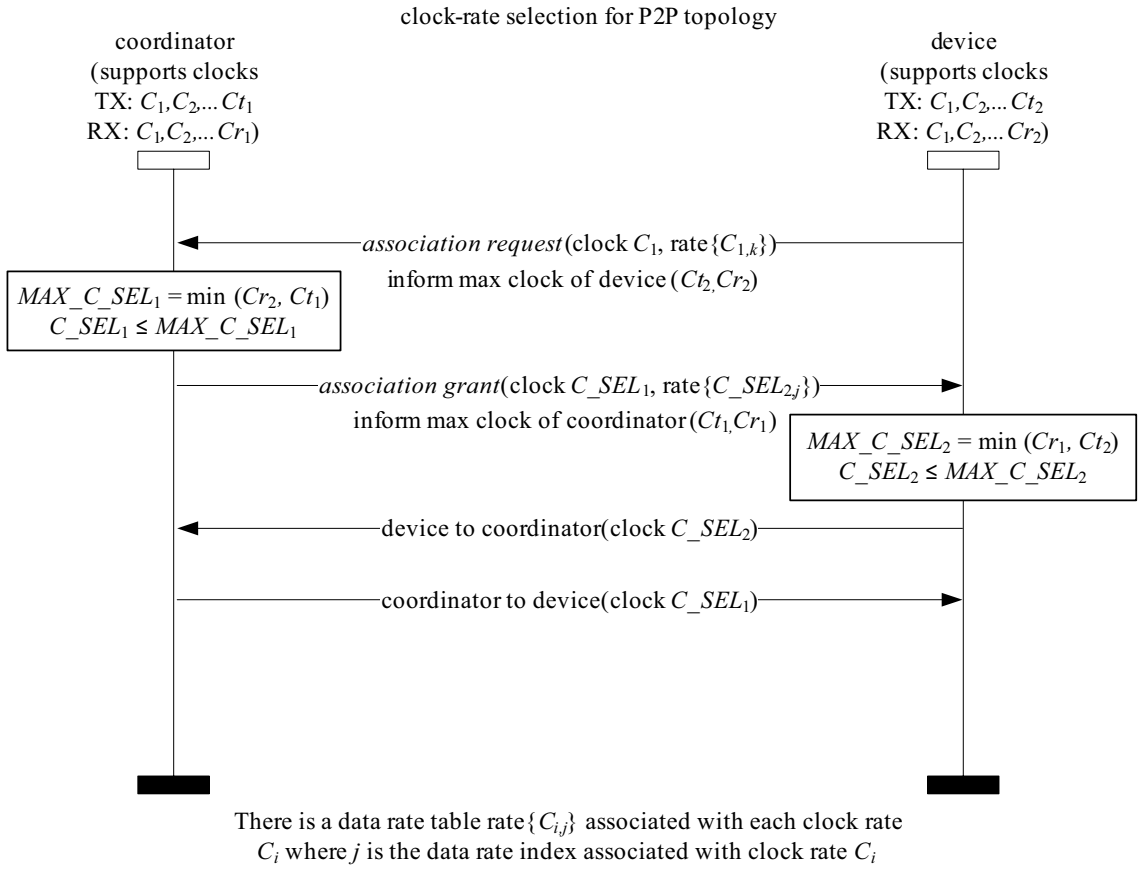


Figure 96—Clock-rate selection for P2P topology (without explicit notification)

6.5.2 Optical-clock-rate selection for star topology

6.5.2.1 Explicit notification

Figure 97 shows the optical clock-rate selection for a star topology. In this case, let us assume device 1 to be a coordinator. The coordinator will send a broadcast message via a beacon to all nodes, such as devices 2 and 3, and inform them of its supported clock rates. The CAP always uses the lowest clock rate C_1 for uplink contention. The coordinator and the devices communicate the selected clock frequencies during the CFP using clock rate C_1 before switching to the selected clock frequencies. The information about the coordinator capabilities is broadcast using the capabilities IE. The current clock in use, and any change of clock, is communicated via the clock rate change notification.

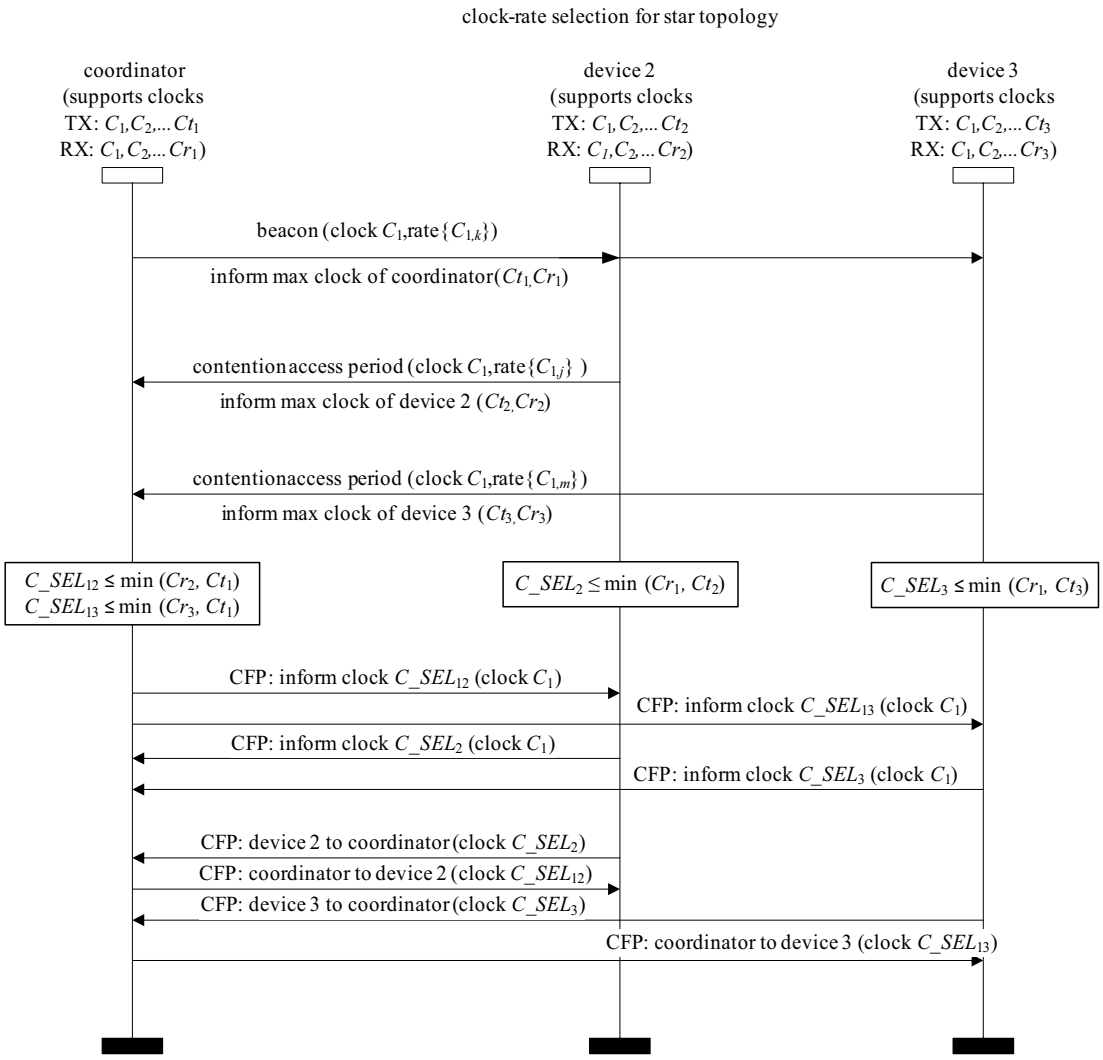


Figure 97—Clock-rate selection for star topology (explicit notification)

6.5.2.2 Without explicit notification

Similar to the P2P topology, the clock-rate selection for the star topology can also occur without explicit notification, as shown in Figure 98.

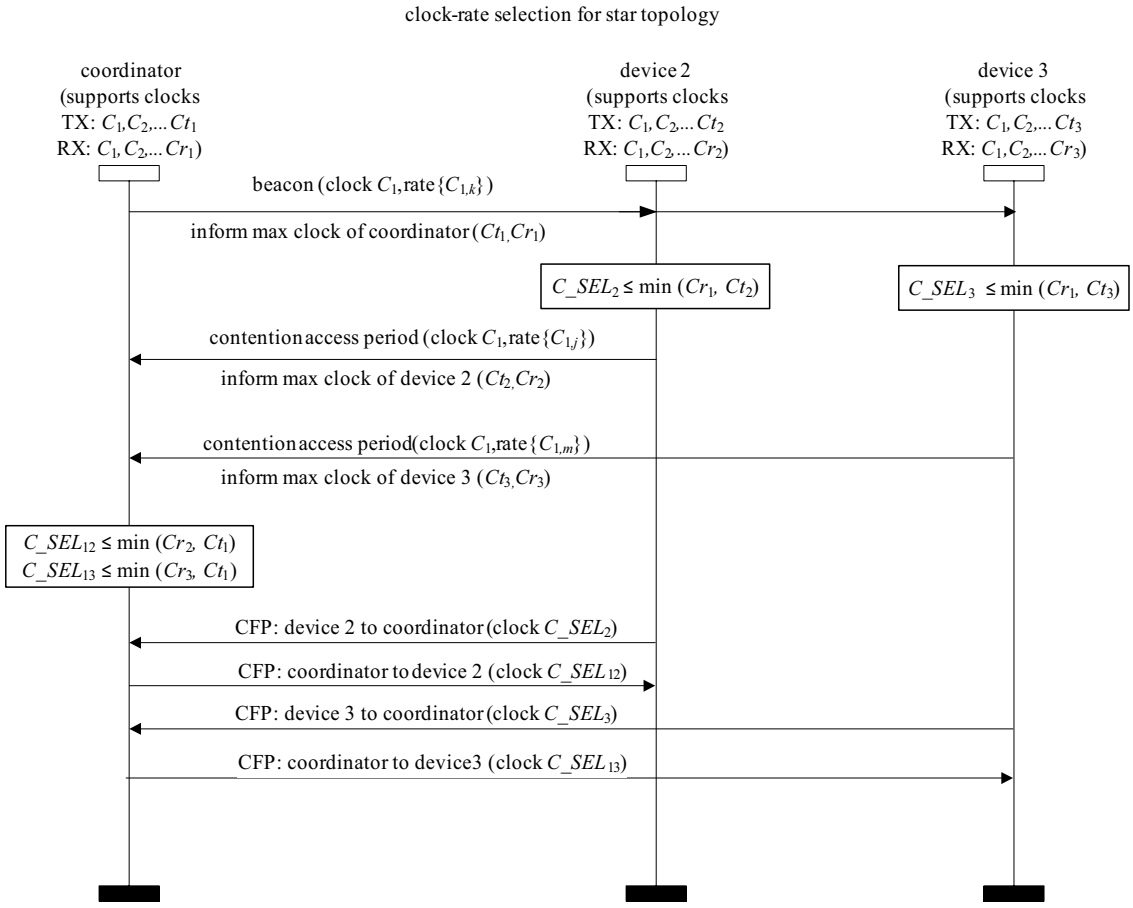


Figure 98—Clock-rate selection for star topology (without explicit notification)

6.5.3 Clock-rate selection for multicast topology

6.5.3.1 Explicit notification

Figure 99 and Figure 100 show the clock-rate selection for multicast topologies assuming bi-directional communication.

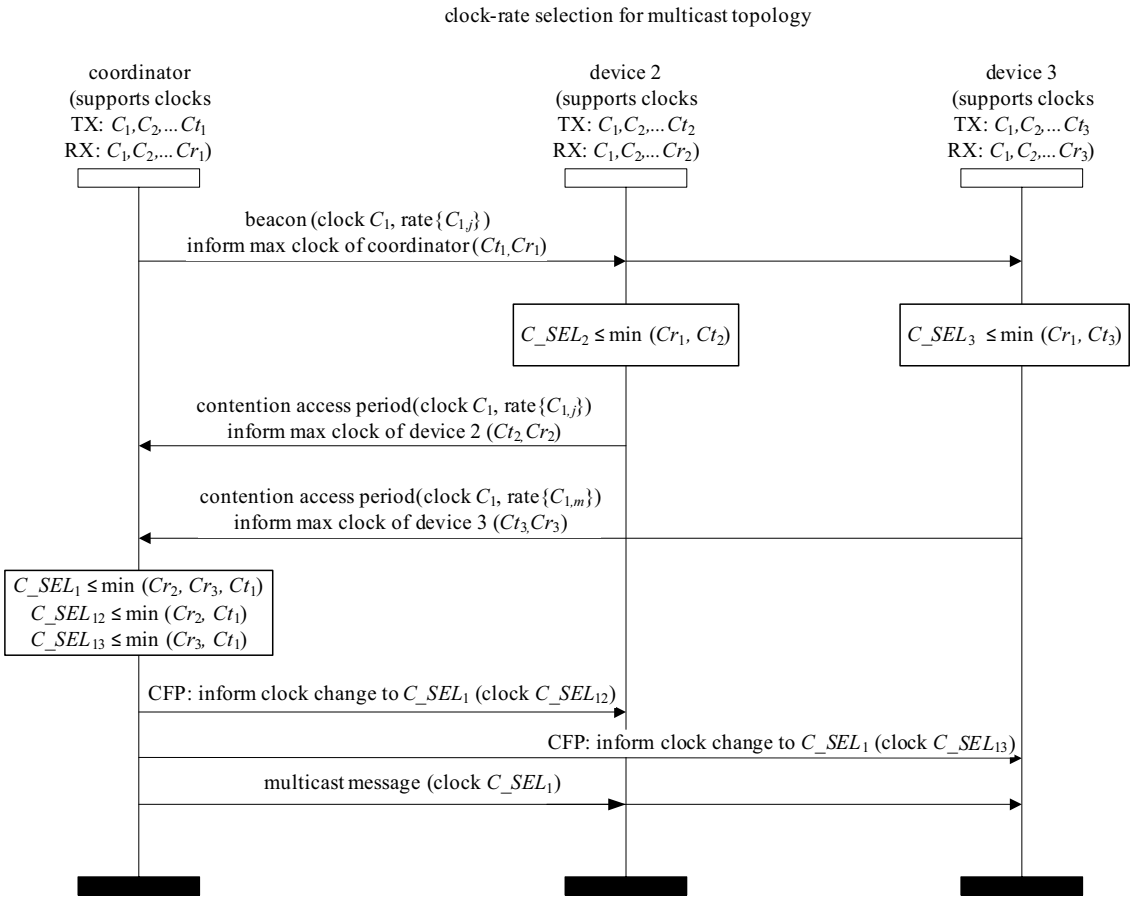


Figure 99—Clock-rate selection for multicast (assuming bi-directional communication)

6.5.3.2 Without explicit notification

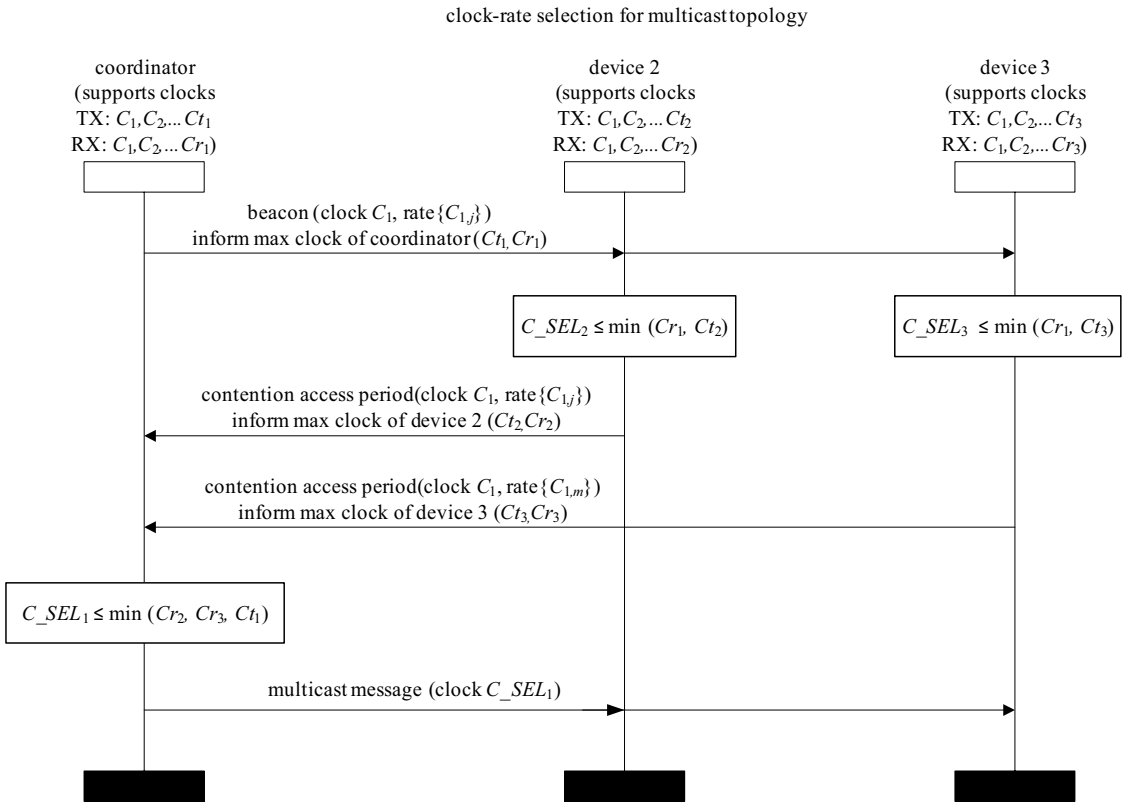


Figure 100—Clock-rate selection for multicast (bi-directional communication and no explicit notification)

6.6 Message sequence charts illustrating MAC-PHY interaction

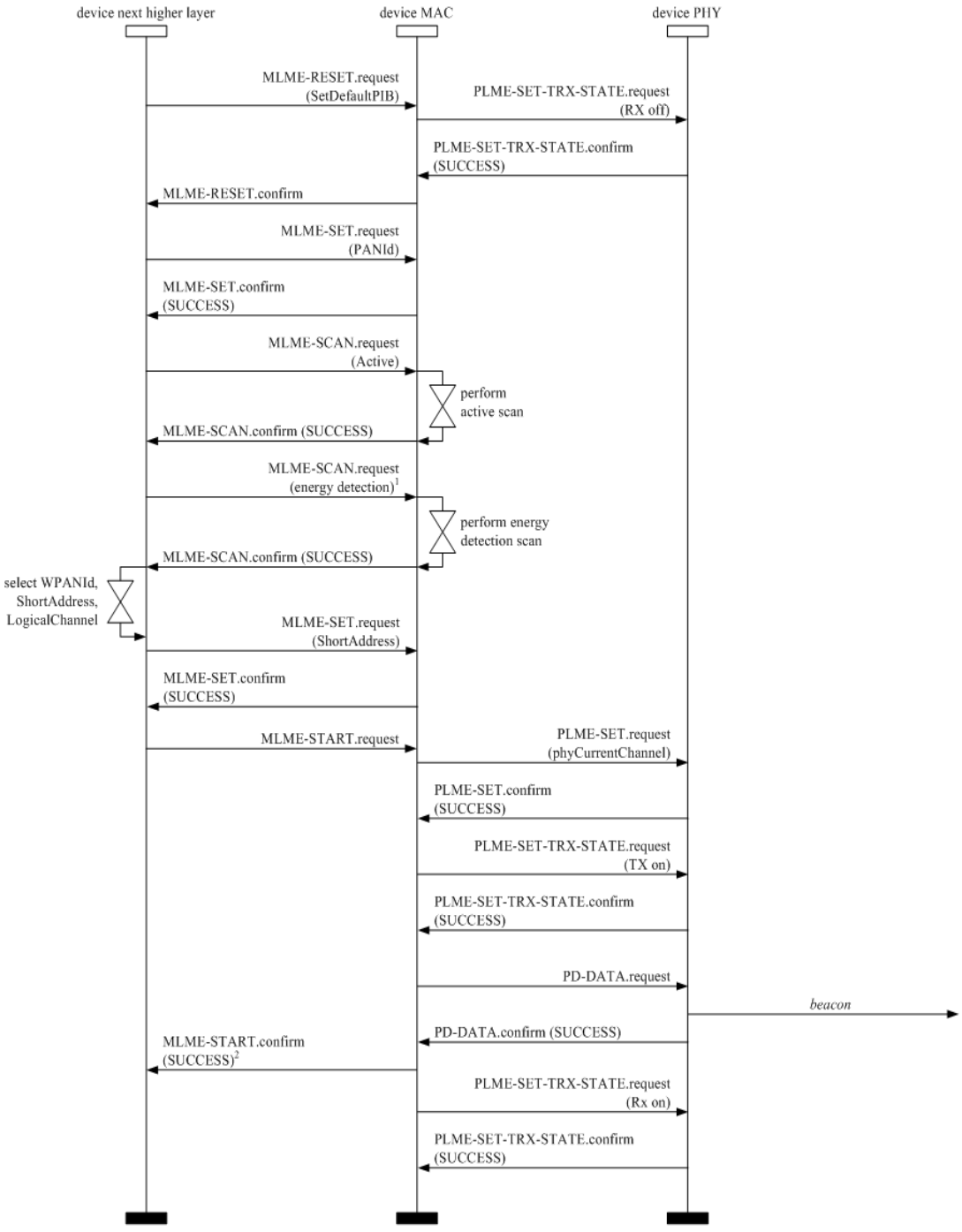
This subclause illustrates the main tasks specified in this standard. Each task is described by use of a message sequence chart to illustrate the chronological order, rather than the exact timing, of the primitives required for each task.

The primitives necessary for the coordinator to start a new VPAN are shown in Figure 101. The first action the next higher layer takes after resetting the MAC sublayer is to initiate a scan to search for other VPANs in the area. An active scan is required. The steps for performing an active scan are shown in Figure 105.

Once a new VPAN is established, the coordinator is ready to accept requests from other devices to join the VPAN. Figure 102 shows the primitives issued by a device requesting association, while Figure 103 illustrates the steps taken by a coordinator allowing association. In the process of joining a VPAN, the device requesting association will perform either a passive or an active scan to determine which VPANs in the area are allowing association; Figure 104 and Figure 105 detail the primitives necessary to complete a passive scan and an active scan, respectively.

The primitives necessary for transmitting and receiving a single data frame are shown next. The actions taken by the originator of the frame are shown in Figure 106, while the actions taken by the recipient are shown in Figure 107.

.



[1] The energy detection scan is optional.
[2] The MLME-START.confirm and PLME-SET-TRX-STATE.request primitives may not need to be in the order shown.

Figure 101—VPAN start message sequence chart—coordinator

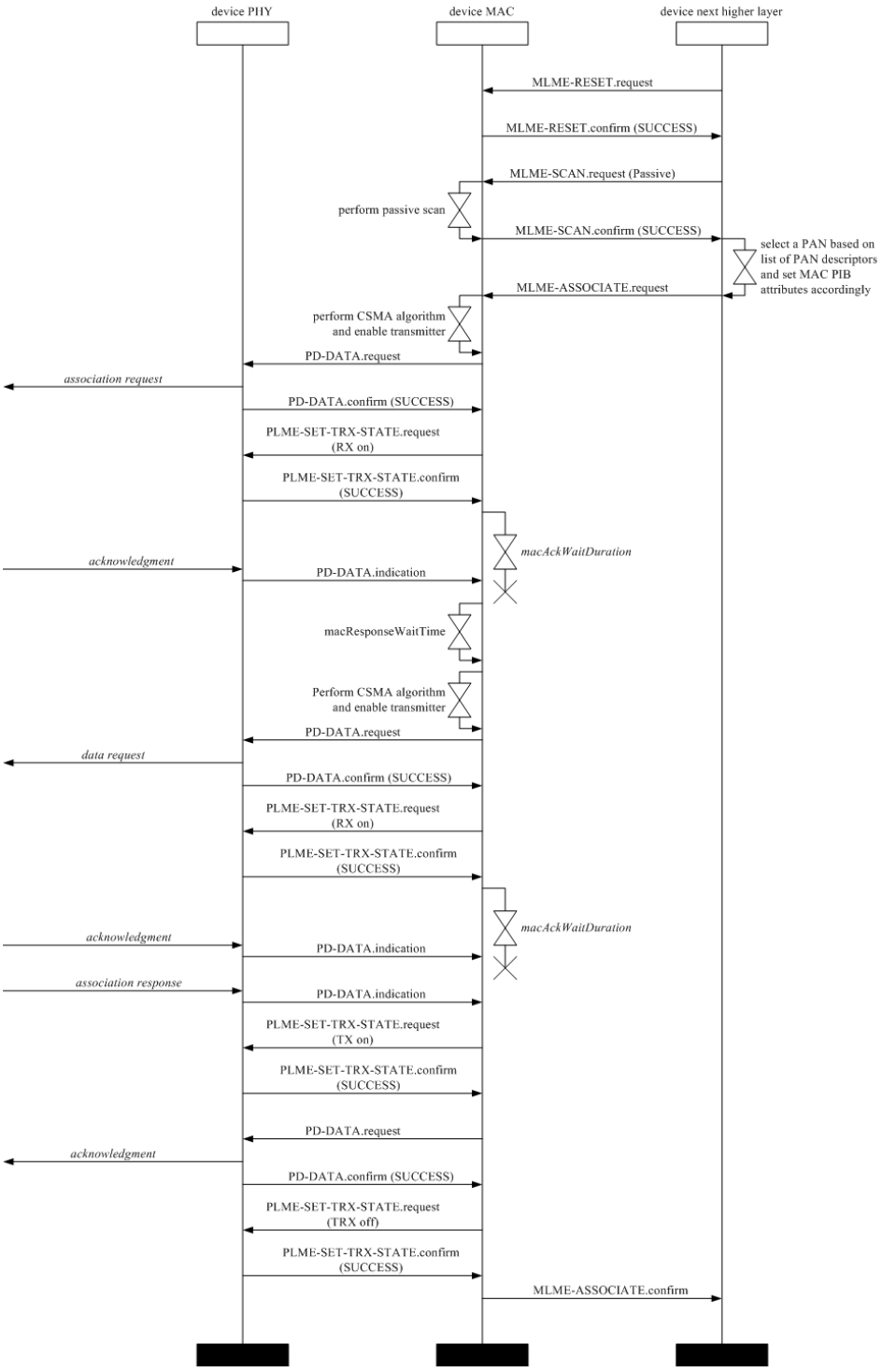


Figure 102—Association message sequence chart—device

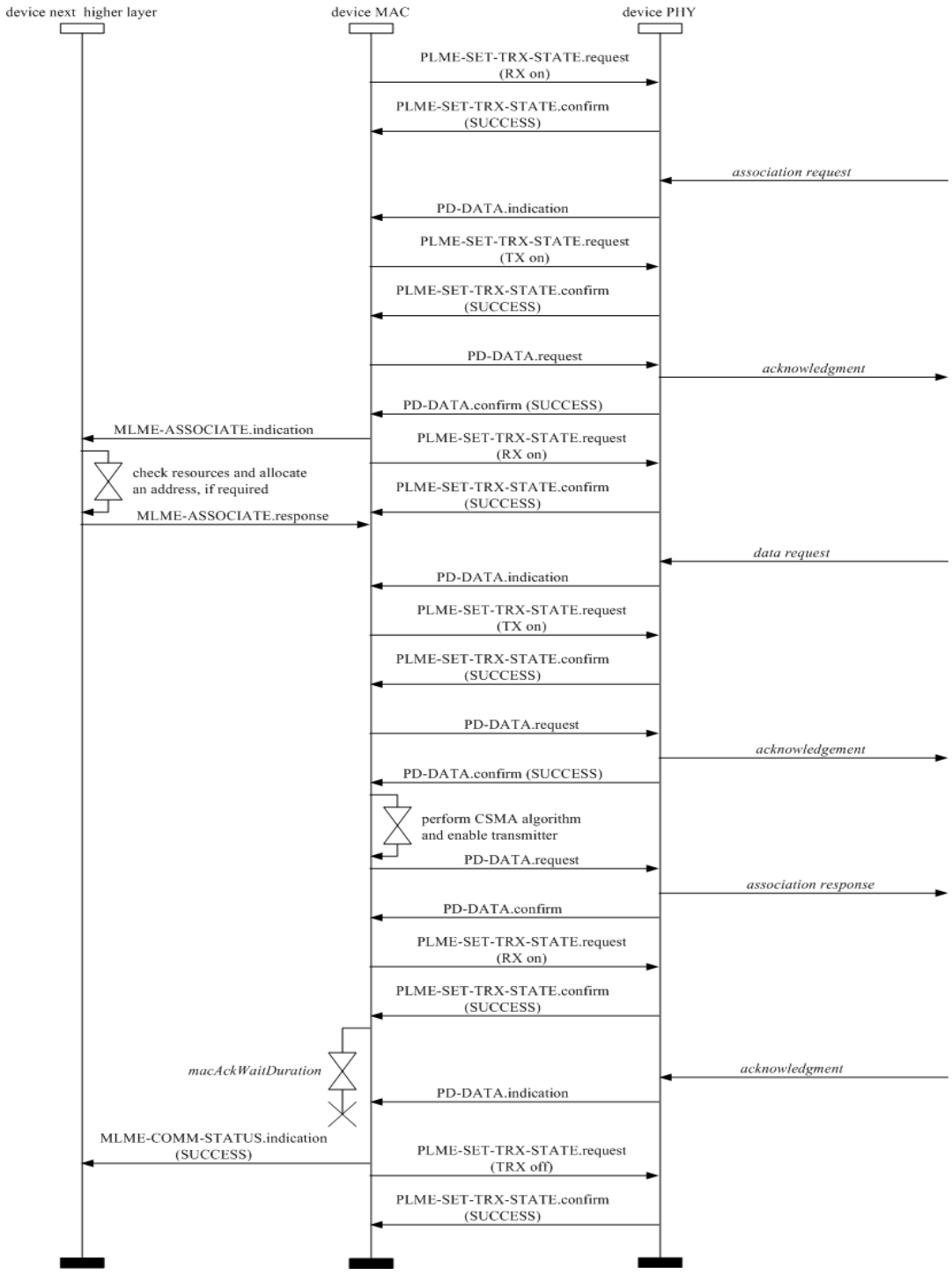


Figure 103—Association message sequence chart—coordinator

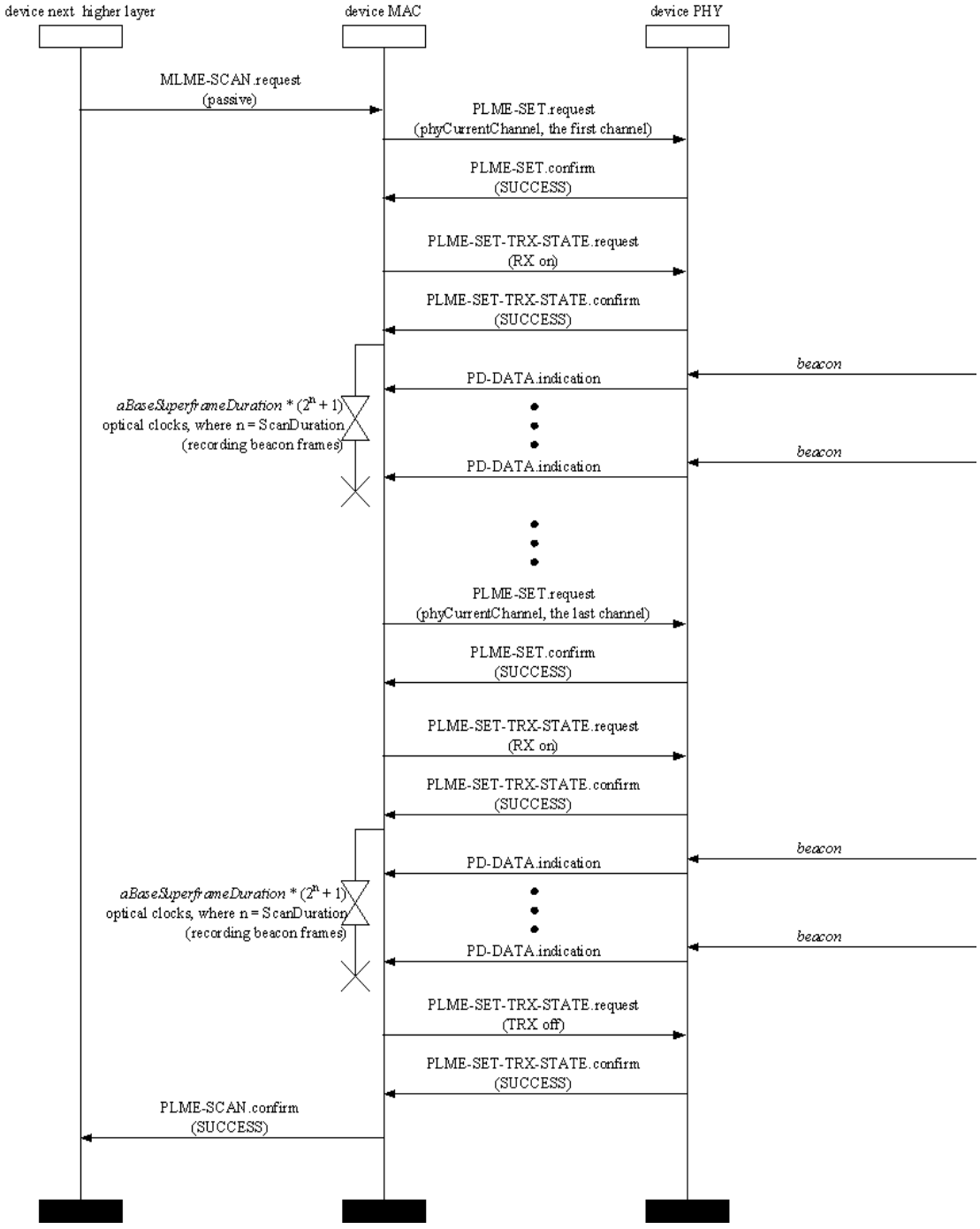


Figure 104—Passive scan message sequence chart

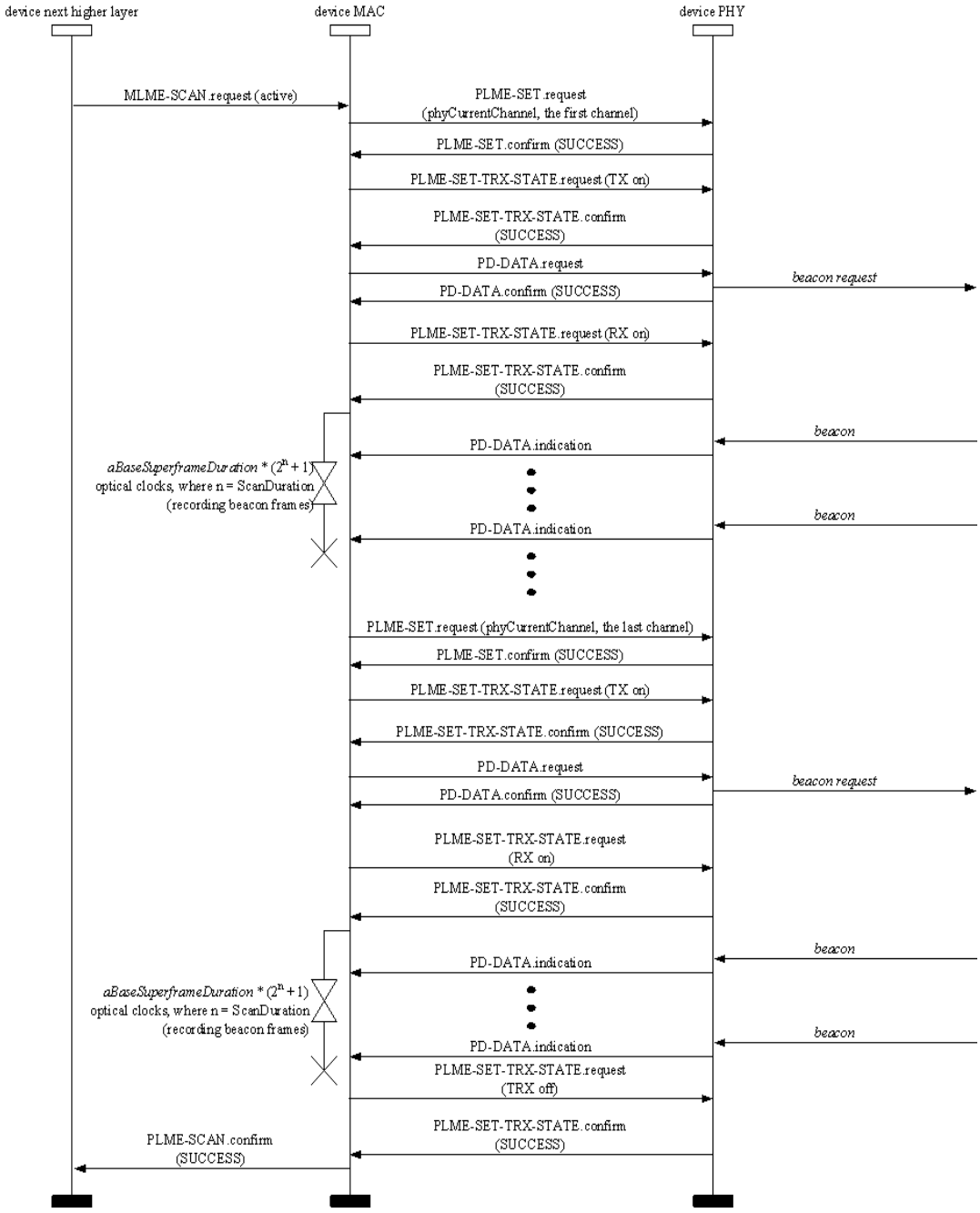


Figure 105—Active scan message sequence chart

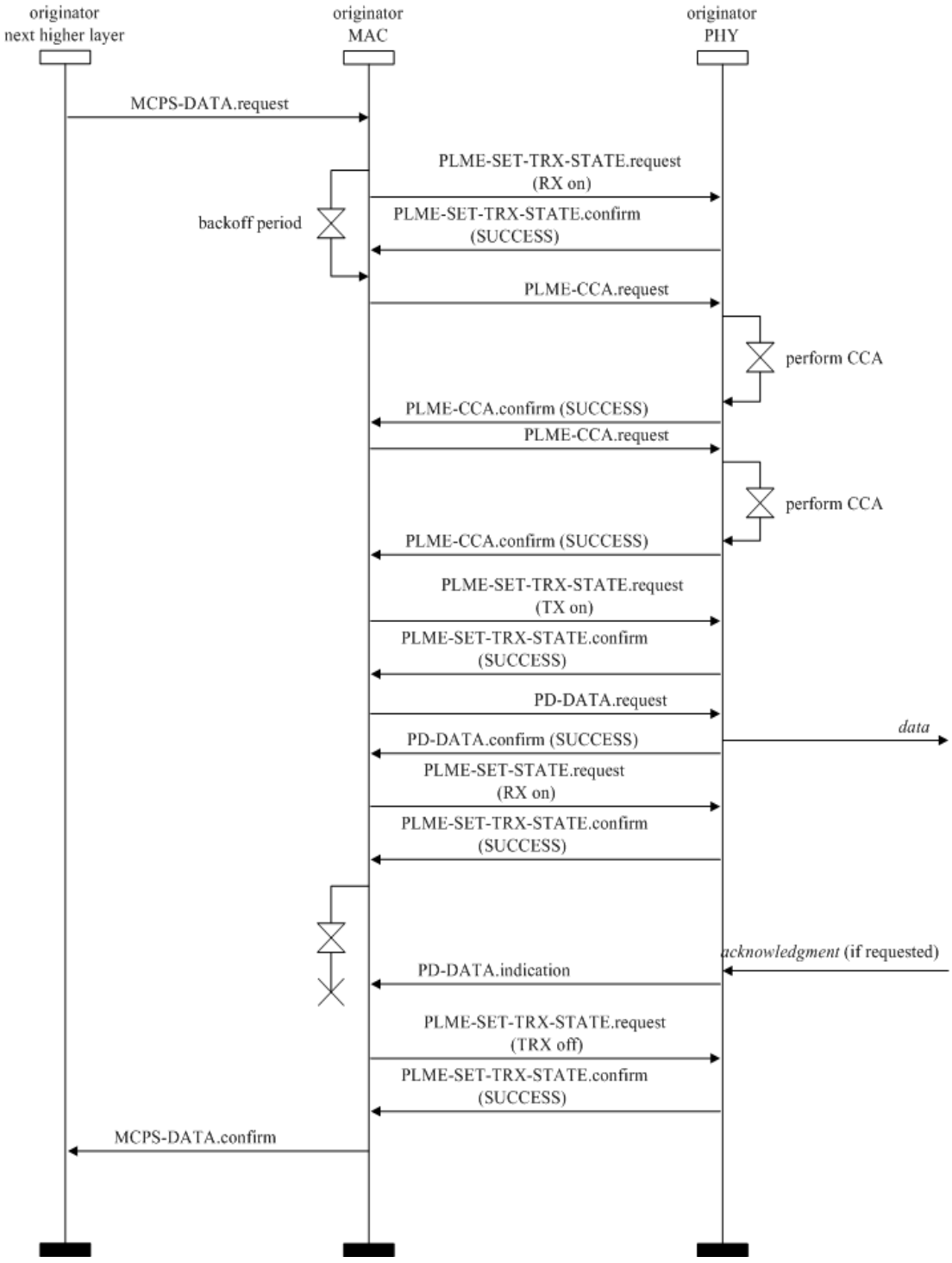


Figure 106—Data-transmission message sequence chart—originator

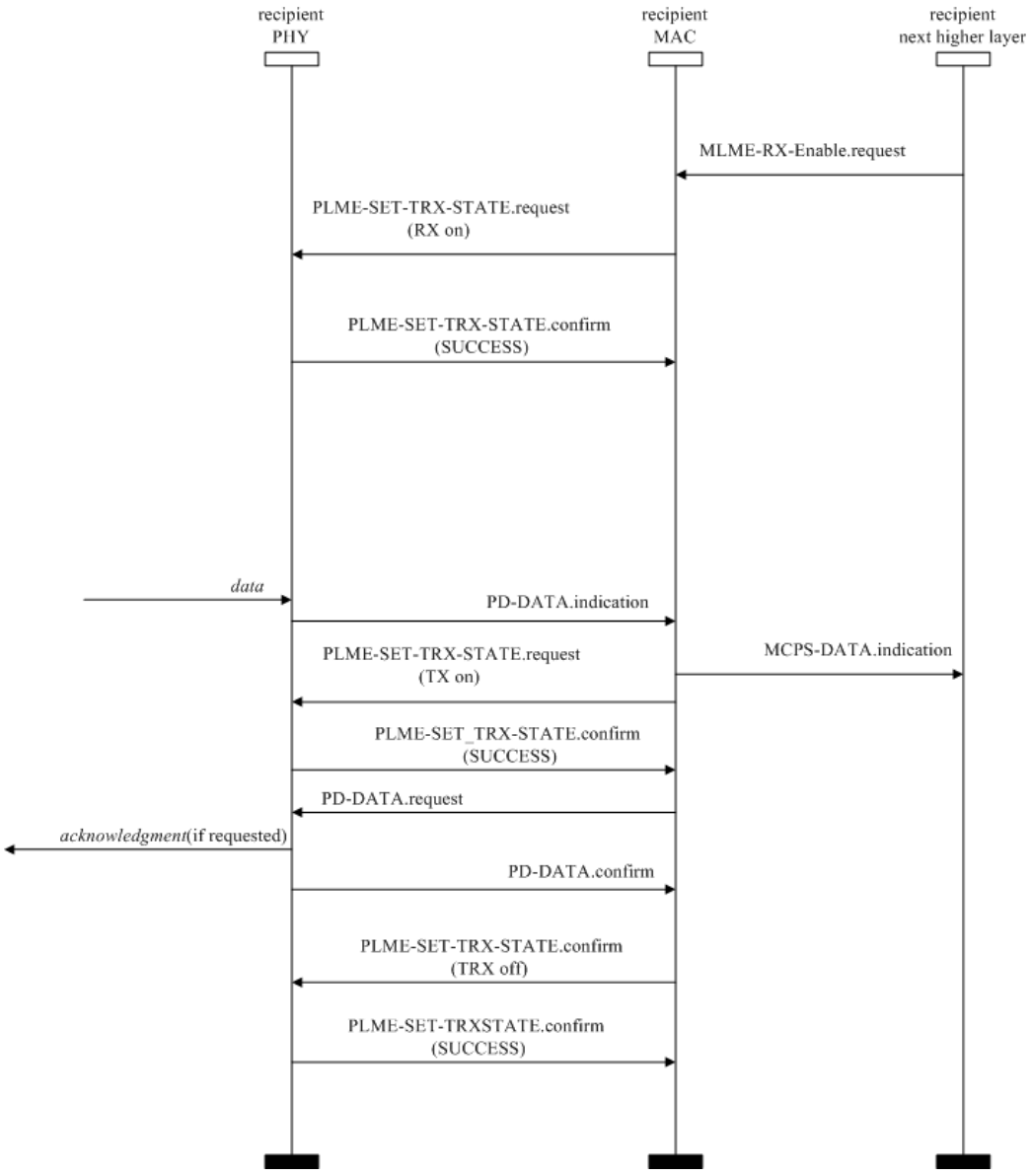


Figure 107—Data-transmission message sequence chart—recipient

7. Security suite specifications

7.1 Overview

The MAC sublayer is responsible for providing security services on specified incoming and outgoing frames when requested to do so by the higher layers. This standard supports the following security services (as defined in 4.6 for definitions):

— Data confidentiality

— Data authenticity
— Replay protection

The information determining how to provide the security is found in the security-related PIB (as defined in Table 66 in 7.5.1).

7.2 Functional description

A device may optionally implement security. A device that does not implement security shall not provide a mechanism for the MAC sublayer to perform any cryptographic transformation on incoming and outgoing frames nor require any PIB attributes associated with security. A device that implements security shall provide a mechanism for the MAC sublayer to provide cryptographic transformations on incoming and outgoing frames using information in the PIB attributes associated with security when the *macSecurityEnabled* attribute is set to TRUE.

If the MAC sublayer is required to transmit a frame or receives an incoming frame, the MAC sublayer shall process the frame as specified in 7.2.1 and 7.2.3, respectively.

7.2.1 Outgoing frame security procedure

The inputs to this procedure are the frame to be secured and the SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters from the originating primitive or automatic request PIB attributes. The outputs from this procedure are the status of the procedure and, if this status is SUCCESS, the secured frame. If outgoing frame security procedure is not successful, the frame is discarded

The outgoing frame security procedure involves the following steps as applicable:

a) If the Security Enabled subfield of the frame control field of the frame to be secured is set to zero, the procedure shall set the security level to zero.
b) If the Security Enabled subfield of the frame control field of the frame to be secured is set to one, the procedure shall set the security level to the SecurityLevel parameter. If the resulting security level is zero, the procedure shall return with a status of UNSUPPORTED_SECURITY.
c) If the *macSecurityEnabled* attribute is set to FALSE and the security level is not equal to zero, the procedure shall return with a status of UNSUPPORTED_SECURITY.
d) The procedure shall determine whether the frame to be secured satisfies the constraint on the maximum length of MAC frames, as follows:
    1) The procedure shall set the length *M*, in octets, of the Authentication field to zero if the security level is equal to zero and shall determine this value from the security level and Table 64 otherwise.
    2) The procedure shall determine the length AuxLen, in octets, of the auxiliary security header (see 7.4) using KeyIdMode and the security level.
    3) The procedure shall determine the data expansion as AuxLen + *M*.
    4) The procedure shall check whether the length of the frame to be secured, including data expansion and FCS, is less than or equal to *aMaxPHYFrameSize*. If this check fails, the procedure shall return with a status of FRAME_TOO_LONG.
e) If the security level is zero, the procedure shall set the secured frame to be the frame to be secured and return with the secured frame and a status of SUCCESS.
f) The procedure shall set the frame counter to the *macFrameCounter* attribute. If the frame counter has the value 0xffffffff, the procedure shall return with a status of COUNTER_ERROR and all keys associated with the device shall be reinitialized and updated as discussed in 7.5.5.
g) The procedure shall obtain the key using the outgoing frame key retrieval procedure as described in 7.2.2. If that procedure fails, the procedure shall return with a status of UNAVAILABLE_KEY.
h) The procedure shall insert the auxiliary security header into the frame, with fields set as follows:

1) The Security Level subfield of the Security Control field shall be set to the security level.
2) The Key Identifier Mode subfield of the Security Control field shall be set to the KeyIdMode parameter.
3) The Frame Counter field shall be set to the frame counter.
4) If the KeyIdMode parameter is set to a value not equal to zero, the Key Source and Key Index subfields of the Key Identifier field shall be set to the KeySource and KeyIndex parameters, respectively.

i) The procedure shall then use *aExtendedAddress*, the frame counter, the security level, and the key to produce the secured frame according to the transformation process known as CCM* [or the extension of CCM, which is the combined counter with CBC-MAC (i.e., cipher block chaining message authentication code) mode of operation] that is described in the security operations (see 7.3.4).
   1) If the SecurityLevel parameter specifies the use of encryption (as defined in Table 64 in 7.4.2.1), the encryption operation shall be applied only to the actual payload field within the MSDU, i.e., the Beacon Payload field (see 5.2.2.1.8), Command Payload field (see 5.2.2.4.3), or Data Payload field (see 5.2.2.2.2), depending on the frame type. The corresponding payload field is passed to the CCM* transformation process described in 7.3.4 as the unsecured payload (as defined in Table 61 in 7.3.4.2). The resulting encrypted payload shall substitute the original payload.
   2) The remaining fields in the MSDU part of the frame shall be passed to the CCM* transformation process described in 7.3.4 as the nonpayload fields (see Table 61).
   3) The ordering and exact manner of performing the encryption and integrity operations and the placement of the resulting encrypted data or integrity code within the MSDU field shall be as defined in 7.3.4.
j) The procedure shall increment the frame counter by one and set the *macFrameCounter* attribute to the resulting value.
k) The procedure shall return with the secured frame and a status of SUCCESS.

7.2.2 Outgoing frame key retrieval procedure

The inputs to this procedure are the frame to be secured and the KeyIdMode, KeySource, and KeyIndex parameters from the originating primitive. The outputs from this procedure are a passed or failed status and, if passed, a key.

The outgoing frame key retrieval procedure involves the following steps as applicable:

a) If the KeyIdMode parameter is set to 0x00 (implicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:
   1) If the Destination Addressing Mode subfield of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the short address is used), the key lookup data shall be set to the 2-octet Source VPAN Identifier field of the frame right-concatenated (see B.2.1. of IEEE Std 802.15.4-2006) with the 2-octet *macVPANCoordShortAddress* attribute right-concatenated with the single octet 0x00. The key lookup size shall be set to five.
   2) If the Destination Addressing Mode subfield of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the key lookup data shall be set to the 8-octet *macVPANCoordExtendedAddress* attribute right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet 0x00. The key lookup size shall be set to nine.
   3) If the Destination Addressing Mode subfield of the frame control field of the frame is set to 0x02, the key lookup data shall be set to the 2-octet Destination VPAN Identifier field of the frame right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 2-octet Destination Address field of the frame right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

4) If the Destination Addressing Mode subfield of the frame control field of the frame is set to 0x03, the key lookup data shall be set to the 8-octet Destination Address field of the frame right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet 0x00. The key lookup size shall be set to nine.

b) If the KeyIdMode parameter is set to a value not equal to 0x00 (explicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:
   1) If the KeyIdMode parameter is set to 0x01, the key lookup data shall be set to the 8-octet *macDefaultKeySource* attribute right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet KeyIndex parameter. The key lookup size shall be set to nine.
   2) If the KeyIdMode parameter is set to 0x02, the key lookup data shall be set to the 4-octet KeySource parameter right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet KeyIndex parameter. The key lookup size shall be set to five.
   3) If the KeyIdMode parameter is set to 0x03, the key lookup data shall be set to the 8-octet KeySource parameter right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet KeyIndex parameter. The key lookup size shall be set to nine.

c) The procedure shall obtain the KeyDescriptor by passing the key lookup data and the key lookup size to the KeyDescriptor lookup procedure as described in 7.2.5. If that procedure returns with a failed status, this procedure shall also return with a failed status.

d) The MAC sublayer shall set the key to the Key element of the KeyDescriptor.

e) The procedure shall return with a passed status, having obtained the key identifier and the key.

NOTE—For broadcast frames, the outgoing frame key retrieval procedure will result in a failed status if implicit key identification is used. Hence, explicit key identification should be used for broadcast frames.

7.2.3 Incoming frame security procedure

The input to this procedure is the frame to be unsecured. The outputs from this procedure are the unsecured frame, the security level, the key identifier mode, the key source, the key index, and the status of the procedure. All outputs of this procedure are assumed to be invalid unless and until explicitly set in this procedure. It is assumed that the PIB attributes associating KeyDescriptors in *macKeyTable* with a single, unique device or a number of devices will have been established by the next higher layer. The incoming frame security procedure involves the following steps:

a) If the Security Enabled field of the frame control field of the frame to be unsecured is set to zero, the procedure shall set the security level to zero.

b) If the Security Enabled field of the Frame Control field of the frame to be unsecured is set to one, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of UNSUPPORTED_LEGACY.

c) If the Security Enabled field of the Frame Control field of the frame to be unsecured is set to one, the procedure shall set the security level and the key identifier mode to the corresponding fields of the Security Control field of the auxiliary security header of the frame to be unsecured, and the key source and key index to the corresponding fields of the Key Identifier field of the auxiliary security header of the frame to be unsecured, if present. If the resulting security level is zero, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of UNSUPPORTED_SECURITY.

d) If the *macSecurityEnabled* attribute is set to FALSE, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of SUCCESS if the security level is equal to zero and with a status of UNSUPPORTED_SECURITY otherwise.

e) The procedure shall determine whether the frame to be unsecured meets the minimum security level by passing the security level, the frame type, and, depending on whether the frame is a MAC command frame, the first octet of the MSDU (i.e., command frame identifier for a MAC command frame) to the incoming security level checking procedure as described in 7.2.8. If that procedure

fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of IMPROPER_SECURITY_LEVEL.

f) If the security level is set to zero, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of SUCCESS.

g) The procedure shall obtain the KeyDescriptor, DeviceDescriptor, and KeyDeviceDescriptor using the incoming frame security material retrieval procedure described in 7.2.4. If that procedure fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of UNAVAILABLE_KEY.

h) The procedure shall determine whether the frame to be unsecured conforms to the key usage policy by passing the KeyDescriptor, the frame type, and, depending on whether the frame is a MAC command frame, the first octet of the MSDU (i.e., command frame identifier for a MAC command frame) to the incoming key usage policy checking procedure as described in 7.2.9. If that procedure fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of IMPROPER_KEY_TYPE.

i) If the Exempt element of the DeviceDescriptor is set to FALSE and if the incoming security level checking procedure of step e) above had as output the “conditionally passed” status, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of IMPROPER_SECURITY_LEVEL.

j) The procedure shall set the frame counter to the Frame Counter field of the auxiliary security header of the frame to be unsecured. If the frame counter has the value 0xffffffff, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of COUNTER_ERROR.

k) The procedure shall determine whether the frame counter is greater than or equal to the FrameCounter element of the DeviceDescriptor. If this check fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of COUNTER_ERROR.

l) The procedure shall then use the ExtAddress element of the DeviceDescriptor, the frame counter, the security level, and the Key element of the KeyDescriptor to produce the unsecured frame according to the CCM* inverse transformation process described in the security operations (see 7.3.5).
   1) If the security level specifies the use of encryption (as defined in Table 64 in 7.4.2.1), the decryption operation shall be applied only to the actual payload field within the MSDU, i.e., the Beacon Payload field (see 5.2.2.1.8), Command Payload field (see 5.2.2.4.3), or Data Payload field (see 5.2.2.2.2), depending on the frame type. The corresponding payload field shall be passed to the CCM* inverse transformation process described in 7.3.5 as the secure payload.
   2) The remaining fields in the MSDU part of the frame shall be passed to the CCM* inverse transformation process described in 7.3.5 as the nonpayload fields.

m) If the CCM* inverse transformation process fails, the procedure shall set the unsecured frame to be the frame to be unsecured and return with a status of SECURITY_ERROR.

n) The procedure shall increment the frame counter by one and set the FrameCounter element of the DeviceDescriptor to the resulting value.

o) If the FrameCounter element is equal to 0xffffffff, the procedure shall set the Blacklisted element of the KeyDeviceDescriptor.

p) The procedure shall return with the unsecured frame and a status of SUCCESS.

7.2.4 Incoming frame security material retrieval procedure

The input to this procedure is the frame to be unsecured. The outputs from this procedure are a passed or failed status and, if passed, a KeyDescriptor, a DeviceDescriptor, and a KeyDeviceDescriptor.

The incoming frame security material retrieval procedure involves the following steps as applicable:

a) If the Key Identifier Mode subfield of the Security Control field of the auxiliary security header of the frame is set to 0x00 (implicit key identification), the procedure shall determine the key lookup data and the key lookup size as follows:
   1) If the source address mode of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the

short address is used), the key lookup data shall be set to the 2-octet Destination VPAN Identifier field of the frame right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 2-octet *macVPANCoordShortAddress* attribute right-concatenated with the single octet 0x00. The key lookup size shall be set to five.

2) If the source address mode of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the key lookup data shall be set to the 8-octet *macVPANCoordExtendedAddress* attribute right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet 0x00. The key lookup size shall be set to nine.
3) If the source address mode of the frame control field of the frame is set to 0x02, the key lookup data shall be set to the 2-octet Source VPAN Identifier field of the frame, or to the 2-octet Destination VPAN Identifier field of the frame if the VPAN ID Compression subfield of the frame control field of the frame is set to one, right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 2-octet Source Address field of the frame right-concatenated with the single octet 0x00. The key lookup size shall be set to five.
4) If the source address mode of the frame control field of the frame is set to 0x03, the key lookup data shall be set to the 8-octet Source Address field of the frame right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the single octet 0x00. The key lookup size shall be set to nine.

b) If the Key Identifier Mode subfield of the Security Control field of the auxiliary security header of the frame is set to a value not equal to 0x00 (explicit key identification), the procedure shall determine the key lookup data and key lookup size as follows:
1) If the key identifier mode is set to 0x01, the key lookup data shall be set to the 8-octet *macDefaultKeySource* attribute right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to nine.
2) If the key identifier mode is set to 0x02, the key lookup data shall be set to the right-concatenation (see B.2.1 of IEEE Std 802.15.4-2006) of the 4-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to five.
3) If the key identifier mode is set to 0x03, the key lookup data shall be set to the right-concatenation (see B.2.1 of IEEE Std 802.15.4-2006) of the 8-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header. The key lookup size shall be set to nine.

c) The procedure shall obtain the KeyDescriptor by passing the key lookup data and the key lookup size to the KeyDescriptor lookup procedure as described in 7.2.5. If that procedure returns with a failed status, the procedure shall also return with a failed status.

d) The procedure shall determine the device lookup data and the device lookup size as follows:
1) If the source address mode of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to a value in the range 0x0000–0xfffd (i.e., the short address is used), the device lookup data shall be set to the 2-octet Destination VPAN Identifier field of the frame right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 2-octet *macVPANCoordShortAddress* attribute. The device lookup size shall be set to four.
2) If the source address mode of the frame control field of the frame is set to 0x00 and the *macVPANCoordShortAddress* attribute is set to 0xfffe (i.e., the extended address is used), the device lookup data shall be set to the 8-octet *macVPANCoordExtendedAddress* attribute. The device lookup size shall be set to eight.
3) If the source address mode of the frame control field of the frame is set to 0x02, the device lookup data shall be set to the 2-octet Source VPAN Identifier field of the frame, or to the 2-octet Destination VPAN Identifier field of the frame if the VPAN ID Compression subfield of the frame control field of the frame is set to one, right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the 2-octet Source Address field of the frame. The device lookup size shall be set to four.

4) If the source address mode of the frame control field of the frame is set to 0x03, the device lookup data shall be set to the 8-octet Source Address field of the frame. The device lookup size shall be set to eight.

e) The procedure shall obtain the DeviceDescriptor and the KeyDeviceDescriptor by passing the KeyDescriptor, the device lookup data, and the device lookup size to the blacklist checking procedure as described in 7.2.6. If that procedure returns with a failed status, the procedure shall also return with a failed status.

f) The procedure shall return with a passed status having obtained the KeyDescriptor, the DeviceDescriptor, and the KeyDeviceDescriptor.

7.2.5 Key descriptor lookup procedure

The inputs to this procedure are the key lookup data and the key lookup size. The outputs from this procedure are a passed or failed status and, if passed, a KeyDescriptor.

The KeyDescriptor lookup procedure involves the following steps as applicable:

a) For each KeyDescriptor in the *macKeyTable* attribute and for each KeyIdLookupDescriptor in the KeyIdLookupList of the KeyDescriptor, the procedure shall check whether the LookupDataSize element of the KeyIdLookupDescriptor indicates the same integer value, shown in Table 72, as the key lookup size and whether the LookupData element of the KeyIdLookupDescriptor is equal to the key lookup data. If both checks pass (i.e., there is a match), the procedure shall return with this (matching) KeyDescriptor and a passed status.

b) The procedure shall return with a failed status.

7.2.6 Blacklist checking procedure

The inputs to this procedure are the KeyDescriptor, the device lookup data, and the device lookup size. The outputs from this procedure are a passed or failed status and, if passed, a DeviceDescriptor and a KeyDeviceDescriptor.

The blacklist checking procedure involves the following steps as applicable:

a) For each KeyDeviceDescriptor in the KeyDeviceList of the KeyDescriptor:
   1) The procedure shall obtain the DeviceDescriptor using the DeviceDescriptorHandle element of the KeyDeviceDescriptor.
   2) If the UniqueDevice element of the KeyDeviceDescriptor is set to TRUE, the procedure shall return with the DeviceDescriptor, the KeyDeviceDescriptor, and a passed status if the BlackListed element of the KeyDeviceDescriptor is set to FALSE, or the procedure shall return with a failed status if this Blacklisted element is set to TRUE.
   3) If the UniqueDevice element of the KeyDeviceDescriptor is set to FALSE, the procedure shall execute the DeviceDescriptor lookup procedure as described in 7.2.7, with the device lookup data and the device lookup size as inputs. If the corresponding output of that procedure is a passed status, the procedure shall return with the DeviceDescriptor, the KeyDevice-Descriptor, and a passed status if the Blacklisted element of the KeyDeviceDescriptor is set to FALSE, or the procedure shall return with a failed status if this Blacklisted element is set to TRUE.

b) The procedure shall return with a failed status.

7.2.7 Device descriptor lookup procedure

The inputs to this procedure are the DeviceDescriptor, the device lookup data, and the device lookup size. The output from this procedure is a passed or failed status.

The DeviceDescriptor lookup procedure involves the following steps as applicable:

a) If the device lookup size is four and the device lookup data is equal to the VPAN ID element of the DeviceDescriptor right-concatenated (see B.2.1 of IEEE Std 802.15.4-2006) with the ShortAddress element of the Device-Descriptor, this procedure shall return with a passed status.
b) If the device lookup size is eight and the device lookup data is equal to the ExtAddress element of the DeviceDescriptor, this procedure shall return with a passed status.
c) The procedure shall return with a failed status.

7.2.8 Incoming security level checking procedure

The inputs to this procedure are the incoming security level, the frame type and the command frame identifier. The output from this procedure is a passed, failed, or "conditionally passed" status.

The incoming security level checking procedure involves the following steps as applicable:

a) It is recommended to use MIC for all secure messages as defined in Table 64. For each SecurityLevelDescriptor in the *macSecurityLevelTable* attribute:
    1) If the frame type is not equal to 0x03 and the frame type is equal to the FrameType element of the SecurityLevelDescriptor, the procedure shall compare the incoming security level (as SEC1) with the SecurityMinimum element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 7.4.2.1. If this comparison fails (i.e., evaluates to FALSE), the procedure shall return with a "conditionally passed" status if the DeviceOverrideSecurityMinimum element of the SecurityLevelDescriptor is set to TRUE and the security level is set to zero and with a failed status otherwise.
    2) If the frame type is equal to 0x03, the frame type is equal to the FrameType element of the SecurityLevelDescriptor, and the command frame identifier is equal to the CommandFrame-Identifier element of the SecurityLevelDescriptor, the procedure shall compare the incoming security level (as SEC1) with the SecurityMinimum element of the SecurityLevelDescriptor (as SEC2) according to the algorithm described in 7.4.2.1. If this comparison fails (i.e., evaluates to FALSE), the procedure shall return with a "conditionally passed" status if the DeviceOverrideSecurityMinimum element of the SecurityLevelDescriptor is set to TRUE and the security level is set to zero and with a failed status otherwise.
b) The procedure shall return with a passed status.

7.2.9 Incoming key usage policy checking procedure

The inputs to this procedure are the KeyDescriptor, the frame type, and the command frame identifier. The output from this procedure is a passed or failed status.

The incoming key usage policy checking procedure involves the following steps as applicable:

a) For each KeyUsageDescriptor in the KeyUsageList of the KeyDescriptor:
    1) If the frame type is not equal to 0x03 and the frame type is equal to the FrameType element of the KeyUsageDescriptor, the procedure shall return with a passed status.
    2) If the frame type is equal to 0x03, the frame type is equal to the FrameType element of the KeyUsageDescriptor, and the command frame identifier is equal to the CommandFrame-Identifier element of the KeyUsageDescriptor, the procedure shall return with a passed status.
b) The procedure shall return with a failed status.

7.3 Security operations

This subclause describes the parameters for the CCM* security operations, as specified in Annex A of IEEE Std 802.15.4-2006.

7.3.1 Integer and octet representation

The integer and octet representation conventions specified in Annex A of IEEE Std 802.15.4-2006 are used throughout 7.3.

7.3.2 CCM* nonce

The CCM* nonce is a 13-octet string and is used for the advanced encryption standard (AES)-CCM* mode of operation (see B.2.2 of IEEE Std 802.15.4-2006). The nonce shall be formatted as shown in Figure 108, with the left most field in the figure defining the first (and left most) octets and the right most field defining the last (and right most) octet of the nonce.

Octets: 8	**4**	**1**
Source address	Frame counter	Security level

Figure 108—CCM* nonce

The source address shall be set to the extended address *aExtendedAddress* of the device originating the frame, the frame counter to the value of the respective field in the auxiliary security header (see 7.4), and the security level to the security level identifier corresponding to the Security Level subfield of the Security Control field of the auxiliary security header as defined in Table 64.

The source address, frame counter, and security level shall be represented as specified in 7.3.1.

7.3.3 CCM* prerequisites

Securing a frame involves the use of the CCM* mode encryption and authentication transformation, as described in B.4.1. of IEEE Std 802.15.4-2006. Unsecuring a frame involves the use of the CCM* decryption and authentication checking process, as described in B.4.2 of IEEE Std 802.15.4-2006. The prerequisites for the CCM* forward and inverse transformations are as follows:

— The underlying block cipher shall be the AES encryption algorithm as specified in B.3.1 of IEEE Std 802.15.4-2006.
— The bit ordering shall be as defined in 7.3.1.
— The length in octets of the Length field *L* shall be 2 octets.
— The length of the Authentication field *M* shall be 0 octets, 4 octets, 8 octets, or 16 octets, as required.

The length of the Authentication field *M* for the CCM* forward transformation and the CCM* inverse transformation is determined from Table 64, using the Security Level subfield of the Security Control field of the auxiliary security header of the frame.

7.3.4 CCM* transformation data representation

This subclause describes how the inputs and output of the CCM* forward transformation, as described in B.4.1 of IEEE Std 802.15.4-2006, are formed.

The inputs are as follows:

— Key
— Nonce
— *a* data
— *m* data

The output is *c* data.

7.3.4.1 Key and nonce data inputs

The Key data for the CCM* forward transformation is passed by the outgoing frame security procedure described in 7.2.1. The nonce data for the CCM* transformation is constructed as described in 7.3.2.

7.3.4.2 *a* data and *m* data

In the CCM* transformation process, the data fields shall be applied as in Table 61.

Table 61—*a* data and *m* data for all security levels

Security level identifier	*a* data	*m* data
0x00	None	None
0x01	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x02	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x03	MHR || Auxiliary security header || Nonpayload fields || Unsecured payload fields	None
0x04	None	Unsecured payload fields
0x05	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields
0x06	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields
0x07	MHR || Auxiliary security header || Nonpayload fields	Unsecured payload fields

7.3.4.3 *c* data output

In the CCM* transformation process, the data fields that are applied, or right-concatenated and applied, represent octet strings.

The secured payload fields right-concatenated with the authentication tag shall substitute the unsecured payload field in the original unsecured frame to form the secured frame (see Table 62).

Table 62—*c* data for all security levels

Security level identifier	*c* data
0x00	None
0x01	MIC-32
0x02	MIC-64
0x03	MIC-128
0x04	Secured payload fields
0x05	Secured payload fields || MIC-32
0x06	Secured payload fields || MIC-64
0x07	Secured payload fields || MIC-128

7.3.5 CCM* inverse transformation data representation

This subclause describes how the inputs and output of the CCM* inverse transformation, as described in C.4.2 of IEEE Std 802.15.4-2006, are formed.

The inputs are as follows:

— Key
— Nonce
— *c* data
— *a* data

The output is *m* data.

7.3.5.1 Key and nonce data inputs

The key data for the CCM* inverse transformation is passed by the incoming frame security procedure described in 7.2.3. The nonce data for the CCM* transformation is constructed as described in 7.3.2.

7.3.5.2 *c* data and *a* data

In the CCM* inverse transformation process, the data fields shall be applied as in Table 63.

7.3.5.3 *m* data output

The *m* data shall then substitute secured payload fields and authentication tag in the original secured frame to form the unsecured frame.

7.4 Auxiliary Security header

The Auxiliary Security Header field has a variable length and contains information required for security processing, including a Security Control field, a Frame Counter field, and a Key Identifier field. The Auxiliary Security Header field shall be present only if the Security Enabled subfield of the frame control field is set to one. The Auxiliary Security Header field shall be formatted as illustrated in Figure 109.

Table 63—*c* data and *a* data for all security levels

Security level identifier	*c* data	*a* data
0x00	None	None
0x01	MIC-32	MHR || Auxiliary security header || Nonpayload fields ||Secured payload fields
0x02	MIC-64	MHR || Auxiliary security header || Nonpayload fields || Secured payload fields
0x03	MIC-128	MHR || Auxiliary security header || Nonpayload fields || Secured payload fields
0x04	Secured payload fields	MHR || Auxiliary security header || Nonpayload fields
0x05	Secured payload fields || MIC-32	MHR || Auxiliary security header || Nonpayload fields
0x06	Secured payload fields || MIC-64	MHR || Auxiliary security header || Nonpayload fields
0x07	Secured payload fields || MIC-128	MHR || Auxiliary security header || Nonpayload fields

Octets: 1	4	0/1/5/9
Security Control	Frame Counter	Key Identifier

Figure 109—Format of the auxiliary security header

7.4.1 Integer and octet representation

The auxiliary security header is a MAC frame field (see 5.2.1.7) and, therefore, uses the representation conventions specified in 5.2.

7.4.2 Security Control field

The Security Control field is 1 octet in length and is used to provide information about what protection is applied to the frame. The Security Control field shall be formatted as shown in Figure 110.

Bit: 0–2	3–4	5–7
Security Level	Key Identifier Mode	Reserved

Figure 110—Security Control field format

7.4.2.1 Security Level subfield

The Security Level subfield indicates the actual frame protection that is provided. This value can be adapted on a frame-by-frame basis and allows for varying levels of data authenticity (to allow minimization of security overhead in transmitted frames where required) and for optional data confidentiality. The cryptographic protection offered by the various security levels is shown in Table 64. When nontrivial protection is required, replay protection is always provided.

Table 64—Security levels available to the MAC sublayer

Security level identifier	Security Control field (Figure 110) b_2 b_1 b_0	Security attributes	Data confidentiality	Data authenticity (including length *M* of authentication tag, in octets)
0x00	'000'	None	OFF	NO (*M* = 0)
0x01	'001'	MIC-32	OFF	YES (*M* = 4)
0x02	'010'	MIC-64	OFF	YES (*M* = 8)
0x03	'011'	MIC-128	OFF	YES (*M* = 16)
0x04	'100'	ENC	ON	NO (*M* = 0)
0x05	'101'	ENC-MIC-32	ON	YES (*M* = 4)
0x06	'110'	ENC-MIC-64	ON	YES (*M* = 8)
0x07	'111'	ENC-MIC-128	ON	YES (*M* = 16)

Security levels can be ordered according to the corresponding cryptographic protection offered. Here, a first security level SEC1 is greater than or equal to a second security level SEC2 if and only if SEC1 offers at least the protection offered by SEC2, both with respect to data confidentiality and with respect to data authenticity. The statement "SEC1 is greater than or equal to SEC2" shall be evaluated as TRUE if both of the following conditions apply:

a) Bit position b_2 in SEC1 is greater than or equal to bit position b_2 in SEC2 (where Encryption OFF < Encryption ON).
b) The integer value of bit positions b_1 b_0 in SEC1 is greater than or equal to the integer value of bit positions b_1 b_0 in SEC2 [where increasing integer values indicate increasing levels of data authenticity provided, i.e., message integrity code (MIC)-0 < MIC-32 < MIC-64 < MIC-128].

Otherwise, the statement shall be evaluated as FALSE.

For example, ENC-MIC-64 ≥ MIC-64 is TRUE because ENC-MIC-64 offers the same data authenticity protection as MIC-64, plus confidentiality. On the other hand, MIC-128 ≥ ENC-MIC-64 is FALSE because even though MIC-128 offers stronger data authenticity than ENC-MIC-64, it offers no confidentiality.

7.4.2.2 Key Identifier Mode subfield

The Key Identifier Mode subfield indicates whether the key that is used to protect the frame can be derived implicitly or explicitly; furthermore, it is used to indicate the particular representations of the Key Identifier field (see 7.4.4) if derived explicitly. The Key Identifier Mode subfield shall be set to one of the values listed

in Table 65. The Key Identifier field of the auxiliary security header (see 7.4.4) shall be present only if this subfield has a value that is not equal to 0x00.

Table 65—Values of the Key Identifier mode

Key Identifier mode	Key Identifier Mode subfield b_1 b_0	Description	Key Identifier field length (octets)
0x00	'00'	Key is determined implicitly from the originator and recipient(s) of the frame, as indicated in the frame header.	0
0x01	'01'	Key is determined from the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header in conjunction with *macDefaultKeySource*.	1
0x02	'10'	Key is determined explicitly from the 4-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header.	5
0x03	'11'	Key is determined explicitly from the 8-octet Key Source subfield and the 1-octet Key Index subfield of the Key Identifier field of the auxiliary security header.	9

7.4.3 Frame Counter field

The Frame Counter field is 4 octets in length and represents the *macFrameCounter* attribute of the originator of a protected frame. It is used to provide semantic security of the cryptographic mechanism used to protect a frame and to offer replay protection.

7.4.4 Key Identifier field

The Key Identifier field has a variable length and identifies the key that is used for cryptographic protection of outgoing frames, either explicitly or in conjunction with implicitly defined side information. The Key Identifier field shall be present only if the Key Identifier Mode subfield of the Security Control field of the auxiliary security header (see 7.4.2.2) is set to a value different from 0x00. The Key Identifier field shall be formatted as illustrated in Figure 111.

Octets: 0/4/8	1
Key Source	Key Index

Figure 111—Format for the Key Identifier field, if present

7.4.4.1 Key Source subfield

The Key Source subfield, when present, is either 4 octets or 8 octets in length, according to the value specified by the Key Identifier Mode subfield of the Security Control field (see 7.4.2.2), and indicates the originator of a group key.

7.4.4.2 Key Index subfield

The Key Index subfield is 1 octet in length and allows unique identification of different keys with the same originator.

It is the responsibility of each key originator to make sure that actively used keys that it issues have distinct key indices and that the key indices are all different from 0x00.

7.5 Security-related MAC PIB attributes

The security-related MAC PIB attributes contain the following:

— Key table (*macKeyTable*, *macKeyTableEntries*)
— Device table (*macDeviceTable*, *macDeviceTableEntries*)
— Minimum security level table (*macSecurityLevelTable*, *macSecurityLevelTableEntries*)
— Frame counter (*macFrameCounter*)
— Automatic request attributes (*macAutoRequestSecurityLevel*, *macAutoRequestKeyIdMode*, *macAutoRequestKeySource*, *macAutoRequestKeyIndex*)
— Default key source (*macDefaultKeySource*)
— coordinator address (*macVPANCoordExtendedAddress, macVPANCoordShortAddress*)

7.5.1 PIB security material

The PIB security-related attributes are presented in Table 66, Table 67, Table 68, Table 69, Table 70, Table 71, and Table 72.

Table 66—Security-related MAC PIB attributes

Attribute	Identifier	Type	Range	Description	Default
macKeyTable	0x71	List of Key-Descriptor entries (see Table 67)	—	A table of KeyDescriptor entries, each containing keys and related information required for secured communications.	(empty)
macKeyTableEntries	0x72	Integer	Implementation specific	The number of entries in *macKeyTable*.	0
macDeviceTable	0x73	List of Device-Descriptor entries (see Table 71)	—	A table of Device-Descriptor entries, each indicating a remote device with which this device securely communicates.	(empty)
macDeviceTable-Entries	0x74	Integer	Implementation specific	The number of entries in *macDeviceTable*.	0

Table 66—Security-related MAC PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description	Default
macSecurity-LevelTable	0x75	Table of SecurityLevel Descriptor entries (see Table 70)	—	A table of SecurityLevel-Descriptor entries, each with information about the minimum security level expected depending on incoming frame type and subtype.	(empty)
macSecurity-LevelTableEntries	0x76	Integer	Implementa-tion specific	The number of entries in *macSecurityLevelTable*.	0
macFrameCounter	0x77	Integer	0x00000000–0xffffffff	The outgoing frame counter for this device.	0x00000000
macAutoRequest-SecurityLevel	0x78	Integer	0x00–0x07	The security level used for automatic data requests.	0x06
macAutoRequest-KeyIdMode	0x79	Integer	0x00–0x03	The key identifier mode used for automatic data requests. This attribute is invalid if the *macAuto-RequestSecurityLevel* attribute is set to 0x00.	0x00
macAutoRequest-KeySource	0x7a	As specified by the *mac-AutoRequest-KeyIdMode* parameter	—	The originator of the key used for automatic data requests. This attribute is invalid if the *macAuto-RequestKeyIdMode* element is invalid or set to 0x00.	All octets 0xff
macAutoRequest-KeyIndex	0x7b	Integer	0x01–0xff	The index of the key used for automatic data requests. This attribute is invalid if the *macAuto-RequestKeyIdMode* attribute is invalid or set to 0x00.	All octets 0xff
macDefaultKey-Source	0x7c	Set of 8 octets	—	The originator of the default key used for key identifier mode 0x01.	All octets 0xff
macVPANCoord-ExtendedAddress	0x7d	IEEE address	An extended 64-bit IEEE address	The 64-bit address of the coordinator.	—
macVPANCoordShor t-Address	0x7e	Integer	0x0000–0xffff	The 16-bit short address assigned to the coordinator. 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.	0x0000

Table 67—Elements of KeyDescriptor

Name	Type	Range	Description
KeyIdLookupList	List of KeyId-LookupDescriptor entries (see Table 72)	—	A list of KeyIdLookupDescriptor entries used to identify this KeyDescriptor.
KeyIdLookupListEntries	Integer	Implementation specific	The number of entries in KeyIdLookupList.
KeyDeviceList	List of KeyDevice-Descriptor entries (see Table 69)	—	A list of KeyDeviceDescriptor entries indicating which devices are currently using this key, including their blacklist status.
KeyDeviceListEntries	Integer	Implementation specific	The number of entries in KeyDeviceList.
KeyUsageList	List of KeyUsage-Descriptor entries (see Table 68)	—	A list of KeyUsageDescriptor entries indicating the frame types with which this key may be used.
KeyUsageListEntries	Integer		The number of entries in KeyUsageList.
Key	Set of 16 octets	—	The actual value of the key.

Table 68—Elements of KeyUsageDescriptor

Name	Type	Range	Description
FrameType	Integer	0x00–0x03	As defined in 5.2.1.1.2.
CommandFrameIdentifier	Integer	0x00–0x09	As defined in Table 10.

Table 69—Elements of KeyDeviceDescriptor

Name	Type	Range	Description
DeviceDescriptorHandle	Integer	Implementa-tion specific	Handle to the DeviceDescriptor corresponding to the device (see Table 71).
UniqueDevice	Boolean	TRUE or FALSE	Indication of whether the device indicated by DeviceDescriptorHandle is uniquely associated with the KeyDescriptor, i.e., it is a link key as opposed to a group key.
Blacklisted	Boolean	TRUE or FALSE	Indication of whether the device indicated by DeviceDescriptorHandle previously communicated with this key prior to the exhaustion of the frame counter. If TRUE, this indicates that the device shall not use this key further because it exhausted its use of the frame counter used with this key.

Table 70—Elements of SecurityLevelDescriptor

Name	Type	Range	Description
FrameType	Integer	0x00–0x03	As defined in 5.2.1.1.2.
CommandFrameIdentifier	Integer	0x00–0x09	As defined in Table 10.
SecurityMinimum	Integer	0x00–0x07	The minimal required/expected security level for incoming MAC frames with the indicated frame type and, if present, command frame type (as defined in Table 64 in 7.4.2.1).
DeviceOverrideSecurity-Minimum	Boolean	TRUE or FALSE	Indication of whether originating devices for which the Exempt flag is set may override the minimum security level indicated by the SecurityMinimum element. If TRUE, this indicates that for originating devices with Exempt status, the incoming security level zero is acceptable, in addition to the incoming security levels meeting the minimum expected security level indicated by the SecurityMinimum element.

Table 71—Elements of DeviceDescriptor

Name	Type	Range	Description
VPANId	Device VPAN ID	0x0000–0xffff	The 16-bit VPAN identifier of the device in this DeviceDescriptor.
ShortAddress	Device short address	0x0000–0xffff	The 16-bit short address of the device in this DeviceDescriptor. A value of 0xfffe indicates that this device is using only its extended address. A value of 0xffff indicates that this value is unknown.
ExtAddress	IEEE address	Any valid 64-bit device address	The 64-bit IEEE extended address of the device in this DeviceDescriptor. This element is also used in unsecuring operations on incoming frames.
FrameCounter	Integer	0x00000000–0xffffffff	The incoming frame counter of the device in this DeviceDescriptor. This value is used to ensure sequential freshness of frames.
Exempt	Boolean	TRUE or FALSE	Indication of whether the device may override the minimum security level settings defined in Table 70.

7.5.2 Key table

The key table holds key descriptors (keys with related key-specific information) that are required for security processing of outgoing and incoming frames. Key-specific information in the key table is identified based on information explicitly contained in the requesting primitive or in the received frame, as described in the outgoing frame key retrieval procedure (see 7.2.2) and the incoming frame security material retrieval procedure (see 7.2.4), as well as in the KeyDescriptor lookup procedure (see 7.2.5).

Table 72—Elements of KeyIdLookupDescriptor

Name	Type	Range	Description
LookupData	Set of 5 or 9 octets	—	Data used to identify the key.
LookupDataSize	Integer	0x00–0x01	A value of 0x00 indicates a set of 5 octets; a value of 0x01 indicates a set of 9 octets.

7.5.3 Device table

The device table holds device descriptors (device-specific addressing information and security-related information) that, when combined with key-specific information from the key table, provide all the keying material needed to secure outgoing (see 7.2.1) and unsecure incoming frames (see 7.2.3). Device-specific information in the device table is identified based on the originator of the frame, as described in the DeviceDescriptor lookup procedure (see 7.2.7), and on key-specific information, as described in the blacklist checking procedure (see 7.2.6).

7.5.4 Minimum security level table

The minimum security level table holds information regarding the minimum security level the device expects to have been applied by the originator of a frame, depending on frame type and, if it concerns a MAC command frame, the command frame identifier. Security processing of an incoming frame will fail if the frame is not adequately protected, as described in the incoming frame security procedure (see 7.2.3) and in the incoming security level checking procedure (see 7.2.8).

7.5.5 Frame counter

The 4-octet frame counter is used to provide replay protection and semantic security of the cryptographic building block used for securing outgoing frames. The frame counter is included in each secured frame and is one of the elements required for the unsecuring operation at the recipient(s). The frame counter is incremented each time an outgoing frame is secured, as described in the outgoing frame security procedure (see 7.2.1). When the frame counter reaches its maximum value of 0xffffffff, the associated keying material can no longer be used, thus requiring all keys associated with the device to be updated. This provides a mechanism for ensuring that the keying material for every frame is unique and, thereby, provides for sequential freshness.

7.5.6 Automatic request attributes

Automatic request attributes hold all the information needed to secure outgoing frames generated automatically and not as a result of a higher layer primitive, as is the case with automatic data requests.

7.5.7 Default key source

The default key source is information commonly shared between originator and recipient(s) of a secured frame, which, when combined with additional information explicitly contained in the requesting primitive or in the received frame, allows an originator or a recipient to determine the key required for securing or unsecuring this frame, respectively. This provides a mechanism for significantly reducing the overhead of security information contained in secured frames in particular use cases as shown in 7.2.2 and 7.2.4.

7.5.8 Coordinator address

The address of the coordinator is information commonly shared between all devices in a VPAN, which, when combined with additional information explicitly contained in the requesting primitive or in the received frame, allows an originator of a frame directed to the coordinator or a recipient of a frame originating from the coordinator to determine the key and security-related information required for securing or unsecuring, respectively, this frame as shown in 7.2.2 and 7.2.4.

8. PHY layer specification

8.1 Overview

This clause specifies three PHY options for IEEE Std 802.15.7.

The PHY is responsible for the following tasks:

— Activation and deactivation of the VLC transceiver
— WQI for received frames
— Channel selection
— Data transmission and reception
— Error correction
— Synchronization

Constants and attributes that are specified and maintained by the PHY are written in the text of this clause in italics. Constants have a general prefix of “a”, e.g., *aMaxPHYFrameSize*, and are listed in Table 99. Attributes have a general prefix of “phy”, e.g., *phyCurrentChannel*, and are listed in Table 100.

This subclause specifies requirements that are common to all of the IEEE 802.15.7 PHYs.

8.2 Operating modes

A compliant IEEE 802.15.7 PHY shall implement at least one of the PHY I or PHY II mandatory modes (as defined in Clause 10 and Clause 11) given in Table 73 and Table 74. A device implementing the PHY III mode in Table 75 shall also implement PHY II mode for coexistence as summarized in 4.4.1.2. The PHY modulation modes may operate in the presence of dimming. Modulation using OOK under dimming provides constant range and variable data rate by inserting compensation time as defined in 4.4.3.1.4 while modulation using VPPM under dimming provides constant data rate and variable range by adjusting the pulse width as summarized in 4.4.3.1.5.

As shown in Table 73, Table 74, and Table 75, the standard provides channel coding support for error correction. PHY I supports concatenated coding with Reed-Solomon (RS) and convolutional coding (CC) since it has been designed for outdoor use with short frames. PHY II and PHY III support only RS coding. PHY I and PHY II also support a run length limited (RLL) code to provide DC balance, clock recovery, and flicker mitigation.

In addition to modulation and coding, multiple optical rates are provided for all PHY types in order to support a broad class of optical transmitters (LEDs) for various applications. The choice of optical rate used for communication is decided by the MAC during device discovery. The MAC shall select the optical clock rate for communication during the optical clock-rate selection process as defined in 6.5. The preamble shall be sent at clock rate chosen by the TX and supported by the RX. The preamble is a time domain sequence

and does not have any modulation, channel coding, or line coding. The PHY header shall be sent at the lowest data rate for the chosen clock rate. The clock rate does not change through the frame between the preamble, header, and payload.

Table 73—PHY I operating modes

Modulation	RLL code	Optical clock rate	FEC		Data rate
			Outer code (RS)	Inner code (CC)	
OOK	Manchester	200 kHz	(15,7)	1/4	11.67 kb/s
			(15,11)	1/3	24.44 kb/s
			(15,11)	2/3	48.89 kb/s
			(15,11)	none	73.3 kb/s
			none	none	100 kb/s
VPPM	4B6B	400 kHz	(15,2)	none	35.56 kb/s
			(15,4)	none	71.11 kb/s
			(15,7)	none	124.4 kb/s
			none	none	266.6 kb/s

Table 74—PHY II operating modes

Modulation	RLL code	Optical clock rate	FEC	Data rate
VPPM	4B6B	3.75 MHz	RS(64,32)	1.25 Mb/s
			RS(160,128)	2 Mb/s
		7.5 MHz	RS(64,32)	2.5 Mb/s
			RS(160,128)	4 Mb/s
			none	5 Mb/s
OOK	8B10B	15 MHz	RS(64,32)	6 Mb/s
			RS(160,128)	9.6 Mb/s
		30 MHz	RS(64,32)	12 Mb/s
			RS(160,128)	19.2 Mb/s
		60 MHz	RS(64,32)	24 Mb/s
			RS(160,128)	38.4 Mb/s
		120 MHz	RS(64,32)	48 Mb/s
			RS(160,128)	76.8 Mb/s
			none	96 Mb/s

Table 75—PHY III operating modes

Modulation	Optical clock rate	FEC	Data rate
4-CSK	12 MHz	RS(64,32)	12 Mb/s
8-CSK		RS(64,32)	18 Mb/s
4-CSK	24 MHz	RS(64,32)	24 Mb/s
8-CSK		RS(64,32)	36 Mb/s
16-CSK		RS(64,32)	48 Mb/s
8-CSK		none	72 Mb/s
16-CSK		none	96 Mb/s

8.3 General requirements

8.3.1 Wavelength band plan

A compliant device shall operate with peak radiated energy within the visible light spectrum defined as being from 380 nm to 780 nm. A compliant device shall operate in one or several visible light frequency bands as summarized in Table 76.

Table 76—Visible light wavelength band plan

Wavelength (nm)		Spectral width (nm)	Code
380	478	98	000
478	540	62	001
540	588	48	010
588	633	45	011
633	679	46	100
679	726	47	101
726	780	54	110
Reserved			111

The codes in Table 76 are used to indicate the wavelengths containing the spectral peak for the transmitted frame and are indicated in the PHY header. This information may be used by the receiver for optimizing its performance. The standard also supports use of wide bandwidth optical transmitters (such as white LEDs) that can transmit on multiple bands or have leakage in other bands using the concepts of channel aggregation and guard channels, as discussed in 5.1.2.5.

8.3.2 Optical mapping

A high switching level from the PHY, applied to the light source, shall result in a high radiated intensity. A low switching level from the PHY, applied to the light source, shall result in a reduced radiated intensity. The

extinction ratio, defined as the ratio of the high radiated intensity to the low radiated intensity, is at the discretion of the implementer.

8.3.3 Maximum error tolerance for multiple optical sources

If multiple optical sources are used for communication, it is recommended the optical sources have similar frequency responses in order to assist communication. The digital input to all the optical sources from the PHY shall be synchronized. Figure 112 shows the allowable spread at the output of the optical sources, assuming a synchronized digital input. The maximum spread at the average signal intensity level during the rise and fall time at the output of the optical sources shall not vary by more than 12.5% of the clock period.

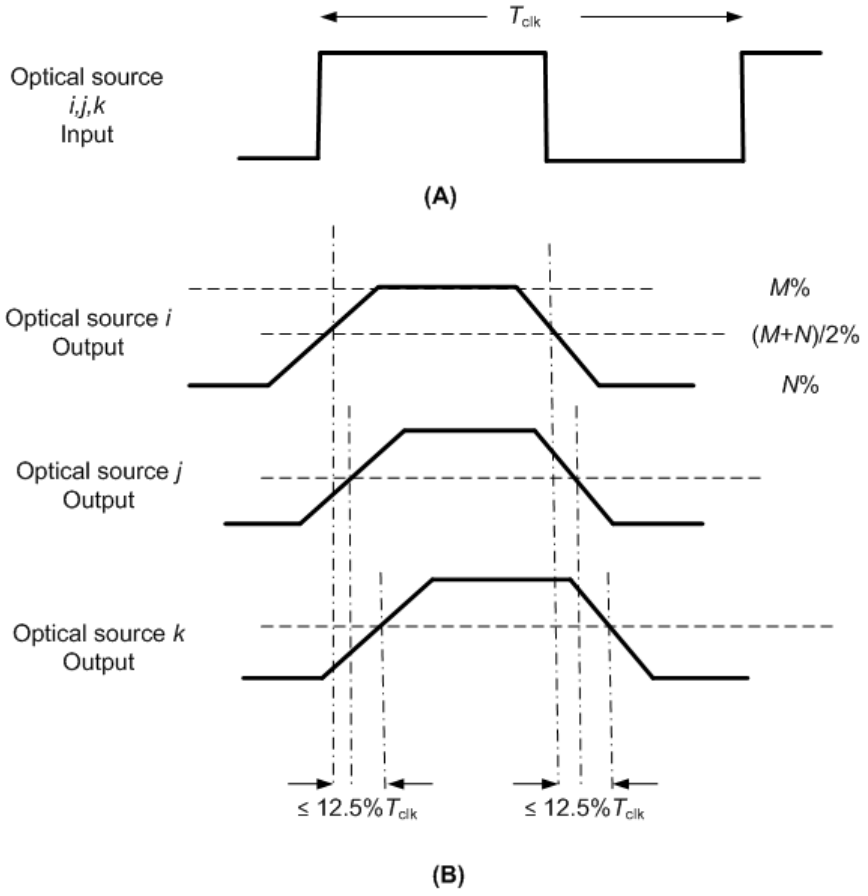


Figure 112—Maximum allowable spread at the output of optical sources

8.3.4 Minimum LIFS, SIFS, and RIFS periods

An interframe spacing (IFS) is used to provide spacing between adjacent frames. The minimum spacing between frames is dependent on the MAC mode of operation. The standard provides three types of interframe spacing: long (LIFS), short (SIFS), and reduced (RIFS). For peer-to-peer and star topologies, the SIFS, LIFS, and RIFS period is based on the currently negotiated optical clock rate by the MAC before starting data communication. Once the optical clock rate is selected, the SIFS, LIFS, and RIFS period is fixed to the values shown in Table 77. The clock-rate negotiation for a peer-to-peer and star topology is provided in 6.5. For a star topology, the beacon and CAP periods are defined at the lowest optical clock rate to ensure fair access to the medium. For a broadcast topology, the IFS is defined based on the optical clock rate chosen for broadcasting data to other devices. The minimum LIFS, SIFS, and RIFS periods for each of the PHYs are shown in Table 77. A detailed description, use, and illustration of LIFS, SIFS, and RIFS is shown in Figure 16.

Table 77—Minimum LIFS, SIFS, and RIFS periods

PHY	*macMinLIFSPeriod*	*macMinSIFSPeriod*	*macMinRIFSPeriod*	Units
PHY I	400	120	40	optical clocks
PHY II	400	120	40	optical clocks
PHY III	400	120	40	optical clocks

8.3.5 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be as shown in Table 99 and shall be measured at the air interface from the trailing edge of the last clock of the last transmission until the receiver is ready to begin the reception of the next PHY frame.

8.3.6 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be as shown in Table 99 and shall be measured at the air interface from the trailing edge of the last clock of the received frame until the transmitter is ready to begin transmission of the resulting acknowledgment. Actual transmission start times are specified by the MAC sublayer.

8.3.7 Transmit data clock frequency tolerance

The transmitted data clock frequency tolerance shall be ±20 ppm maximum.

8.3.8 Wavelength quality indicator (WQI)

8.3.8.1 OOK and VPPM WQI support

The WQI measurement is a characterization of the strength and/or quality of a received frame. The measurement may be implemented using receiver energy detection (ED), a signal-to-noise ratio estimation, or a combination of these methods. The use of the WQI result by the network or application layers is not specified in this standard. The WQI measurement shall be performed for each received frame, and the result shall be reported to the MAC sublayer using the PD-DATA.indication, specified in 9.3.3, as an integer ranging from 0x00 to 0xff. The minimum and maximum WQI values (0x00 and 0xff) should be associated with the lowest and highest quality IEEE 802.15.7 signals detectable by the receiver, and WQI values in

between should be uniformly distributed between these two limits. At least seven unique values of WQI shall be used. WQI value shall indicate the band plan ID, as given by the value in the PHY header of the received frame. A single WQI value set consists of band plan ID and corresponding WQI value as defined in Table 22.

8.3.8.2 CSK wavelength quality indication support

A device shall be capable of estimating the link quality of the received color channel, where the color quality shall be defined as an estimate of the SNR available after the CDR and will include all implementation losses associated with that particular receiver architecture (quantization noise, channel estimation errors, etc.). All estimated values, when measured under static channel conditions, shall be monotonically increasing with signal strength over the entire reporting range. Note that the estimates may exhibit saturation behavior at values higher than that required for highest data rate operation. Finally, the link quality estimates shall be made on a frame-by-frame basis. No bounds on absolute accuracy with respect to an external reference plane are intended or implied by this specification.

8.3.9 Clear channel assessment (CCA)

The IEEE 802.15.7 PHY may provide the capability to perform CCA according to at least one of the following three methods:

a) CCA Mode 1: Energy above threshold. CCA may report a busy medium upon detecting any energy above the energy detect threshold.
b) CCA Mode 2: Carrier sense only. CCA may report a busy medium only upon the detection of a signal with the modulation characteristics of IEEE Std 802.15.7. This signal may be above or below the energy detect threshold.
c) CCA Mode 3: Carrier sense with energy above threshold. CCA may report a busy medium only upon the detection of a signal with the modulation characteristics of IEEE Std 802.15.7 with energy above the energy detect threshold. See 4.3 for conceptual guidance.

For any of the CCA modes, if the PLME-CCA.request primitive, specified in 9.2.1, is received by the PHY during reception of a PPDU, CCA may report a busy medium. PPDU reception is considered to be in progress following detection of the preamble, and it remains in progress until the number of octets specified by the decoded PHR has been received.

A busy channel may be indicated by the PLME-CCA.confirm primitive with a status of BUSY as specified in 9.2.2.

A clear channel may be indicated by the PLME-CCA.confirm primitive with a status of IDLE as specified in 9.2.2.

The PHY PIB attribute *phyCCAMode* may indicate the appropriate operation mode as specified in Table 99.

8.4 Data modes

The PHY shall support the following normal data transmission modes as shown in Figure 113:

a) Single mode
b) Packed mode
c) Burst mode

In addition, there is a fourth mode for data transfer called “dimmed OOK” mode, which is used for data transfer while dimming in conjunction with OOK.

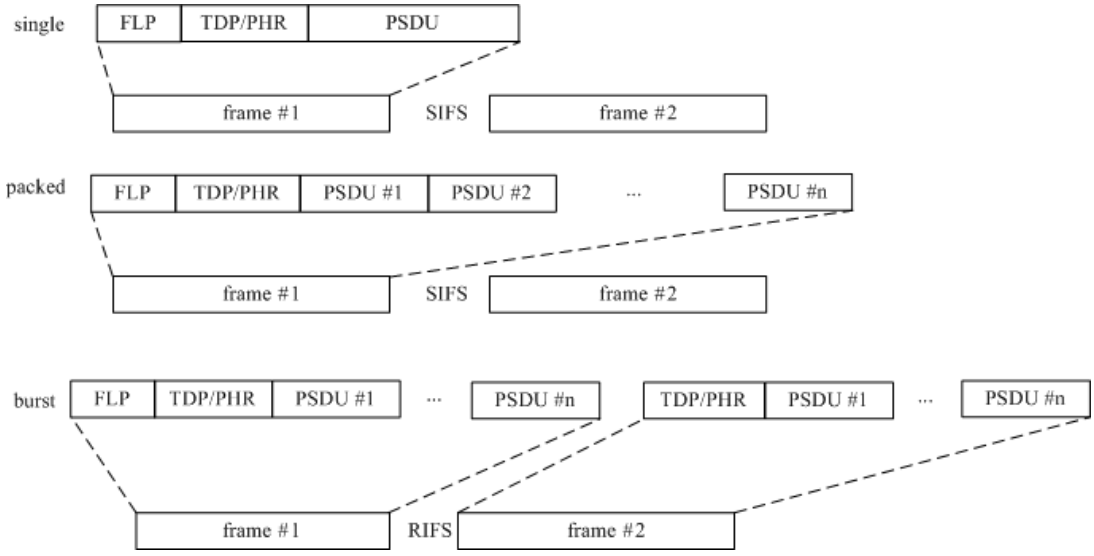


Figure 113—Data modes supported by the MAC (single, packed, burst)

The single mode transfers one PPDU per frame. This may be used for very short data communication such as acknowledgments, association, beaconing, or for information broadcast mode, for example.

The packed mode contains multiple PPDUs per frame and is used to send multiple consecutive PPDUs to the same destination within the frame for high throughput. Thus, the overhead of sending multiple MAC and PHY headers to the same destination is eliminated in this mode, providing higher MAC efficiency. This can be used in most modes as the preferred means of data communication.

The burst data mode uses a reduced length PHY preamble, as defined in 8.6.1, after the first frame in the burst. In addition, the RIFS is used between frames instead of the SIFS. The shorter preamble increases the efficiency and throughput in this mode.

The dimmed OOK mode is used to support data transfer under dimming requirements, as summarized in 4.4.3.1.4.

8.5 Dimming and flicker mitigation

A compliant IEEE 802.15.7 device shall honor all dimming requests from the upper layer. The dimming request from the upper layers to the PHY shall be indicated using the PHY PIB attribute, *phyDim* as shown in Table 100. The PHY shall support dimming using one of the techniques specified in either 8.5.1 or 8.5.2, when the *phyDim* PHY PIB attribute is set.

8.5.1 Dimming during idle time

The dimming during idle time is supported to avoid flicker and is achieved by the methods described in 4.4.3.1.1.

8.5.1.1 Idle pattern and compensation time dimming

An in-band or out-of-band idle pattern whose duty cycle variation results in brightness variation may be optionally inserted between the data frames for light dimming. Note that the concept of out-of-band includes

the option of using an un-modulated DC bias to maintain properly dimmed visibility. The compensation time (which means “ON” or “OFF” time of a light source) can be also inserted into either the idle pattern or into the data frame (if using the dimmed OOK mode) to reduce or increase the average brightness of a light source.

8.5.1.2 Visibility pattern dimming

The visibility pattern is an in-band idle pattern and is sent as part of the payload of the CVD frame as defined in 5.2.2.5. A set of 11 base low resolution patterns with 10% step size shall be used for dimming using visibility patterns. Any set of 11 base low resolution visibility patterns of any length can be used as long as there is no conflict between the visibility pattern and a valid RLL code. A set of 11 patterns are provided in Table 78 as an example for 8B10B code. The low resolution patterns shall be used to develop high resolution visibility patterns by averaging them across time to generate the required high resolution pattern. For example, if visibility patterns are available at 10% resolution, then a 25% visibility pattern can be attained for example, by alternately sending a 20% visibility pattern followed by a 30% visibility pattern. This method guarantees all visibility patterns will retain the same properties as the base low resolution visibility patterns. The high resolution visibility pattern shall be provided by using the low resolution patterns using the algorithm specified in Figure 114. The visibility patterns are repeated to satisfy the frame length as mentioned in the PHY header.

Table 78—Example of visibility patterns for 8B10B code

Visibility pattern	Percentage visibility
11111 11111	100%
11110 11111	90%
11110 11110	80%
11101 11100	70%
11001 11100	60%
10001 11100	50%
00001 11100	40%
00001 11000	30%
00001 10000	20%
00001 00000	10%
00000 00000	0%

Let the following values be defined as follows:

— Visibility patterns: $V_0, V_1, \ldots, V_K$

— Desired visibility = dv (expressed as a percentage value) e.g., for a 25.3% visibility, $dv = 25.3$

Desired precision = $p, p \leq 0, p$ is an integer (expressed as a logarithm value) e.g., for 0.01%,

$$\text{sel1pat} = \left\lfloor \frac{dv * K}{100} \right\rfloor$$

$$\text{sel2pat} = \left\lceil \frac{dv * K}{100} \right\rceil$$

$$\text{reppat2} = 10^{-p}\left(dv - \frac{100 * \text{sel1pat}}{K} \right)$$

$$\text{reppat1} = 10^{1-p} - \text{reppat2}$$

Then, to achieve visibility dv:

— repeat V_{sel1pat} reppat1 times, and

— repeat V_{sel2pat} reppat2 times.

Figure 114—Algorithm for achieving 0.1% dimming resolution with visibility patterns

8.5.2 Dimming during data transmission time

The dimming technologies on data transmission time depend on the PHY modulation schemes and are designed to avoid flicker. As stated in 8.5, all devices shall honor dimming requests but a device shall not be required to support communication for any dimming request. In this case a device may issue a disassociation notification command (see 5.3.3) with the reason given in Table 13. Due to non-linear human eye response to light, dimming levels as low as 0.1% shall be supported (square law phenomenon).

8.5.2.1 CSK-mode dimming

The CSK-mode dimming is described in 4.4.3.1.3. In CSK, total average power of multiple light sources is constant. For dimming control, the instantaneous power per light source is changed in order to adjust the average intensity to the required level. CSK keeps the center color of the color constellation with required intensity. A color stabilization scheme for illuminators is also provided in 8.5.4.

8.5.2.2 OOK-mode dimming

The OOK-mode dimming is described in 4.4.3.1.4. The OOK-mode dimming is supported by using the dimmed OOK bit field set in the PHY header as explained in Table 82 in 8.6.2. An arbitrary dimming level accuracy can be achieved by the combined use of the compensation length (described in 8.6.4.2), and optical mapping and extinction ratio (described in 8.3.2). If any requested dimming results in unsatisfactory performance (e.g., flicker generation or color shifting) while trying to maintain compliance to this standard, then the device shall disassociate from the network. The dimming method used by an unassociated device is out-of-scope of this standard.

8.5.2.3 VPPM-mode dimming

VPPM-mode dimming is described in 4.4.3.1.5. VPPM modulation, as implemented here (see 10.6), supports a dimming-level resolution of 10%. To support a dimming resolution of 0.1%, as prescribed in 5.3.10, the VPPM PHY shall use the algorithm provided below.

The algorithm relies on the following symbols: VS_0, VS_1, VS_2, … VS_{10}. VS_0 corresponds to the light source being turned off (*macDim* = 0) and VS_{10} corresponds to the light source fully being turned on (*macDim* = 1000). VS_1 to VS_9 are the VPPM symbols for $d = 0.1$ to 0.9 (see 10.6).

a) Choose the dimming level *macDim* (see Table 60).

b) First, determine the type of the corresponding symbols, viz. $k_1 = \lfloor macDim/100 \rfloor$

and $k_2 = \lceil macDim/100 \rceil$, where $\lfloor \bullet \rfloor$ stands for rounding to the next lower integer and $\lceil \bullet \rceil$ for rounding to the next higher integer.

c) Next, calculate the number of how often each symbol is to be sent: rep_2 = $macDim - 100 \times k_1$

and rep_1 = 100 – rep_2 .

d) Then, to achieve the desired dimming level *macDim*:

— Sequentially assign VS_{k1} rep_1 times, and then,
— assign VS_{k2} rep_2 times.
— If the number of VPPM data symbols, to be sent, is not modulo 100, then add VPPM idle-pattern symbols so that the number of VPPM symbols to be sent becomes multiples of 100. The configurations of VPPM data and idle-pattern symbols are shown in Figure 115.

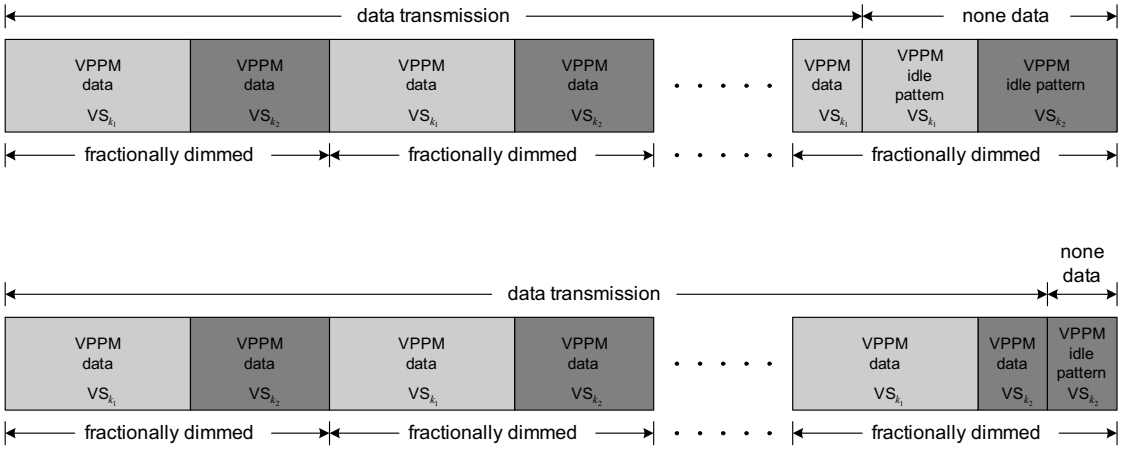


Figure 115—Sequential proportional cycling between two duty symbols to achieve fractional dimming with 0.1% accuracy in VPPM-mode

The upper panel shows padding with VS_{k1} and then VS_{k2} idle patterns. The lower panel shows padding with VS_{k2} idle-patterns.

Note that during data transmission time, only VPPM symbols between VS_1 and VS_9 can carry data information, as shown in Table 104. This is because VS_0 (light full off) and VS_{10} (light full on) cannot carry

data information because there are no transitions during these two symbols. Therefore, when a *macDim* value less than 100 is required, data information is carried only by VS_1 symbols. Similarly, data information is carried only by VS_9 symbols when a *macDim* value greater than 900 is required. All dimming requests must be honored even if data transmission is not possible. It is recommended that the receiver changes its matched filter in step with the change in the transmitter-symbol shape in order to enable optimum detection.

By default, a 50% duty cycle shall be used for VPPM. If dimming is supported using VPPM modulation, a dimming notification command shall be sent by the MAC. Both the TX and RX shall use the above algorithm for VPPM dimming. The transmitter shall honor all dimming requests from the upper layer. It is recommended that the transmitter uses the receiver's capability information as provided in 5.3.19.1.1 for VPPM dimming support. This information is obtained during the device discovery process described in 5.1.2.4.

8.5.3 Flicker mitigation

Flicker mitigation can be divided into intra-frame mitigation and interframe flicker mitigation as described in 4.4.3.2.

Intra-frame flicker mitigation refers to mitigation flicker within the transmission of a data frame. Intra-frame flicker in OOK is avoided by the use of the dimmed OOK mode as described in 8.5.2.2**,** and RLL coding as described in 8.2. VPPM inherently does not cause any interframe flicker and also uses a RLL code. Interframe flicker is avoided in CSK by ensuring constant average power across multiple light sources along with scrambling and the high optical clock rates (MHz) at which this modulation is used.

Interframe flicker mitigation applies to both data transmission (RX mode) and idle periods. While idling, visibility patterns or idle patterns as described in 8.5.1 may be used to ensure light emission by the VLC transmitters have the same average brightness over adjacent MFTPs as during data transmission. These patterns can be modulated in-band or out-of-band as in 8.5.1.1.

When the dimmer setting is changed above the MAC sublayer, the MAC and PHY layers adjust the data transmission and idle time transmission to adjust to the new dimmer settings. A summary of the different mitigation techniques for interframe and intra-frame flicker is provided in Table 79.

Table 79—Flicker mitigation for various modulation modes

Flicker mitigation	Data transmission (Intra-frame flicker)	Idle or RX periods (Interframe flicker)
OOK modulation	Dimmed OOK mode, RLL code	Idle/visibility patterns
VPPM modulation	VPPM guarantees no intra-frame flicker, RLL code	
CSK modulation	Constant average power across multiple light sources, scrambler, high optical clock rates (MHz)	

8.5.4 CSK color stabilization at the transmitter

This mode is optional and is used for PHY III devices. The control-loop model for the color-stabilization scheme is shown in Figure 116. The goal of this control mechanism is to stabilize the center of gravity of the CSK constellation diagram as described in 12.4. Visibility patterns, as described in 8.5.1.2, are sent from the tri-band TX of PHY III to a tri-band RX. An optional back link is used to relay these signals back to the tri-

band TX, where they are used to correct the LED driving currents in such a way that the center of gravity of the constellation diagram is moved back to its initial position.

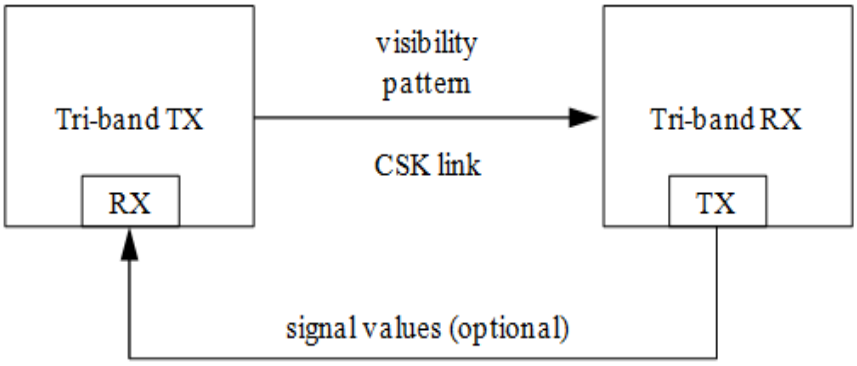


Figure 116—Control loop for a color-stabilized CSK link

Figure 117 shows the details of the color stabilization mechanism. Upon transmission of visibility patterns, the received signals after the D/As are relayed back to the CSK transmitter. In a color-stabilization module, which is out of scope of this standard, compensation factors c for each band are calculated.[8] Thereafter, all signal values outputted by the xy-to-(i,j,k) converter are multiplied by the respective compensation factors.

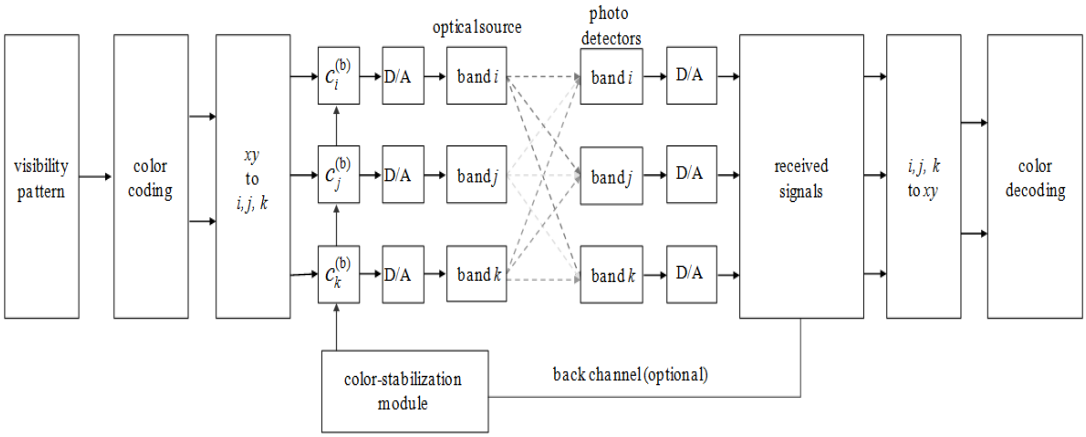


Figure 117—Color stabilization link implementation

In the color-stabilization mode, the visibility patterns to be used are in-band idle patterns with 100% visibility (as described in 8.5.1.2). The xy values of the emitted light coincides with the color chosen for the visibility pattern phase. The length of the visibility pattern in the CVD frame, as described in 4.4.3.1.2, is chosen so that thermal equilibrium in all band emitters is reached before sending the next CVD frame. The received signal (see Figure 117) is only acquired for the last sent bit of the last visibility pattern or an average over a suitable number of last bits.

[8]The calculation of the compensation factors is outside the scope of this standard. Examples for such calculations can be found elsewhere in the literature (Walewski [B14]).

8.6 PPDU format

For convenience, the PPDU frame structure is presented so that the left most field as written in this standard shall be transmitted or received first. All multiple octet fields shall be transmitted or received least significant octet first and each octet shall be transmitted or received least significant bit (LSB) first. The same transmission order should apply to data fields transferred between the PHY layer and MAC sublayer. The PPDU frame structure shall be formatted as illustrated in Figure 118.

Preamble (see 8.6.1)	**PHY header (see 8.6.2)**	**HCS (see 8.6.3)**	**Optional fields (see 8.6.4)**	**PSDU (see 8.6.5)**
SHR	PHR			PHY payload

Figure 118—Format of the PPDU

8.6.1 Preamble field

The preamble field is used by the transceiver to obtain optical clock synchronization with an incoming message. The standard defines one fast locking pattern (FLP) followed by choice of four topology dependent patterns (TDPs) for the purposes of distinguishing different PHY topologies. The MAC shall select the optical clock rate for communication during the clock-rate selection process as defined in 6.5. The preamble shall be sent at a clock rate chosen by the TX and supported by the RX. The preamble is a time domain sequence and does not have any channel coding or line coding.

The preamble first starts with a FLP of at least 64 alternate ones and zeros. The FLP is fixed to start as a "1010…" pattern i.e., it ends with a '0'. This maximum transition sequence is used to lock the CDR circuit. The fast locking pattern length shall not exceed the maximum as shown in Figure 119. After the fast locking pattern, four repetitions of one of four TDPs (defined in Figure 120) shall be sent. The TDP shall be 15 bits in length and the TDP shall be inverted every other repetition to provide DC balance.

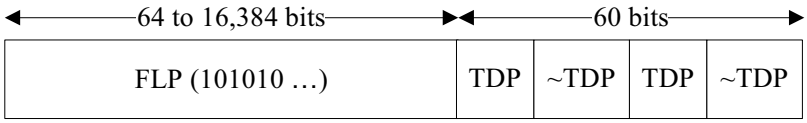


Figure 119—Preamble transmission

P1 : 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0
P2 : 0 0 1 0 1 1 1 0 1 1 1 1 1 1 0
P3 : 1 0 0 1 1 0 0 0 0 0 1 0 0 1 1
P4 : 0 1 0 0 0 0 1 1 0 1 0 0 1 0 1

Figure 120—TDPs for various topologies

The preambles shall be transmitted using an OOK modulation. The preamble field for single data mode and packed data mode shall be formatted as illustrated in Figure 119. For PHY III, all the three light sources shall transmit the same preamble pattern simultaneously in the supported frequency bands within the error tolerance specified in 8.3.3.

For the burst mode transmission, the FLP shall be included only for the first frame. Subsequent frames shall not include the FLP in the burst mode since the receiver is already synchronized to the transmitter. This reduces the preamble length by at least half and provides higher throughput at the MAC layer. The preamble field for burst data mode shall be formatted as illustrated in Figure 121.

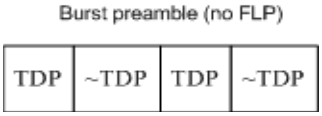


Figure 121—Burst preamble transmission

The TDP used for a specific a topology is defined in Table 80. The topologies are given in 4.2.

Table 80—TDP assignments for various topologies

TDP	Topology
P1	Topology independent (visibility)
P2	Peer-to-peer
P3	Star
P4	Broadcast

The same preamble sequences shall be used for all PHY types. The number of repetitions of the FLP can be extended by the MAC during idle time or for different operating modes for better synchronization or to provide visibility or image array receiver based device discovery.

8.6.2 PHY header

The PHY header, as shown in Table 81, shall be transmitted with an OOK modulation. For PHY III, all light sources shall transmit the same header contents simultaneously within the error tolerance specified in 8.3.3 and the band plan ID field shall be set to be the band plan ID of the lowest wavelength. The MAC shall select the optical clock rate for communication during the clock-rate selection process, as defined in 6.5. The PHY header shall be sent at the lowest data rate for the chosen optical clock rate. The clock rate does not change throughout the frame between the preamble, header, and payload. If the dimmed OOK extension bit is set in the PHY header for dimming support, additional fields are transmitted after the PHY header as shown in Table 82.

Table 81—PHY header

PHY header fields	Bit-width	Explanation on usage
Burst mode	1	Reduce preamble and IFS
Channel number	3	Band plan ID
MCS ID	6	Provide information about PHY type and data rate
PSDU length	16	Length up to *aMaxPHYFrameSize*
Dimmed OOK extension	1	Information on compensation time, resync, and length of sub-frame
Reserved fields	5	Future use

Table 82—Dimmed OOK extension

Extension fields	Bit-width	Explanation on usage
Compensation length	10	Compensation length in optical clocks
Resync length	4	Number of resync optical clocks
Subframe length	10	Length of subframe in optical clocks
OFCS	8	Optional field check sequence

8.6.2.1 Burst mode

The burst mode bit indicates that the next frame following the current frame is part of the burst mode. Refer to 5.2.2.2 for more detailed information.

8.6.2.2 Channel number

The channel number indicates the code used from Table 76. The channel number field for PHY III shall be the band plan ID of the lowest wavelength. Refer to 8.3.1 for more detailed information.

8.6.2.3 MCS ID

The modulation and coding scheme (MCS) ID shall be indicated in the PHY header based on Table 83.

8.6.2.4 Length of PSDU field

The PSDU length field specifies the total number of octets contained in the PSDU. It is a value between 0 and *aMaxPHYFrameSize* as shown in 9.5.1.

Table 83—MCS ID

	MCS indication	PHY	Data rate	Unit
0	000000	I	11.67	kb/s
1	000001		24.44	
2	000010		48.89	
3	000011		73.3	
4	000100		100	
5	000101		35.56	
6	000110		71.11	
7	000111		124.4	
8	001000		266.6	
16	010000	II	1.25	Mb/s
17	010001		2	
18	010010		2.5	
19	010011		4	
20	010100		5	
21	010101		6	
22	010110		9.6	
23	010111		12	
24	011000		19.2	
25	011001		24	
26	011010		38.4	
27	011011		48	
28	011100		76.8	
29	011101		96	
32	100000	III	12	Mb/s
33	100001		18	
34	100010		24	
35	100011		36	
36	100100		48	
37	100101		72	
38	100110		96	
others		reserved		

8.6.2.5 Dimmed OOK extension

The dimmed OOK bit shall be set to one when supporting dimming while using OOK modulation. The dimmed OOK bit shall be set when the MAC PIB attribute, *macUseDimmedOOKmode*, as defined in Table 60, indicates the dimmed OOK mode usage. The dimmed OOK extension bit indicates that more optional fields are present at the end of the header. These fields are described in 8.6.4.2, 8.6.4.3, 8.6.4.4, and 8.6.4.5.

8.6.3 Header check sequence (HCS)

The PHY header shall be protected with a 2 octet CRC-16 HCS. A schematic of the CRC processing used for HCS calculation is shown in Annex C. The HCS bits shall be processed in the transmit order. The registers shall be initialized to all ones.

8.6.4 Optional fields

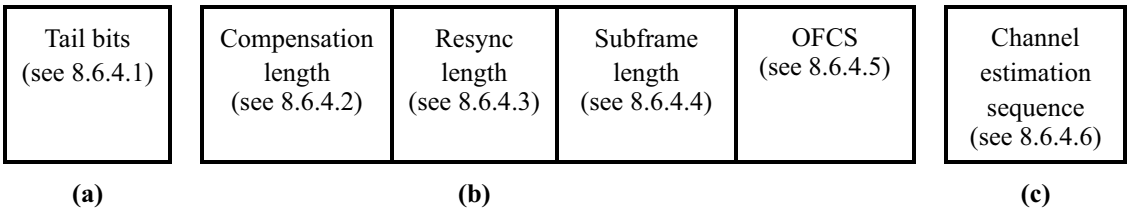


Figure 122—PPDU option fields

The optional fields shall be formatted as shown in Figure 122. The optional fields in Figure 122 (a) shall be transmitted only when PHY I is used with an optical clock of 200 kHz based on the MCS ID chosen in the PHR. The optional fields in Figure 122 (b) shall be transmitted after the tail bits only if the dimmed OOK bit is set in the PHR. The optional field in Figure 122 (c) shall be transmitted only if PHY III is selected based on the MCS ID chosen in the PHR. The dimmed OOK mode shall not be used with PHY III. i.e., the optional fields (b) and (c) shall never be used simultaneously. Optional fields (a) and (c) shall also never be transmitted simultaneously since they correspond to different PHY types.

8.6.4.1 Tail bits

Six tail bits of zeros shall be added after the HCS when PHY I is used with an optical clock rate of 200 kHz.

8.6.4.2 Compensation length

The compensation length has a 10-bit value, which indicates the number of compensation symbols at the optical clock rate. The values of these compensation symbols are user defined. When used, this field shall be set to a value between 0 to 1023.

8.6.4.3 Resync length

The resync length has a 4-bit value, which indicates the number of resync symbols at the optical clock rate. The resync pattern used is the same as the FLP. When used, this field shall be set to a value from 0 to 15, with a default value of 15.

8.6.4.4 Subframe length and generation

The subframe length has a 10-bit value, which indicates the number of uncoded data bits in the subframe. When used, this field shall be set to a value of 0 to 1023. The subframes shall be generated at the transmitter after the FCS has be determined and the FEC has been applied. The FEC and FCS shall not include the compensation symbols and the resync symbols. All subframes shall have the same length except for the last subframe, which may be truncated to meet the frame length.

8.6.4.5 Optional field check sequence generation

The PPDU optional field check sequence (OFCS) value is calculated across the compensation length, resync length and subframe length fields (as shown in Figure 122a) and inserted into the OFCS field.

The OFCS field shall be an 8-bit sequence (ITU-T I.432.1). It shall be the remainder of the division (modulo 2) by the generator polynomial $x^8 + x^2 + x + 1$ of the product x^8 multiplied by the content of the header excluding the OFCS field.

The initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division of the header, excluding the OFCS field, by the generator polynomial. The resulting remainder is the 8-bit OFCS.

8.6.4.6 Channel estimation sequence

The channel estimation sequences are three optional 8-bit sequences and are used only for PHY III operation. The information about PHY III is obtained after decoding the PHY header. The channel estimation sequence details are discussed in 12.9.

8.6.5 PSDU field

The PSDU field has a variable length and carries the data of the PHY frame. The FCS is appended if the PSDU has a non-zero byte payload. Six tail bits of zeros are attached to end of the PSDU, if PHY I is used with data rates of 11.67 kb/s, 24.44 kb/s, or 48.89 kb/s. The structure of the PSDU field is as shown in Figure 123.

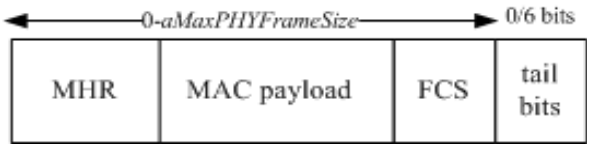


Figure 123—PSDU field structure

The complete PPDU format for the PHY is shown in Figure 124.

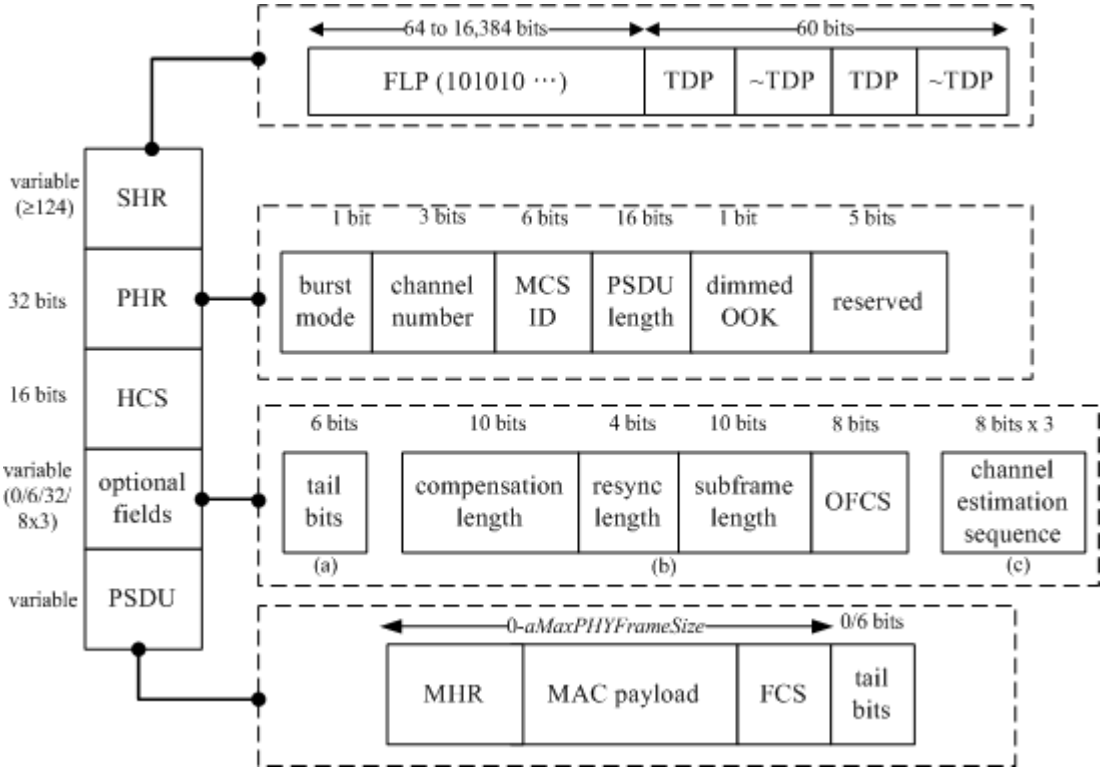


Figure 124—PPDU structure

9. PHY service specifications

9.1 Overview

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

Figure 125 depicts the components and interfaces of the PHY.

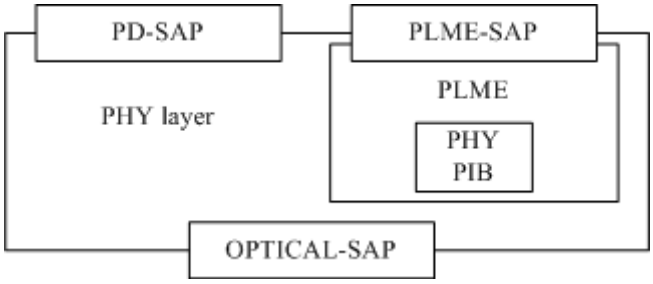


Figure 125—PHY layer service access points

The PHY provides two services, accessed through two SAPs: the PHY data service, accessed through the PHY data SAP (PD-SAP), and the PHY management service, accessed through the PLME's SAP (PLMESAP). The optical SAP provides an interface between the PHY layer and the optical channel and is not specified in this standard. Any required light source drivers are considered to be part of the optical channel.

9.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME or the DME and the PLME. Table 84 lists the primitives supported by the PLME-SAP. These primitives are discussed in the subclauses referenced in Table 84.

Table 84—PLME-SAP primitives

PLME-SAP primitive	Request	Confirm
PLME-CCA	9.2.1	9.2.2
PLME-GET	9.2.3	9.2.4
PLME-SET	9.2.5	9.2.6
PLME-SET-TRX-STATE	9.2.7	9.2.8
PLME-SWITCH	9.2.9	9.2.10

9.2.1 PLME-CCA.request

The PLME-CCA.request primitive requests that the PLME perform a CCA as defined in 8.3.9.

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

PLME-CCA.request ()

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

9.2.1.1 When generated

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

9.2.1.2 Effect on receipt

If the receiver is enabled on receipt of the PLME-CCA.request primitive, the PLME will cause the PHY to perform a CCA.

9.2.2 PLME-CCA.confirm

The PLME-CCA.confirm primitive reports the results of a CCA.

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

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

Table 85 specifies the parameters for the PLME-CCA.confirm primitive.

Table 85—PLME-CCA.confirm parameters

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

9.2.2.1 When generated

The PLME-CCA.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-CCA.request primitive. When the PHY has completed the CCA, the PLME will issue the PLME-CCA.confirm primitive with a status of either BUSY or IDLE, depending on the result of the CCA.

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

9.2.2.2 Effect on receipt

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

9.2.3 PLME-GET.request

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

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

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

Table 86 specifies the parameters for the PLME-GET.request primitive.

Table 86—PLME-GET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	As defined in Table 100	The identifier of the PHY PIB attribute to get.

9.2.3.1 Appropriate usage

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

9.2.3.2 Effect on receipt

On receipt of the PLME-GET.request primitive, the PLME will attempt to retrieve the requested PHY PIB attribute from its database.

9.2.4 PLME-GET.confirm

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

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

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

Table 87 specifies the parameters for the PLME-GET.confirm primitive.

Table 87—PLME-GET.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE	The result of the request for PHY PIB attribute information.
PIBAttribute	Enumeration	As defined in Table 100	The identifier of the PHY PIB attribute to get.
PIBAttributeValue	Various	Attribute specific	The value of the indicated PHY PIB attribute to get.

9.2.4.1 When generated

The PLME-GET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-GET.request primitive. If the identifier of the PIB attribute is not found in the database, the PLME will issue the PLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

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

9.2.4.2 Effect on receipt

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

9.2.5 PLME-SET.request

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

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

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

Table 88 specifies the parameters for the PLME-SET.request primitive.

Table 88—PLME-SET.request parameters

Name	Type	Valid range	Description
PIBAttribute	Enumeration	As defined in Table 100	The identifier of the PIB attribute to set.
PIBAttributeValue	Various	Attribute specific, as defined in Table 100	The value of the indicated PIB attribute to set.

9.2.5.1 When generated

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

9.2.5.2 Effect on receipt

On receipt of the PLME-SET.request primitive, the PLME will attempt to write the given value to the indicated PHY PIB attribute in its database.

9.2.6 PLME-SET.confirm

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

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

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

Table 89 specifies the parameters for the PLME-SET.confirm primitive.

Table 89—PLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER	The status of the attempt to set the requested PIB attribute.
PIBAttribute	Enumeration	As defined in Table 100	The identifier of the PIB attribute being confirmed.

9.2.6.1 When generated

The PLME-SET.confirm primitive is generated by the PLME and issued to its MLME in response to a PLME-SET.request primitive.

If the PIBAttribute parameter specifies an attribute that is not found in the database, as shown in Table 100, the PLME will issue the PLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttibuteValue parameter specifies a value that is out of the valid range for the given attribute, the PLME will issue the PLME-SET.confirm primitive with a status of INVALID_PARAMETER.

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

9.2.6.2 Effect on receipt

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

9.2.7 PLME-SET-TRX-STATE.request

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

a) Transceiver disabled (TRX_OFF)
b) Transmitter enabled (TX_ON)
c) Receiver enabled (RX_ON)

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

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

Table 90 specifies the parameters for the PLME-SET-TRX-STATE.request primitive.

Table 90—PLME-SET-TRX-STATE.request parameters

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

9.2.7.1 When generated

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

9.2.7.2 Effect on receipt

On receipt of the PLME-SET-TRX-STATE.request primitive, the PLME will cause the PHY to attempt to change to the requested state.

9.2.8 PLME-SET-TRX-STATE.confirm

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

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

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

Table 91 specifies the parameters for the PLME-SET-TRX-STATE.confirm primitive.

Table 91—PLME-SET-TRX-STATE.confirm parameters

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

9.2.8.1 When generated

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

9.2.8.2 Effect on receipt

On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MLME is notified of the result of its request to change the internal operating state of the transceiver.

If the state change is accepted, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status of SUCCESS. If this primitive requests a state that the transceiver is already configured, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status indicating the current state, i.e., RX_ON, TRX_OFF, or TX_ON. If this primitive is issued with RX_ON or TRX_OFF argument and the PHY is busy transmitting a PPDU, the PHY will issue the PLME-SET-TRXSTATE.confirm primitive with a status BUSY_TX and defer the state change until the end of transmission. If this primitive is issued with TX_ON or TRX_OFF argument and the PHY is in RX_ON state and has already received a valid preamble, the PHY will issue the PLME-SET-TRX-STATE.confirm primitive with a status BUSY_RX and defer the state change until the end of reception of the PPDU. If this primitive is issued with FORCE_TRX_OFF, the PHY will cause the PHY to go the TRX_OFF state irrespective of the state the PHY is in.

9.2.9 PLME-SWITCH.request

The PLME-SWITCH.request primitive request is used by the DME to request that the PHY entity select the switch to enable the appropriate cells in the SW-BIT-MAP. The semantics of the PLME-SWITCH.request primitive are as follows:

```
PLME-SWITCH.request     (
                        SW-BIT-MAP,
                        DIR
                        )
```

Table 92 specifies the parameters for the PLME-SET-TRX-STATE.request primitive.

Table 92—PLME-SWITCH.request parameters

Name	Type	Valid range	Description
SW-BIT-MAP	Vector of ‘*n*’ × ‘*m*’entries	Boolean	One bit for each optical source or photodetector and is dependent on the direction. Setting the k^{th} bit to a “1” brings the corresponding optical source or photodetector into the cell group. ‘*n*’ is the number of cells and ‘*m*’ is the number of distinct data streams from the PHY. The value of ‘*m*’ is three for PHY III.
DIR		Boolean	‘0’ is for TX and ‘1’ is for ‘RX’

9.2.9.1 When generated

The PLME-SWITCH.request primitive is generated by the DME and issued to its PLME when the current cell selection needs to be changed.

9.2.9.2 Effect on receipt

On receipt of the PLME-SWITCH.request primitive, the PLME will cause the PHY to attempt to change to the cell.

9.2.10 PLME-SWITCH.confirm

The PLME-SWITCH.confirm primitive reports the result of a request to change the currently operating cell.

The semantics of the PLME-SWITCH.confirm primitive are as follows:

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

Table 93 specifies the parameters for the PLME-SWITCH.confirm primitive.

Table 93—PLME-SWITCH.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS	The result of the request to change the cell.

9.2.10.1 When generated

The PLME-SWITCH.confirm primitive is generated by the PLME and issued to its DME after attempting to change the cell.

9.2.10.2 Effect on receipt

On receipt of the PLME-SWITCH.confirm primitive, the DME is notified of the result of its request to change the currently operating cell.

If the PHY switch is able to select the new cell, the PHY will issue the PLME-SWITCH.confirm primitive with a status of SUCCESS.

9.3 PHY data service

The PD-SAP supports the transport of MPDUs between a local MAC sublayer and a local PHY layer entity. Table 94 lists the primitives supported by the PD-SAP.

Table 94—PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	9.3.1	9.3.2	9.3.3

9.3.1 PD-DATA.request

The PD-DATA.request primitive requests the transfer of data from the MAC sublayer to form a PSDU at the local PHY entity.

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

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

Table 95 specifies the parameters for the PD-DATA.request primitive.

Table 95—PD-DATA.request parameters

Name	Type	Valid range	Description
psduLength	Unsigned Integer	0–*aMaxPHYFrameSize*	The number of octets in the PSDU to be transmitted by the PHY entity.
psdu	Set of octets		The set of octets forming the PSDU to be transmitted by the PHY entity.
bandplanID	Unsigned Integer	0–6	Color band channel of PSDU.

9.3.1.1 When generated

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

9.3.1.2 Effect on receipt

The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PPDU.

9.3.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of data from a local MAC sublayer entity.

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

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

Table 96 specifies the parameters for the PD-DATA.confirm primitive.

Table 96—PD-DATA.confirm parameters

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

9.3.2.1 When generated

The PD-DATA.confirm primitive is generated by the PHY entity and issued to its MAC sublayer entity in response to a PD-DATA.request primitive. Provided the transmitter is enabled (TX_ON state), the PHY will first construct a PPDU containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission, it will issue the PD-DATA.confirm primitive with a status of SUCCESS.

If the PD-DATA.request primitive is received while the receiver is enabled (RX_ON state) or if the transceiver is disabled (TRX_OFF state), the PHY entity will issue the PD-DATA.confirm primitive with a status of RX_ON or TRX_OFF, respectively.

9.3.2.2 Effect on receipt

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

9.3.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of data from the PHY to the local MAC sublayer entity.

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

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

Table 97 specifies the parameters for the PD-DATA.indication primitive.

9.3.3.1 When generated

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

9.3.3.2 Effect on receipt

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

Table 97—PD-DATA.indication parameters

Name	Type	Valid	Description
psduLength	Unsigned Integer	0–*aMaxPHYFrameSize*	The number of octets contained in the PSDU received by the PHY entity.
psdu	Set of octets	—	The set of octets forming the PSDU received by the PHY entity.
ppduLinkQuality	Unsigned Integer	0x00–0xff	Wavelength quality indication (WQI) value measured during reception of the PPDU as defined in 5.3.19.2.

9.4 PHY enumeration description

Table 98 shows a description of the PHY enumeration values defined in the PHY specification.

Table 98—PHY enumeration description

Enumeration	Value	Description
BUSY	0x00	The CCA attempt has detected a busy channel.
BUSY_RX	0x01	The transceiver is asked to change its state while receiving.
BUSY_TX	0x02	The transceiver is asked to change its state while transmitting.
FORCE_TRX_OFF	0x03	The transceiver is to be switched off.
IDLE	0x04	The CCA attempt has detected an idle channel.
INVALID_PARAMETER	0x05	A SET/GET request was issued with a parameter in the primitive that is out of the valid range.
RX_ON	0x06	The transceiver is in, or is to be configured into, the receiver enabled state.
SUCCESS	0x07	The request completed successfully.
TRX_OFF	0x08	The transceiver is in, or is to be configured into, the transceiver disabled state.
TX_ON	0x09	The transceiver is in, or is to be configured into, the transmitter enabled state.
UNSUPPORTED_ATTRIBUTE	0x0a	A SET/GET request was issued with the identifier of an attribute that is not supported.

9.5 PHY constants and PIB attributes

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

9.5.1 PHY constants

The constants that define the characteristics of the PHY are presented in Table 99. These constants are hardware dependent and shall not be changed during operation.

Table 99—PHY constants

Constant	Description	Value
aMaxPHYFrameSize	The maximum PSDU size (in octets) the PHY shall be able to receive.	1023 for PHY I, 65535 for PHY II, III
aTurnaroundTime-TX-RX	TX-to-RX maximum turnaround time (as defined in 8.3.5)	zero optical clock cycles
aTurnaroundTime-RX-TX	RX-to-TX maximum turnaround time (as defined in 8.3.6)	PHY I: ≤ 240 optical clock cycles, PHY II, III: ≤ 5120 optical clock cycles

9.5.2 PHY PIB attributes

The PHY PIB comprises the attributes required to manage the PHY of a device. Each of these attributes can be read or written using the PLME-GET.request and PLME-SET.request primitives, respectively. The attributes contained in the PHY PIB are presented in Table 100.

Table 100—PHY PIB attributes

Attribute	Identifier	Type	Range	Description
phyCurrentChannel	0x00	Integer	0–6	The wavelength used for all following transmissions and receptions (as defined in 8.3.1).
phyCCAMode	0x01	Octet	enumerated	b0=CCA mode 1 b1=CCA mode 2 b2=CCA mode 3 b3–b7=reserved The CCA modes are defined in 8.3.9.
phyDim	0x02	Integer	0–1000	0 is 0% or no visibility and 1000 is 100% visibility (full brightness).
phyUseExtendedMode	0x03	Integer	0–1	This attribute is set to a one to indicate that an extended preamble or visibility pattern is to be used. Otherwise, it is set to zero.

Table 100—PHY PIB attributes *(continued)*

Attribute	Identifier	Type	Range	Description
phyColorFunction	0x04	256 by 3 matrix of integer	The row index ranges from 0 to 255 and the elements range from 0 to 255.	A table with three columns per row. The first row is the index, the second and the third columns define the color.
phyBlinkingNotification-Frequency	0x05	Integer	0–10	The frequency of blinking notification: 0: 0.25 Hz 1: 0.5 Hz 2: 0.75 Hz 3: 1 Hz 4: 1.25 Hz 5: 1.5 Hz 6: 1.75 Hz 7: 2 Hz 8: 2.25 Hz 9: 2.5 Hz 10: 2.75 Hz

10. PHY I specifications

PHY I is targeted towards applications requiring low data rates as shown in Table 73. For PHY I, the PHY header shall be sent at 11.67 kb/s if the 200 kHz optical clock rate is selected or at 35.56 kb/s if the 400 kHz optical clock rate is selected. Support for 11.67 kb/s at 200 kHz optical clock is mandatory.

10.1 Reference modulator diagram

A reference implementation of the modulator is shown in Figure 126.

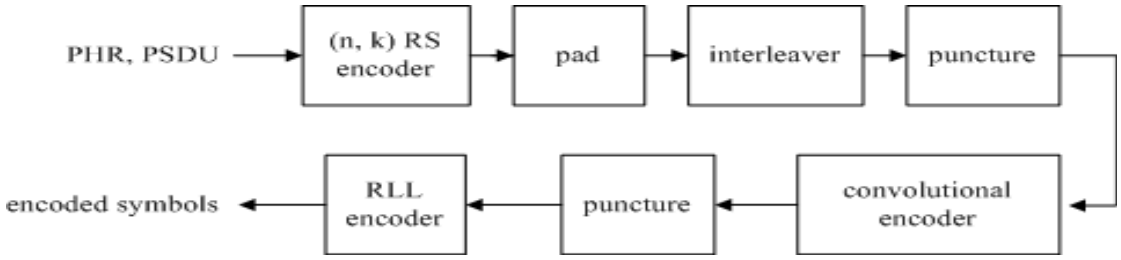


Figure 126—Reference modulator diagram for PHY I

For PHY I, concatenated coding is used with a combination of convolutional outer code and a RS inner code. The RS encoder output is padded with zeros to form an interleaver boundary. The padded zeros are then punctured (discarded) and the result is sent to the inner convolutional encoder. The PHR and PSDU parts of the frame are subject to the FEC for error protection. The PHR is encoded using parameters corresponding to the lowest data rate for the currently negotiated clock rate.

10.2 Outer forward error correction encoder

Systematic RS codes are used for the PHY I outer FEC with GF(16), generated by the polynomial x^4+x+1. The generators for the RS(n, k) codes for PHY I (see Table 73) are given in Table 101, where α is a primitive element in GF(16).

Table 101—Generator polynomials

(n,k)	g(x)
(15,11)	$x^4+\alpha^{13}x^3+\alpha^6x^2+\alpha^3x+\alpha^{10}$
(15,7)	$x^8+\alpha^{14}x^7+\alpha^2x^6+\alpha^4x^5+\alpha^2x^4+\alpha^{13}x^3+\alpha^5x^2+\alpha^{11}x^1+\alpha^6$
(15,4)	$x^{11}+\alpha^9x^{10}+\alpha^8x^9+\alpha^4x^8+\alpha^9x^7+\alpha^{13}x^6+\alpha^4x^5+\alpha^{12}x^4+\alpha^4x^3+\alpha^5x^2+\alpha^3x+\alpha^6$
(15,2)	$x^{13}+\alpha^3x^{12}+\alpha^8x^{11}+\alpha^9x^{10}+\alpha^2x^9+\alpha^4x^8+\alpha^{14}x^7+\alpha^6x^6+\alpha^{10}x^5+\alpha^7x^4+\alpha^{13}x^3+\alpha^{11}x^2+\alpha^5x+\alpha$

The Reed-Solomon code may be shortened for the last block if it does not meet the block size requirements. No zero padding is required for the RS code. A shortened RS code is used for frame sizes not matching code word boundaries via the following operation to minimize padding overhead.

Starting with a RS(n,k) code, one can get an RS(n-s, k-s) shortened code as follows:

a) Pad the *k-s* RS symbols with *s* zero RS symbols.
b) Encode using RS(n, k) encoder.
c) Delete the padded zeros (do not transmit them).
d) At the decoder, add the zeros, then decode.

10.3 Interleaving and puncturing block

A block interleaver is used as an interleaver between the inner convolutional code and the outer RS code as shown in Figure 127. The interleaver is of a fixed height *n* but has a flexible depth *D*, dependent on the frame size. The flexible depth of the interleaver and the puncturing block after the interleaver is used to minimize padding overhead.

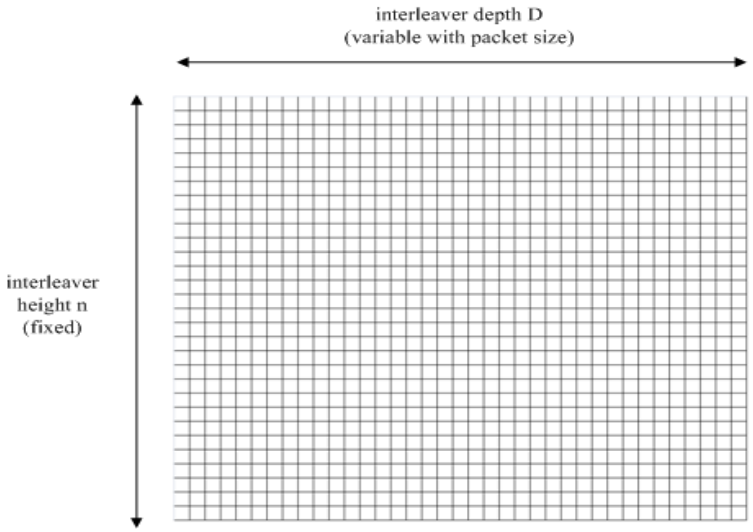


Figure 127—Interleaver for PHY I

The following parameters are used to describe the interleaver:

n: RS codeword length
k: Number of information data symbols in a RS codeword
q: Number of elements in the Galois field: GF(q)
L_{frame}: Input frame size in bytes
S_{frame}: Number of symbols at the input of the RS encoder
S: Number of symbols from the output of the shortened RS encoder
S_{block}: The size of the interleaver used
D: The interleaving depth
i: Ordered indices take the values 0, 1, …, S_{block}–1
$l(i)$: Interleaved indices
p: Number of zero RS symbols
t: Ordered indices take the values 0, 1, …, p
$z(t)$: Locations of the bits to be punctured at the output of the interleaver before transmission

The interleaver and the locations to be punctured are described by the equations shown in Equation (3).

$$
\begin{aligned}
S_{frame} &= \left\lceil \frac{L_{frame} * 8}{\log_2(q)} \right\rceil \\
S &= n * \left\lceil \frac{S_{frame}}{k} \right\rceil - \left(k - \left(S_{frame} \bmod k\right)\right) \\
D &= \left\lceil \frac{S}{n} \right\rceil \\
S_{block} &= n * D \\
p &= n - (S \bmod n) \\
l(i) &= (i \bmod D) * n + \left\lfloor \frac{i}{D} \right\rfloor ; \text{ for } i = 0,1,..,(S_{block} - 1) \\
z(t) &= (n - p + 1) * D + t * D - 1; \text{ for } t = 0,1,..,p-1
\end{aligned}
\tag{3}
$$

The length of the frame is communicated to the receiver in the header so that the receiver can adaptively adjust the interleaver based on the frame sizes. When the data rates corresponding to transmissions using the concatenated codes are used, the header shall also be interleaved according to procedure shown in Equation (3). Since the length of the header is fixed, the receiver can deinterleave the header without explicit transmission of the header length.

10.4 Inner forward error correction encoder

The inner code is a based on a rate-1/3 mother convolutional code of constraint length seven (K=7) with generator polynomial $g_0 = 133_8$; $g_1 = 171_8$; $g_2 = 165_8$, as shown in Figure 128.

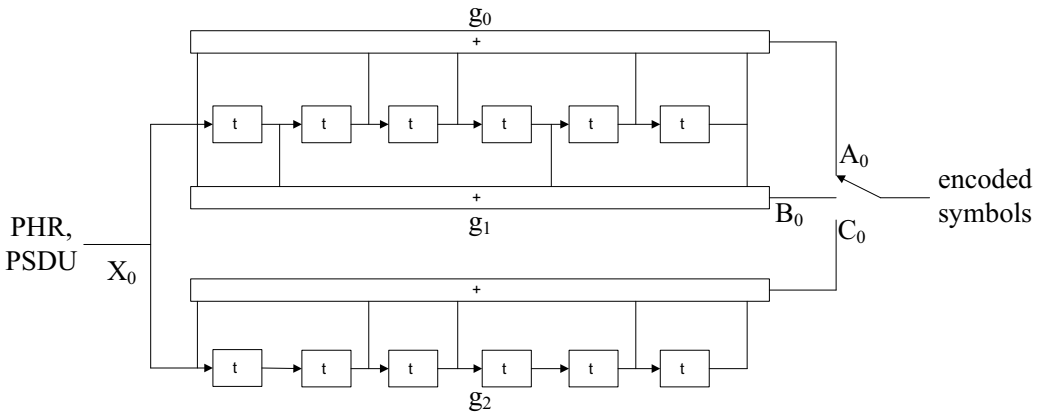


Figure 128—Rate-1/3 mother convolutional code with constraint length 7

Six tail bits of zeros shall be added at the end of the encoding in order to terminate the convolutional encoder to an all zeros state. The tail bit of zeros shall be applied to both the header and the payload when the inner convolutional code is used.

10.4.1 Rate-1/4 code

The rate-1/4 code is obtained by puncturing the rate-1/3 mother code to a rate-1/2 code, as shown in Figure 129, and then using a simple repetition code as shown in Figure 130.

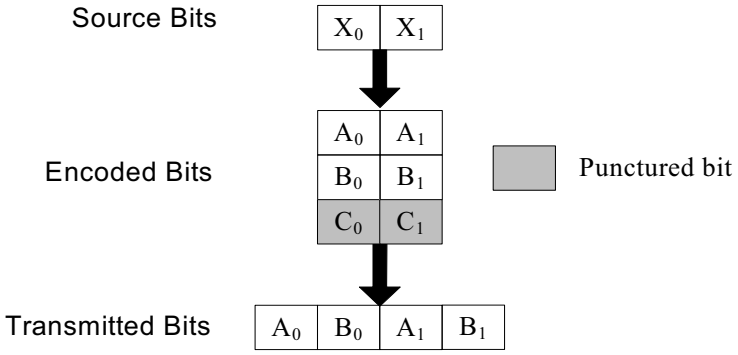


Figure 129—Puncturing pattern to obtain rate-1/2 code

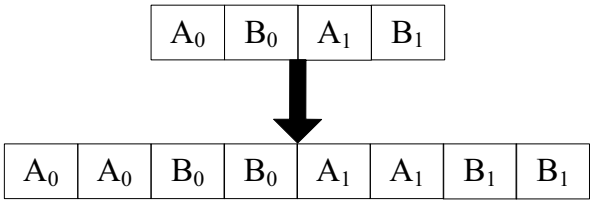


Figure 130—Repetition pattern used to obtain the effective rate-1/4 code

10.4.2 Rate-1/3 code

The rate-1/3 code is obtained by using the outputs of the rate-1/3 mother code shown in Figure 128.

10.4.3 Rate-2/3 code

The rate-2/3 code is obtained by puncturing the rate-1/3 mother code, as shown in Figure 131.

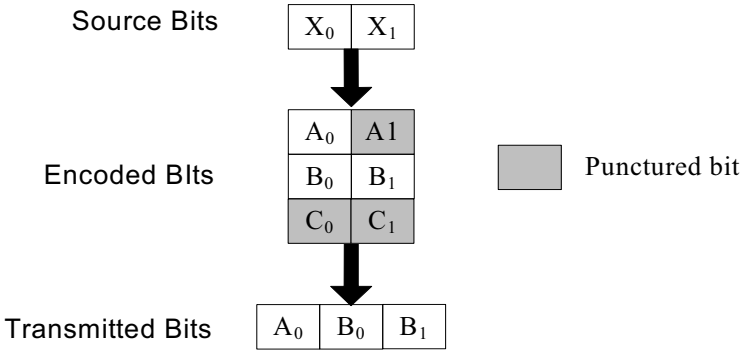


Figure 131—Puncturing pattern to obtain rate-2/3 code

10.5 Run-length limiting encoder

10.5.1 4B6B encoding for VPPM modes

All VPPM PHY I modes shall use 4B6B encoding. The 4B6B expands 4-bit to 6-bit encoded symbols with DC balance. The counts of 1 and 0 in every VPPM encoded symbol is always equal to 3. Table 102 defines the 4B6B code.

Table 102—Mapping input 4B to output 6B

4B (input)	6B (output)	Hex
0000	001110	0
0001	001101	1
0010	010011	2
0011	010110	3
0100	010101	4
0101	100011	5
0110	100110	6
0111	100101	7
1000	011001	8
1001	011010	9
1010	011100	A
1011	110001	B
1100	110010	C
1101	101001	D
1110	101010	E
1111	101100	F

The features of the 4B6B code are as follows:

a) Always 50% duty cycle during one encoded symbol
b) DC balanced run length limiting code
c) Error detection capability
d) Run length is limited to four
e) Allows reasonable clock recovery

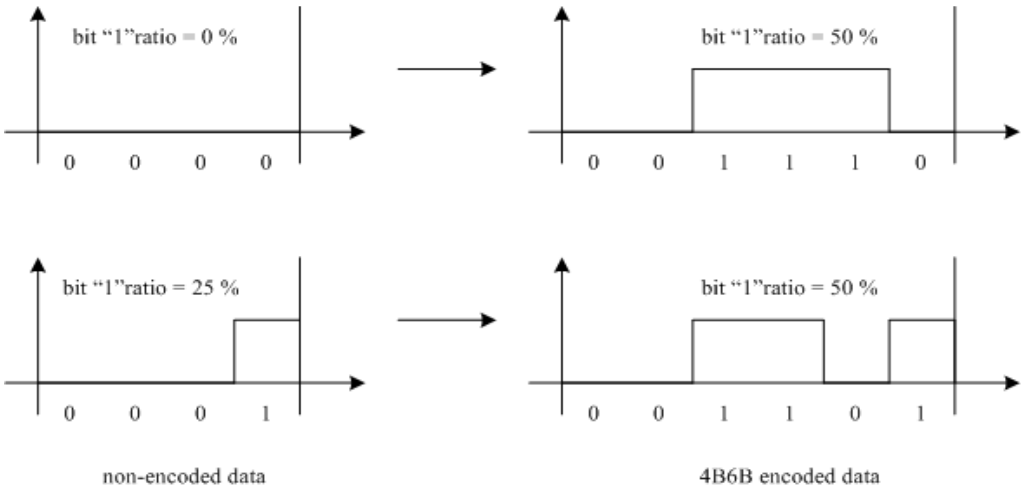


Figure 132—Illustrative comparison between non-encoded and 4B6B encoded symbols

10.5.2 Manchester encoding for OOK mode

All OOK PHY I modes shall use Manchester DC balancing encoding. The Manchester code expands each bit into an encoded 2-bit symbol as shown in Table 103.

Table 103—Manchester encoding

bit	Manchester symbol
0	01
1	10

10.6 Data mapping for VPPM

The data mapping for VPPM shall be defined as in Table 104. The physical value mapped from the logical data '0' has a transition from 'high' to 'low', and the physical value mapped from the logical data '1' has a transition from 'low' to 'high', as shown in Table 104. 'Low' and 'high' values are defined in 8.3.2. The variable d in Table 104 is the VPPM duty cycle, and it is assigned by the VPPM-mode dimming mechanism described in 8.5.2.3. It can be varied in steps of 0.1.

Table 104—Definition of data mapping for VPPM mode

Logical value	Physical value d is the VPPM duty cycle ($0.1 \le d \le 0.9$)	
0	High	$0 \le t < dT$
	Low	$dT \le t < T$
1	Low	$0 \le t < (1 - d)T$
	High	$(1 - d)T \le t < T$

11. PHY II specifications

PHY II is targeted towards applications requiring high data rates, as shown in Table 74. For PHY II, the PHY header shall be sent at one of the following data rates: 1.25 Mb/s, 2.5 Mb/s, 6 Mb/s, 12 Mb/s, 24 Mb/s, or 48 Mb/s, depending on the selected optical clock rate. Support for 1.25 Mb/s at an optical clock of 3.75 MHz is mandatory.

11.1 Reference modulator diagram

A reference implementation is in Figure 133. The PHR and PSDU parts of the frame are subject to the FEC for error protection. The PHR is encoded using parameters corresponding to the lowest data rate for the currently negotiated clock rate.

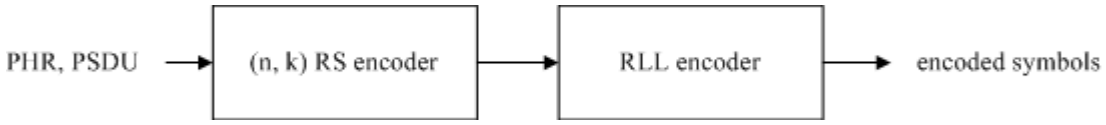


Figure 133—Reference modulator diagram for PHY II

11.2 Forward error correction encoder

A systematic Reed-Solomon code operating on GF(256) shall be used for PHY II to correct errors and increase the system reliability. The Reed-Solomon code may be shortened for the last block if it does not meet the block size requirements, as specified for PHY I in 10.2. No zero padding is required for the RS code.

The Reed-Solomon code is defined over GF(256) with a primitive polynomial $x^8+x^4+x^3+x^2+1$. The generator for the RS(160, 128) code and the RS(64, 32) code is given by

$$g(x) = x^{32}+\alpha^{11}x^{31}+\alpha^{8}x^{30}+\alpha^{109}x^{29}+\alpha^{194}x^{28}+\alpha^{254}x^{27}+\alpha^{173}x^{26}+\alpha^{11}x^{25}+\alpha^{75}x^{24}+\alpha^{218}x^{23}+\alpha^{148}x^{23}+\alpha^{149}x^{21}+\alpha^{44}x^{20}+\alpha^{0}x^{19}+\alpha^{137}x^{18}+\alpha^{104}x^{17}+\alpha^{43}x^{16}+\alpha^{137}x^{15}+\alpha^{203}x^{14}+\alpha^{99}x^{13}+\alpha^{176}x^{12}+\alpha^{59}x^{11}+\alpha^{91}x^{10}+\alpha^{194}x^{9}+\alpha^{84}x^{8}+\alpha^{53}x^{7}+\alpha^{248}x^{6}+\alpha^{107}x^{5}+\alpha^{80}x^{4}+\alpha^{28}x^{3}+\alpha^{215}x^{2}+\alpha^{251}x+\alpha^{18} \quad (4)$$

where α is a primitive element in GF(256).

For the VPPM modes using 4B6B encoding, the RS code word (d1,..., d8) from the GF(256) RS code is broken into 2 nibbles (d1,.., d4) and (d5,.., d8). These nibbles are sent LSB first to the 4B6B encoder as shown in Figure 134.

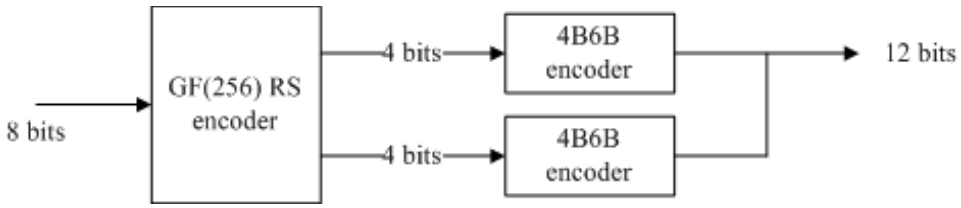


Figure 134—GF(256) RS encoder usage with 4B6B encoder

11.3 Run-length limiting encoder

All PHY II VPPM modes shall use 4B6B encoding as defined in 10.5.1. All OOK PHY II modes shall use 8B10B encoding as specified in ANSI/INCITS 373.

11.4 Data mapping for VPPM

All PHY II VPPM modes shall use data mapping as defined in 10.6.

12. PHY III specifications

The data rates supported by PHY III are shown in Table 75. For PHY III, the PHY header shall be sent at 12 Mb/s if the 12 MHz optical clock rate is selected or at 24 Mb/s if the 24 MHz optical clock rate is selected. Support for 12 Mb/s at 12 MHz is mandatory. PHY III devices shall utilize PHY II devices for device discovery. After all devices in the network are discovered and if all of them support PHY III, the coordinator can decide to switch to PHY III mode of operation. In addition, PHY III devices shall exchange their supported bands for CSK operation with the coordinator and the coordinator shall verify that the frequency bands supported in all PHY III devices in the network support reliable CSK communication. This is to ensure that transmission on two optical frequency bands of the transmitting device does not fall within one optical filter band of the receiving device for CSK operation, leading to communication errors during CSK operation.

12.1 Reference modulator diagram

Figure 135 shows the CSK system configuration for PHY III with light sources of three colors (bands i, j, and k). After scrambling and channel coding, data is transformed into *xy* values, according to the mapping rule on the *xy* color coordinates by the color coding block. The PHR and PSDU parts of the frame are subject to the FEC block for error protection. The PHR is encoded using parameters corresponding to the lowest data rate for the currently negotiated clock rate. The channel estimation sequence is transmitted after the PHR as shown in Figure 122b.

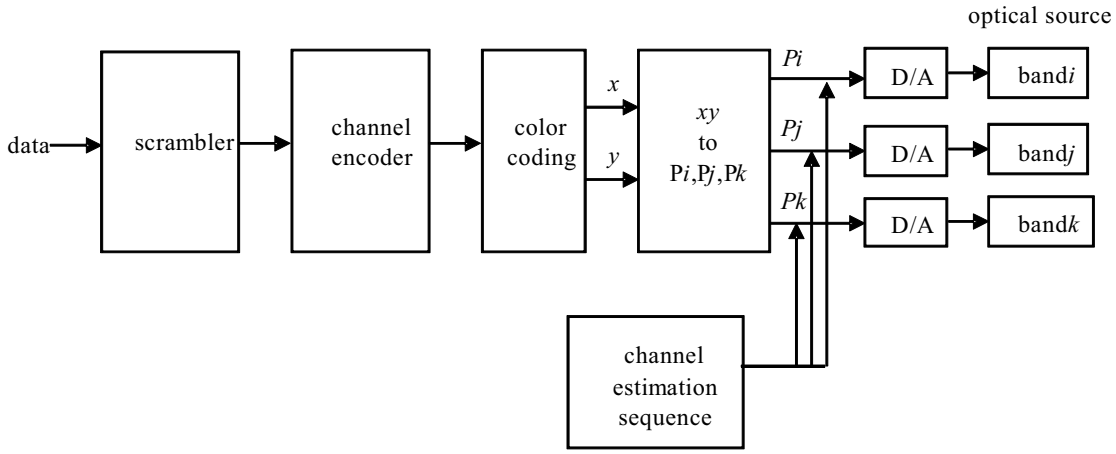


Figure 135—CSK system diagram for PHY III

12.2 Scrambler

A scrambler shall be used to ensure pseudo-random data for the PHY III. The scrambler shall be applied to the entire PSDU. In addition, the scrambler shall be initialized to a seed value dependent on the topology dependent pattern at the beginning of the PSDU. The polynomial generator, $g(D)$, for the pseudo-random binary sequence (PRBS) generator shall be: $g(D) = 1 + D^{14} + D^{15}$, where D is a single bit delay element. Using this generator polynomial, the corresponding PRBS, $x[n]$, is generated as in Equation (5).

$$x[n] = x[n - 14] \oplus x[n - 15],\ n = 0, 1, 2, \ldots \tag{5}$$

where "$\oplus$" denotes modulo-2 addition. The following sequence defines the initialization vector, x_{init}, which is specified by the parameter "seed value" in Table 105:

$$x_{init} = [x_i[-1]\ x_i[-2] \ldots x_i[-14] x_i[-15]], \ldots$$

where $x_i[-k]$ represents the binary initial value at the output of the k^{th} delay element. The scrambled data bits, v_m, are defined in Figure 136 and shall be calculated as:

$$v[m] = s[m] \oplus x[m],\ m = 0, 1, 2, \ldots$$

where $s[m]$ represents the non-scrambled data bits. The side-stream de-scrambler at the receiver shall be initialized with the same initialization vector, x_{init}, used in the transmitter scrambler. The initialization vector is determined from the TDP.

The 15-bit initialization vector or seed value shall correspond to the seed identifier as defined in Table 105, corresponding to the TDP pattern. The seed values shall be incremented in a roll-over fashion for each frame sent by the PHY. For example, if the seed value used is the seed corresponding to P3 in the first frame, the seed value corresponding to P4 is used in the second frame, seed value corresponding to P1 is used in the third frame and so on. All consecutive frames, including retransmissions, shall be sent with a different initial seed value.

Table 105—Scrambler seed selection

TDP	Seed Value *xinit* = *xi*[-1] *xi*[-2]… *xi*[-15]	PRBS Output First 16 bits *x*[0] *x*[1] … *x*[15]
P1	0011 1111 1111 111	0000 0000 0000 1000
P2	0111 1111 1111 111	0000 0000 0000 0100
P3	1011 1111 1111 111	0000 0000 0000 1110
P4	1111 1111 1111 111	0000 0000 0000 0010

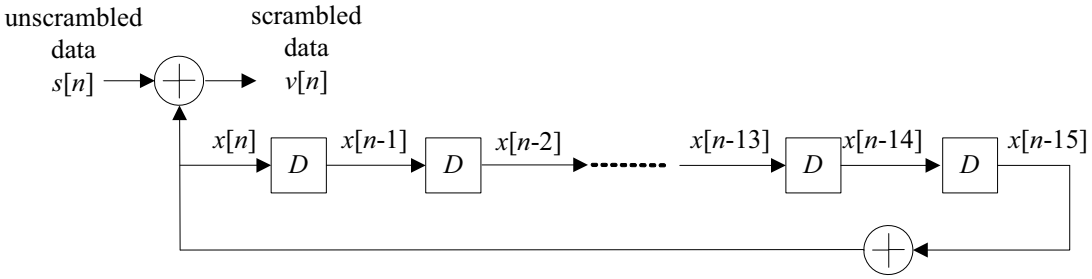


Figure 136—Scrambler block diagram

12.3 Channel encoder

When used, the channel encoding for PHY III is obtained using the ½ RS(64, 32) code as defined in 11.2.

12.4 CSK constellation overview

The CSK signal is generated by using three color light sources out of the seven color bands that are defined in 8.3.1. The three vertices of the CSK constellation triangle are decided by the center wave length of the three color bands on *xy* color coordinates. It is possible that some of the optical sources would have a spectral peak at a different frequency than the center of the bandplan. It is also possible that the spectrum of the optical source would be distributed among over multiple frequency bands. Implementers of CSK systems can select the color band based on the center wave length of the actual optical source. Table 106 shows the *xy* color coordinates values assuming the optical source is chosen with the spectral peak occurring at the center of each of the seven color bands. The color calibration function in 12.9 can compensate color coordinate errors caused by the drifting of the optical source characteristics and cancel any interference between the three colors.

Figure 137 shows the center of color bands of Table 106 on *xy* color coordinates.

Table 106—*xy* color coordinates

Band (nm)	Code	Center (nm)	(*x*, *y*)
380–478	000	429	(0.169, 0.007)
478–540	001	509	(0.011, 0.733)
540–588	010	564	(0.402, 0.597)
588–633	011	611	(0.669, 0.331)
633–679	100	656	(0.729, 0.271)
679–726	101	703	(0.734, 0.265)
726–780	110	753	(0.734, 0.265)

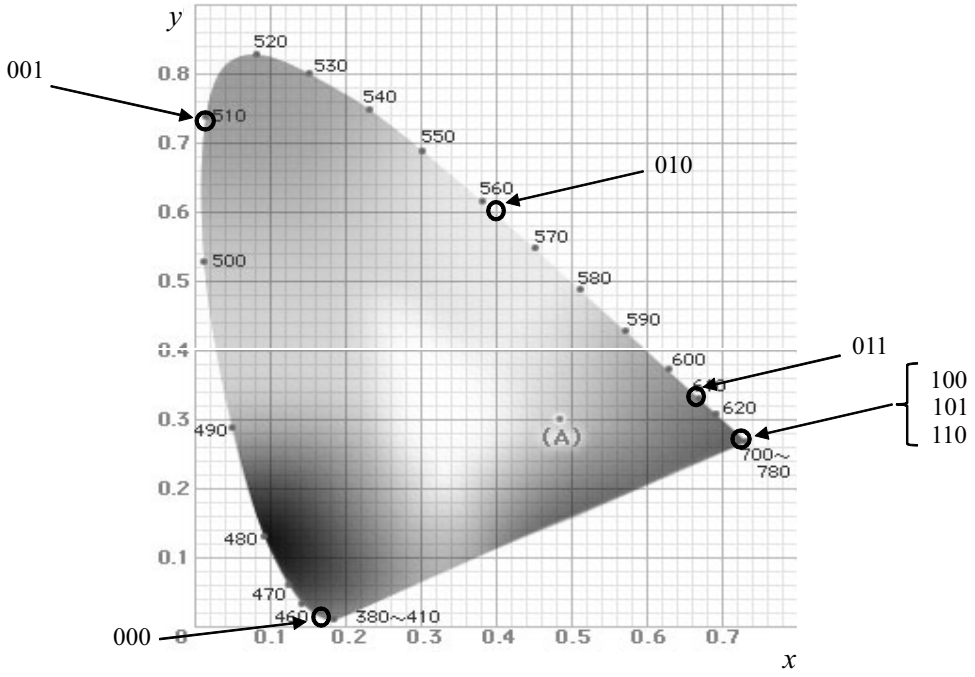


Figure 137—Center of color bands on *xy* color coordinates

12.5 CSK constellation design rules

12.5.1 Design rule for 4-CSK

4-CSK symbol points are defined by the design rule in Figure 138. Points I, J, and K show the center of the three color bands on *xy* color coordinates in Table 106. In Figure 138, *x*-axis and *y*-axis are the relative value. S0 to S3 are four symbol points of 4-CSK. S1, S2, and S3 are three vertices of the triangle IJK. S0 is

the centroid of the triangle IJK. The absolute values for 4-CSK for multiple combinations of the optical sources assuming the spectral peak of the optical source is at the center of the bandplan can be obtained in Yokoi, et al. [B17].

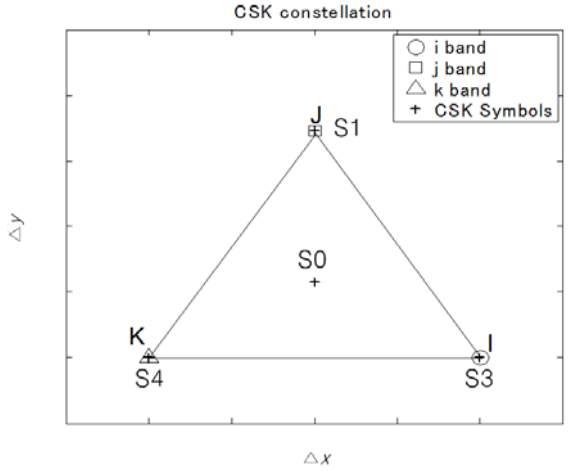


Figure 138—Constellation design rule for 4-CSK

12.5.2 Design rule for 8-CSK

8-CSK symbol points are defined by the design rule in Figure 139. Points I, J, and K show the center of the three color bands on *xy* color coordinates in Table 106. S0 to S7 are 8 symbol points of 8-CSK. S0, S4, and S7 are three vertices of the triangle IJK. S1 and S2 are points that divide side JK and side JI in the ratio 1:2. Point B and C are midpoints of the line JI and line JK. S6 is a midpoint of the line KI. Point A is the centroid of the triangle B-S6-I. Point D is the centroid of the triangle C-K-S6. S3 is a point that divides line AB in the ratio 1:2. S5 is a point that divides line DC in the ratio 1:2.

The absolute values for 8-CSK for multiple combinations of the optical sources assuming the spectral peak of the optical source is at the center of the bandplan can be obtained in Yokoi, et al. [B17].

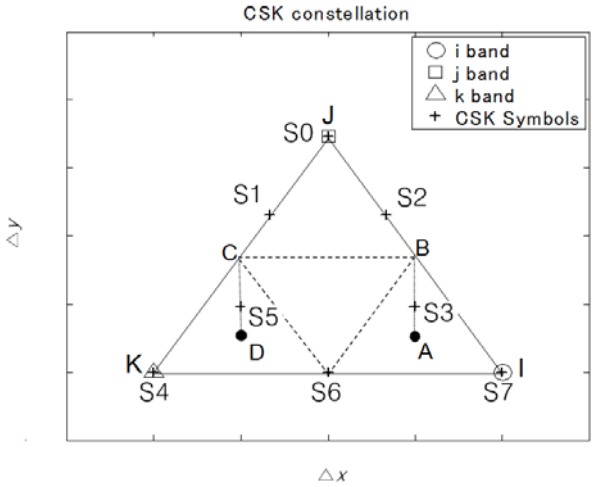


Figure 139—Constellation design rule for 8-CSK

12.5.3 Design rule for 16-CSK

16-CSK symbol points are defined by the design rule in Figure 140. Points I, J, and K show the center of the three color bands on *xy* color coordinates in Table 106. S0 to S15 are 16 symbol points of 16-CSK. S5, S10, and S15 are three vertices of the triangle IJK. S2 and S8 are points that divide side JK in one third. S3 and S12 are points that divide side JI in one third. S11 and S14 are points that divide side KI in one third. S0 is the centroid of the triangle IJK. S1, S4, S6, S7, S9, and S13 are the centroids of each of the smaller triangles. The absolute values for 16-CSK for multiple combinations of the optical sources assuming the spectral peak of the optical source is at the center of the bandplan can be obtained in Yokoi, et al. [B17].

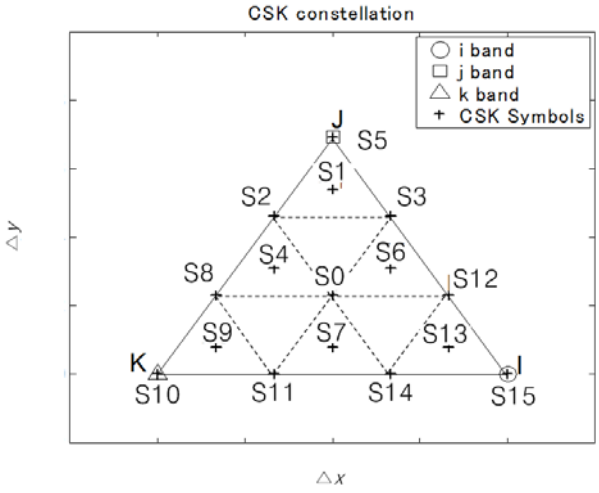


Figure 140—Constellation design rule for 16-CSK

12.6 Data mapping for CSK

4-CSK data mapping is shown in Figure 141. Two bits are assigned per symbol.

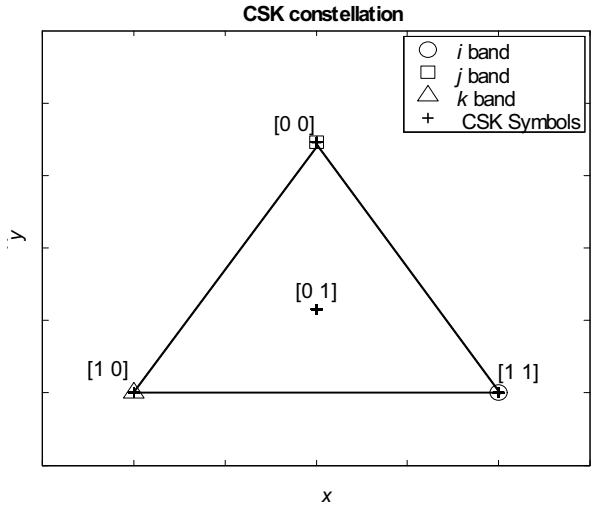


Figure 141—Data mapping for 4-CSK

8-CSK data mapping is shown in Figure 142. Three bits are assigned per symbol.

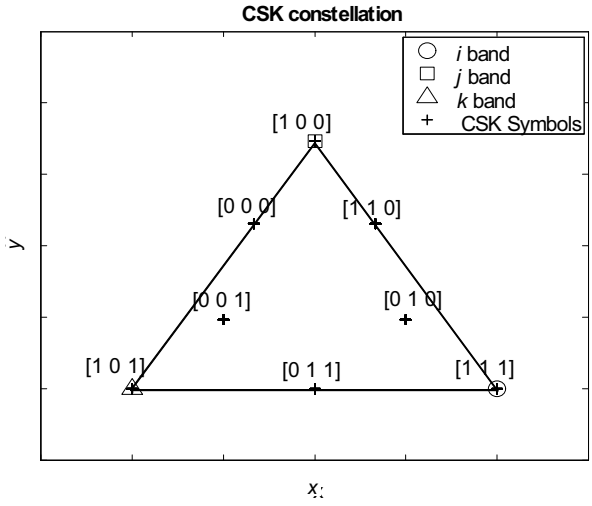


Figure 142—Data mapping for 8-CSK

16-CSK data mapping is shown in Figure 143. Four bits are assigned per symbol.

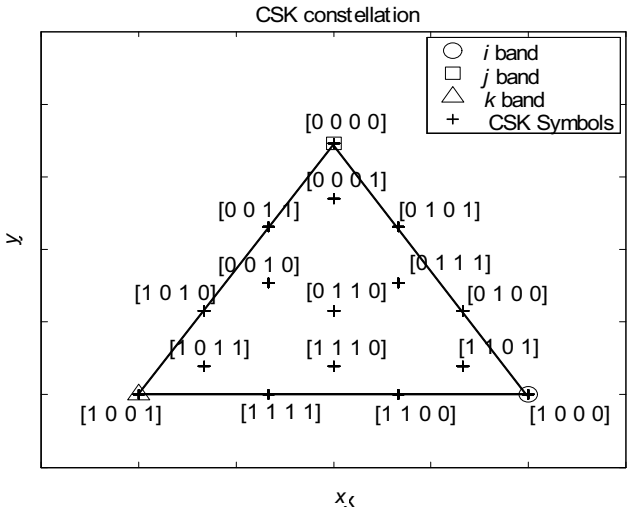


Figure 143—Data mapping for 16-CSK

12.7 Valid color band combinations

The CSK constellation is decided by the combination of the three color bands. Certain combinations that cannot make a triangle on the *xy* color coordinates are excluded, such as (110-101-100) or (100-011-010). Table 107 shows valid color band combinations that can make triangles for CSK constellations.

Table 107—Valid color band combinations for CSK

	Band *i*	**Band *j***	**Band *k***
1	110	010	000
2	110	001	000
3	101	010	000
4	101	001	000
5	100	010	000
6	100	001	000
7	011	010	000
8	011	001	000
9	010	001	000

Figure 144 shows an example of the CSK constellation triangle when color codes (110, 010, 000) are used.

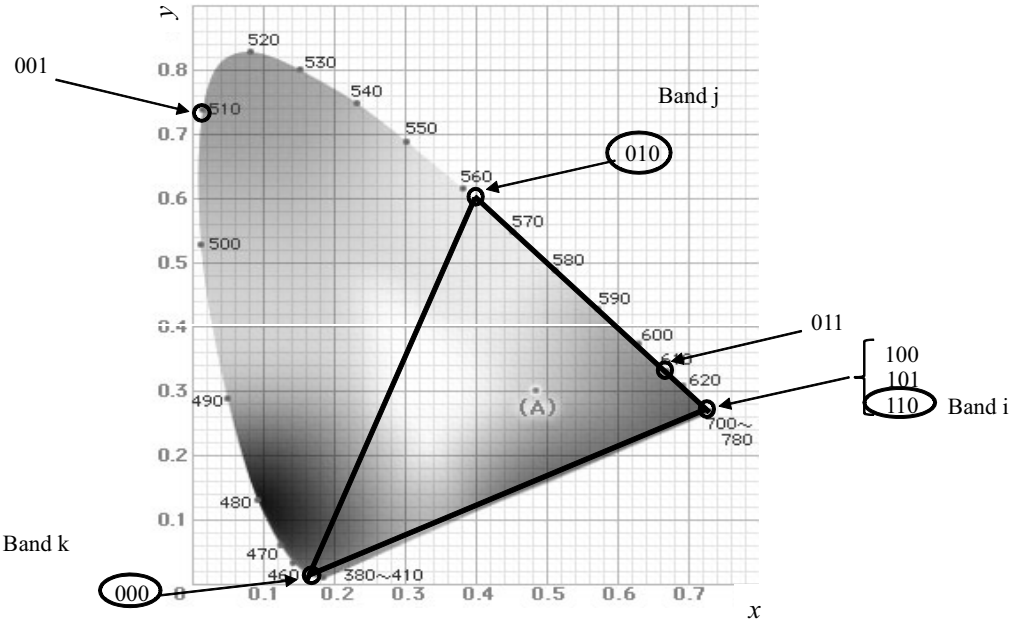


Figure 144—Valid CSK constellation example for codes (110, 010, 000)

Table 108 shows color band combination and the *xy* coordinate values when color codes (110, 010, 000) are used. Figure 145 shows the CSK constellation points when color codes (110, 010, 000) are used.

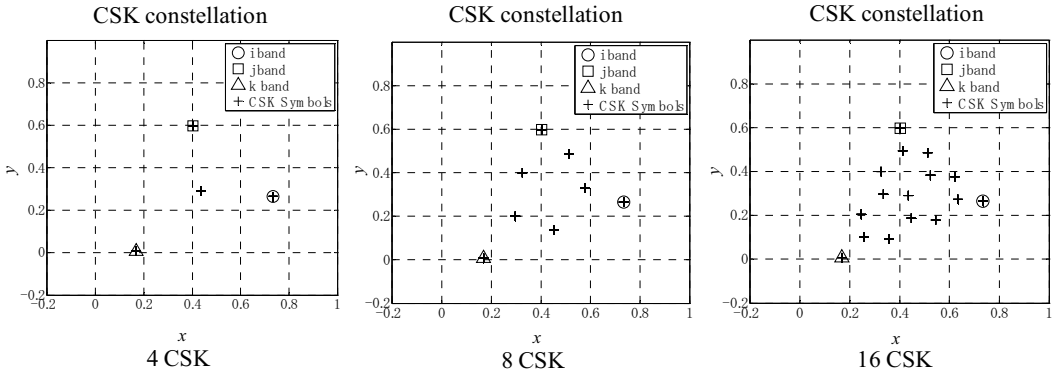


Figure 145—CSK constellation made by color band combinations

Table 108—Color band combination example for (110, 010, 000)

Center of band (*x*,*y*)	***xy* coordinate values of symbols**		
	4-CSK [data] – (*xp*,*yp*)	**8-CSK** [data] – (*xp*,*yp*)	**16-CSK** [data] – (*xp*,*yp*)
(0.734 0.265) (0.402 0.597) (0.169 0.007)	[0 0] – (0.402 0.597) [0 1] – (0.435 0.290) [1 0] – (0.169 0.007) [1 1] – (0.734 0.265)	[0 0 0] – (0.324 0.400) [0 0 1] – (0.297 0.200) [0 1 0] – (0.579 0.329) [0 1 1] – (0.452 0.136) [1 0 0] – (0.402 0.597) [1 0 1] – (0.169 0.007) [1 1 0] – (0.513 0.486) [1 1 1] – (0.734 0.265)	[0 0 0 0] – (0.402 0.597) [0 0 0 1] – (0.413 0.495) [0 0 1 0] – (0.335 0.298) [0 0 1 1] – (0.324 0.400) [0 1 0 0] – (0.623 0.376) [0 1 0 1] – (0.513 0.486) [0 1 1 0] – (0.435 0.290) [0 1 1 1] – (0.524 0.384) [1 0 0 0] – (0.734 0.265) [1 0 0 1] – (0.169 0.007) [1 0 1 0] – (0.247 0.204) [1 0 1 1] – (0.258 0.101) [1 1 0 0] – (0.546 0.179) [1 1 0 1] – (0.634 0.273) [1 1 1 0] – (0.546 0.179) [1 1 1 1] – (0.357 0.093)

12.8 CSK color mapping

Figure 146 shows the CIE1931 *xy* color coordinates (CIE 1932 [B13]) with the color mapping for 4-point CSK (4CSK). In this case, four color points are defined.

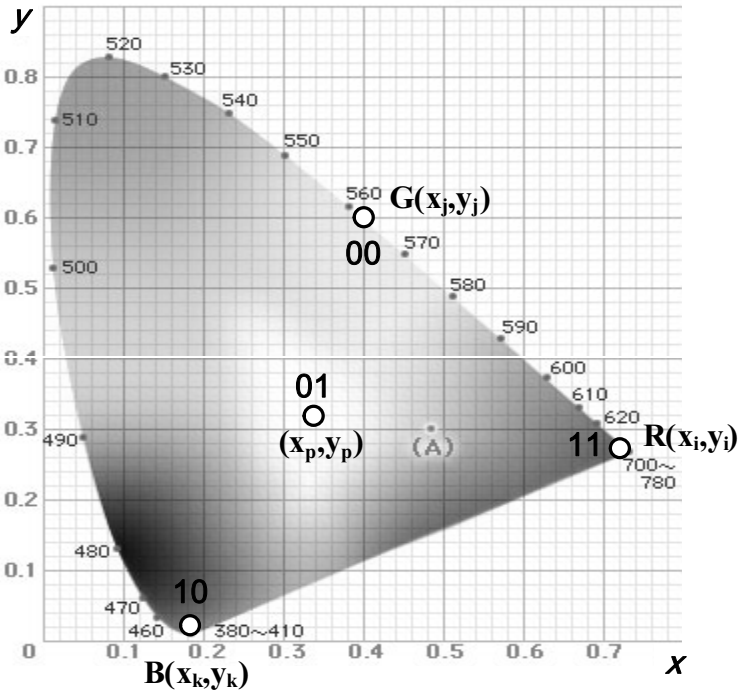


Figure 146—CIE 1931 *xy* color coordinates

The points (x_i,y_i), (x_j,y_j), (x_k,y_k) shows the *xy* coordinates of three light sources. The point (x_p,y_p) shows the one of the allocated color points in 4-CSK. The color point (x_p,y_p) is generated by the intensity of the three light sources P_i, P_j, and P_k in Figure 135. These *xy* values are transformed into intensity P_i, P_j, and P_k. The relation between the coordinates and the intensity is shown in Equation (6). In the receiver side, *xy* values are calculated from the received light powers of three colors, and *xy* values are decoded into the received data.

$$\begin{aligned} x_p &= P_i \cdot x_i + P_j \cdot x_j + P_k \cdot x_k \\ y_p &= P_i \cdot y_i + P_j \cdot y_j + P_k \cdot y_k \\ P_i + P_j + P_k &= 1 \end{aligned} \tag{6}$$

12.9 CSK calibration at the receiver

The VLC system could have some degradation, for example, multi-color imbalance, multi-color interference, or other error on *xy* color coordinates caused by ambient light or the light device characteristics; therefore, a CSK compensation method at the receiver is provided in the standard using color calibration for performance improvement. Figure 147 shows the CSK system with color calibration.

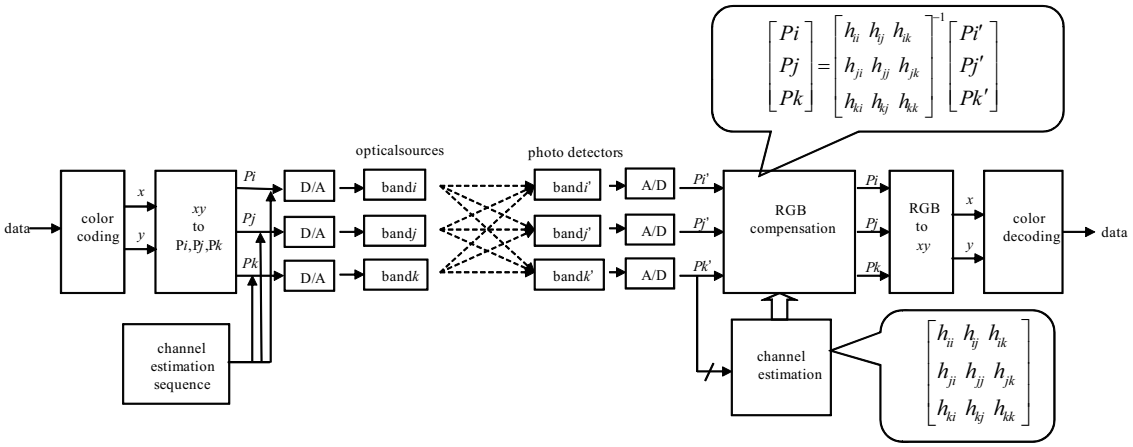


Figure 147—CSK system with color calibration

Before data communication, the system estimates the channel propagation matrix using orthogonal sequences included in the channel estimation sequence. The channel propagation matrix is a 3x3 square matrix as shown in Equation (7).

$$\begin{bmatrix} h_{ii} & h_{ij} & h_{ik} \\ h_{ji} & h_{jj} & h_{jk} \\ h_{ki} & h_{kj} & h_{kk} \end{bmatrix} \tag{7}$$

The propagation deviation can be compensated by multiplying the received signal with the inverted channel matrix as shown in Equation (8).

$$\begin{bmatrix} Pi \\ Pj \\ Pk \end{bmatrix} = \begin{bmatrix} h_{ii} & h_{ij} & h_{ik} \\ h_{ji} & h_{jj} & h_{jk} \\ h_{ki} & h_{kj} & h_{kk} \end{bmatrix}^{-1} \begin{bmatrix} Pi' \\ Pj' \\ Pk' \end{bmatrix} \tag{8}$$

Walsh codes shall be used for channel estimation as shown in Figure 148. During the transmission of the channel estimation sequence, the light sources are modulated with OOK according to the Walsh codes. Three Walsh code sequences of length 4 are provided for the three bands used for CSK. W(1,4) = {1,–1,1,–1}, W(2,4) = {1,1,–1,–1}, W(3,4) = {1,–1,–1,1} are the three Walsh codes that shall be used for channel estimation. W(1,4), W(2,4) and W(3,4) shall be used for band *i*, *j*, *k* respectively. Each bit of the Walsh code shall be transmitted twice. Accurate channel estimation can be obtained by averaging the two bits.

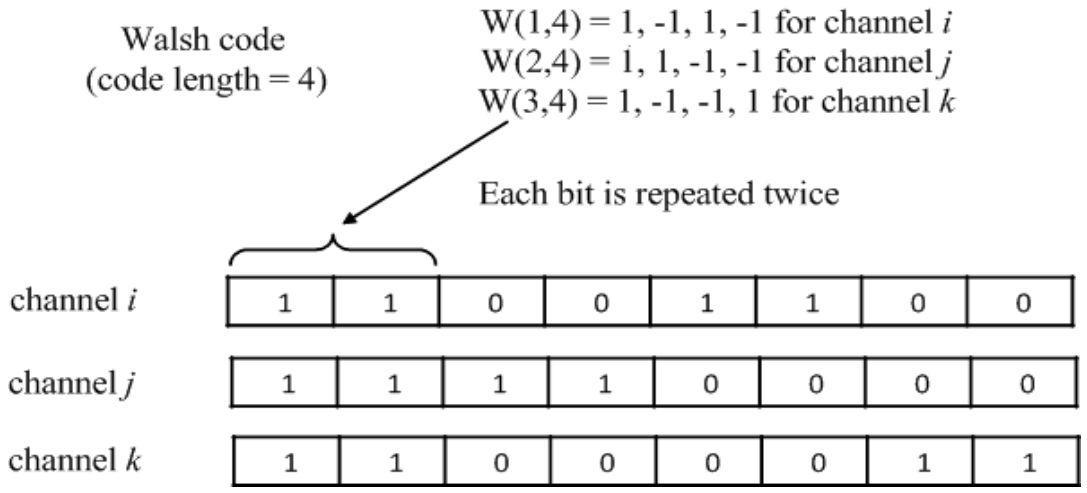


Figure 148—Walsh codes for color calibration

Annex A

(informative)

Bibliography

A.1 General

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Berman, S. M., Greenhouse, D. S., Bailey, I. L., Clear, R. and Raasch, T. W., "Human electroretinogram responses to video displays, fluorescent lighting and other high frequency sources," Optometry and Vision Science 68:645–662, 1991.

[B2] CIE (1932) Commission internationale de l'Eclairage proceedings, 1931. Cambridge University Press, Cambridge.

[B3] Colman, R. S., Frankel, F., Ritvo, E., and Freeman, B., "The effects of fluorescent and incandescent illumination upon repetitive behaviors in autistic children," J. Autism Childhood Schz. 6:157–162, 1976.

[B4] Harding, G. F. A. and Jeavons, P., "Photosensitive Epilepsy," Mac Keith Press, 1994.

[B5] IEEE 802.15 document 15-10-0151-01-0007, "High-power high-bandwidth linear driving circuit for VLC applications," Baumgartner, Robert, et. al., https://mentor.ieee.org/802.15/dcn/10/15-10-0151-01-0007-high-power-high-bandwidth-linear-driving-circuit-for-vlc-applications.pdf.

[B6] IEEE Std 802.11™-2007, 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.

[B7] 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) Specifications for High Rate Wireless Personal Area Networks (VPANs).

[B8] ISO/IEC 7498-1:1994, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model.[9]

[B9] Kang, T. G., et. al., "White Paper: IEEE 802.15.7 VLC Regulations," July 2010, IEEE802.15 contribution 15-10-0615-01-0007, https://mentor.ieee.org/802.15/dcn/10/15-10-0615-01-0007-white-paper-ieee-802-15-7-vlc-regulations.pdf.

[B10] Küller, R. and Laike, T., "The impact of flicker from fluorescent lighting on well-being, performance, and physiological arousal," Ergonomics 41(4): 433–447, 1998.

[B11] Lindner, H. and Kropf, S., "Asthenopic complaints associated with fluorescent lamp illumination (FLI): The role of individual disposition," Lighting Res. Technol. Vol. 25:59–69, 1993.

[9]ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[B12] Schubert, E. Fred, “Light Emitting Diodes,” Cambridge University Press, 2003.

[B13] Stone, P. T., “Fluorescent lighting and health,” Lighting Res. Technol. 24:55–61, 1990.

[B14] Walewski, J. W., “Color stabilization for CSK by use of visibility frames,” January 2011, IEEE802.15 contribution 15-1-0262-05-0007, https://mentor.ieee.org/802.15/dcn/11/15-10-0262-05-0007-color-stabilization-for-csk-by-use-of-visibility-frames.pdf.

[B15] Wilkins, A. J., I. Nimmo-Smith, A. Slater, and L. Bedocs, “Fluorescent lighting headaches and eye-strain,” Lighting Res. Technol. Vol. 21:11–18, 1989.

[B16] Wright, P. S., “An overview of harmonic and flicker emission standards and their associated measurements,” Power Eng. J. 15(2): 87–93, 2001.

[B17] Yokoi, A., et. al., “More description about CSK constellation,” March 2011, IEEE 802.15 contribution 15-11-0247-00-0007, https://mentor.ieee.org/802.15/dcn/11/15-11-0247-00-0007-csk-constellation-in-all-color-band-combinations.pdf.

A.2 Regulatory documents

[B18] ANSI C82.77-2002, American National Standard for Harmonic Emission Limits—Related Power Quality Requirements for Lighting Equipment.[10]

[B19] ANSI C82.SSL1 Operational Characteristics and Electrical Safety of SSL Power Supplies and Drivers.

[B20] ANSI Z136.1, American National Standard for Safe Use of Lasers (2007).

[B21] ANSI Z136.6, American National Standard for Safe Use of Lasers Outdoors (2005).

[B22] ARIB STD-T66, Second Generation Low Power Data Communication System/Wireless LAN System 1999.12.14 (H11.12.14) Version 1.0.[11]

[B23] BS EN 29241-3:1993, Ergonomic requirements for office work with visual display terminals (VDTs). Visual display requirements.

[B24] EN61000-3-3: Electronic compatibility (EMC) Part 3: Limits—Section 3: Limitation of voltage fluctuations and flicker in low voltage supply systems for equipment with rated current 16 A and smaller. International Electrotechnical Commission, 1994.

[B25] ERC Recommendation 70-03, Relating to the Use of Short Range Devices (SRDs), April 2002.[12]

[B26] IEC 1000-3-3 (1994), Part 3: Limits—Section 3: Limitation of voltage fluctuations and flicker in low-voltage supply systems for equipment with rated current 16 A.

[B27] IEC 1000-3-5: Electromagnetic compatibility. Part 3: Limits—Section 5: Limitation of voltage fluctuations and flicker in low-voltage power supply systems for equipment with rated current greater than 16 A. 1994.

[10]ANSI C82 publications are available from NEMA publications are available from the National Electrical Manufacturers Association, 1300 N. 17th St., Ste. 1847, Rosslyn, VA 22209, USA.

[11]ARIB publications are available from the Association of Radio Industries and Businesses (http://www.arib.or.jp).

[12]ERC publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (http://global.ihs.com/).

[B28] IEC 60050-112 ed1.0 (2010-01) TC/SC 1—International Electrotechnical Vocabulary—Part 112: Quantities and units.

[B29] IEC 60050-300 ed1.0 (2001-07), International Electrotechnical Vocabulary—Electrical and electronic measurements and measuring instruments—Part 311: General terms relating to measurements—Part 312: General terms relating to electrical measurements—Part 313: Types of electrical measuring instruments—Part 314: Specific terms according to the type of instrument.

[B30] IEC 60050-845 ed1.0 (1987-12), International Electrotechnical Vocabulary, Lighting.

[B31] IEC 60825-1 ed2.0 (2007-03), Safety of laser products—Part 1: Equipment classification and requirements.

[B32] IEC 60825-12 ed1.0 (2004-02), Safety of laser products—Part 12: Safety of free space optical communication systems used for transmission of information.

[B33] IEC 61347-1 ed2.0 (2007-01), Lamp control gear—Part 1: General and safety requirements.

[B34] IEC 61347-2-13 ed1.0 (2006-05), Lamp control gear—Part 2-13: Particular requirements for d.c. or a.c. supplied electronic control gear for LED modules

[B35] IEC 62384-am1 ed1.0 (2009-07) Amendment 1—DC or AC supplied electronic control gear for LED modules—Performance requirements.

[B36] IEC 62471 ed1.0 (2006-07), Photobiological safety of lamps and lamp systems.

[B37] IEC/TR 62471-2 ed1.0 (2009-08), Photobiological safety of lamps and lamp systems—Part 2: Guidance on manufacturing requirements relating to non-laser optical radiation safety.

[B38] IEC/TS 61000-3-5 Ed. 2.0 b Cor.2:2010 Corrigendum 2—Electromagnetic compatibility (EMC)—Part 3-5: Limits—Limitation of voltage fluctuations and flicker in low-voltage power supply systems for equipment with rated current greater than 75 A.

[B39] ANSI/IESNA RP-16-05, Nomenclature and Definitions for Illuminating Engineering.[13]

[B40] ANSI/IESNA RP-27.1-05, Photobiological Safety for Lamps and Lamp Systems—General Requirements.

[B41] IESNA LM-79-08, IES Approved Method for the Electrical and Photometric Measurements of Solid-State Lighting Products.

[B42] IESNA LM-80-08, IES Approved Method: Measuring Lumen Maintenance of LED Light Sources.

[B43] IESNA TM-16-05, Technical Memorandum on Light Emitting Diode (LED) Sources and Systems.

[B44] Recommendation ITU-R RA.1630, Technical and operational characteristics of ground-based astronomy systems for use in sharing studies with active services between 10 THz and 1000 THz.

[B45] Recommendation ITU-R RS.1744, Technical and operational characteristics of ground-based meteorological aids systems operating in the frequency range 272–750 THz.[14]

[13]IESNA is the Illuminating Engineering Society Standards from ANSI. IESNA publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[14]This recommendation replaces ITU-R SA.1744. ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

Annex B

(normative)

Service-specific convergence sublayer (SSCS)

B.1 IEEE 802.2 convergence sublayer

The IEEE 802.2 convergence sublayer exists above the IEEE 802.15.7 MCPS. This sublayer provides an interface between an instance of an IEEE 802.2 LLC sublayer and the IEEE 802.15.7 MCPS.

B.1.1 MA-UNITDATA.request

The MA-UNITDATA.request primitive requests the transfer of a LLC protocol data unit (LPDU) (i.e., MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or multiple peer IEEE 802.2 Type 1 LLC sublayer entities in the case of a group address.

The semantics of the MA-UNITDATA.request primitive is as follows:

```
MA-UNITDATA.request (
                    SrcAddr,
                    DstAddr,
                    RoutingInformation,
                    data,
                    priority,
                    ServiceClass
                    )
```

Table B.1 specifies the parameters for the MA-UNITDATA.request primitive.

Table B.1—MA-UNITDATA.request parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU is being transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
data	Set of octets	—	The set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

B.1.1.1 Appropriate usage

The MA-UNITDATA.request primitive is generated by a local IEEE 802.2 Type 1 LLC sublayer entity when an LPDU (MSDU) is to be transferred to a peer IEEE 802.2 Type 1 LLC sublayer entity or entities.

B.1.1.2 Effect on receipt

On receipt of the MA-UNITDATA.request primitive, the MAC sublayer entity shall begin the transmission of the supplied MSDU.

The MAC sublayer first builds an MPDU to transmit from the supplied arguments. The MPDU shall be transmitted using the unslotted CSMA-CA algorithm in the contention period of the frame and without requesting a handshake.

If the unslotted CSMA-CA algorithm indicates a busy channel, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of CHANNEL_ACCESS_FAILURE. If the MPDU was successfully transmitted, the MAC sublayer shall issue the MA-UNITDATA-STATUS.indication primitive with a status of SUCCESS.

B.1.2 MA-UNITDATA.indication

The MA-UNITDATA.indication primitive indicates the transfer of an LPDU (i.e., MSDU) from the MAC sublayer to the local IEEE 802.2 Type 1 LLC sublayer entity.

The semantics of the MA-UNITDATA.indication primitive is as follows:

```
MA-UNITDATA.indication (
                       SrcAddr,
                       DstAddr,
                       RoutingInformation,
                       data,
                       ReceptionStatus,
                       priority,
                       ServiceClass
                       )
```

Table B.2 specifies the parameters for the MA-UNITDATA.indication primitive.

Table B.2—MA-UNITDATA.indication parameters

Name	Type	Valid range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been received.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU is being transferred.
RoutingInformation	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

Table B.2—MA-UNITDATA.indication parameters *(continued)*

Name	Type	Valid range	Description
data	Set of octets	—	The set of octets forming the MSDU received by the MAC sublayer entity.
ReceptionStatus	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
priority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

B.1.2.1 When generated

On receipt of a data frame at the local MAC sublayer entity, the FCS field is checked. If it is valid, the MAC sublayer shall issue the MA-UNITDATA.indication primitive to the IEEE 802.2 Type 1 LLC sublayer entity, indicating the arrival of a MSDU. If the FCS is not valid, the frame shall be discarded, and the IEEE 802.2 Type 1 LLC sublayer entity shall not be informed.

B.1.2.2 Appropriate usage

The appropriate usage of the MA-UNITDATA.indication primitive by the IEEE 802.2 Type 1 LLC sublayer entity is not specified in this standard.

B.1.3 MA-UNITDATA-STATUS.indication

The MA-UNITDATA-STATUS.indication primitive reports the results of a request to transfer a LPDU (MSDU) from a local IEEE 802.2 Type 1 LLC sublayer entity to a single peer IEEE 802.2 Type 1 LLC sublayer entity or to multiple peer IEEE 802.2 Type 1 LLC sublayer entities.

The semantics of the MA-UNITDATA-STATUS.indication primitive is as follows:

```
MA-UNITDATA-STATUS.indication   (
                                SrcAddr,
                                DstAddr,
                                status,
                                ProvPriority,
                                ProvServiceClass
                                )
```

Table B.3 specifies the parameters for the MA-UNITDATA-STATUS.indication primitive.

B.1.3.1 When generated

The MA-UNITDATA-STATUS.indication primitive is generated by the MAC sublayer entity in response to an MA-UNITDATA.request primitive issued by the IEEE 802.2 Type 1 LLC sublayer.

Table B.3—MA-UNITDATA-STATUS.indication parameters

Name	Type	Valid Range	Description
SrcAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity from which the MSDU has been transferred.
DstAddr	IEEE address	Any valid IEEE address	The individual IEEE address of the entity to which the MSDU has been transferred.
status	Enumeration	SUCCESS, TRANSMISSION_PENDING, NO_BEACON, or CHANNEL_ACCESS_FAILURE	The status of the last MSDU transmission.
ProvPriority	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.
ProvServiceClass	—	null	This parameter is not used by the MAC sublayer and shall be specified as a null value.

B.1.3.2 Appropriate usage

The receipt of the MA-UNITDATA-STATUS.indication primitive by the IEEE 802.2 Type 1 LLC sublayer entity signals the completion of the current data transmission.

Annex C

(normative)

Cyclic redundancy check

The CRC field is 2 octets in length. The CRC shall be calculated using the following standard generator polynomial of degree 16:

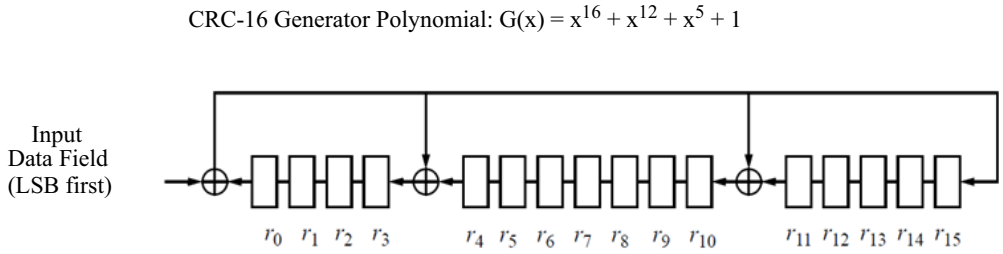
$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1 \quad \text{(C.1)}$$

The CRC shall be calculated for transmission using the following algorithm:

— Let $M(x) = b_0 x^{k-1} + b_1 x^{k-2} + \ldots + b_{k-2} x + b_{k-1}$ be the polynomial representing the sequence of bits for which the checksum is to be computed.
— Multiply $M(x)$ by x^{16}, giving the polynomial $x^{16} \times M(x)$.
— Divide $x^{16} \times M(x)$ modulo 2 by the generator polynomial, $G_{16}(x)$, to obtain the remainder polynomial, $R(x) = r_0 x^{15} + r_1 x^{14} + \ldots + r_{14} x + r_{15}$.
— The CRC field is given by the coefficients of the remainder polynomial, *R(x)*.

Here, binary polynomials are represented as bit strings, in highest polynomial degree first order.

A typical implementation is depicted in Figure C.1.



1. Initialize the remainder register (r_0 through r_{15}) to all ones.
2. Shift the data into the divider in the order of transmission (LSB).
3. After the last bit of the data field is shifted into the divider, the remainder register contains the CRC.
4. The CRC is appended to the data so that r_0 is transmitted first.

Figure C.1—Typical CRC implementation

Annex D

(normative)

Channel assignment

Table D.1 shows the bit patterns that are used to assign multiple channels for VLC.

Table legend: X=not used and O=used

Table D.1—Multiple channel assignment table

Bit	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7
0000000	No multiple channel mode						
0000001	X	X	X	X	X	X	O
0000010	X	X	X	X	X	O	X
0000011	X	X	X	X	X	O	O
0000100	X	X	X	X	O	X	X
0000101	X	X	X	X	O	X	O
0000110	X	X	X	X	O	O	X
0000111	X	X	X	X	O	O	O
0001000	X	X	X	O	X	X	X
0001001	X	X	X	O	X	X	O
0001010	X	X	X	O	X	O	X
0001011	X	X	X	O	X	O	O
0001100	X	X	X	O	O	X	X
0001101	X	X	X	O	O	X	O
0001110	X	X	X	O	O	O	X
0001111	X	X	X	O	O	O	O
0010000	X	X	O	X	X	X	X
0010001	X	X	O	X	X	X	O
0010010	X	X	O	X	X	O	X
0010011	X	X	O	X	X	O	O
0010100	X	X	O	X	O	X	X
0010101	X	X	O	X	O	X	O
0010110	X	X	O	X	O	O	X
0010111	X	X	O	X	O	O	O
0011000	X	X	O	O	X	X	X

Table D.1—Multiple channel assignment table *(continued)*

Bit	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7
0011001	X	X	O	O	X	X	O
0011010	X	X	O	O	X	O	X
0011011	X	X	O	O	X	O	O
0011100	X	X	O	O	O	X	X
0011101	X	X	O	O	O	X	O
0011110	X	X	O	O	O	O	X
0011111	X	X	O	O	O	O	O
0100000	X	O	X	X	X	X	X
0100001	X	O	X	X	X	X	O
0100010	X	O	X	X	X	O	X
0100011	X	O	X	X	X	O	O
0100100	X	O	X	X	O	X	X
0100101	X	O	X	X	O	X	O
0100110	X	O	X	X	O	O	X
0100111	X	O	X	X	O	O	O
0101000	X	O	X	O	X	X	X
0101001	X	O	X	O	X	X	O
0101010	X	O	X	O	X	O	X
0101011	X	O	X	O	X	O	O
0101100	X	O	X	O	O	X	X
0101101	X	O	X	O	O	X	O
0101110	X	O	X	O	O	X	O
0101111	X	O	X	O	O	O	X
0110000	X	O	O	X	X	X	X
0110001	X	O	O	X	X	X	O
0110010	X	O	O	X	X	O	X
0110011	X	O	O	X	X	O	O
0110100	X	O	O	X	O	X	X
0110101	X	O	O	X	O	X	O
0110110	X	O	O	X	O	O	X
0110111	X	O	O	X	O	O	O
0111000	X	O	O	O	X	X	X
0111001	X	O	O	O	X	X	X

Table D.1—Multiple channel assignment table *(continued)*

Bit	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7
0111010	X	O	O	O	X	X	X
0111011	X	O	O	O	X	X	X
0111100	X	O	O	O	O	X	X
0111101	X	O	O	O	O	X	X
0111110	X	O	O	O	O	O	X
0111111	X	O	O	O	O	O	O
1000000	O	X	X	X	X	X	X
1000001	O	X	X	X	X	X	O
1000010	O	X	X	X	X	O	X
1000011	O	X	X	X	X	O	O
1000100	O	X	X	X	O	X	X
1000101	O	X	X	X	O	X	O
1000110	O	X	X	X	O	O	X
1000111	O	X	X	X	O	O	O
1001000	O	X	X	O	X	X	X
1001001	O	X	X	O	X	X	O
1001010	O	X	X	O	X	O	X
1001011	O	X	X	O	X	O	O
1001100	O	X	X	O	O	X	X
1001101	O	X	X	O	O	X	O
1001110	O	X	X	O	O	O	X
1001111	O	X	X	O	O	O	O
1010000	O	X	O	X	X	X	X
1010001	O	X	O	X	X	X	O
1010010	O	X	O	X	X	O	X
1010011	O	X	O	X	X	O	O
1010100	O	X	O	X	O	X	X
1010101	O	X	O	X	O	X	O
1010110	O	X	O	X	O	O	X
1010111	O	X	O	X	O	O	O
1011000	O	X	O	O	X	X	X
1011001	O	X	O	O	X	X	O
1011010	O	X	O	O	X	O	X

Table D.1—Multiple channel assignment table *(continued)*

Bit	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7
1011011	O	X	O	O	X	O	O
1011100	O	X	O	O	O	X	X
1011101	O	X	O	O	O	X	O
1011110	O	X	O	O	O	O	X
1011111	O	X	O	O	O	O	O
1100000	O	O	X	X	X	X	X
1100001	O	O	X	X	X	X	O
1100010	O	O	X	X	X	O	X
1100011	O	O	X	X	X	O	O
1100100	O	O	X	X	O	X	X
1100101	O	O	X	X	O	X	O
1100110	O	O	X	X	O	O	X
1100111	O	O	X	X	O	O	O
1101000	O	O	X	O	X	X	X
1101001	O	O	X	O	X	X	O
1101010	O	O	X	O	X	O	X
1101011	O	O	X	O	X	O	O
1101100	O	O	X	O	O	X	X
1101101	O	O	X	O	O	X	O
1101110	O	O	X	O	O	O	X
1101111	O	O	X	O	O	O	O
1110000	O	O	O	X	X	X	X
1110001	O	O	O	X	X	X	O
1110010	O	O	O	X	X	O	X
1110011	O	O	O	X	X	O	O
1110100	O	O	O	X	O	X	X
11101001	O	O	O	X	O	X	O
1110110	O	O	O	X	O	O	X
1110111	O	O	O	X	O	O	O
1111000	O	O	O	O	X	X	X
1111001	O	O	O	O	X	X	O
1111010	O	O	O	O	X	O	X
1111011	O	O	O	O	X	O	O

Table D.1—Multiple channel assignment table *(continued)*

Bit	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7
1111100	O	O	O	O	O	X	X
1111101	O	O	O	O	O	X	O
1111110	O	O	O	O	O	O	X
1111111	O	O	O	O	O	O	O

An example is shown in Figure D.1 where it is assumed that red, green and blue are available at the optical sources. If a certain optical source uses HP1 (00001) and another optical source in the adjacent cell uses HP2 (00011), then hopping pattern application in the adjacent cell is that HP1 operates R in first frame or time slot, B in second frame or time slot, G in third frame or time slot, but HP2 is operating at G in first frame or time slot, G and R in second frame or time slot, R and B in third frame or time slot. This mechanism can avoid interference between optical sources. Also the hopping pattern application is not limited to one frame or one time slot. A hopping pattern across multiple frames or time slots is fine.

Table D.2 shows a hopping pattern example applicable to VLC. If coordinator assign pattern '00001' to a device by using H_pattern, then the device's frame or time slot moves according to the hopping pattern. Also one hopping pattern or multiple hopping patterns can be assigned to one user.

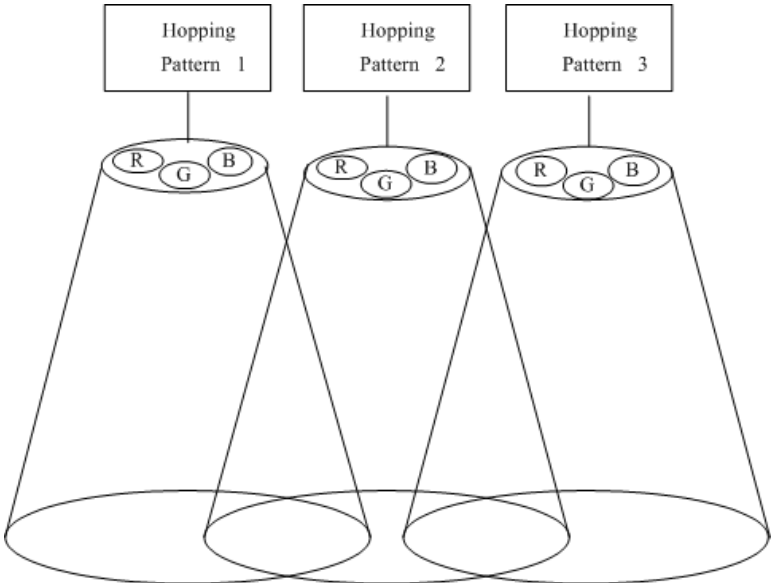


Figure D.1—Hopping pattern assignment

Table D.2—Example of hopping pattern assignment

Pattern	00001	00011	00101
Frame/time slot	**HP1**	**HP2**	**HP3**
1	R	G	B
2	B	G/R	B
3	G	R/B	G
4	G/R	B	G/R
5	G/R	R	G/B
6	R/B	G	R/B
7	G	B	R
8	B	R	G
9	R	G/B	R

Annex E

(informative)

Considerations for VLC using LED displays

E.1 Introduction—Dynamic displays vs. addressed displays

This annex discusses two types of LED involved displays: dynamic displays and addressed displays. The fundamental different between the two has to do with the amount of time the pixel is illuminated. In dynamic displays the pixels on a line are illuminated once every frame for T_frame/N_line seconds where N_line is the number of lines controlled by a line scan controller and T_frame is the time to sweep N_line once; that is, the pixels on a line are operated dynamically so that they are only on for a fraction of the total frame time (hence the name “dynamic”). In the addressed display, the pixel is illuminated for the duration of the frame and is readdressed once per frame for possible state change.

E.2 Dynamic displays

E.2.1 Operation mechanism

In general, a dynamic display consists of a host controller, a line scan controller, a display data buffer, and LED matrix, as shown in Figure E.1. The line scan controller selects a line for display, and the display data buffer transmits state information, such as the on/off state or the color selection, to each LED pixel on the selected line. The line scan controller determines the active time of each line, where the active time indicates whether the LED pixels on the selected line are switched ON or OFF as per the display data buffer for the active time of the selected line. Therefore, VLC using a dynamic display is tightly coupled with the active time.

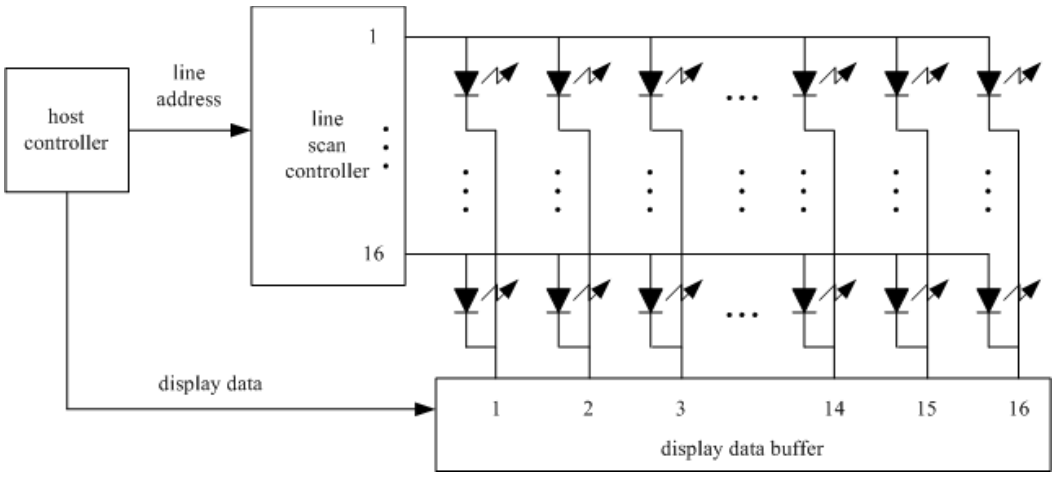


Figure E.1—General architecture of LED signboard operated by the dynamic display mechanism

It is well known that most of PC monitors and TV display images use 60 displayed-frames per second. In the case of PC monitors or TVs, the total displayed-frame number (i.e., different display information) is actually 30 frames per second because each displayed-frame is transmitted twice. The display mechanism of a dynamic display is similar to a PC monitor or TV. Assuming that a dynamic display consists of 16x16 lines and displays images or text through 60 displayed-frames per second, with 16 lines per displayed-frame, the active time slot period for VLC assigned to each line is 1.042 ms per displayed-frame, as shown in Figure E.2. Therefore, an active time slot can transmit 130 byte at 1 Mbit/s, as shown in Figure E.3.

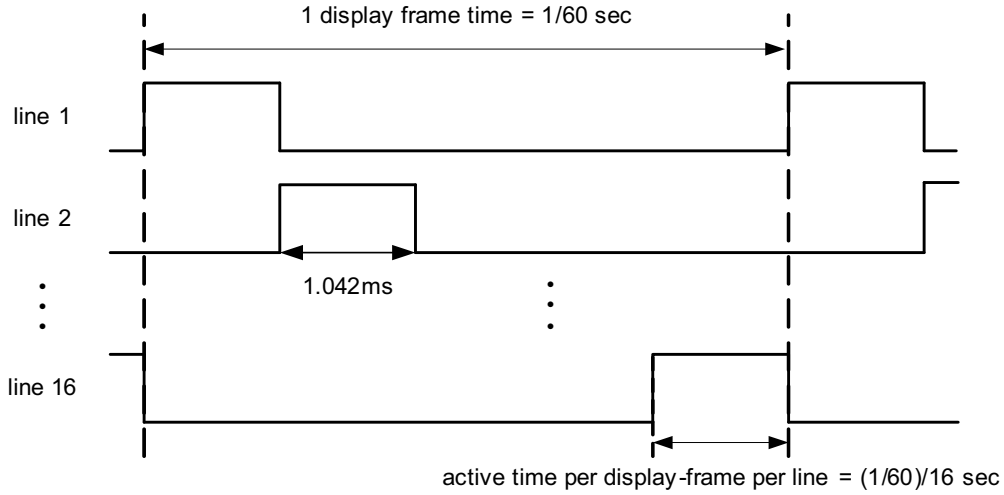


Figure E.2—Operation mechanism of dynamic display

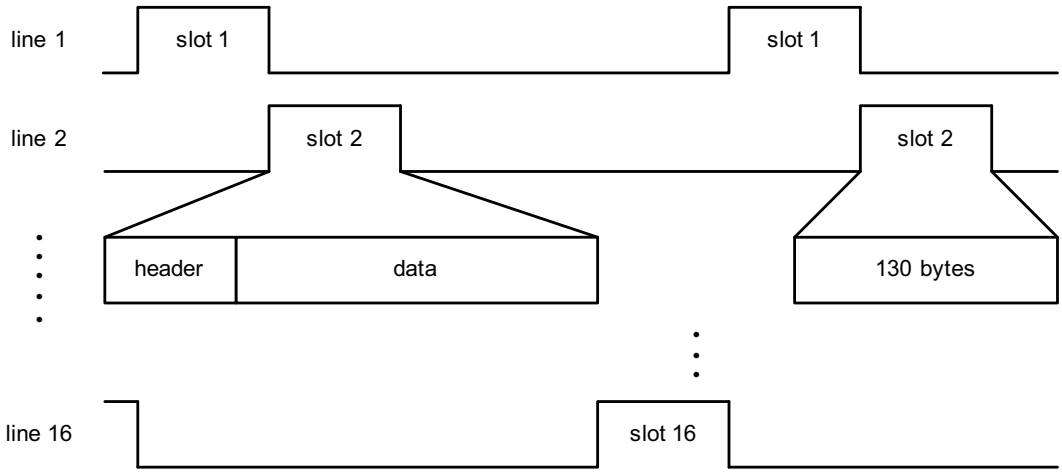


Figure E.3—An example of VLC data transmission on a dynamic display

E.2.2 Reduced brightness mitigation on VLC dynamic displays

The VLC dynamic display may be less bright than the non-VLC dynamic display because of the VLC modulation during the active time periods. Therefore, it is important to minimize the reduction of the average brightness of the dynamic display during the shortest time period that the human can distinguish.

This is done to keep the display performing adequately for its intended function. Figure E.4 shows an example of an operational mechanism to mitigate the average brightness reduction per display frame time that can arise for a dynamic display having 16x16 lines operating as shown in Figure E.2 (i.e., the VLC data is carried during the active time of each line). The average brightness per display frame time in Figure E.4 is twice as large as that in Figure E.2 because the active time is twice as long.

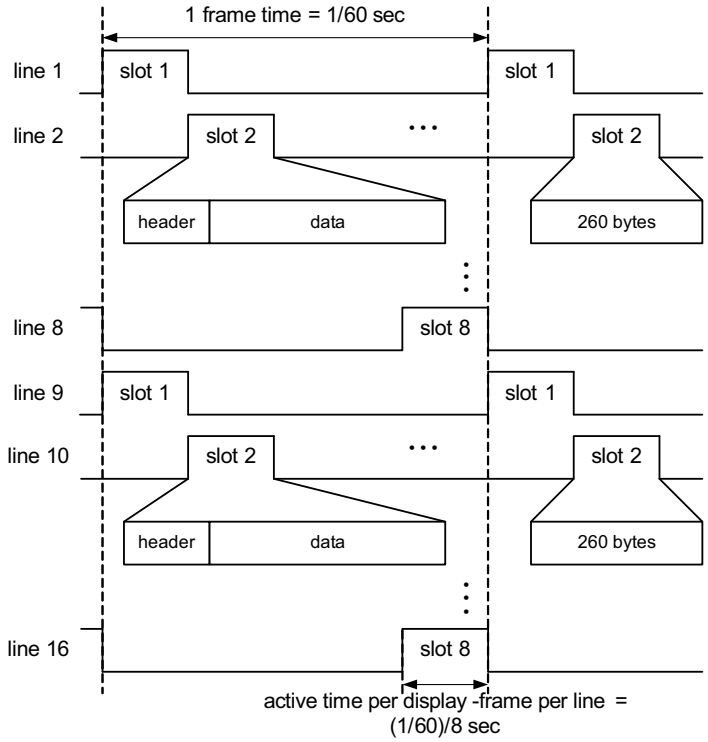


Figure E.4—An example for the mitigation of the average brightness reduction on the VLC dynamic display

E.2.3 VLC application using dynamic displays

A VLC enabled dynamic display can be used in the broadcast topology. The VLC broadcast topology in this standard consists of mainly the beacon and the downlink, as shown in Figure 14. Therefore, the VLC broadcast topology using a dynamic display can be constructed by the assignment of the active time slots and the use of GTS field in the beacon frame. Figure E.5 shows the VLC broadcast topology construction using the dynamic display. The active time slot #1 is assigned to the beacon and the active time slots from #2 to #8 are assigned to the downlink in Figure E.5. The GTS fields of the beacon frame can be used to indicate the GTS number, GTS length, and GTS direction for the broadcast topology. Multiple GTS slots can also be used depending upon the desired service level, the subscriber's grade, and the QoS policy.

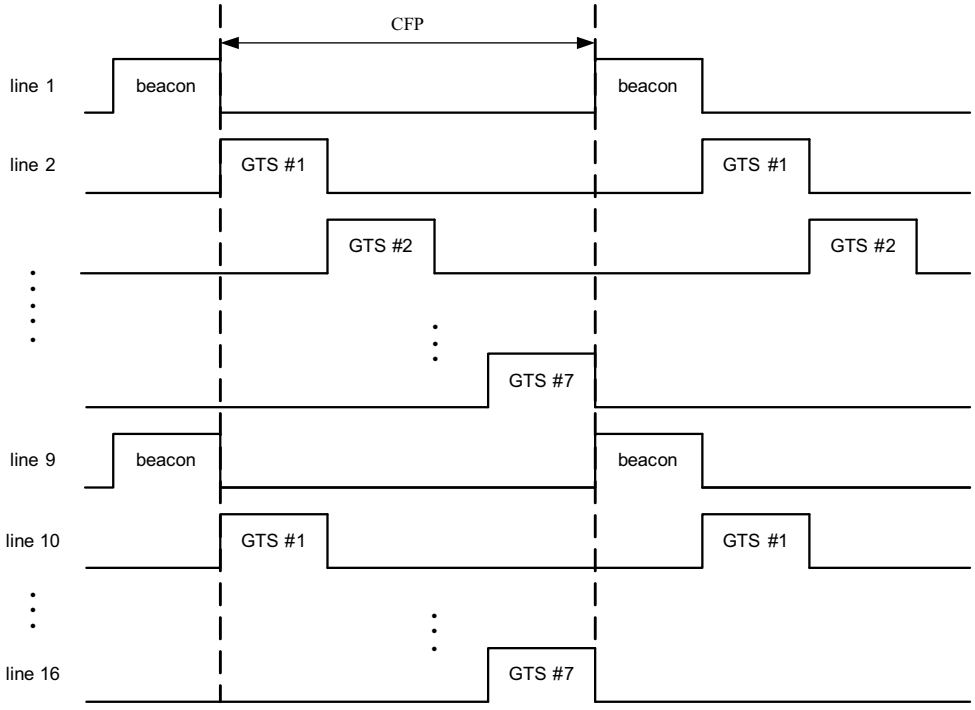


Figure E.5—VLC broadcast topology construction using the dynamic display

E.3 Addressed displays

In an addressed display a particular pixel is addressed (i.e., refreshed) once every frame; that is, once addressed the pixel maintains its current state until readdressed during the next frame. The role of the LED can be either to provide pixel back lighting, as in the case of an LCD (liquid crystal display), or the pixel can be formed directly from a LED device as in the case of LED signage (which would typically be implemented as a pixel constructed from a compound RGB LED). Generally the addressed pixels in a frame are serially updated and there is no need for a retrace blanking interval as in the old days of cathode ray tubes.

E.3.1 LCD display using LED backlighting modulation

In LED backlighting, there are numerous LEDs that provide illuminance for all LCD pixels, while the intensity of a particular pixel is determined by the transmittance of the pixel LCD. This means the data modulation of the LED backlighting is going to radiate from all the pixels. The radiation is a density (mW/ unit area) and the best performance occurs when the sensor views the whole screen (i.e., ingests the most power). Viewing the whole screen also provides intensity averaging over all the pixels, which is advantageous since in the some scenes various pixels might go dark. Given enough area averaging, there no particular relationship between the data rate and the frame refresh rate.

E.3.2 LED pixel modulation

One has the option of either modulating individual pixels or groups of pixels. In other words, one can generate at least two display sections that can transmit different data. One could even go further and create intermediate sections that transmit an aggregate of the data transmitted in the adjacent sections. For this to

work one needs a detector capable of spatially resolving such sections. Partitioning of the display is done by using the cell mechanism and the PHY SAP, which were introduced for mobility support (see 5.1.11). For example, the cell partitioning in Figure 33 can be interpreted as a 2x2 display with four sections if the transmitters are operated in the broadcast mode. Also, wavelength division multiplexing of RGB pixels can be enabled with this mechanism, so that each color carries a different data stream. Averaging over an area of the sign allows some insensitivity to scene-to-scene pixel intensity variation.

Annex F

(informative)

Receiver performance variation on multi-color channels

Many applications using colored light sources can be considered in VLC. For example, Figure F.1 describes the scenario that a VLC receiver receives some information from a traffic signal light sources with color “A” and color “B”. Figure F.2 describes that an user with a VLC receiver can get the audio information from a color “A” lamp, the video information from a color “B” lamp, or the navigation information from a color “C” lamp. Figure F.3 shows that the multiplexing technologies such as Wavelength Division Multiplexing (WDM) can be applied to VLC applications using colorful light sources.

VLC services, using multiple color channels according to the VLC band plan, should be attained by only one VLC receiver. It is undesirable if a VLC receiver exhibits better receiver performance only on, for example, the color “A” channel but it does not exhibit the same performance on the color “B” or color “C” channel as on the color “A” channel. Therefore, a uniform performance on each color channel may be desired.

There are two main factors influencing the performance variation of a multi-color VLC receiver. One is the conversion relations between the radiometric and photometric units. The other is the photo sensitivity characteristics of a photo-detector, such as a Si photo-detector that depends on the wavelength variation, assuming such photo-detectors will be used as a receiver in VLC.

First, suppose that there are two light sources or VLC transmitters with red color and green color respectively and a VLC receiver to perceive the variation of received powers under the multiple color channels which originates from the conversion relations between the radiometric and photometric units. Also suppose that each color light radiates from two light sources at the same divergence angle, and they radiate with the same luminous flux, (lumens), so that the human eye senses the same brightness when simultaneously viewing the two light sources respectively at the same distance. However, when each color light radiates with the same luminous flux then the radiation of the two light sources in radiometric power (Watts) are each different, which is the origin of the CIE sensitivity curves. The CIE sensitivity curves indicate what the human eye senses, and it turns out that the green light is perceived as being brighter than red light when the radiometric radiation powers of the two light sources are equal. Therefore, the radiometric received powers of a receiver are different on red and green channels, respectively, even though the divergence angles and luminous fluxes of two light sources are equal, the same receiver is used, and the distances between the receiver and the light source are equal.

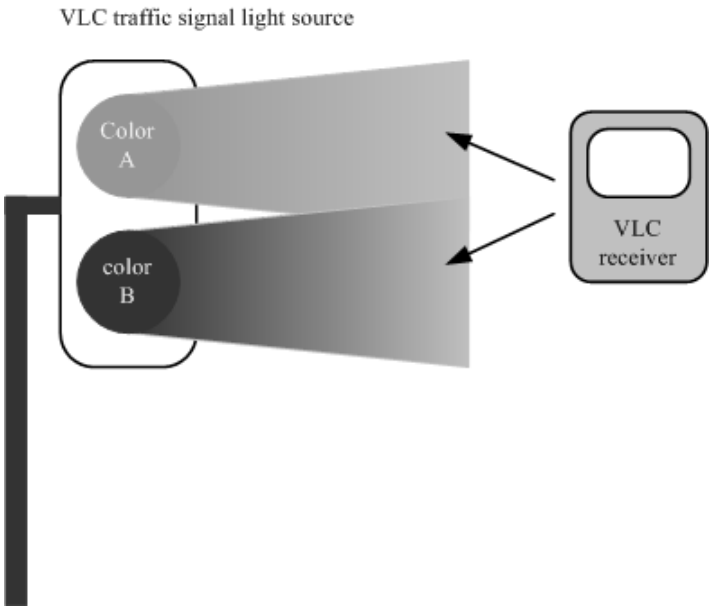


Figure F.1—VLC application using traffic signal light sources

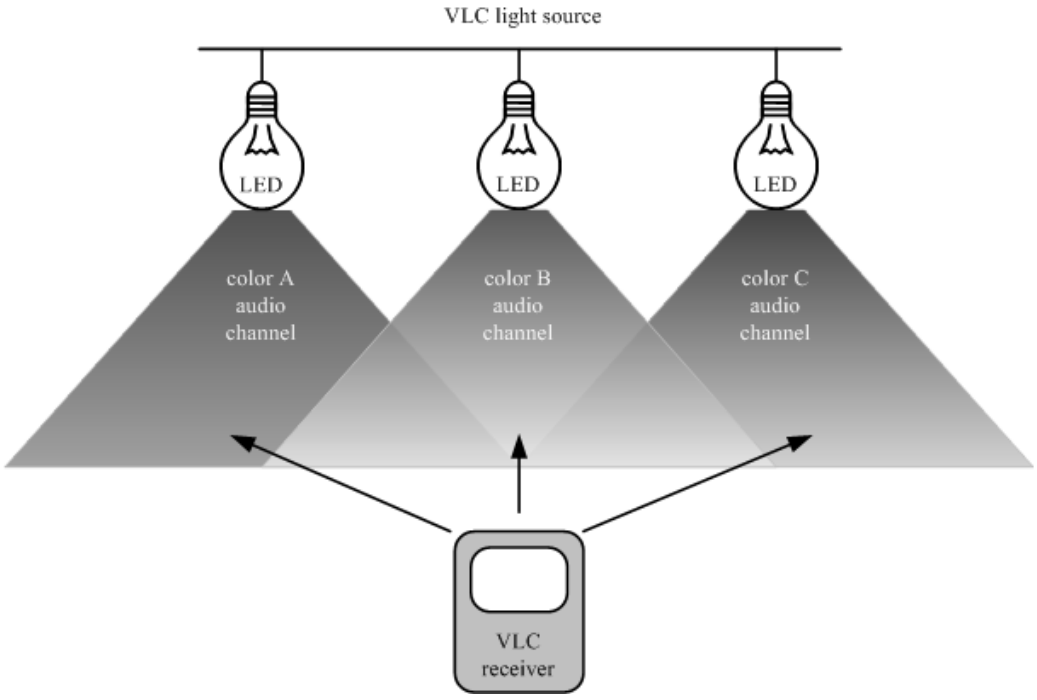


Figure F.2—VLC application using colored light sources

Figure F.3—VLC application using WDM technology

Of course, the VLC light source power can be also defined in radiometric unit, Watt. However, it is more reasonable that the light source power is defined in photometric unit because the light sources will be used not only for a VLC transmitter but also for illumination or visual display related to human eye.

Table F.1 describes the receiver input powers, calculated in Watts, from the assumption of 1 lumen on each of the seven color bands (given in Table 76). The assumption that the lights have only monochromatic component (shown as example wavelength in Table F.1) on each color band is also used for simple calculations. V(λ) is the human eye sensitivity function, which indicates CIE sensitivity curves (Schubert [B12]).

Table F.1—Calculated color channel power at receiver

Wavelength band (nm)		Spectral width (nm)	Example wavelength (nm)	V(λ) at example wavelength	Receiver input power (Watts) @ 1m
380	478	98	430	0.0273	0.0536
478	540	62	510	0.5030	0.0029
540	588	48	565	0.9788	0.0015
588	633	45	610	0.5030	0.0029
633	679	46	655	0.0817	0.0179
679	726	47	700	0.0041	0.3571
726	780	54	750	0.0001	14.6413

The second factor causing the performance variation of a VLC receiver across multiple color channels is the photo sensitivity characteristics of optical receivers, such as Si photo-detectors, which is wavelength dependent. Figure F.4 shows the photo sensitivity characteristics of a Si photo-detector according to the wavelength variation. It has been known that the photo sensitivity value of Si photo-detector is higher on longer wavelength than on shorter wavelength in the visible band as shown in Figure F.4. Figure F.4 shows that a Si photo-detector produce more electrical current on red color channel than on green or blue color channel even though the radiometric received powers on each color channel are equal.

Table F.2 shows the photo-detector output current obtained from both the wavelength dependence of photo sensitivity shown in Figure F.4 and the conversion relations between the radiometric and photometric units described in Table F.1. The photo-detector output currents in Table F.2 were calculated only at 430 nm, 510 nm, and 655 nm among the example wavelengths in seven color bands, as shown in Table F.1, for convenience.

Table F.2 indicates that a VLC receiver with Si photo-detector performs differently on multiple color channels even though the radiometric received powers are equal on each color channel. Therefore, two main factors, the unit conversion and the photo sensitivity of a photo-detector depending on wavelength, need to be sufficiently considered in order that the performance of a VLC receiver can be maintained uniformly on multiple color channels.

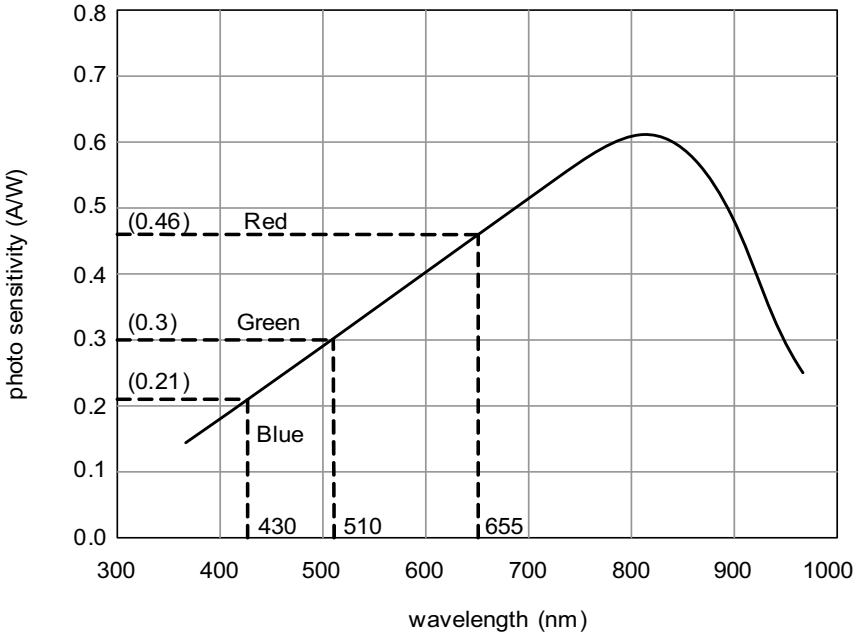


Figure F.4—Typical Si photo-detector wavelength sensitivity

Table F.2—Photo-detector current from Figure F.4 with conditions of Table F.1

Example wavelength (nm)	Receiver input power (Watt) @ 1 m	Photo sensitivity (A/W)	Photo-detector output current (mA) @ 1 lm
430	0.0536	0.21	11.26
510	0.0029	0.30	0.87
655	0.0179	0.46	8.23