International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.9959
(01/2015)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Access networks – In premises networks

Short range narrow-band digital radiocommunication transceivers – PHY, MAC, SAR and LLC layer specifications

Recommendation ITU-T G.9959

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999
Metallic access networks	G.9700–G.9799
Optical line systems for local and access networks	G.9800–G.9899
In premises networks	**G.9900–G.9999**

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.9959

Short range narrow-band digital radiocommunication transceivers – PHY, MAC, SAR and LLC layer specifications

Summary

Recommendation ITU-T G.9959 specifies the physical (PHY), medium access control (MAC), segmentation and reassembly (SAR), and logical link control (LLC) layers for short range narrow-band digital radiocommunication transceivers (TRXs). This Recommendation contains the non-radio (frequency) related aspects of the radiocommunication TRX. Sub 1 GHz TRXs claiming compliance with this specification shall also comply with Annex A of this Recommendation.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.9959	2012-02-13	15	11.1002/1000/11529
1.1	ITU-T G.9959 (2012) Amd.1	2013-10-07	15	11.1002/1000/12006
2.0	ITU-T G.9959	2015-01-13	15	11.1002/1000/12399

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

© ITU 2015

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Recommendation ITU-T G.9959

Short range narrow-band digital radiocommunication transceivers – PHY, MAC, SAR and LLC layer specifications

1 Scope

This Recommendation specifies the PHY, MAC, SAR and LLC layers for short range narrow-band digital radio communication transceivers (TRXs). The Recommendation specifies sub 1 GHz TRXs which shall be interoperable with TRXs complying with Annex A of this Recommendation.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 alien domain: Any group of non-ITU-T G.9959 nodes connected to the same or a different medium (wired or wireless). These domains can be used as backbones to the ITU-T G.9959 network or as separate networks. The bridging function to an alien domain, as well as coordination with an alien domain to avoid mutual interference is beyond the scope of this Recommendation.

3.2.2 broadcast: A type of communication where a node sends a MAC frame which is simultaneously received by all other nodes within a direct range. In a multi-hop domain, some nodes of the domain may not receive the broadcast frame.

3.2.3 channel: A transmission path between nodes. Logically a channel is an instance of a communication medium being used for the purpose of passing data between two or more nodes.

3.2.4 clear channel assessment (CCA): Provided by the receiver, a CCA indicates if the medium is busy, e.g., if a PHY frame is currently transmitted on the medium by another node.

3.2.5 data: Bits or bytes transported over the medium or via a reference point that individually convey information. Data includes user (application) data and any other auxiliary information (overhead, control, management, etc.). Data does not include bits or bytes that, by themselves, do not convey any information, such as the preamble.

3.2.6 data rate: The rate, in bits per second, at which data is transmitted by a node onto the medium. Data rate is calculated only for time periods of continuous transmission.

3.2.7 domain: A collection of ITU-T G.9959 nodes comprising the domain master and all those nodes that are registered with the same domain master. In the context of this Recommendation, use of the term 'domain' without a qualifier means 'ITU-T G.9959 domain', and use of the term 'alien domain' means 'non-ITU-T G.9959 domain'.

3.2.8 domain ID: A unique identifier of a domain. Refer to HomeID.

3.2.9 domain master: A node with extended management capabilities which allows it to handle registration and maintenance of the nodes in its domain.

3.2.10 home area network (HAN): A network capable of connecting devices in home premises.

3.2.11 HomeID: Information unit used as a domain ID in ITU-T G.9959 networks.

3.2.12 inclusion: The process of adding a new node to a domain in a way so that the node can communicate with other nodes in the domain and filter out traffic from other domains.

3.2.13 inter-domain bridge (IDB): A bridging function to interconnect nodes of two different domains.

3.2.14 industrial, scientific and medical (ISM) band: Frequency bands for industrial, scientific and medical use, allocated by the ITU.

3.2.15 latency: A measure of the delay from the instant that a frame has been transmitted through a reference point of the transmitter protocol stack to the instant when a frame reaches the corresponding reference point of the receiver protocol stack.

3.2.16 logical (functional) interface: An interface in which the semantic, syntactic, and symbolic attributes of information flows are defined. Logical interfaces do not define the physical properties of signals used to represent the information. It is defined by a set of primitives.

3.2.17 medium: The radio waves carrying the signals. Walls and other building components may affect the quality of the medium. Nodes communicating via the same medium may interfere with each other.

3.2.18 multicast: A type of communication where a node sends a MAC frame which is simultaneously received by one or more other nodes in the domain.

3.2.19 network: One or more, potentially overlapping, domains.

3.2.20 node: Any device that contains an ITU-T G.9959 transceiver. In the context of this Recommendation, the term 'node' without a qualifier means an 'ITU-T G.9959 node', and the term 'alien node' means a 'non-ITU-T G.9959 node'.

3.2.21 node ID: A unique identifier allocated to a node during its registration in a domain.

3.2.22 physical interface: An interface defined in terms of the physical properties of the signals used to represent the information transfer. A physical interface is defined by signal parameters like power (power spectrum density) and timing.

3.2.23 primitives: Variables and functions used to define logical interfaces and reference points.

3.2.24 reference point: A location in a signal flow, either logical or physical, that provides a common point for observation and or measurement of the signal flow.

3.2.25 symbol frame: A frame composed of bits of a single modulation symbol period.

3.2.26 symbol rate: The rate, in symbols per second, at which modulation symbols are transmitted by a node onto a medium. Symbol rate is calculated only for time periods of continuous transmission.

3.2.27 transmission overhead: A part of the available data rate used to support transmission over the media (e.g., preamble, inter-frame gaps, and silent periods).

3.2.28 unicast: A type of communication when a node sends the frame to another single node.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACK	Acknowledgement
AL	Always Listening
CCA	Clear Channel Assessment
CRC	Cyclic Redundancy Code
Dst	Destination
EOF	End of Frame
EHR	End Header
EIRP	Effective Isotropic Radiated Power
FCS	Frame Check Sequence
FER	Frame Error Rate
FL	Frequently Listening
FLN	Frequently Listening Node
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
HAN	Home Area Network
ID	Identification
IDB	Inter-Domain Bridge
ISM	Industrial, Scientific and Medical
LBT	Listen Before Talk
LLC	Logical Link Control
LSB	Least Significant Bit
MAC	Medium Access Control
MD	MAC Data
MD-SAP	MAC Data – Service Access Point
MDI	Medium-Dependent Interface
MFR	MAC Footer
MHR	MAC Header
MIB	Management Information Base
MLME	MAC Layer Management Entity
MLME-SAP	MAC Layer Management Entity – Service Access Point
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
MSDU	MAC Service Data Unit
NPDU	Network Layer Protocol Data Unit
NRZ	Non Return to Zero

OSI	Open System Interconnection
PD	PHY Data
PD-SAP	PHY Data – Service Access Point
PDU	Protocol Data Unit
PHY	Physical Layer
PLME-SAP	Physical Layer Management Entity – Service Access Point
PMI	Physical Medium-Independent interface
PPDU	PHY Protocol Data Unit
PPM	Parts Per Million
PSDU	PHY Service Data Unit
R1	Data Rate Type 1 (9.6 kbit/s)
R2	Data Rate Type 2 (40 kbit/s)
R3	Data Rate Type 3 (100 kbit/s)
RF	Radio Frequency
RX	Receive/Receiver
SAP	Service Access Point
SAR	Segmentation and Reassembly
SDU	Service Data Unit
SHR	Start Header
SOF	Start of Frame
Src	Source
TRX	Transceiver
TX	Transmit/Transmitter

5 Conventions

None.

6 Network architecture and reference models

6.1 Network architecture and topology

6.1.1 Basic principles of ITU-T G.9959 networking

The following are the basic principles of the ITU-T G.9959 network architecture:

1. The network is divided into domains:
   - The division of physical nodes into domains is logical.
   - Domains may fully or partially overlap as there is no physical separation.
   - The number of domains is limited by the 32-bit HomeID identifier.
   - Each domain is identified by a unique HomeID.
   - Nodes of different domains may communicate with each other via inter-domain bridges (IDB).

- Operation of different domains is handled by individual domain masters.

2. The domain is a set of nodes connected to the same medium:
   - One node in the domain operates as a domain master.
   - Each domain may contain up to 232 nodes (including the domain master).
   - Each node in the domain is identified by a NodeID that is unique inside the domain.
   - All nodes that belong to the same domain are identified by the same HomeID. A node can belong to only one domain.
   - Nodes of the same domain may communicate with each other either directly or via other nodes of the same domain.
3. Nodes of different ITU-T G.9959 domains:
   - Nodes in different domains can communicate via IDB. The IDB function is a bridging function associated with one dedicated node in each network domain.

The details of domain operation rules and the functionalities of domain master and endpoint nodes are beyond the scope of this Recommendation.

Alien domains and bridges to alien domains are beyond the scope of this Recommendation.

The scope of this Recommendation is limited to the PHY, MAC, segmentation and reassembly (SAR), and LLC layers of ITU-T G.9959 radiocommunication TRXs.

6.1.2 Network architecture and topology

An example architectural model of a network is presented in Figure 6-1. It comprises the local area network (LAN), IDB and home area networks (HAN). Each HAN is one ITU-T G.9959 domain.

Each HAN is connected to a LAN via an IDB.

NOTE – This architectural model is exclusively for reference purposes and does not limit the use of ITU-T G.9959 TRXs for other network architectures.

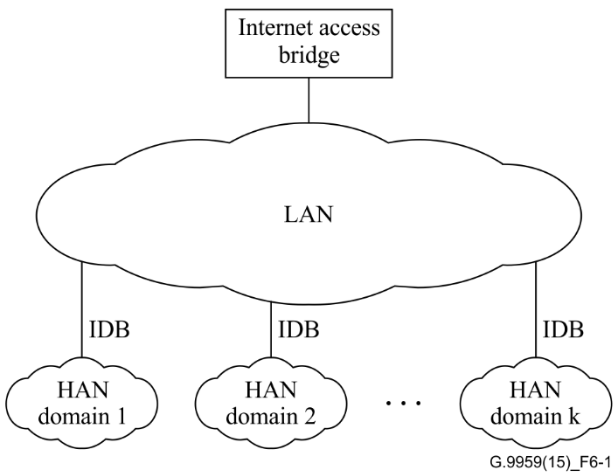

Figure 6-1 – Generic network architecture (example)

6.1.2.1 Generic LAN architecture

The LAN interconnects HAN domains via IDB. Each domain shall be associated with a particular set of nodes interfacing with the same medium.

The LAN may connect to alien domains. Alien domains may be bridged to ITU-T G.9959 domains using IDBs. The specification of LAN technologies and IDBs is beyond the scope of this Recommendation.

6.1.2.2 Generic HAN architecture

The HAN implements a single domain. Each domain is associated with a particular set of ITU-T G.9959 nodes. A particular node can belong to only one domain. Nodes of different HAN domains may communicate via IDBs.

Two or more HAN domains may overlap: nodes of overlapping domains "see" transmissions of each other and may interfere with each other at the radio frequency (RF) level.

6.2 Reference models

6.2.1 Protocol reference model of a TRX

The protocol reference model of a TRX is presented in Figure 6-2. It includes four reference points: the data link layer interface (DLI), the MAC layer interface (MLI), the physical medium-independent interface (PMI), and the medium-dependent interface (MDI).

The MDI is a physical interface defined in terms of the physical signals transmitted over a medium (clause 6.2.3).

The PMI is both medium independent and application independent. The PMI, MLI and DLI interfaces are defined as functional interfaces, in terms of sets of primitives exchanged across the interface.

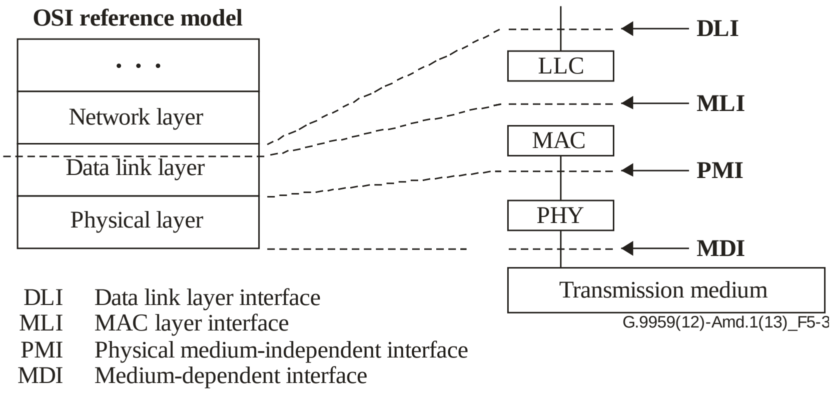

Figure 6-2 – Protocol reference model of an ITU-T G.9959 TRX

The logical link control (LLC) layer enables access of different instances of network protocol stacks to the MAC layer.

The medium access control (MAC) layer controls access of the node to the medium using the medium access protocols defined. The MAC layer also provides checksum protection to the MAC information.

The physical layer (PHY) provides bit rate adaptation (data flow control) between the MAC and PHY and adds PHY-related control and management overhead. The PHY layer provides encoding of the PHY frame content (header and payload) and modulates the encoded PHY frames for transmission over the medium.

6.2.2 Functional description of the interface

This clause contains the functional description of the TRX interfaces (reference points) based on the protocol reference model presented in Figure 6-3. The interfaces shown in Figure 6-3 are defined in this Recommendation.

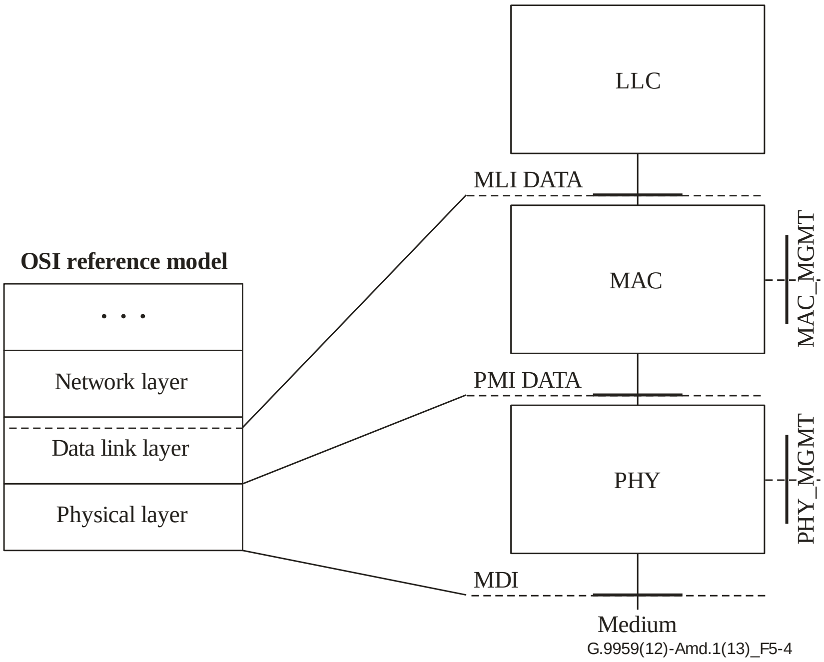

Figure 6-3 − TRX reference points related to PHY/MAC

The reference model in Figure 6-3 shows interfaces related to the application data path (MLI_DATA, PMI_DATA, and MDI) and the management interfaces between data and management planes of the PHY (PHY_MGMT). All interfaces are specified as reference points in terms of primitive flows exchanged between the corresponding entities. The description does not imply any specific implementation of the interfaces.

6.2.3 Functional model of a TRX

The functional model of a TRX is presented in Figure 6-4.

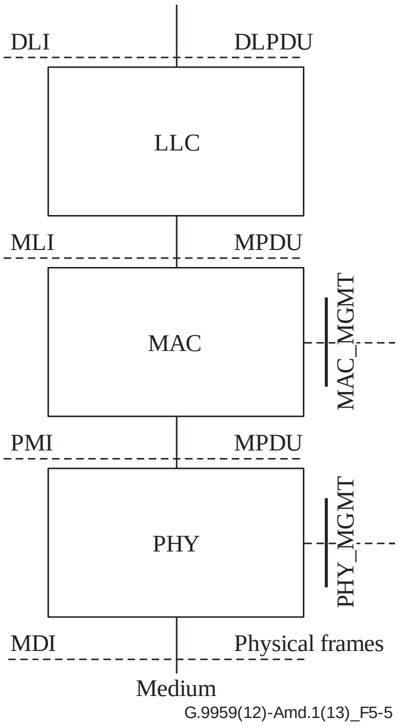

Figure 6-4 − Functional model of an ITU-T G.9959 TRX

The detailed description of the PHY layer is presented in clause 7.1. The detailed description of the MAC layer is presented in clause 8.1. The MAC layer interface (MLI) deviates from the open system interconnection (OSI) reference stack in that it exchanges MAC PDUs (MPDU) with the MAC layer rather than MAC service data units (MSDUs). This allows upper layers to perform mesh routing, segmentation, and IP header compression operations, based on information carried in the MPDU header. The detailed description of the functional model of the LLC layer is presented in clause 9.1.

6.3 Operation modes

G.9959 nodes may operate in two different receiving modes: always listening (AL) and frequently listening (FL). An ITU-T G.9959 node may operate in either of the two modes and dynamically alternate between the two modes.

In AL mode, the receiver stays on at all time.

In FL mode, the receiver is turned off most of the time. At a regular interval, the receiver is turned on for a short duration. This mode saves energy while still allowing for frame reception. The drawback of FL mode is an increased transmission latency due to the low receiver duty cycle.

7 PHY specification

7.1 PHY specification of 'sub 1 GHz' TRXs

7.1.1 General

The PHY layer defines modulation schemes, data rates, synchronization methods and a frame format for use in low-power, low-bandwidth control networks.

7.1.1.1 Features of the PHY layer

The PHY layer is responsible for the following tasks:

– assignment of RF profiles to physical channels
– activation and deactivation of the radio TRX
– transmission and reception
– clear channel assessment (CCA)
– frequency selection
– link quality assessment for received frames

The RF TRX shall be able to operate in a one, two or three channel configuration in license-free RF bands (refer to clause 7.1.2).

The PHY shall provide two services: (1) the PHY data (PD) service accessed via the PHY data service access point (PD-SAP) and (2) the PHY management service accessed via the physical layer management entity service access point (PLME-SAP). The PD service enables the transmission and reception of PHY protocol data units (PPDUs) over the physical radio channel. See clause 7.1.2.4 for a detailed description.

Constants and attributes that are specified and maintained by the PHY are written in *italics*. Constants have a general prefix of "aPhy", e.g., *aPhyMaxFrameSizeR1*, and are listed in Table 7-27. Attributes have a general prefix of "phy", e.g., *phyCurrentTxChannel*, and are listed in Table 7-28.

7.1.1.2 Data wrapping

The PHY layer inserts outgoing data into a physical RF frame format. When receiving frames the incoming data is extracted from the RF frame structure and forwarded for the upper layers. Refer to clause 7.1.3.1.

Data from the upper layers is passed to the PHY layer as a PHY service data unit (PSDU). The PSDU is prefixed by the PHY with a start header (SHR). The SHR contains the preamble sequence and start of frame (SOF) fields. The preamble sequence enables the RF receiver to achieve symbol synchronization. Finally, an end header (EHR) is appended for some data rates. The SHR, PSDU and EHR together form the PPDU.

7.1.1.3 Concept of service primitives

This clause provides a brief overview of the concept of service primitives (operations). Refer to [b-ITU-T X.210] for more detailed information. The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 7-1, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

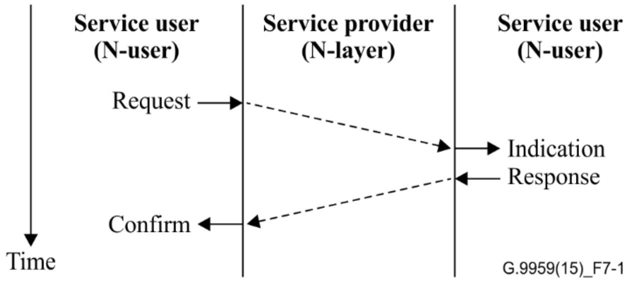

Figure 7-1 – Service primitives

The services are specified by describing the information flow between the N-user and the N-layer. This information flow is modelled 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 service access point (SAP) associated with an N-user. Service primitives convey the required information by providing a particular service. These service primitives are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation.

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 four generic types:

– Request: the request primitive is passed from the N-user to the N-layer to request that a service is initiated.

– Indication: the indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event.

– Response: the response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive.

– Confirm: the confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

7.1.2 Transceiver front-end specifications

7.1.2.1 RF profiles

An RF profile defines one or more data rates for use in a given radio channel. The definition of specific regional frequencies is outside the scope of this Recommendation.

The list of RF profiles is specified in Table 7-1. A TRX shall support up to 3 radio channels, each characterized by an RF profile. Each channel shall have a unique RF profile assigned to it. Depending on the actual region, RF Profiles may allow communication at one or more data rates and one or more channels.

Table 7-1 – RF profiles

RF profile	Region	Channel configurations	Centre frequency	R3	R2	R1	FL (Note)
0	n/a		n/a				
1	European Union	1,2	f_{EU1}	√			
2			f_{EU2}		√	√	
3			f_{EU2}		√		√
4	United States	1,2	f_{US1}	√			
5			f_{US2}		√	√	
6			f_{US2}		√		√
7	Hong Kong	1,2	f_{HK1}	√			
8			f_{HK1}		√	√	
9			f_{HK1}		√		√
10	Australia and New Zealand	1,2	f_{ANZ1}	√			
11			f_{ANZ2}		√	√	
12			f_{ANZ2}		√		√
13	Malaysia	1,2	f_{MY1}	√			
14			f_{MY1}		√	√	
15			f_{MY1}		√		√
16	India	1,2	f_{IN1}	√			
17			f_{IN1}		√	√	
18			f_{IN1}		√		√
19	Japan	3	f_{JP1}	√			√
20			f_{JP2}	√			√
21			f_{JP3}	√			√
22	Israel	1,2	f_{IL1}	√			
23			f_{IL1}		√	√	
24			f_{IL1}		√		√
25	Korea	3	f_{KR1}	√			√
26			f_{KR2}	√			√
27			f_{KR3}	√			√
28	Russia	1,2	f_{RU1}	√			

Table 7-1 – RF profiles

RF profile	Region	Channel configurations	Centre frequency	R3	R2	R1	FL (Note)
29	China	1,2	f_{RU1}		√	√	
30			f_{RU1}		√		√
31			f_{CN1}	√			
32			f_{CN1}		√	√	
33			f_{CN1}		√		√
NOTE – FL mode is only applicable for the indicated RF profiles. AL mode may be used with any RF profile.							

7.1.2.2 Data rates

The PHY shall comply with the data rate and accuracy requirements listed in Table 7-2.

Table 7-2 – Data rate and data rate accuracy

Data rate	Bit rate	Symbol rate	Accuracy
R1	9.6 kbit/s	19.2 kbaud	±27 parts per million (ppm)
R2	40 kbit/s	40 kbaud	±27 ppm
R3	100 kbit/s	100 kbaud	±27 ppm

7.1.2.3 Channel configurations

A compliant node shall operate in one of the channel configurations listed in Table 7-3.

Table 7-3 – channel configurations

Channel configuration	Number of channels	Data rate		
		R3	R2	R1
1	1	–	Ch B	Ch B
2	2	Ch A	Ch B	Ch B
3	3	Ch A Ch B Ch C	–	–

Table 7-1 and Table 7-3 shall be used in combination.

Example:

A node intended for the EU region may use configuration 2 which provides two alternative communication channels. The following RF profiles are available:

- RF profile 1 at the frequency f_{EU1} (Ch A) supporting data rate R3 in AL mode only.
- RF profile 2 at the frequency f_{EU2} (Ch B) supporting data rates R1 and R2 in AL mode only.
- RF profile 3 at the frequency f_{EU2} (Ch B) supporting data rate R2 in AL and FL mode.

This imaginary node runs on battery but has to support low-latency communication. Thus, FL mode is preferred. When listening or transmitting to other FL nodes, the node uses RF Profile 3 (Ch B). When sending to AL nodes, the node uses RF Profile 1 (Ch A) or RF Profile 2 (Ch B).

7.1.2.4 Modulation and encoding

The PHY shall employ frequency shift keying (FSK) for RF modulation at data rates R1 and R2. The PHY shall employ Gaussian frequency shift keying (GFSK) for RF modulation at data rate R3.

Manchester code shall be used for data symbol encoding at data rate R1 and non-return-to-zero (NRZ) shall be used for data symbol encoding at data rates R2 and R3.

The modulation and the coding format are summarized in Table 7-4.

Table 7-4 – Modulation and coding format

Data Rate	Modulation	Coding	Frequency offset	Separation	Symbols
R1	FSK	Manchester	20 kHz	40 kHz ±20%	Binary
R2	FSK	NRZ	0 kHz	40 kHz ±20%	Binary
R3	GFSK, BT = 0.6	NRZ	0 kHz	58 kHz ±20%	Binary

The mapping of NRZ symbols to the physical medium is given in Table 7-5:

Table 7-5 – NRZ symbol mapping

Symbol	Frequency
0	$f_{\text{center frequency}}$ + separation/2
1	$f_{\text{center frequency}}$ – separation/2

The mapping of Manchester symbols to the physical medium is given in Table 7-6:

Table 7-6 – Manchester symbol mapping

Symbol	Frequency
0	Transition from ($f_{\text{center frequency}} + f_{\text{offset}}$ – separation/2) to ($f_{\text{center frequency}} + f_{\text{offset}}$ + separation/2)
1	Transition from ($f_{\text{center frequency}} + f_{\text{offset}}$ + separation/2) to ($f_{\text{center frequency}} + f_{\text{offset}}$ – separation/2)

Table 7-6 refers to a frequency offset, f_{offset}. The frequency offset is specified in Table 7-4.

7.1.2.5 Transmitter (TX) and receiver requirements

Unless stated otherwise, all RF power measurements, either transmit or receive, shall be made at the antenna connector. The measurements shall be made with equipment that is either matched to the impedance of the antenna connector or corrected for any mismatch. For devices without an antenna connector, the measurements shall be interpreted as effective isotropic radiated power (EIRP) (i.e., a 0 dBi gain antenna), and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

7.1.2.5.1 Transmit frequency error

Frequency error is defined as the difference between the measured transmitted centre frequency and the actual regional centre frequency.

The frequency error shall not exceed ±27 ppm.

7.1.2.5.2 Transmit power adjustments (conducted)

The TX output power level shall conform to local regulations. The output power level allowed by local regulations is denoted as "nominal." Actual absolute power levels are out of scope of this Recommendation.

The TX shall be able to adjust its output power in steps of 2 dB in the range between the nominal transmit power down to (nominal output power – 10 dB).

The TX shall further be able to reduce its output power to at least (nominal output power – 20 dB).

7.1.2.5.3 Receiver sensitivity

To ensure a minimum RF link budget, the receiver shall be capable of receiving a standard test frame at a minimum power level.

The standard test frame and test conditions shall be as specified in Table 7-7.

The minimum receiver sensitivity for each data rate shall be as specified in Table 7-8.

Table 7-7 – Test conditions

Term	Definition	Conditions
Standard test frame	PHY frame used for testing sensitivity.	PHY frame with at least four bytes of random payload data.
Frame error rate (FER)	Average frame loss.	Average measured over standard test frames.
Receiver sensitivity	Threshold input signal power that yields a specific FER.	FER < 1% Power measured at antenna terminals. Interference not present.

Table 7-8 – Minimum receiver sensitivity

Rate	Minimum receiver sensitivity
R1	–95 dBm
R2	–92 dBm
R3	–89 dBm

7.1.2.5.4 Clear channel assessment

The PHY shall be able to perform a CCA with a threshold of –80 dBm. If the RF channel is found to be idle, the PHY may transmit its data.

In a given deployment, a listen before talk (LBT) operation based on CCA shall comply with actual regional RF regulatory requirements, e.g., listening period and threshold.

7.1.2.5.5 Receiver spurious requirement

A TRX shall limit its RF emissions when in receiver (RX) mode. Emissions near the centre frequency may affect the ability of other nearby devices to receive weak signals.

A receiver shall not emit more than –70 dBm within ±1 MHz from the centre frequency as shown in Figure 7-2. The measurement bandwidth shall be 100 kHz.

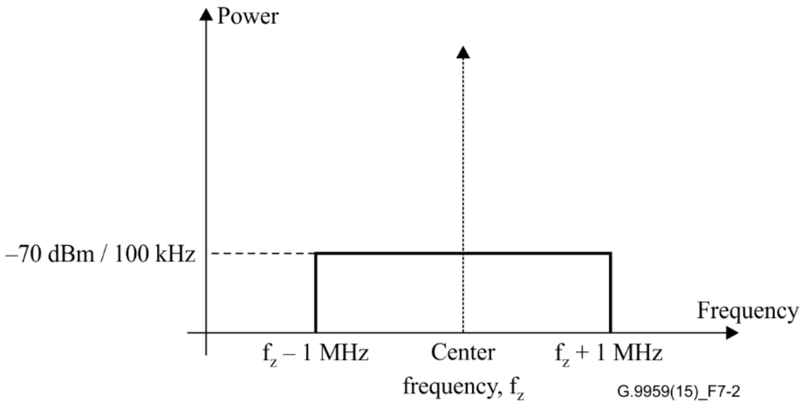

Figure 7-2 – Receiver spurious limit

7.1.2.5.6 Receiver blocking

Blocking is a measure of the capability of the receiver to receive the intended modulated signal without experiencing degradation due to the presence of another unwanted input signal.

A conforming implementation shall be able to pass a blocking test as described below for all data rates.

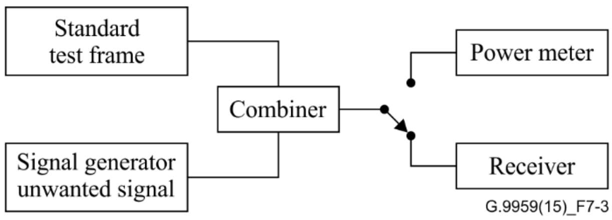

Figure 7-3 – Receiver blocking test definition

Method of measurement:

– Standard test frames shall be transmitted at the nominal frequency using the modulation specified for the actual data rate. Its power is adjusted down to a power level which is 3 dB higher than the sensitivity level defined in Table 7-8.

– The blocking test signal shall be a carrier transmitted at a specific offset frequency as defined in Table 7-9. The blocking test signal power shall be increased until the receiver experiences an FER that corresponds to the sensitivity level.

Limits:

Table 7-9 – Blocking limits

Frequency offset	Limits
±1 MHz	–44 dBm
±2 MHz	–34 dBm
±5 MHz	–27 dBm
±10 MHz	–25 dBm

7.1.2.5.7 Receiver saturation

The receiver saturation power level is the maximum power level, in decibels relative to 1 mW, present at the input of the receiver. A receiver shall meet the FER criterion in Table 7-7 while receiving at an input power level greater than or equal to 0 dBm to sustain "zero" distance between two or more devices.

7.1.2.5.8 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be measured from the trailing edge of the last transmitted symbol to the leading edge of the first symbol of the received preamble.

The TX-to-RX turnaround time shall be less than *aPhyTurnaroundTimeTXRX* (see Table 7-27).

Latency estimations shall be calculated on the 99th percentile of all latency measurements.

7.1.2.5.9 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured from the trailing edge of the last received symbol to the leading edge of the first transmitted preamble symbol.

The RX-to-TX turnaround time shall be more than *aPhyTurnaroundTimeRXTX* (see Table 7-27).

Latency estimations shall be calculated on the 99th percentile of all latency measurements.

7.1.3 PPDU format

The general PPDU frame structure is outlined in Figure 7-4. The frame format is depicted in the order in which it is transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from $k - 1$ (leftmost and most significant) down to 0 (rightmost and least significant), where the length of the field is k bits. Bytes within each multi-byte field are numbered from 1 (leftmost and most significant) up to $n - 1$ (rightmost and least significant), where the length of the field is n bytes. Bits within each byte are numbered from 7 (leftmost and most significant) down to 0 (rightmost and least significant).

7.1.3.1 General PHY frame format

The PPDU shall be formatted as illustrated in Figure 7-4.

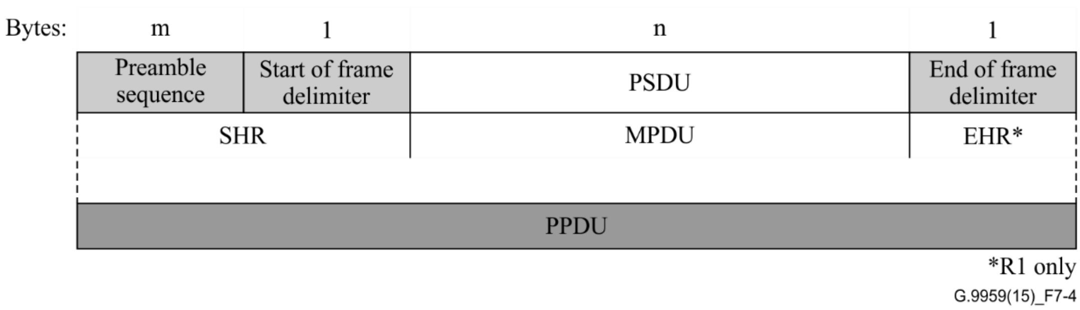

Figure 7-4 – PPDU format

7.1.3.2 Preamble field

The preamble field allows a receiver to obtain symbol synchronization. The preamble field shall be composed of a sequence of bytes containing the binary pattern "01010101". Figure 7-5 shows the logical bit waveform of the Manchester encoded preamble pattern for data rate R1.

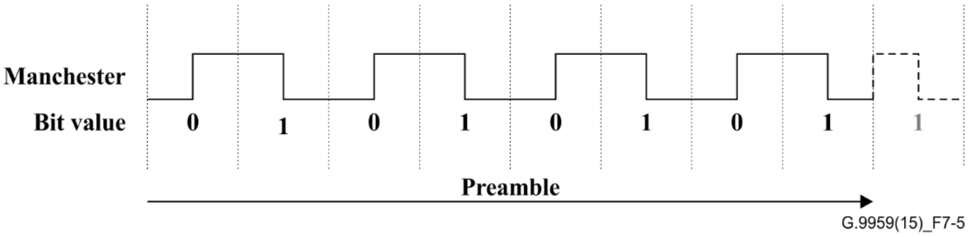

Figure 7-5 – Manchester encoded preamble pattern (R1)

Figure 7-6 shows the logical bit waveform of the NRZ encoded preamble pattern for data rates R2 and R3.

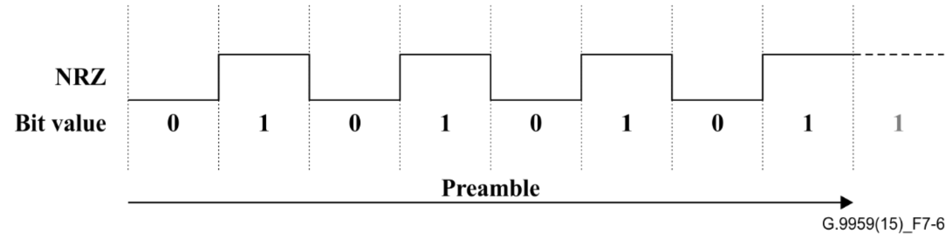

Figure 7-6 – NRZ encoded preamble pattern (R2, R3)

The preamble length shall comply with Table 7-10. The values allow a receiver to scan all channels and obtain synchronization at an arbitrary channel during the duration of the preamble.

Table 7-10 – Minimum Preamble length

Channel configurations	Rate	Minimum Preamble length in bytes		
		Singlecast/ broadcast	Multicast	Beam
1	R1 R2 R3	10 10 n/a	10 20 n/a	n/a 20 n/a
2	R1 R2 R3	10 10 40	10 20 40	n/a 20 n/a
3	R1 R2 R3	n/a n/a 24	n/a n/a 24	n/a n/a 8

7.1.3.3 Start of frame (SOF) field

The SOF is an 8-bit field terminating the preamble field and the start of the PSDU. The SOF shall be formatted as the logical bit pattern illustrated in Table 7-11.

Table 7-11 – Format of the SOF field

Bit number	7	6	5	4	3	2	1	0
Value	1	1	1	1	0	0	0	0

7.1.3.4 PSDU field

The PSDU field has a variable length and carries the data of the PHY frame.

7.1.3.5 End of frame (EOF) delimiter field

The EOF delimiter field shall be sent only when transmitting at data rate R1. The field shall carry a sequence of 8 Manchester code violations each denoted E. Each violation, E, shall be a symbol without transition. Refer to Figure 7-7.

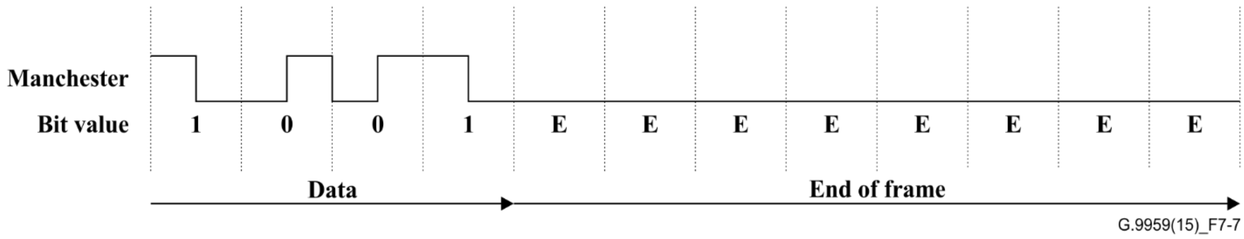

Figure 7-7 – EOF delimiter pattern (R1 only)

7.1.4 PHY service specifications

The PHY layer shall provide two services, accessed via two SAPs: the PD, accessed via the PHY data SAP (PD-SAP), and the PHY management service, accessed via the PHY layer management entity SAP (PLME-SAP). The PLME is responsible for maintaining a database of managed objects pertaining to the PHY. This database is referred to as the PHY management information base (MIB). Figure 7-8 shows the components and interfaces of the PHY.

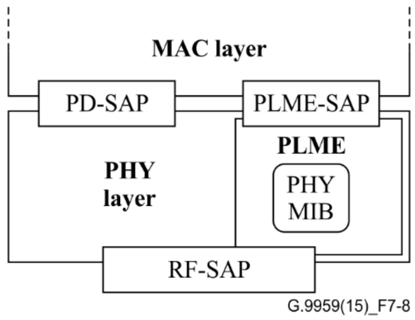

Figure 7-8 – PHY reference model

7.1.4.1 PD service

The PD-SAP supports the transport of MPDUs between peer MAC entities. Table 7-12 lists the primitives supported by the PD-SAP.

Table 7-12 – PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	clause 7.1.4.1.1	clause 7.1.4.1.2	clause 7.1.4.1.3

7.1.4.1.1 PD-DATA.request

The PD-DATA.request primitive requests the transfer of an MPDU (i.e., the PSDU) from the MAC entity to the PHY media.

7.1.4.1.1.1 Semantics of the PD request primitive

The semantics of the PD-DATA.request primitive shall be as follows:

```
PD-DATA.request    (
                   psduChannel,
                   psduRate,
                   psduLength,
                   psdu
                   )
```

Table 7-13 specifies the parameters for the PD-DATA.request primitive.

Table 7-13 – PD-DATA.request parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B, C according to the applied RF profile (see Tables 7-1 and 7-3)	Channel to use
psduRate	Enumeration	Data rate R1, R2, R3 according to the applied RF profile (see Tables 7-1 and 7-3)	Data rate to use
psduLength	Byte	≤ *aPhyMaxFrameSizeRx*, where Rx is R1, R2, or R3 depending on actual data rate	Number of bytes to transmit
psdu	Byte Array	–	PSDU to transmit

7.1.4.1.1.2 When generated

The PD-DATA.request primitive is generated by the MAC entity and issued to the PHY entity to request the transmission of an MPDU.

7.1.4.1.1.3 Effects on receipt

The receipt of the PD-DATA.request primitive by the PHY entity shall cause the transmission of the supplied PSDU. Provided the TX is enabled (TX_ON mode), the PHY shall construct a PPDU, containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission, it shall 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 mode) or if the TRX is disabled (TRX_OFF mode), the PHY entity shall issue the PD-DATA.confirm primitive with a status of RX_ON or TRX_OFF, respectively.

7.1.4.1.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of a PSDU from the MAC entity to the physical media.

7.1.4.1.2.1 Semantics of the PD confirm primitive

The semantics of the PD-DATA.confirm primitive shall be as follows:

```
PD-DATA.confirm    (
                   status
                   )
```

Table 7-14 specifies the parameters for the PD-DATA.confirm primitive.

Table 7-14 – PD-DATA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF	Transmission was successful Error: Currently receiving Error: Transmitter is disabled

7.1.4.1.2.2 When generated

The PD-DATA.confirm primitive is generated by the PHY entity and issued to the MAC entity in response to a PD-DATA.request primitive. The PD-DATA.confirm primitive shall return a status of either SUCCESS, indicating that the transmit request was successful, or an error code of RX_ON or TRX_OFF.

7.1.4.1.2.3 Effects on receipt

The PD-DATA.confirm primitive allows the MAC entity to take proper action when the transmission has been completed.

7.1.4.1.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of a PSDU from the PHY to the local MAC entity.

7.1.4.1.3.1 Semantics of the PD indication primitive

The semantics of the PD-DATA.indication primitive shall be as follows:

```
PD-DATA.indication    (
                      psduByte
                      )
```

Table 7-15 specifies the parameters for the PD-DATA.indication primitive.

Table 7-15 – PD-DATA.indication parameters

Name	Type	Valid range	Description
psduByte	Byte	–	One byte of the PSDU received by the PHY entity

7.1.4.1.3.2 When generated

The PD-DATA.indication primitive is generated by the PHY entity and issued to the MAC entity to transfer one received PSDU byte.

7.1.4.1.3.3 Effect on receipt

On receipt of the PD-DATA.indication primitive, the MAC entity is notified of the arrival of MPDU data. The MAC layer shall monitor incoming bytes until a complete MPDU has been received. The MAC layer shall use the MPDU Length field to determine the length of the MPDU. The MAC layer shall verify the frame check sequence (FCS) before issuing an MD-DATA.indication to higher layers.

7.1.4.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME and the PLME. Table 7-16 lists the primitives supported by the PLME-SAP.

Table 7-16 – PLME-SAP primitives

PLME-SAP Primitive	Request	Confirm	Indication	Response
PLME-SOF	–	–	clause 7.1.4.2.1	–
PLME-GET-CCA	clause 7.1.4.2.2	clause 7.1.4.2.3	–	–
PLME-GET	clause 7.1.4.2.4	clause 7.1.4.2.5	–	–
PLME-SET-TRX-MODE	clause 7.1.4.2.6	clause 7.1.4.2.7	–	–
PLME-SET	clause 7.1.4.2.8	clause 7.1.4.2.9	–	–

7.1.4.2.1 PLME-SOF.indication

The PLME-SOF.indication primitive indicates the reception of a SOF delimiter from the PHY to the MAC entity.

7.1.4.2.1.1 Semantics for the service primitive

The semantics of the PLME-SOF.indication primitive shall be as follows:

```
PLME-SOF.indication     (
                        psduChannel,
                        psduRate
                        )
```

Table 7-17 specifies the parameters for the PD-DATA.indication primitive.

Table 7-17 – PLME-SOF.indication parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B, C	Channel from which the PSDU was received
psduRate	Enumeration	Data rate R1, R2, R3	Data rate of the received PSDU

7.1.4.2.1.2 When generated

The PLME-SOF.indication primitive is generated by the PLME and issued to the MLME whenever a SOF delimiter is detected by the PHY.

7.1.4.2.1.3 Effect on receipt

The MAC entity is notified of the reception of a SOF delimiter. This information may be used by the MAC entity for preparing frame reception and inhibiting transmissions.

7.1.4.2.2 PLME-GET-CCA.request

The PLME-GET-CCA.request primitive requests a CCA for a specified channel.

7.1.4.2.2.1 Semantics for the service primitive

The semantics of the PLME-GET-CCA.request primitive shall be as follows:

```
PLME-GET-CCA.request    (
                        channel
                        )
```

Table 7-18 specifies the parameters for the PLME-GET-CCA.request primitive.

Table 7-18 – PLME-GET-CCA.request parameters

Name	Type	Valid range	Description
Channel	Enumeration	Channel A, B, or C according to the applied RF profile (see Tables 7-1 and 7-2)	The physical channel on which the CCA is to be performed

7.1.4.2.2.2 When generated

The PLME-GET-CCA.request primitive is generated by the MLME and issued to the PLME to query the availability of the specified channel.

7.1.4.2.2.3 Effect on receipt

On receipt of the PLME-GET-CCA.request primitive, the PLME should perform a CCA for the specified channel. When the operation is completed, the PLME shall issue a PLME-GET-CCA.confirm advertising the status.

7.1.4.2.3 PLME-GET-CCA.confirm

The PLME-GET-CCA.confirm primitive reports the result of a CCA request.

7.1.4.2.3.1 Semantics for the service primitive

The semantics of the PLME-GET-CCA.confirm primitive shall be as follows:

```
PLME-GET-CCA.confirm   (
                       status
                       )
```

Table 7-19 specifies the parameters for the PLME-GET-CCA.confirm primitive.

Table 7-19 – PLME-GET-CCA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	CCA_CLEAR, CCA_NOT_CLEAR, CCA_RX_OFF	Channel is ready for transmission Channel is not ready for transmission Receiver is disabled

7.1.4.2.3.2 When generated

The PLME-GET-CCA.confirm primitive shall be generated by the PLME in response to a PLME-GET-CCA.request primitive. The PLME-GET-CCA.confirm primitive may return the status of values CCA_CLEAR, CCA_NOT_CLEAR or CCA_RX_OFF. The CCA_RX_OFF status shall be returned if the TRX is not in RX mode (and thus, unable to perform a CCA).

7.1.4.2.3.3 Effect on receipt

The MLME is notified of the result of the CCA operation. This information may be used by the MAC entity for channel availability evaluation or for deciding whether to transmit now.

7.1.4.2.4 PLME-GET.request

The PLME-GET.request primitive requests the value of the specified PHY MIB attribute.

7.1.4.2.4.1 Semantics for the service primitive

The semantics of the PLME-GET.request primitive shall be as follows:

PLME-GET.request (
PhyMibAttribute
)

Table 7-20 specifies the parameters for the PLME-GET.request primitive.

Table 7-20 – PLME-GET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See Table 7-28	Identifier of the PHY MIB attribute

7.1.4.2.4.2 When generated

The PLME-GET.request primitive shall be generated by the MLME and issued to the PLME to request information from the PHY MIB.

7.1.4.2.4.3 Effect on receipt

On receipt of the PLME-GET.request primitive, the PLME should retrieve the value of the specified PHY MIB attribute.

7.1.4.2.5 PLME-GET.confirm

The PLME-GET.confirm primitive reports the result of a PHY MIB attribute request.

7.1.4.2.5.1 Semantics for the service primitive

The semantics of the PLME-GET.confirm primitive shall be as follows:

PLME-GET.confirm (
status,
PhyMibAttribute,
PhyMibAttributeValue
)

Table 7-21 specifies the parameters for the PLME-GET.confirm primitive.

Table 7-21 – PLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE	MIB value is available MIB value does not exist
PhyMibAttribute	Enumeration	Refer to Table 7-28	Identifier of the PHY MIB attribute
PhyMibAttributeValue	Various	Attribute specific	Value of the specified PHY MIB attribute

7.1.4.2.5.2 When generated

The PLME-GET.confirm primitive shall be generated by the PLME in response to a PLME-GET.request primitive.

If a non-existent PHY MIB attribute is requested, the PLME shall issue the PLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested PHY MIB attribute exists, the PLME shall issue the PLME-GET.confirm primitive with a status of SUCCESS as well as the MIB attribute identifier and its value.

7.1.4.2.5.3 Effect on receipt

On receipt of the PLME-GET.confirm primitive, the MLME shall be notified of the result of the PHY MIB attribute request. If the request was successful, the MLME may use the returned MIB attribute value.

7.1.4.2.6 PLME-SET-TRX-MODE.request

The PLME-SET-TRX-MODE.request primitive requests that the PHY entity changes the operating mode of the TRX. The TRX may be set to one of the modes outlined in Table 7-22.

7.1.4.2.6.1 Semantics for the service primitive

The semantics of the PLME-SET-TRX-MODE.request primitive shall be as follows:

```
PLME-SET-TRX-MODE.request   (
                            mode
                            )
```

Table 7-22 specifies the parameters for the PLME-SET-TRX-MODE.request primitive.

Table 7-22 – PLME-SET-TRX-MODE.request parameters

Name	Type	Valid values	Description
Mode	Enumeration	RX_ON, TRX_OFF, FORCE_TRX_OFF, TX_ON	Enable receiver if possible Disable receiver and transmitter if possible Force receiver and transmitter off Enable transmitter if possible

7.1.4.2.6.2 When generated

The PLME-SET-TRX-MODE.request primitive may be generated by the MLME and issued to the PLME to change the operational mode of the TRX.

7.1.4.2.6.3 Effect on receipt

On receipt of the PLME-SET-TRX-MODE.request primitive, the PHY should change the TRX operation mode.

If the PHY is busy receiving or transmitting, the PHY shall ignore the mode request.

If this primitive is issued with FORCE_TRX_OFF, the PHY shall set the TRX mode to TRX_OFF irrespective of the current mode.

7.1.4.2.7 PLME-SET-TRX-MODE.confirm

The PLME-SET-TRX-MODE.confirm primitive shall report the operating mode of the TRX in response to a PLME-SET-TRX-MODE.request.

If the TRX operation mode is changed, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with a status of SUCCESS.

If the TRX is requested to change to the current operation mode, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with a status advertising the current mode, i.e., RX_ON, TRX_OFF, or TX_ON.

If the TRX is requested to change to the RX_ON or TRX_OFF mode and the PHY is busy transmitting, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with the status BUSY_TX.

If the TRX is requested to change to the TX_ON or TRX_OFF mode and the PHY is busy receiving, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with the status BUSY_RX.

7.1.4.2.7.1 Semantics for the service primitive

The semantics of the PLME-SET-TRX-MODE.confirm primitive shall be as follows:

```
PLME-SET-TRX-MODE.confirm   (
                            status
                            )
```

Table 7-23 specifies the parameters for the PLME-SET-TRX-MODE.confirm primitive.

Table 7-23 – PLME-SET-TRX-MODE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF, TX_ON, BUSY_RX, BUSY_TX	Change request was successful Receiver is already enabled Receiver and TX are disabled TX is already enabled Receiver is enabled and busy TX is enabled and busy

7.1.4.2.7.2 When generated

The PLME-SET-TRX-MODE.confirm primitive is generated by the PLME and issued to the MLME.

7.1.4.2.7.3 Effect on receipt

The MLME is notified of the operating mode of the TRX.

The PLME-SET-TRX-MODE.confirm primitive may advertise the status BUSY_RX or BUSY_TX. This indicates that the request for a new operation mode was ignored.

7.1.4.2.8 PLME-SET.request

The PLME-SET.request primitive may be issued to request that the specified PHY MIB attribute is set to the specified value.

7.1.4.2.8.1 Semantics for the service primitive

The semantics of the PLME-SET.request primitive shall be as follows:

```
PLME-SET.request   (
                   PhyMibAttribute,
                   PhyMibAttributeValue
                   )
```

Table 7-24 specifies the parameters for the PLME-SET.request primitive.

Table 7-24 – PLME-SET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See clause 7.1.5.2	Identifier of the PHY MIB attribute
PhyMibAttributeValue	Various	Attribute specific	Value of the PHY MIB attribute

7.1.4.2.8.2 When generated

The PLME-SET.request primitive is generated by the MLME and issued to the PLME to set the specified PHY MIB attribute.

7.1.4.2.8.3 Effect on receipt

On receipt of the PLME-SET.request primitive, the PLME should set the specified PHY MIB attribute to the specified value.

If a non-existent PHY MIB attribute is specified, the PLME shall not change any MIB attribute.

If the specified value is invalid for the specified PHY MIB attribute, the PLME shall not change the PHY MIB attribute value.

If the specified PHY MIB attribute is changed, the change should take effect immediately.

7.1.4.2.9 PLME-SET.confirm

The PLME-SET.confirm primitive shall report the result of a requested PHY MIB attribute change.

7.1.4.2.9.1 Semantics for the service primitive

The semantics of the PLME-SET.confirm primitive shall be as follows:

```
PLME-SET.confirm    (
                    status,
                    PhyMibAttribute
                    )
```

Table 7-25 specifies the parameters for the PLME-SET.confirm primitive.

Table 7-25 – PLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER	The MIB parameter was set The MIB parameter does not exist Invalid value for this MIB parameter
PhyMibAttribute	Enumeration	Refer to Table 7-28	The identifier of the PHY MIB attribute

7.1.4.2.9.2 When generated

The PLME-SET.confirm primitive shall be generated by the PLME and issued to the MLME in response to a PLME-SET.request primitive.

If a non-existent PHY MIB attribute is specified, the PLME shall advertise a status of UNSUPPORTED_ATTRIBUTE.

If the specified value is invalid for the specified PHY MIB attribute, the PLME shall advertise a status of INVALID_PARAMETER.

If the specified PHY MIB attribute is updated, the PLME shall advertise a status of SUCCESS.

7.1.4.2.9.3 Effect on receipt

The MLME is notified of the result of the PHY MIB attribute change request. The MLME should verify that the status parameter advertises the status value SUCCESS. Refer to Table 7-25.

7.1.4.3 PHY enumerations description

Table 7-26 shows PHY enumeration values defined for the PHY layer.

Table 7-26 – PHY enumerations description

Enumeration	Description
BUSY	CCA operation detected a busy channel
BUSY_RX	TRX is receiving and cannot change its mode (unless forced)
BUSY_TX	TRX is transmitting and cannot change its mode (unless forced)
CCA_CLEAR	CCA operation detected a clear channel
FORCE_TRX_OFF	TRX shall be switched off; even if it is currently receiving or transmitting data
INVALID_PARAMETER	The specified parameter is out of range for the actual MIB attribute
CCA_NO_CLEAR	CCA operation detected a busy channel
RX_ON	Receiver is enabled (used for command as well as status)
CCA_RX_OFF	CCA request failed: The receiver is disabled
SUCCESS	Operation was successful
TRX_OFF	Command or Status indicating that the TRX is disabled
TX_ON	Command or Status indicating that the TX is enabled
UNSUPPORTED_ATTRIBUTE	The specified MIB attribute is not supported

7.1.5 PHY constants and MIB attributes

This clause specifies the constants and attributes relating to the PHY layer.

7.1.5.1 PHY constants

The PHY shall comply with the constants defined in Table 7-27.

Table 7-27 – PHY constants

Constants	Description	Value
PacketaPhyMaxFrameSizeR1	Maximum PSDU size for data rate R1	64 bytes
aPhyMaxFrameSizeR2	Maximum PSDU size for data rate R2	64 bytes
aPhyMaxFrameSizeR3	Maximum PSDU size for data rate R3	170 bytes
aPhyTurnaroundTimeTXRX	TX-to-RX maximum turnaround time (see clause 7.1.2.5.8).	1 ms
aPhyTurnaroundTimerRXTX	RX-to-TX minimum turnaround time (see clause 7.1.2.5.9).	1 ms

7.1.5.2 PHY MIB attributes

The PHY (MIB) comprises the attributes required to manage the PHY. Each of these attributes may be read or written using the PLME-GET.request and PLME-SET.request primitives, respectively. The attributes contained in the PHY MIB are presented in Table 7-28.

Table 7-28 – PHY MIB attributes

Attribute	Type	Range	Description
phyCurrentTxChannel	Enumeration	A, B, C (see Table 7-3)	TX channel to use
phyMapChannelA	Enumeration	Available RF profiles (see Table 7-1)	Apply RF profile to channel A
phyMapChannelB	Enumeration	Available RF profiles (see Table 7-1)	Apply RF profile to channel B
phyMapChannelC	Enumeration	Available RF profiles (see Table 7-1)	Apply RF profile to channel C
phyTransmitPower	Enumeration	Valid output power levels (refer to clause 7.1.2.5.2)	Transmit power to use

8 MAC layer specification

8.1 MAC specifications of the 'sub 1 GHz' TRXs

8.1.1 General

The MAC layer defines a half duplex protocol for acknowledged wireless communication in a low-cost control network. The MAC layer targets "soft” real-time applications which are not time critical in nature. The MAC layer supports on-demand communication to battery operated nodes.

MAC Protocol Data Units (MPDU) carry one small header in order to conserve bandwidth. While presented as one header, a few fields are used by higher layers. These fields are carried transparently and ignored by the MAC layer.

8.1.1.1 Features of the MAC layer

The features of the MAC layer are channel access, frame validation, acknowledged frame delivery, and retransmission.

The MAC is responsible for handling the following:

– domain identification (ID)
– node ID
– collision avoidance algorithm
– backoff algorithm
– automatic retransmission in case of transmission errors
– support for battery operation

The MAC layer provides two services: the MAC data (MD) service, accessed through the MAC layer data service access point (MD-SAP) and the MAC management service interfacing with the MAC layer management entity service access point (MLME-SAP). The MD service enables reliable transmission and reception of MPDUs across the PD service.

Constants and attributes that are specified and maintained by the MAC are written in the text of this clause in *italics*. Constants have a general prefix of "aMac", e.g., *aMacMaxMSDUSizeR1*, and are listed in Tables 8-18 and 8-19. Attributes have a general prefix of "mac", e.g., *macHomeID*, and are listed in Table 8-20.

8.1.1.2 Bootstrapping

A unique 32-bit identifier called the HomeID is used to identify individual domains.

NodeIDs are unique within a given domain. A NodeID is an 8-bit short address. A primary node hands out the HomeID and unique NodeIDs to all other nodes included in the domain.

8.1.1.3 Concept of service primitives

This clause provides a brief overview of the concept of service primitives (operations). Refer to [b-ITU-T X.210] for more detailed information.

The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 8-1, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

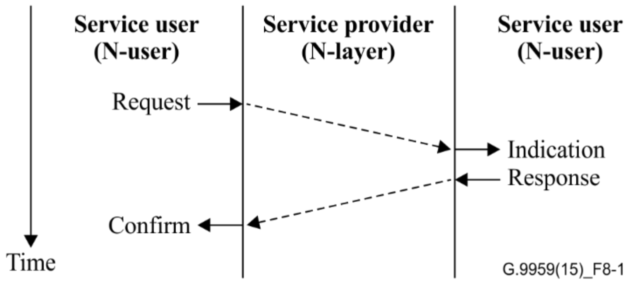

Figure 8-1 – Service primitives

The services are specified by describing the information flow between the N-user and the N-layer. This information flow is modelled 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 SAP associated with an N-user. Service primitives convey the required information by providing a particular service. These service primitives are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation.

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 four generic types:

– Request: the request primitive is passed from the N-user to the N-layer to request that a service is initiated.
– Indication: the indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event.
– Response: the response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive.
– Confirm: the confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

8.1.1.4 Functional overview

The MAC features of an ITU-T G.9959 network include data transfer model, frame structure, robustness and power consumption.

8.1.1.4.1 MPDU formats

A number of MPDU formats are defined.

Figure 8-2 shows the structure of the general MPDU. The MSDU contains the payload data from higher layers. The MSDU is prepended with a MAC header (MHR) and appended with a MAC footer (MFR). The MHR, MSDU, and MFR together constitute the MPDU.

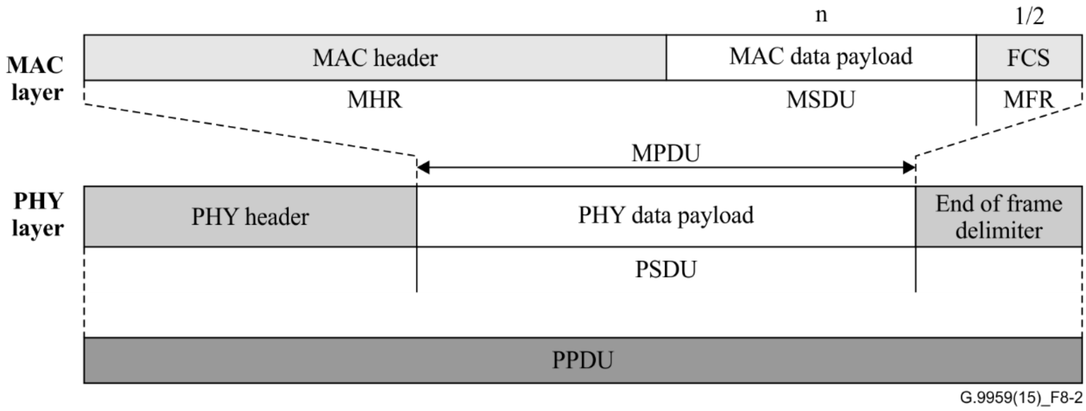

Figure 8-2 – General MPDU structure

The MPDU is passed to the PHY layer as a PSDU.

8.1.1.4.1.1 Singlecast MPDU

The singlecast MPDU uses the general frame structure (Figure 8-2).

8.1.1.4.1.2 Acknowledgement (ACK) MPDU

The ACK MPDU uses the general MPDU structure (Figure 8-2). The MSDU may have a length of zero bytes when used for acknowledgement.

8.1.1.4.1.3 Multicast MPDU

A multicast header explicitly identifies each target node via a dedicated multicast addressing bitmask.

8.1.1.4.2 Network robustness

The ITU-T G.9959 MAC layer employs various mechanisms to ensure robustness in data transmission. These mechanisms of the MAC layer are backoff algorithm, frame ACK, data verification, and frame retransmission.

8.1.1.4.2.1 CCA

A node shall query the availability of the channel from the PHY layer before transmitting. If the channel is found to be idle, the node may transmit its data. If the channel is found busy, the node shall wait for idle channel before transmitting.

8.1.1.4.2.2 ACK

A successful reception and validation of an MPDU may be confirmed with an ACK MPDU. If the destination (Dst) node receives an MPDU containing bit errors, the message shall not be acknowledged. An ACK request shall be used to indicate the need for ACK.

8.1.1.4.2.3 Retransmission

If two or more nodes transmit MPDUs simultaneously, a collision may occur and the MPDUs may not reach their destinations.

If the source (Src) node does not receive an ACK, it shall assume that the transmission was unsuccessful and in response it shall retry the MPDU transmission up to *aMacMaxFrameRetries* times.

In order to avoid another collision, each TX shall delay a retransmission by a random delay.

8.1.1.4.2.4 Multi-hop routing

The MAC layer defines MPDU header fields for support of multi-hop routing. Routing is outside the scope of this Recommendation.

8.1.1.4.2.5 Data validation

An 8-bit non-correcting FCS mechanism is employed to detect bit errors for data rates R1 and R2. A 16-bit non-correcting FCS mechanism is employed to detect bit errors for data rate R3. Refer to clause 7.1.2.2 for details on data rates.

8.1.1.4.3 Power consumption considerations

One category of battery-powered nodes spend most of their operational life in sleep mode. Such nodes may periodically wake up and poll other nodes to get pending messages. The PHY layer AL mode is used for listening during wake-up polling.

Other low-power nodes may require a more responsive behaviour than can be achieved via periodic wake-up. Such nodes may use the PHY FL mode for incoming messages.

8.1.1.4.3.1 Communication with a frequently listening node (FLN)

Battery-powered devices may need to be reachable at any time. Nodes that are listening at regular intervals are said to operate in FL mode. The PHY layer provides an extended preamble sequence that allows an FLN to operate at a very low duty cycle while still being reachable.

8.1.2 MAC layer service specification

The MAC layer provides an interface between higher layers, typically the network layer, and the PHY layer. The MLME provides the service interfaces through which MAC layer management functions may be invoked. The MLME is responsible for maintaining a database of managed objects pertaining to the MAC layer. This database is referred to as the MAC MIB. Figure 8-3 depicts the components and interfaces of the MAC layer.

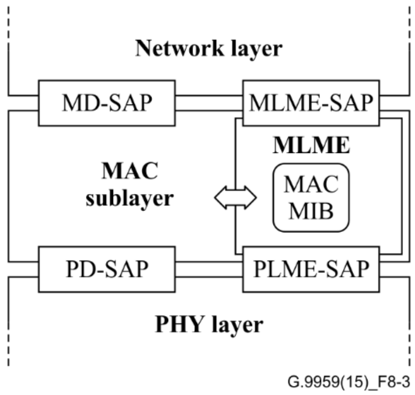

Figure 8-3 – MAC layer reference model

The MAC layer shall provide two services to the network layer, accessed through two SAPs:

– The MD service, accessed through the MD-SAP;

– The MAC management service, accessed through the MLME-SAP.

8.1.2.1 MD service

The MD-SAP supports the transport of a network layer protocol data unit (NPDU) (i.e., the MSDU) from the network layer to the PHY layer. Table 8-1 lists the primitives supported by the MD-SAP.

Table 8-1 – MD-SAP primitives

MD-SAP Primitive	Request	Confirm	Indication
MD-DATA	8.1.2.1.1	8.1.2.1.2	8.1.2.1.3

8.1.2.1.1 MD-DATA.request

The MD-DATA.request primitive requests the transfer of an NPDU (i.e., MSDU) from the network layer to the PHY entity.

8.1.2.1.1.1 Semantics of the service primitive

The semantics of the MD-DATA. request primitive shall be as follows:

```
MD-DATA.request  (SrcHomeID,
                  SrcNodeID,
                  DstNodeID,
                  msduLength,
                  msdu,
                  SequenceNumber,
                  TxType,
                  TxOptions,
                  BeamOption,
                  RateOption,
                  ChannelOption,
                  MulticastOption
                  )
```

Table 8-2 specifies the parameters for the MD-DATA.request primitive.

Table 8-2 – MD-Data.request parameters

Name	Type	Valid range	Description
SrcHomeID	Byte Array	0x00000000..0xFFFF FFFF	Domain identifier to use
SrcNodeID	Byte	0x00..0xE8	Source node identifier to use: 0x00: Uninitialized NodeID 0x01..0xE8: NodeID 0xE9..0xFE: Reserved
DstNodeID	Byte	0x00..0xFF	Destination node identifier to use: 0x00: Reserved 0x01..0xE8: NodeID 0xE9..0xFE: Reserved 0xFF: Broadcast NodeID
msduLength	Byte	≤ *aMacMaxMSDU SizeX*	Number of bytes to transmit
msdu	Byte Array	–	MSDU to transmit
SequenceNumber	Byte	–	Unique number of this frame The same value shall be used for retransmissions
TxType	Byte	0x01..0x03	Transmission type to use: 0x01 = Singlecast or broadcast transmission

Table 8-2 – MD-Data.request parameters

Name	Type	Valid range	Description
			0x02 = Multicast transmission 0x03 = Acknowledged transmission
TxOptions	Byte	0x01, 0x02, 0x04	Transmission options for this MSDU The control word is formed as a bitwise OR of one or more of the following values: 0x01 = Routed frame transmission 0x02 = Acknowledged transmission 0x04 = Low power transmission (Note 1)
BeamOption	Byte	0x00, 0x02..0x04	Battery support options 0x00: No beam 0x02: Short continuous beam 0x03: Long continuous beam 0x04: Fragmented beam
ChannelOption	Enumeration	Channel A,B, C (Note 2)	Channel to use
RateOption	Enumeration	Data Rate R1, R2, R3 according to the applied RF profile (see Tables 7-1, 7-2 and 7-3)	Data rate to use
MulticastOption	Byte + Bitmap (30 bytes)	–	If TxType is "Multicast transmission", this bitmap indicates the intended recipients
NOTE 1 – Bit 2 (0x04) controls which transmission power level the MAC layer requests from the PHY layer. NOTE 2 – Channels A, B, and C are preconfigured to RF profiles from the list presented in clause 7.1.2.1 via MAC MIB attributes defined in Table 8-20. Channel configuration MAC MIB profiles map directly to PHY MIB profiles.			

8.1.2.1.1.2 When generated

The MD-DATA.request primitive is generated by the network layer and issued to the MAC entity to request the transfer of an NPDU (i.e., MSDU).

The TxOptions.LowPower is an option targeting the PHY layer. The option shall make the MAC request the PHY power level defined by the MAC constant aMacTxPhyPowerLevelLow instead of aMacTxPhyPowerLevelNormal.

8.1.2.1.1.3 Effects on receipt

The receipt of the MD-DATA.request primitive by the MAC entity shall cause the transmission of the supplied MSDU.

The MAC entity builds an MPDU to transmit from the supplied parameters. The TxOptions parameters indicate optional parameters on how the MAC layer data service transmits the supplied MSDU.

The MAC entity shall check for PHY CCA (see clause 8.1.5.1.1). If the PHY PLME-GET-CCA primitive returns TRUE, the MAC entity shall enable the TX by issuing the PLME-SET-TRX-MODE.request primitive with a mode of TX_ON to the PHY. On receipt of the PLME-SET-TRX-MODE.confirm primitive with a status of either SUCCESS or TX_ON, the constructed MPDU may then be transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-

DATA.confirm primitive, the MAC entity shall disable the TX by issuing the PLME-SET-TRX-MODE.request primitive to the PHY.

If the TxOptions parameter specifies that acknowledged transmission is required, the MAC entity shall enable its receiver immediately following the transmission of the MPDU and wait for an ACK for at least *aMacMinAckWaitDuration* symbols. If the MAC entity does not receive an ACK within this time, it shall retransmit the MPDU one or more times as defined by *aMacMaxFrameRetries*. If the MAC entity still does not receive an ACK, it shall discard the MSDU and issue the MD-DATA.confirm primitive with a status of NO_ACK.

If the MPDU was successfully transmitted the MAC entity shall issue the MD-DATA.confirm primitive with a status of SUCCESS.

If the MPDU could not be transmitted due to a busy channel (clause 8.1.5.1.1) the MAC entity shall issue the MD-DATA.confirm primitive with a status of NO_CCA.

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

If the MSDU length is longer than *aMacMaxMSDUSizeX* the MAC entity shall issue the MD-DATA.confirm primitive with a status of FRAME_TOO_LONG.

The MD-Data.request parameters are used to construct the MHR see clause 8.1.3.

8.1.2.1.2 MD-DATA.confirm

The MD-DATA.confirm primitive confirms the transmission of an MSDU.

8.1.2.1.2.1 Semantics of the service primitive

The semantics of the MD-DATA.confirm primitive shall be as follows:

```
MD-DATA.confirm    (
                   status
                   )
```

Table 8-3 specifies the parameters for the MD-DATA.confirm primitive.

Table 8-3 – MD-Data.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_ACK, INVALID_PARAMETER, NO_CCA, FRAME_TOO_LONG	Transmission was successful Transmission was not acknowledged Transmission aborted; invalid parameter Transmission aborted; no clear channel Transmission aborted; frame too long

8.1.2.1.2.2 When generated

The MD-DATA.confirm primitive is generated by the MAC entity in response to an MD-DATA.request primitive. The MD-DATA.confirm primitive shall return a status as defined in Table 8-3.

The primitive may be delayed if the MAC has to wait for an ACK. If the frame is transmitted using beaming (refer to clauses 8.1.3.11 and 8.1.3.13), the delay may be more than one second.

8.1.2.1.2.3 Effects on receipt

The MD-DATA.confirm primitive allows higher layers to take proper action whether the transmission was successful or unsuccessful.

8.1.2.1.3 MD-DATA.indication

The MD-DATA.indication primitive indicates the reception of an MSDU from the MAC entity to higher layer entities.

8.1.2.1.3.1 Semantics of the service primitive

The semantics of the MD-DATA.indication primitive shall be as follows:

```
MD-DATA.indication      (
                        FrmType,
                        SrcHomeID,
                        SrcNodeID,
                        DstNodeID,
                        msduLength,
                        msdu,
                        SequenceNumber,
                        ChannelOption,
                        RateOption,
                        MulticastOption
                        )
```

Table 8-4 specifies the parameters for the MD-DATA.indication primitive.

FrmType indicates if the data delivered to the network layer is a data frame or a beam fragment. Refer to clause 8.1.3.10 for details on beam fragments.

Table 8-4 – MD-DATA.indication parameters

Name	Type	Valid range	Description
FrmType	Enumeration	0x01..0xFF	Code indicating the frame type: 0x00: Uninitialized type 0x01: Singlecast frame 0x02: Multicast frame 0x03: ACK frame 0x04..0x07: Not to be used (Note) 0x08: Routed frame 0x09: Beam fragment 0x0A..0xFF: Reserved
SrcHomeID	Byte Array	0x00000000.. 0xFFFFFFFF The value 0x00000000 is reserved for un-initialized legacy nodes. HomeID 0x00000000 shall not be used for any domain.	The 4-byte domain identifier of the node from which the MSDU was received
SrcNodeID	Byte	0x00..0xFF The value 0x00 is reserved for un-initialized nodes.	NodeID of the originating node 0x00: Uninitialized NodeID 0x01..0xE8: NodeID 0xE9..0xFF: Reserved
DstNodeID (Not applicable	Byte	0x00..0xFF	NodeID of the destination node 0x00: Reserved

Table 8-4 – MD-DATA.indication parameters

Name	Type	Valid range	Description
for multicast frame type)			0x01..0xE8: NodeID 0xE9..0xFE: Reserved 0xFF: Broadcast NodeID
msduChannel	Enumeration	Channel A, B, C	The channel from which the MSDU was received
msduRate	Enumeration	Data rate R1, R2, R3	The bit rate of the received MSDU
msduLength	Byte	≤ *aMacMaxMSDUSizeX*	The number of received bytes
msdu	Byte Array	–	The received MSDU
SequenceNumber	Byte	–	The unique number of this frame. Use same value for retransmissions
MulticastOption	Byte + Bitmap (30 bytes)	–	If TxType is "Multicast transmission", this bitmap indicates the intended recipients
NOTE – These values shall not be used due to backwards compatibility.			

In case of a beam frame, the SrcHomeID field shall be formatted as specified in clause 8.1.3.10.

8.1.2.1.3.2 When generated

The MD-DATA.indication primitive is generated by the MAC entity on receipt of a frame from the PHY layer. If the frame checksum is valid, the frame shall be forwarded to the network layer.

8.1.2.1.3.3 Effects on receipt

The network layer is notified of the arrival of data.

Beam fragments shall also be forwarded to the network layer, which may then forward the beam fragment to higher layers. Higher layers of a node in FL mode may decide to re-enable sleep mode if the NodeID of a beam fragment is intended for another node.

8.1.2.1.4 Data service message sequence charts

Figure 8-4 illustrates the sequence of messages necessary for a successful data transfer between two nodes.

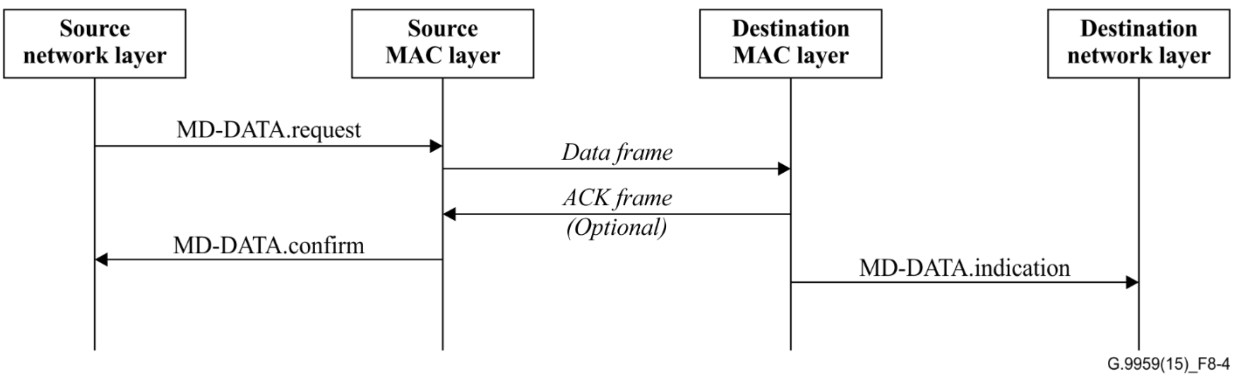

Figure 8-4 – MD service message sequence chart

8.1.2.2 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 8-5 summarizes the primitives supported by the MLME through the MLME-SAP interface.

The primitives are discussed in the clauses referenced in the table.

Table 8-5 – MLME-SAP primitives

MLME-SAP Primitive	Request	Confirm	Indication	Response
MLME-GET	8.1.2.2.1	8.1.2.2.2	–	–
MLME-SET	8.1.2.2.3	8.1.2.2.4	–	–
MLME-RESET	8.1.2.2.5	8.1.2.2.6	–	–

8.1.2.2.1 MLME-GET.request

The MLME-GET.request primitive requests information about a given MAC MIB attribute.

8.1.2.2.1.1 Semantics of the service primitive

The semantics of the MLME-GET.request primitive shall be as follows:

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

Table 8-6 specifies the parameters for the MLME-GET.request primitive.

Table 8-6 – MLME-GET.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table 8-20	The identifier of the MAC MIB attribute to read.

8.1.2.2.1.2 When generated

The MLME-GET.request primitive is generated by the next higher layer and issued to the MLME to obtain information from the MAC MIB.

8.1.2.2.1.3 Effects on receipt

On receipt of the MLME-GET.request primitive, the MLME shall retrieve the requested MAC MIB attribute from its database. If the identifier of the MAC MIB attribute is not found in the database, the MLME shall issue the MLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested MAC MIB attribute is successfully retrieved, the MLME shall issue the MLME-GET.confirm primitive with a status of SUCCESS and the MAC MIB attribute value.

8.1.2.2.2 MLME-GET.confirm

The MLME-GET.confirm primitive reports the result of a MAC MIB attribute request.

8.1.2.2.2.1 Semantics of the service primitive

The semantics of the MLME-GET.confirm primitive shall be as follows:

```
MLME-GET.confirm   (
                   status,
```

MacMibAttribute,
MacMibAttributeValue
)

Table 8-7 specifies the parameters for the MLME-GET.confirm primitive.

Table 8-7 – MLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the request for MAC MIB attribute information
MacMibAttribute	Integer	See Table 8-20	The identifier of the MAC MIB attribute that was read
MacMibAttributeValue	Various	Attribute specific, see Table 8-20	The value of the indicated MAC MIB attribute that was read

8.1.2.2.2.2 When generated

The MLME-GET.confirm primitive is generated by the MLME and issued to the 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 MAC MIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE.

8.1.2.2.2.3 Effects on receipt

The MLME-GET.confirm primitive reports the result of the MAC MIB attribute request. If the request was successful, the requester may use the returned MIB attribute value.

8.1.2.2.3 MLME-SET.request primitives

The MLME-SET.request primitive may be used to request that the specified MAC MIB attribute is set to the specified value.

8.1.2.2.3.1 Semantics of the service primitive

The semantics of the MLME-SET.request primitive shall be as follows:

MLME-SET.request (
MacMibAttribute,
MacMibAttributeValue
)

Table 8-8 specifies the parameters for the MLME-SET.request primitive.

Table 8-8 – MLME-SET.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table 8-20	The identifier of the MAC MIB attribute to write.
MacMibAttributeValue	Various	Attribute specific, see Table 8-20	The value to write to the indicated MAC MIB attribute.

8.1.2.2.3.2 When generated

The MLME-SET.request primitive is generated by the next higher layer and issued to the MLME to set the specified MAC MIB attribute.

8.1.2.2.3.3 Effects on receipt

On receipt of the MLME-SET.request primitive, the MLME should set the specified MAC MIB attribute to the specified value.

If a non-existent MAC MIB attribute is specified, the MLME shall not change any MIB attribute.

If the specified value is invalid for the specified MAC MIB attribute, the MLME shall not change the MAC MIB attribute value.

If the specified MAC MIB attribute is changed, the change should take effect immediately.

8.1.2.2.4 MLME-SET.confirm

The MLME-SET.confirm primitive shall report the result of the requested MAC MIB attribute change.

8.1.2.2.4.1 Semantics of the service primitive

The semantics of the MLME-SET.confirm primitive shall be as follows:

```
MLME-SET.confirm    (
                    status,
                    MacMibAttribute
                    )
```

Table 8-9 specifies the parameters for the MLME-SET.confirm primitive.

Table 8-9 – MLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE INVALID_PARAMETER	Attribute successfully changed Attribute is not supported Attribute value is invalid
MacMibAttribute	Integer	Table 8-20	The identifier of the MAC MIB attribute that was written.

8.1.2.2.4.2 When generated

The MLME-SET.confirm primitive shall be generated by the MLME and issued to the next higher layer in response to an MLME-SET.request primitive.

If a non-existent MAC MIB attribute is specified, the MLME shall advertise a status of UNSUPPORTED_ATTRIBUTE.

If the specified value is invalid for the specified MAC MIB attribute, the MLME shall advertise a status of INVALID_PARAMETER.

If the specified MAC MIB attribute is updated, the MLME shall advertise a status of SUCCESS.

8.1.2.2.4.3 Effects on receipt

The next higher layer is notified of the result of the MAC MIB attribute change request. The next higher layer should verify that the status parameter advertises the status value SUCCESS. Refer to Table 8-9.

8.1.2.2.5 MLME-RESET.request

The MLME-RESET.request primitive allows the next higher layer to request that the MLME performs a reset operation.

8.1.2.2.5.1 Semantics of the service primitive

The semantics of the MLME-RESET.request primitive shall be as follows:

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

Table 8-10 specifies the parameter for the MLME-RESET.request primitive.

Table 8-10 – MLME-RESET.request parameters

Name	Type	Valid range	Description
SetDefaultMIB	Boolean	TRUE FALSE	Reset MAC entity and all MIB attributes Reset MAC entity state machine only

8.1.2.2.5.2 When generated

The MLME-RESET.request primitive is generated by the next higher layer and issued to the MLME to request a reset of the MAC entity to its initial conditions. The MLME-RESET.request primitive is issued when a node is excluded from a domain.

8.1.2.2.5.3 Effects on receipt

On receipt of the MLME-RESET.request primitive, the MLME shall issue the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF. On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MAC entity shall then be set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultMIB parameter is set to TRUE, the MAC MIB attributes are set to their default values.

If the PLME-SET-TRX-STATE.confirm primitive is successful, the MLME shall issue the MLME-RESET.confirm primitive with the status of SUCCESS. Otherwise, the MLME shall issue the MLME-RESET.confirm primitive with the status of DISABLE_TRX_FAILURE.

8.1.2.2.6 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

8.1.2.2.6.1 Semantics of the service primitive

The semantics of the MLME-RESET.confirm primitive shall be as follows:

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

Table 8-11 specifies the parameter for the MLME-RESET.confirm primitive.

Table 8-11 – MLME-RESET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS DISABLE_TRX_FAILURE	MAC entity successfully reset Transceiver could not be reset

8.1.2.2.6.2 When generated

The MLME-RESET.confirm primitive is generated by the MLME and issued to the next higher layer in response to an MLME-RESET.request primitive and following the receipt of the PLME-SET-TRX-STATE.confirm primitive.

8.1.2.2.6.3 Effects on receipt

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of the request to reset the MAC entity. This primitive shall return a status as defined in Table 8-11.

8.1.2.3 MAC enumeration description

This clause explains the meaning of the enumerations used in the primitives defined in the MAC layer specification. Table 8-12 shows a description of the MAC enumeration values.

Table 8-12 – MAC enumerations description

Enumeration	Description
SUCCESS	The requested operation was completed successfully
DISABLE_TRX_FAILURE	The attempt to disable the transceiver has failed
FALSE	Control flag value for disabling a property
FRAME_TOO_LONG	The frame has a length that is greater than *aMacMaxMSDUSizeX*
INVALID_PARAMETER	A parameter in the primitive is out of the valid range
NO_ACK	No acknowledgement was received after *aMacMaxFrameRetries*
NO_CCA	No clear channel access was possible after a period of *macCCARetryDuration*
TRUE	Control flag value for enabling a property
UNSUPPORTED_ATTRIBUTE	A SET/GET request was issued with the identifier of a MAC MIB attribute that is not supported

8.1.3 MPDU format

This clause specifies the format of the MPDU. Each MPDU consists of the following basic components:

1. A MHR, which comprises addresses, frame control and length information;
2. A MD payload, of variable length, which contains information specific to the frame type;
3. An MFR, which contains a FCS.

The MPDU is defined as a sequence of fields. All MPDU formats in this clause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from $k - 1$ (leftmost and most significant) down to 0 (rightmost and least significant), where the length of the field is k bits. Bytes within each multi-byte field are numbered from 1 (leftmost and most significant) up to $n - 1$ (rightmost and least significant), where the length of the field is n bytes. Bits within each byte are numbered from 7 (leftmost and most significant) down to 0 (rightmost and least significant).

The general MPDU format comprises the fields MHR, Data payload and MFR. The general MPDU shall be formatted as illustrated in Figure 8-5.

The MPDU formats for channel configurations 1, 2, and channel configuration 3 are explained in the following.

Singlecast MPDUs shall use the formats outlined in Figures 8-5 and 8-6.

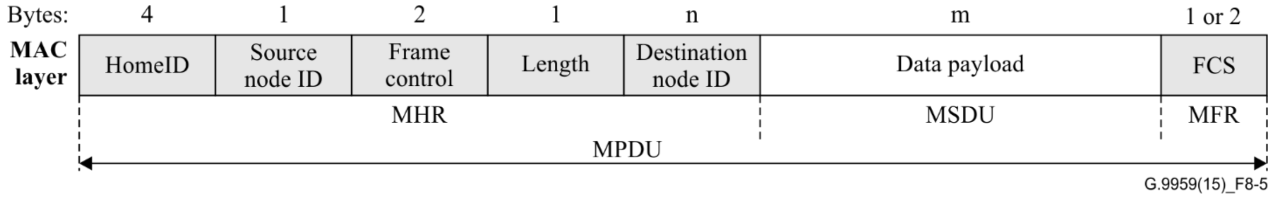

Figure 8-5 – General MPDU format (Channel configurations 1, 2)

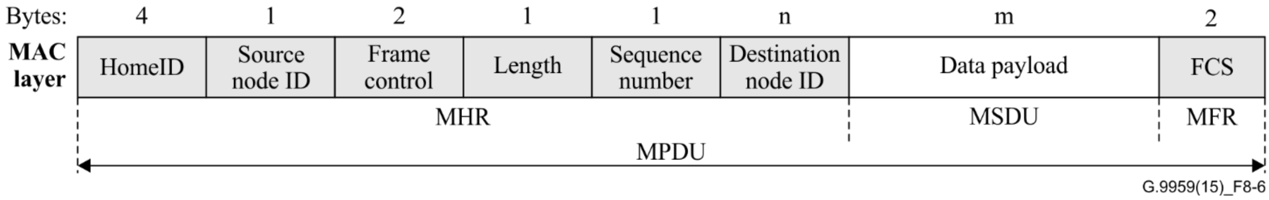

Figure 8-6 – General MPDU format (Channel configuration 3)

The multicast MPDU shall use the formats outlined in Figures 8-7 and 8-8.

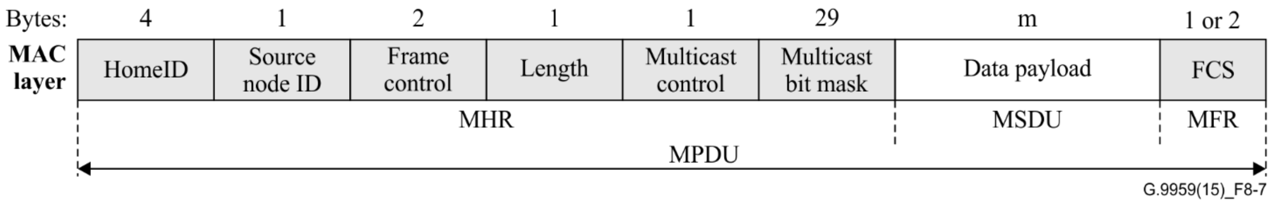

Figure 8-7 – Multicast MPDU format (Channel configurations 1, 2)

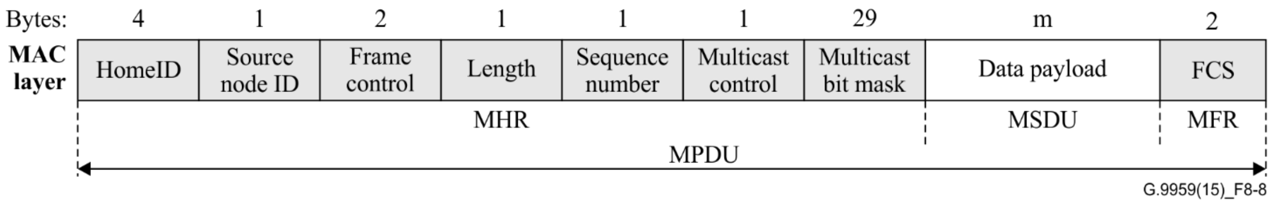

Figure 8-8 – Multicast MPDU format (Channel configuration 3)

Refer to clause 8.1.3.6.1 for details on transmission and processing of multicast MPDUs.

8.1.3.1 HomeID

The HomeID identifier field is 4 bytes in length and specifies the unique domain identifier. All nodes in a domain shall have the same HomeID.

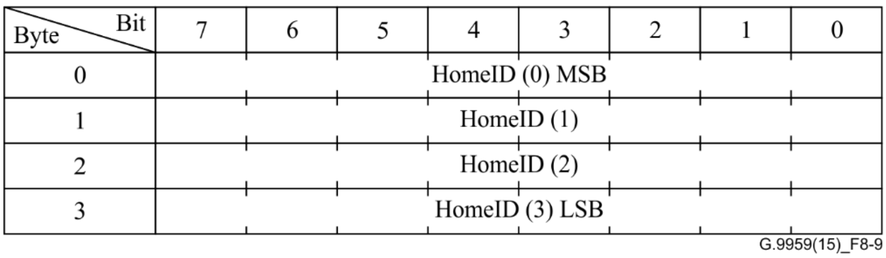

Figure 8-9 – Format of a HomeID field

The MAC layer shall support configuration of a promiscuous mode in which all MPDUs are forwarded to higher layers. Refer to clause 8.1.5.1.3.

8.1.3.2 Source NodeID

The source NodeID is 8 bits in length and shall be a unique identifier of a node in a given domain. Together with the HomeID, the source NodeID identifies the node that originated the MPDU.

Byte \ Bit	7	6	5	4	3	2	1	0
4	Source NodeID							

G.9959(15)_F8-10

Figure 8-10 – Source NodeID field

The source NodeID shall comply with Table 8-13.

Table 8-13 –Source NodeID values

NodeID value	Description
0x00	Uninitialized NodeID
0x01..0xE8	NodeID
0xE9..0xFF	Reserved

8.1.3.3 Frame control

The frame control field is 16 bits in length and contains information defining the frame type and other control flags. The frame control field shall be formatted as illustrated in Figures 8-11 and 8-12.

Byte \ Bit	7	6	5	4	3	2	1	0
5	Routed	ACK Req.	Low power	Speed modified	Header type			
6	Reserved	Beaming Info.		Reserved	Sequence number			

G.9959(15)_F8-11

Figure 8-11 –Frame control field (Channel configurations 1, 2)

Byte \ Bit	7	6	5	4	3	2	1	0
5	ACK Req.	Low power	Reserved		Header type			
6	Reserved	Beaming Info.			Reserved			

G.9959(15)_F8-12

Figure 8-12 – Frame Control field (Channel configuration 3)

8.1.3.3.1 Routed subfield (Channel configurations 1, 2 only)

The routed subfield is 1 bit in length. It shall be set to 0 when the MPDU is not routed and set to 1 when the MPDU is routed. The use of this field is out of scope of this Recommendation.

8.1.3.3.2 ACK Req subfield

The ACK Req subfield is 1 bit in length. It shall be set to 1 when the source node wants the destination node to acknowledge the MPDU and set to 0 when no ACK is needed.

A receiving node shall return an ACK MPDU in response to the ACK request.

8.1.3.3.3 Low power subfield

The Low power subfield is 1 bit in length. It shall be set by the source node. The bit informs a destination node that the actual transmission was using low power.

A receiving node shall return an ACK MPDU in low power in response to the low-power bit.

8.1.3.3.4 Speed modified subfield (Channel configurations 1, 2 only)

The speed modified subfield is 1 bit in length. It shall be set to 1 if an MPDU is sent at a lower speed than supported by the Src and Dst. The field shall not be used for routed and multicast MPDUs. The field shall be set to 0 if the MPDU is sent at the highest speed supported by the source and destination.

8.1.3.3.5 Header type subfield

The header type subfield defines the frame header type.

Table 8-14 – Header type

Header type value	Description
0x1	Singlecast MPDU
0x2	Multicast MPDU
0x3	Acknowledgement MPDU
0x4..0x7	Not to be used (Note)
0x8	Routed MPDU (Channel configuration 3 only)
0x9..0xB	Not to be used (Note)
0xC..0xF	*Reserved*
NOTE – These values shall not be transmitted due to backward compatibility. Received frames shall be forwarded to higher layers.	

A broadcast MPDU is a singlecast MPDU (header type 0x1) carrying DstNodeID = 0xFF; see clause 8.1.3.6.

A receiving node shall forward routed MPDUs to higher layers. The transmission of routed MPDUs is out of scope of this Recommendation.

8.1.3.3.6 Beaming information

The beaming information sub-field shall advertise the capability of a sending FL node to be awakened by a beam if it is asleep. The beaming information sub-field shall be interpreted in combination with the actual channel configuration.

Table 8-14a – Beaming information, channel configurations 1, 2

Channel configuration	Beam information value	Description
1,2	Frame Control[6..5] = "00"	No beam
1,2	Frame Control[6..5] = "01"	Short continuous beam
1,2	Frame Control[6..5] = "10"	Long continuous beam
1,2	Frame Control[6..5] = "11"	Reserved

Table 8-14b – Beaming information, channel configuration 3

Channel configuration	Beam information value	Description
3	Frame Control[6..4] = "000"	Reserved
3	Frame Control[6..4] = "001"	Reserved
3	Frame Control[6..4] = "010"	Reserved
3	Frame Control[6..4] = "011"	Reserved
3	Frame Control[6..4] = "100"	Fragmented beam
3	Frame Control[6..4] = "101"	Reserved
3	Frame Control[6..4] = "110"	Reserved
3	Frame Control[6..4] = "111"	Reserved

8.1.3.3.7 Sequence Number (Channel configurations 1, 2 only)

The Sequence Number is a 4-bit sub-field of the Frame Control field. The Sequence Number is provided by higher layers when transmitting. The MAC layer of a transmitting node shall forward the Sequence Number value transparently to the PHY. The MAC layer of a receiving node shall forward the Sequence Number value transparently to higher layers.

The MAC layer shall use the same sequence number for the initial transmission and for all retransmissions of a given MPDU. The transmitted sequence number shall be in the range 0x1..0xf. The value 0xf shall be followed by the value 0x1.

A receiving node shall accept any sequence number value in the range 0x0..0xf. The receiving node shall return the same value in an ACK MPDU if ACK is requested. A transmitting node shall validate the received sequence number in an ACK MPDU. In order to support legacy implementations, a transmitting node which receives an ACK MPDU shall accept the sequence number value zero irrespective of the sequence number transmitted.

8.1.3.4 Length

The length field is 1 byte in length and shall indicate the length of the MPDU in bytes.

Byte \ Bit	7	6	5	4	3	2	1	0
7	Length							

G.9959(15)_F8-13

Figure 8-13 – Length field

The length is limited to *aMacMaxMSDUSizeX*. The actual values can be found in Table 8-18. A receiving node shall not accept more bytes than the maximum length allowed for the actual data rate.

8.1.3.5 Sequence number (Channel configuration 3 only)

The sequence number is an 8-bit field of the MPDU header (MHR). The MAC layer of a transmitting node shall forward the sequence number value transparently to the PHY. The MAC layer of a receiving node shall forward sequence number value transparently to higher layers.

The MAC layer shall use the same sequence number for the initial transmission and for all retransmissions of a given MPDU. The transmitted sequence number shall be in the range 0x00..0xff. The value 0xff shall be followed by the value 0x00.

A receiving node shall accept any sequence number value in the range 0x00..0xff. The receiving node shall return the same value in an ACK MPDU if ACK is requested. A transmitting node shall validate the received sequence number in an acknowledgement MPDU.

8.1.3.6 Destination NodeID

The destination NodeID shall specify a destination node in the same domain identified by the HomeID.

The destination NodeID value shall comply with Table 8-15.

Table 8-15 – Destination NodeID

NodeID	Node type
0x00	Uninitialized NodeID
0x01..0xE8	NodeID
0xE9..0xFE	Reserved
0xFF	Broadcast NodeID

The DstNodeID value 0xFF may be used for broadcasting a MPDU to all nodes within direct range. The MAC layer of a receiving node shall always forward a broadcast MPDU to higher layers.

The MAC layer shall support the configuration of a promiscuous receiver mode, forwarding all MPDUs to higher layers irrespective of the actual Dst NodeID.

8.1.3.6.1 Multicast destination fields

A multicast MPDU shall carry a multicast control field and a multi-byte multicast bit mask as shown in Figure 8-14.

Byte \ Bit	7	6	5	4	3	2	1	0
8	Address Offset			Number of mask bytes				
9	Mask byte O							
10	Mask byte 1							
11	Mask byte 2							
...	....							
37	Mask byte 28							

G.9959(15)_F8-14

Figure 8-14 – Multicast destination fields

A receiving node shall not acknowledge a multicast MPDU. A sending node shall set the ACK Req bit of a multicast MPDU to zero. Likewise, a sending node shall set the address offset field to zero and the number of mask bytes field shall be set to 29.

A receiving node shall support the values of the Address Offset field outlined in Table 8-16. A receiving node shall support values of the number of Mask Bytes field in the range 1..29.

The address offset field is a 3 bit field. The encoding of the field is outlined in Table 8-16.

Table 8-16 – Multicast address offset encoding

Address offset value	Bit address offset
0	0
1	32
2	64
3	96
4	128
5	160
6	192
7	224

Each bit of a mask byte represents a NodeID. Mask byte 0 represents NodeIDs 1 to 8.

As an example a mask byte 0 value of 0xC5 (in binary: 11000101) covers NodeIDs 1, 3, 7 and 8 while a mask byte 1 value of 0xC5 (in binary: 11000101) covers NodeIDs 9, 11, 15 and 16.

The destination NodeID can be calculated using the following formula.

$$\text{Dst NodeID} = \text{bit address offset} + \text{mask byte number} \times 8 + \text{mask bit number} + 1$$

8.1.3.7 Data Payload

The Data Payload field has a variable length. An ACK MPDU may carry zero or more data payload bytes. A receiving node may derive the length of the Data Payload field from the MPDU Length field.

8.1.3.8 FCS

An 8-bit non-correcting FCS shall be used for validating the MPDU integrity for date rates R1 and R2. The FCS shall be calculated from the HomeID field to the Data Payload field, both included, as shown in Figure 8-15.

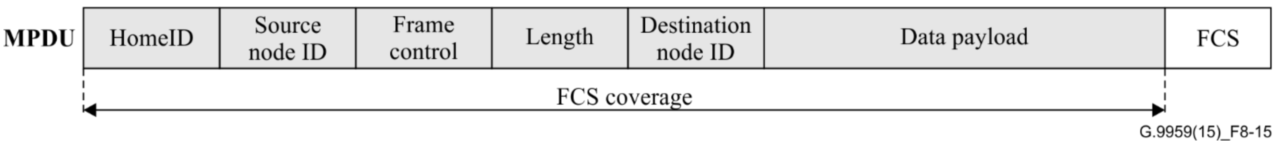

Figure 8-15 – FCS coverage

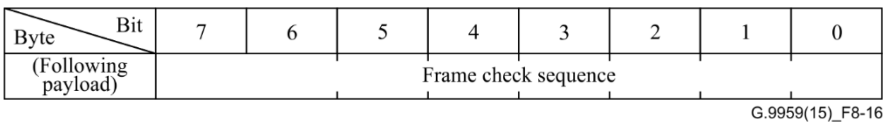

Figure 8-16 – FCS field

The FCS shall be calculated as an odd (XOR'ed) checksum as shown in the following algorithm.

```
BYTE GenerateCheckSum(BYTE *Data, BYTE Length)
{
  BYTE CheckSum = 0xFF;
  for (; Length > 0; Length--)
  {
    CheckSum ^= *Data++;
  }
  return CheckSum;
}
```

8.1.3.9 Cyclic redundancy code (CRC)

A 16-bit non-correcting CRC shall be used for validating the MPDU integrity for data rate R3.

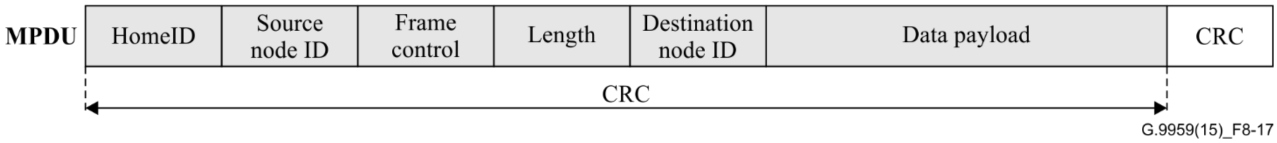

Figure 8-17 – CRC coverage

The CRC-16 generator polynomial shall be:

$$P(x) = x^{16} + x^{12} + x^5 + 1\text{, also known as CRC-CCITT}$$

The CRC 16 shall be calculated over the whole MPDU, except for the preamble, SOF, and the CRC-16 fields.

The CRC-16 generator shall be initialized to 1D0Fh before applying the first byte of a frame. Additional bits shall not be appended to the frame data.

8.1.3.10 Beam frame format

Beam frames may be used to awaken battery powered nodes operating in FL mode. Beam frames are used for several beam types. Beam frames are transmitted back to back to ensure an FL node can detect a beam within a very short time window before returning to sleep.

Each beam frame carries a preamble sequence and an SOF field, just like the start of any other PHY PDU. The SOF is followed by two or three bytes, replacing the HomeID field found in a general MPDU. A receiving node may distinguish a general MPDU from a beam frame by inspecting the MS byte of the HomeID. If this byte carries a beam tag (refer to Table 8-17) this is a beam frame and the following one or two bytes carry beaming relevant information.

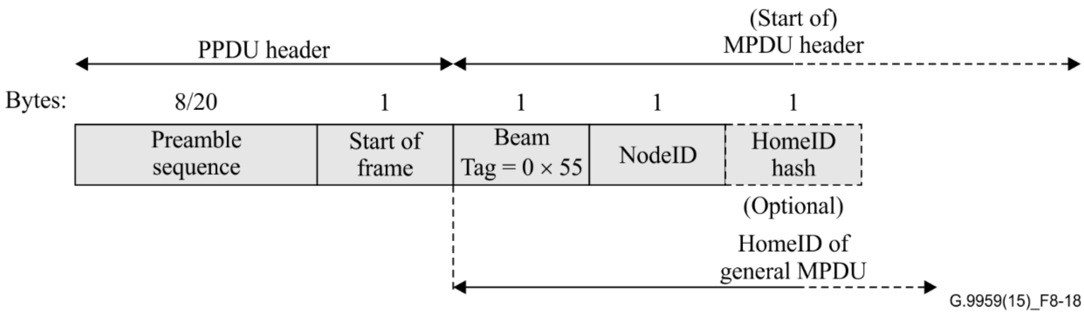

Figure 8-18 – Beam frame (Tag = 0x55)

The preamble sequence length depends on the actual RF profile used. Refer to clauses 7.1.2.1 and 7.1.3.2. A node shall listen long enough to detect a beam frame during worst case conditions around the SOF, beam tag, and beam information. In a worst-case situation, a node in FL mode starts listening just when it is too late to achieve lock to the preamble sequence. In this case, the node has to wait until another preamble sequence starts. Only after achieving lock to the preamble, the SOF and beam tag fields may be correctly decoded.

Each beam frame shall carry the Beam Tag and NodeID fields. The NodeID field should be followed by the optional HomeID Hash field.

Table 8-17 – Beam frame fields

Field name	Value	Description
Beam Tag	0x55	HomeID MSB reserved for beam frame transmission. HomeIDs in the range 0x55000000..0x55FFFFFF shall not be assigned to any domain The Beam Tag value 0x55 shall advertise the presence of a NodeID field and an optional HomeID Hash field
	0x54 (reserved)	HomeID MSB reserved for beam frame transmission HomeIDs in the range 0x54000000..0x54FFFFFF shall not be assigned to any domain The beam tag value 0x54 shall be reserved for future use
NodeID	1..232, 255	Destination NodeID. If the value is 255 or it matches the NodeID of an FL node, the node shall also inspect the HomeID Hash field. If no match is found, the FL node should return to sleep.
HomeID Hash (optional)	Hashed 8-bit version of the HomeID. Ignore if 0x0A, 0x4A or 0x55.	Hash value for preventing NodeID collisions in dense network environments with overlapping domains Refer to the text following this table

It is recommended that support is implemented for the HomeID Hash field presented in Figure 8-18 and Table 8-17.

The HomeID hash value shall be calculated as shown in the following algorithm.

```
BYTE GenerateHomeIdHash(BYTE *HomeId)
{
  BYTE HomeIdHash = 0xFF;
  for (Length = 4; Length > 0; Length--)
  {
    HomeIdHash ^= *HomeId++;
  }
  switch (HomeIdHash)
  {
    Case 0x0A:
    Case 0x4A:
    Case 0x55:
      HomeIdHash++;
  }
  return HomeIdHash;
}
```

A sending node should include a HomeID hash field in the beam frame (Tag 0x55) to assist FL nodes in filtering out beam frames belonging to other domains.

A FL node receiving one of the HomeID hash values 0x0A, 0x4A or 0x55 shall accept the value as a potential match to the actual HomeID and therefore it shall stay awake to receive the MPDU that follows if there is a match for the NodeID. Likewise, a FL node shall stay awake to receive the MPDU that follows if it detects a match to the hashed version of its own HomeID and if there is also a match for the actual NodeID. In case of no match for HomeID hash and/or NodeID, the node may return to sleep immediately.

8.1.3.11 Fragmented beam format

A beam fragment shall comprise a number of beam frames. The beam fragment duration shall be in the range 110-115 ms. Beam frames shall be sent back to back to ensure that the beam fragment can be detected by a node waking up at any moment during the duration of the beam fragment.

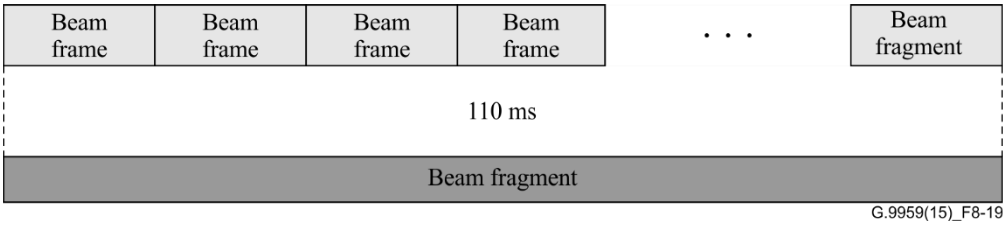

Figure 8-19 – Format of the beam fragment

A fragmented beam shall comprise a number of beam fragments. The next beam fragment shall begin in the range 190-200 ms measured from the beginning of the previous beam fragment. Fragmented beams shall be sent at data rate R3. A receiver shall be able to monitor multiple channels for beam frames.

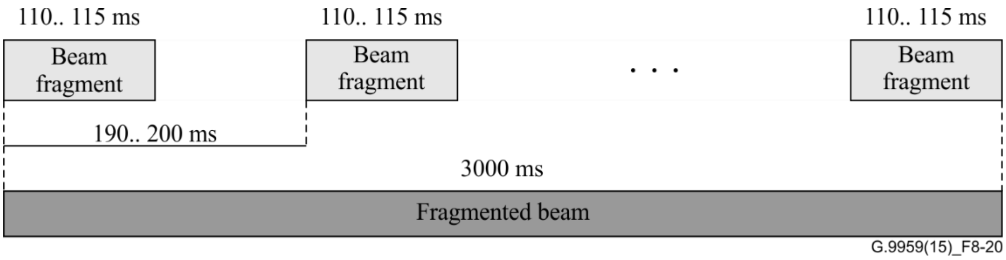

Figure 8-20 – Format of the fragmented beam

Proper TX scheduling allows a sending node to reach sleeping nodes that use a range of wakeup intervals 100 ms, 150 ms, 250 ms, 500 ms, 700 ms, 900 ms. The chosen wakeup interval should be a trade-off between battery lifetime and response time.

A fragmented beam may address any NodeID and also the broadcast NodeID 0xFF. A full fragmented beam shall span 3 000 ms. A singlecast frame shall follow the fragmented beam. A receiving node may interrupt the transmission of a fragmented beam by acknowledging a singlecast beam fragment. A receiving node shall not acknowledge a broadcast beam fragment. A receiving FL node detecting a positive match for the HomeID Hash field of the beam frame may return the ACK MPDU immediately. A receiving FL node not detecting a positive match for the HomeID hash field shall apply a random delay of 0..50 ms to the ACK MPDU to prevent collisions between FL nodes having the same NodeID in different domains.

In response to an ACK MPDU, the originating node shall transmit the MPDU to the receiving node if the source HomeID of the ACK MPDU matches the HomeID of the beaming node. If the HomeID or NodeID does not match, the originating node shall ignore the ACK MPDU.

8.1.3.12 Continuous beam format

A continuous beam is a series of beam frames spanning a fixed period of time. The beam frames shall be sent back to back to prevent other TXs from interrupting the continuous beam. The continuous beam shall be sent at data rate R2.

A continuous beam may address any NodeID and also the broadcast NodeID (0xFF).

There are two types of continuous beams: long and short. A long continuous beam may last for a maximum of 1 160 ms, and a short one may last for a maximum of 300 ms. The recommended duration is 1 100 ms for a long continuous beam and 275 ms for a short continuous beam. A continuous beam shall always be followed by a singlecast frame.

8.1.3.13 Beam formats on channel configurations 1 and 2

FL nodes operating in channel configuration 1 or 2 shall be able to receive and transmit the continuous beam format at data rate R2.

8.1.3.14 Beam formats on channel configuration 3

FL nodes operating in channel configuration 3 shall be able to receive and transmit the fragmented beam format at data rate 3.

8.1.4 MAC constants and MIB attributes

This clause specifies the constants and attributes used by the MAC.

8.1.4.1 MAC constants

The constants used by the MAC layer are presented in Table 8-18. These constants are hardware dependent and cannot be changed during operation.

Table 8-18 – MAC constants in general

Constants	Description	Value
aMacMaxMSDUSizeR1	The maximum singlecast/broadcast MSDU size for data rate R1	54 bytes
aMacMaxMSDUSizeMultiR1	The maximum multicast MSDU size for data rate R1	25 bytes
aMacMaxMSDUSizeR2	The maximum singlecast/broadcast MSDU size for data rate R2	54 bytes
aMacMaxMSDUSizeMultiR2	The maximum multicast MSDU size for data rate R2	25 bytes
aMacMaxMSDUSizeR3	The maximum singlecast/broadcast MSDU size for data rate R3	158 bytes
aMacMaxMSDUSizeMultiR3	The maximum multicast MSDU size for data rate R3	129 bytes
aMacMinAckWaitDuration	The minimum ACK wait duration	aPhyTurnaroundTimeRxTx + (aMacTransferAckTimeTX * (1/data rate))
aMacMaxFrameRetries	The number of retries after a transmission failure	2
aMacMinCCARetryDuration	The minimum duration of clear channel access assessment	1100 ms
aMacTxPhyPowerLevelLow	The value used for phyTransmitPower to transmit at low power	Implementation dependent
aMacTxPhyPowerLevelNormal	The value used for phyTransmitPower to transmit at normal power	Implementation dependent

The term *aMacMaxMSDUSizeX* is used in text that applies to all data rates (*aMacMaxMSDUSizeR1, aMacMaxMSDUSizeR2, aMacMaxMSDUSizeR3, aMacMaxMSDUSizeMultiR1, aMacMaxMSDUSizeMultiR2, aMacMaxMSDUSizeMultiR3)*.

Table 8-19 outlines the constant values to use for message transfer at data rates R1, R2, R3.

Table 8-19 – MAC constants for MPDU transfer

Constants	Description	R1	R2	R3	
				Channel Cfg 1, 2	**Channel Cfg 3**
aMacTransferAckTimeTX (ch)	The number of symbols of an ACK MPDU; including preamble	168 bits	248 bits	416 bits	296 bits
aMacTypicalFrameLengthTX(ch)	The number of symbols of a singlecast MPDU with a data payload of 4 bytes	200 bits	280 bits	448 bits	328 bits
aMacMinRetransmitDelay	Random backoff shall be higher than this value	10 ms			
aMacMaxRetransmitDelay	Random backoff shall be lower than this value	40 ms			

The MAC determines if the PHY is configured for channel configurations 1, 2 or channel configuration 3 operation by evaluating the RF profile mappings configured via the *macMapPhyChannel** MAC MIB parameters and the RF profile table defined for the PHY.

8.1.4.2 MAC MIB attributes

The MAC MIB comprises the attributes required to manage the MAC layer. Each of these attributes can be read or written using the MLME-GET.request and MLME-SET.request primitives respectively. The attributes contained in the MAC MIB are presented in Table 8-20.

Table 8-20 – MAC MIB attributes

Attribute	Type	Range	Description	Default
macCCARetryDuration	Integer	aMacMinCCARetry Duration – infinite	The duration of clear channel access assessment	~ 1100 ms
macHomeID	Byte array	0x00000000.. 0xFFFFFFFF	The HomeID is the unique domain identifier	(random)
macNodeID	Byte	0x00..0xFF	The NodeID is the address of the individual nodes in a domain	0x00
macPromiscuousMode	Boolean	TRUE or FALSE	This indicates whether the MAC layer is in a promiscuous (receive all) mode. A value of TRUE indicates that the	FALSE

Table 8-20 – MAC MIB attributes

Attribute	Type	Range	Description	Default
			MAC layer accepts all frames received from the PHY	
macRxOnWhenIdle	Boolean	TRUE or FALSE	This indicates whether the MAC layer is to enable its receiver at any time	FALSE
macMapPhyChannelA	Integer	Available RF profiles (Refer to clause 7.1.2.1)	Apply RF profile to channel A. Parameter maps directly to the PHY MIB parameter phyMapChannelA.	–
macMapPhyChannelB	Integer	Available RF profiles (Refer to clause 7.1.2.1)	Apply RF profile to channel B. Parameter maps directly to the PHY MIB parameter phyMapChannelB.	–
macMapPhyChannelC	Integer	Available RF profiles (Refer to clause 7.1.2.1)	Apply RF profile to channel C. Parameter maps directly to the PHY MIB parameter phyMapChannelC.	–

8.1.5 MAC functional description

This clause provides a detailed description of the MAC functionality. Throughout this clause, the receipt of an MPDU is defined as the successful receipt of an incoming MPDU from the RF media to the PHY and the successful verification of the FCS by the MAC layer, as described in clause 8.1.3.8.

8.1.5.1 Transmission, reception and acknowledgement

This clause describes the fundamental procedures for transmission, reception and acknowledgement.

8.1.5.1.1 Clear channel access

The MAC layer shall request the channel status from the PHY layer.

A PLME-GET-CCA.request message is used to evaluate the channel. A PLME-GET-CCA.confirm message returns the current channel status.

If the MAC layer finds the channel busy for a period of *macCCARetryDuration* the transmission has failed. This shall be indicated to the network layer via the MD-DATA.confirm primitive with a status of NO_CCA (see clause 8.1.2.1.2).

8.1.5.1.2 Transmission

To avoid RF collisions, the MAC layer shall perform a CCA before transmitting. If the channel is found idle, the MPDU may be transmitted. The sourceHomeID and source NodeID field shall identify the sending node and the destination NodeID field shall identify the destination node.

8.1.5.1.3 Reception and rejection

Each node may choose whether the MAC layer is to enable its receiver during idle periods. During these idle periods, the MAC layer shall still service TRX task requests from the network layer. A TRX task shall be defined as a transmission request, a reception request, or a clear channel access detection. On completion of each TRX task, the MAC layer shall request that the PHY enables or disables its receiver, depending on whether *macRxOnWhenIdle* is set to TRUE or FALSE, respectively.

Due to the broadcast nature of radiocommunications, a node is able to receive and decode transmissions from all nodes that are operating on the same channel(s). The MAC layer shall be able to filter incoming frames and present only the frames that are of interest to the upper layers.

In promiscuous mode, the MAC layer shall pass all MPDUs directly to the network layer. If the MAC layer is not in promiscuous mode (i.e., *macPromiscuousMode* is set to FALSE), it shall only accept MPDUs and issue an MD-DATA.indication to the network layer if the MPDU header contains the HomeID and NodeID of the receiving node. MPDUs shall also be accepted if addressed to the broadcast address or if the NodeID is included in a multicast header.

If the frame type subfield indicates a singlecast MPDU and the ACK request subfield of the frame control field is set to 1, the MAC layer shall send an ACK MPDU.

The MAC layer shall be able to receive beam fragments and forward these to higher layers.

8.1.5.1.3.1 RX filtering

An MPDU shall be discarded if the received MPDU has an invalid FCS value.

An MPDU shall be discarded if it has a length field less than 9 or greater than the maximum size values indicated in Table 8-18.

8.1.5.1.4 Use of ACK

A singlecast MPDU may be sent with the ACK request subfield of the frame control field set to 1. Any multicast or broadcast MPDU shall be sent with the ACK request subfield set to 0.

Sequence number checking shall be done as part of the ACK handling. Refer to clause 8.1.3.3.7.

8.1.5.1.4.1 No ACK

An MPDU transmitted with its ACK request subfield set to 0 shall not be acknowledged by its intended recipient. The originating node shall assume that the transmission was successful. The sequence diagram in Figure 8-21 shows the scenario for transmitting a single MPDU without requiring an ACK.

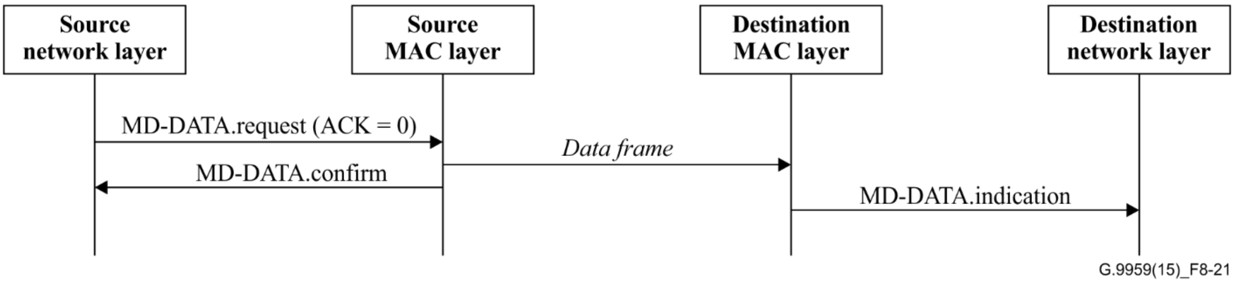

Figure 8-21 – Successful transmission without acknowledgement

8.1.5.1.4.2 ACK

A singlecast MPDU transmitted with the ACK request subfield of its frame control field set to 1 shall be acknowledged by the recipient. If the intended recipient correctly receives the MPDU, it shall return an ACK MPDU. Only the singlecast MPDU may be sent with the ACK request subfield set to 1. For other MPDU types the ACK request subfield shall be ignored by the intended recipient.

The transmission of an ACK MPDU shall not commence before *aPhyTurnaroundTimeRXTX* symbols have elapsed after the reception of the last symbol of the frame. Refer to clause 7.1.2.5.9.

The sequence diagram in Figure 8-22 shows the transmission of a singlecast MPDU with an ACK MPDU.

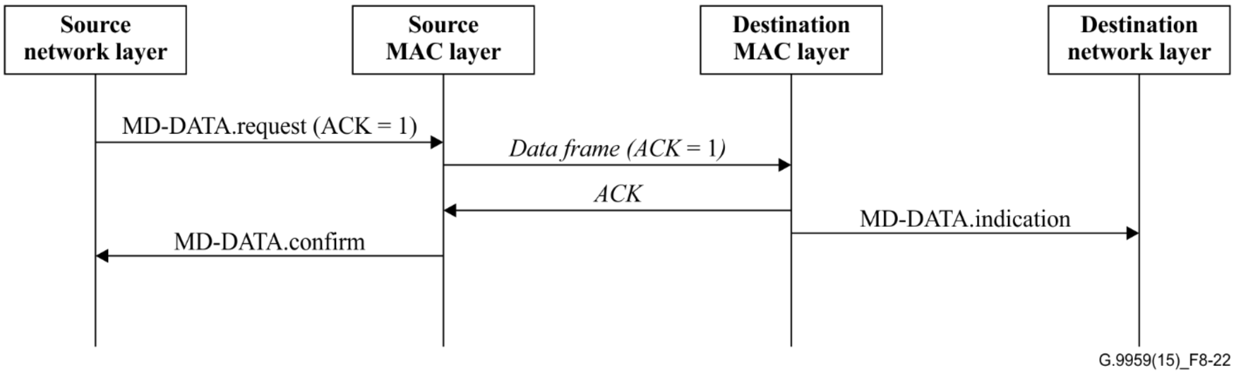

Figure 8-22 – Successful transmission with ACK

8.1.5.1.4.3 Retransmission

A node that sends a singlecast MPDU with its ACK request subfield set to 1 shall wait for a minimum of *aMacMinAckWaitDuration* symbols for the corresponding ACK MPDU to be received. If an ACK MPDU is received within *aMacMinAckWaitDuration* symbols and contains the correct HomeID and source NodeID, the transmission is considered successful, and no further action shall be taken by the originator. If an ACK MPDU is not received within *aMacMinAckWaitDuration* symbols, the transmission attempt has failed. The originator shall repeat the process of transmitting the MPDU and waiting for the ACK MPDU up to *aMacMaxFrameRetries* times. Before retransmitting, the node shall wait for a random backoff period (see clause 8.1.5.1.4.4).

If an ACK MPDU is still not received after *aMacMaxFrameRetries* retransmissions, the MAC layer shall assume the transmission has failed and notify the network layer of the failure. This shall be done via the MD-DATA.confirm primitive with a status of NO_ACK.

8.1.5.1.4.4 Random backoff

If a singlecast MPDU with its ACK request subfield set to 1 or the corresponding ACK MPDU is lost or corrupted, the singlecast MPDU shall be retransmitted. The MAC layer collision avoidance mechanism prevents nodes from retransmitting at the same time. The random delay shall be calculated as a period in the interval *aMacMinRetransmitDelay* .. *aMacMaxRetransmitDelay*; Refer to Table 8-19.

8.1.5.1.5 Idle mode

If the MLME is requested to set *macRxOnWhenIdle* to TRUE, the PHY shall enter RX mode and stay in RX mode when a frame has been transmitted (AL). This is achieved when the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of RX_ON.

If the MLME is requested to set *macRxOnWhenIdle* to FALSE, the PHY shall disable its receiver when a frame has been transmitted. This is achieved by the MLME issuing the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF.

8.1.5.2 Transmission scenarios

Due to the imperfect nature of the radio medium, a transmitted frame does not always reach its intended Dst. Figures 8-23 to 8-25 illustrate three different data transmission scenarios:

– ***Successful transmission***. The originating MAC layer transmits the MPDU to the recipient via the PHY data service. The originating MAC layer waits for *aMacMinAckWaitDuration* symbols. The destination MAC layer receives the MPDU, returns an ACK MPDU, and passes the MPDU to the next higher layer. The originating MAC layer receives the ACK MPDU. The data transfer is now complete, and the originating MAC layer issues a success confirmation to the network layer.

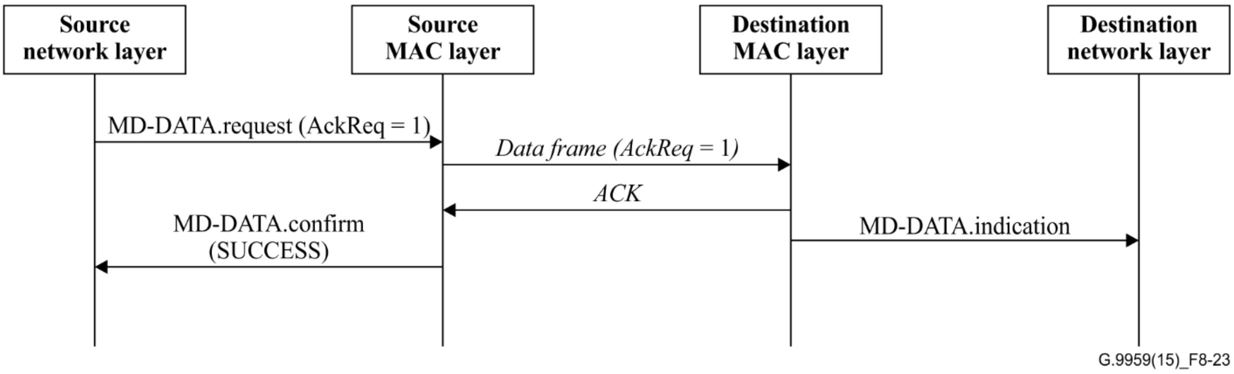

Figure 8-23 – Successful transmission scenario

– ***Lost MPDU***. The originating MAC layer transmits the MPDU to the recipient via the PHY data service. The originating MAC layer waits for *aMacMinAckWaitDuration* symbols. The Dst MAC layer does not receive the MPDU and therefore does not return an ACK MPDU. The timer of the originating MAC layer expires. The transmission has failed and the originator retransmits the MPDU. This sequence is repeated up to *aMacMaxFrameRetries* times. If transmissions fail a total of (1 + *aMacMaxFrameRetries*) times, the originating MAC layer issues a failure confirmation to the network layer.

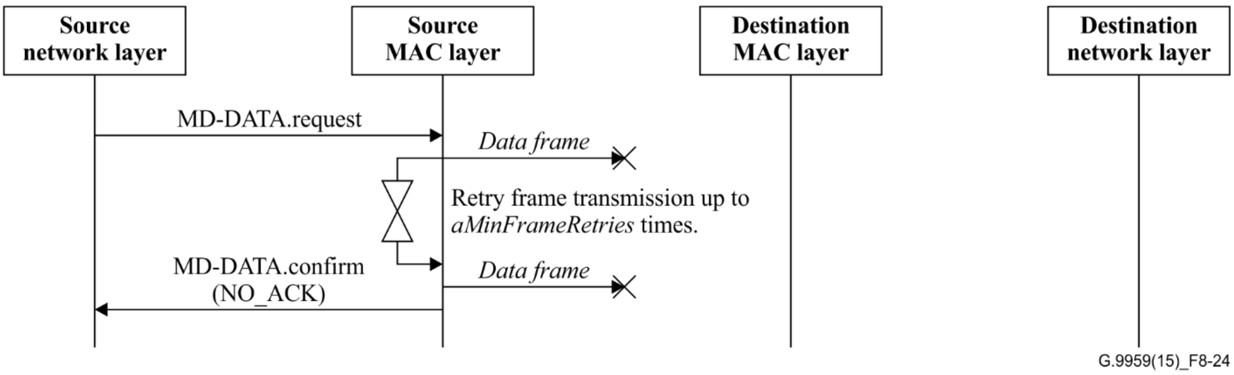

Figure 8-24 – Lost MPDU scenario

– ***LostAck MPDU***. The originating MAC layer transmits the MPDU to the recipient via the PHY data service. The originating MAC layer waits for *aMacMinAckWaitDuration* symbols. The destination MAC layer receives the MPDU, returns an ACK MPDU back to the originator, and passes the MPDU to the network layer. The originating MAC layer does not receive the ACK MPDU and its timer expires. The transmission has failed, and the originator retransmits the MPDU. If transmissions fail a total of (1 + *aMacMaxFrameRetries*) times, the MAC layer issues a failure confirmation to the network layer.

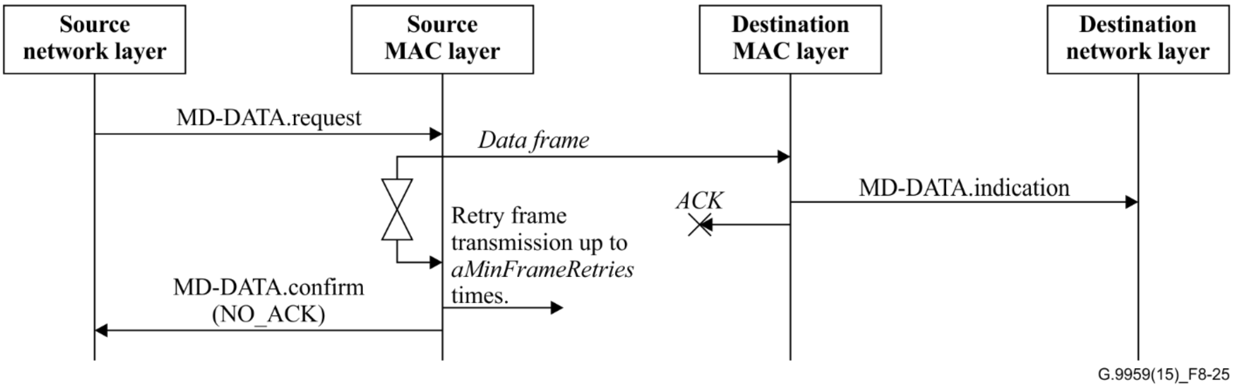

Figure 8-25 – Lost ACK MPDU scenario

9 LLC layer specification

In the OSI 7-layer stack, the LLC layer is the upper part of the data link layer (DLL).

9.1 General

The purpose of the LLC layer is to enable de-multiplexing of incoming MPDUs. The LLC layer shall not change the contents of the data link PDU (DLPDU) payload.

The following DLI interfaces are defined, as shown in Figure 9-1:

- DLI.6LoWPAN: Interface for IPv6 packets carried in 6LoWPAN frames
- DLI.Default: Interface for non-IP PDUs

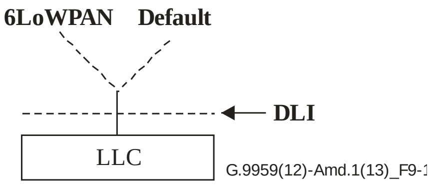

Figure 9-1 – Instances of the DLI interface

The DLI.Default interface shall receive all traffic which is not explicitly targeting other instances of the DLI interface.

9.2 Cascaded adaptation layer architecture

One or more adaptation layers may be injected into the DLL between the MAC and LLC layers, as shown in Figure 9-2. Each adaptation layer serves a specific purpose, such as handling mesh routing, segmentation or security encryption. The function of an adaptation layer may be triggered by the presence of a routing header or specific DLPDU header values. Each adaptation layer shall implement the MLI interface on each of its interfaces. Each adaptation layer may modify or entirely discard an incoming MPDU. An adaptation layer shall transparently forward incoming MPDUs to the next layer in case the actual adaptation layer has no relevant function to perform.

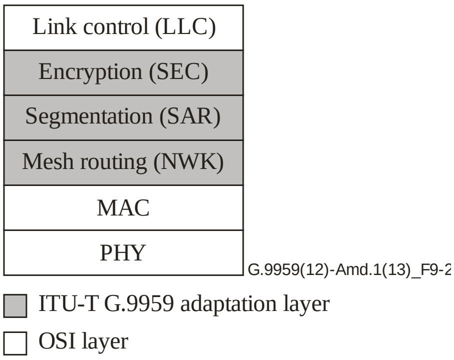

Figure 9-2 – Ordering of adaptation layers

9.3 DLPDU format

The LLC receives an MPDU from the MLI interface. The MAC layer specifies MPDU formats for singlecast and multicast transmission modes (see Figure 9-3).

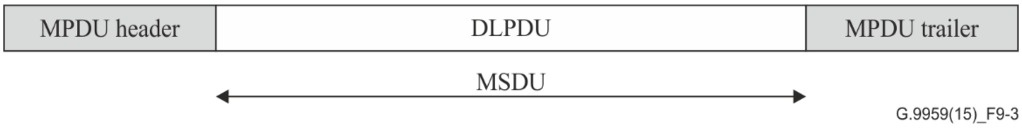

Figure 9-3 – General MPDU format

It is further specified that MPDUs may carry mesh routing information when the "Routed" flag of the frame control field is set or if the "Routed Frame" header type is used. In this case, a routing header is prepended to the DLPDU carried in the MSDU, as shown in Figure 9-4.

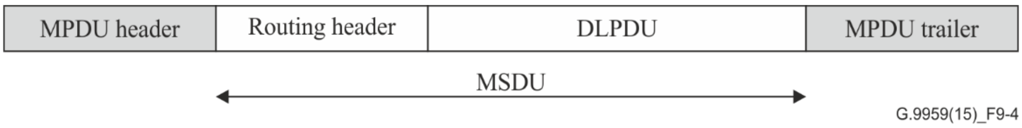

Figure 9-4 – MPDU format for routed frame

For correctly configured nodes, the LLC layer should never receive an MPDU which carries routing information. The LLC layer shall ignore an MPDU which carries routing information. The following text assumes that no routing header is prepended to the MPDU payload when it is received by the LLC layer.

The DLPDU shall carry a DLPDU header, and may carry a payload field, as shown in Figure 9-5. The payload field may have any length up to *aLlcMaxDlpduSize* bytes. The payload field may carry another encapsulated DLPDU.

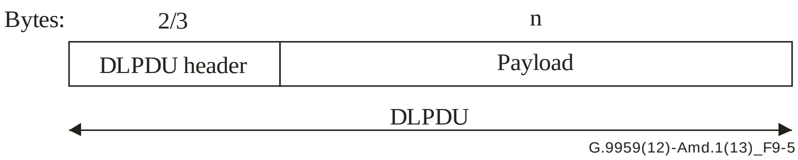

Figure 9-5 – DLPDU format

9.3.1 DLPDU header

The DLPDU header comprises two fields: the Command Class and the Command, as shown in Figure 9-6.

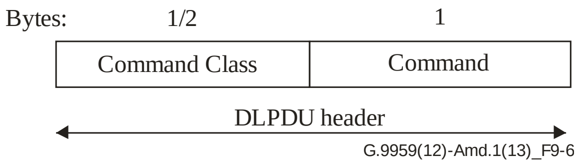

Figure 9-6 – DLPDU header

9.3.1.1 Command Class

The Command Class field may be 1 or 2 bytes in length.

In the general case, the Command Class field shall be 1 byte long. If the first byte of the command class is a value in the range *aLlcFirstExtendedCommandClass1* .. *aLlcLastExtendedCommandClass1*, the DLPDU header contains an extended command class identifier. An extended Command Class field shall be 2 bytes in length.

Table 9-1 – Command class types

Cmd Cls 1	Cmd Cls 2	Command class type
aLlcFirstNormalCommandClass1 .. *aLlcLastNormalCommandClass1*	(not present)	Normal
aLlcFirstExtendedCommandClass1 .. *aLlcLastExtendedCommandClass1*	*aLlcFirstExtendedCommandClass2* .. *aLlcLastExtendedCommandClass2*	Extended

9.3.1.2 Command

The Command field shall be 1 byte in length and specifies one particular command for a given command class.

9.4 Data link interface selection

The LLC layer shall extract the DLPDU contained in an incoming MPDU and forward the DLPDU to a target DLI. The target DLI shall be determined by an evaluation of the DLPDU header Command Class field.

9.4.1 DLI.IPv6 interface selection

The LLC layer shall forward the DLPDU to the DLI.6LoWPAN interface if the DLPDU header Command Class field carries the 6LoWPAN command class identifier *aLlc6lowpanCommandClass*.

9.4.2 DLI.Default interface selection

Any DLPDU which does not match the explicit selection criterion for another DLI interface instance shall be forwarded to the DLI.Default interface.

9.5 LLC layer constants

This clause specifies the constants relating to the LLC layer.

9.5.1 LLC constants

The LLC shall comply with the constants defined in Table 9-2.

Table 9-2 – LLC constants

Constants	Description	Value
aLlcMaxDlpduSize	The maximum DLPDU size to support	1350 bytes
aLlcFirstNormalCommandClass1	Start of Normal command class range, byte 1	0x00
aLlcLastNormalCommandClass1	End of Normal command class range, byte 1	0xF0
aLlcFirstExtendedCommandClass1	Start of Extended command class range, byte 1	0xF1
aLlcLastExtendedCommandClass1	End of Extended command class range, byte 1	0xFF
aLlcFirstExtendedCommandClass2	Start of Extended command class range, byte 2	0x00
aLlcLastExtendedCommandClass2	End of Extended command class range, byte 2	0xFF
aLlc6lowpanCommandClass	6LoWPAN identifier indicating IPv6 payload	0x4F

10 Adaptation layers

As specified in clause 9 ("LLC layer specification"), one or more adaptation layers may be injected in between the MAC and LLC layers.

10.1 SAR adaptation layer

The SAR adaptation layer allows ITU-T G.9959 nodes to exchange long payloads. Examples of such payloads include security certificates for ITU-T G.9959 network bootstrapping or IPv6 packets.

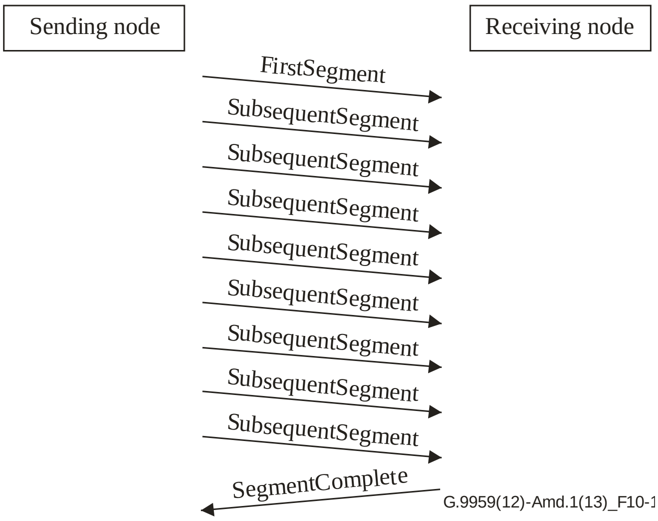

Figure 10-1 – Example of datagram transferred as segments

10.1.1 Terminology overview

The SAR adaptation layer provides segmentation and reassembly of a datagram by splitting up the datagram into segments, as shown in Figure 10-1, each of which fits into a MAC data payload (i.e., the MSDU). The datagram is formatted as a DLPDU with two header fields (Command Class, Command) like any other DLPDU.

The SAR adaptation layer defines the transport service command class. As with all other command classes, the transport service command class is unconcerned whether segments are transported in direct range or by a mesh routing protocol.

10.1.2 SAR adaptation layer specification

The SAR adaptation layer shall provide the following features in order to obtain reliable data transfer of datagrams larger than the MSDU can accommodate.

Typical examples are security encapsulated datagrams and 6LoWPAN encapsulated IPv6 packets.

10.1.2.1 Transparent functionality

The SAR adaptation layer shall support segmentation and reassembly of any datagram up to *aLlcMaxDlpduSize* bytes.

A node implementing the SAR adaptation layer shall automatically determine if a datagram fits into the available MPDU payload length. A node should not use segmentation if a datagram can be sent in one MPDU.

10.1.2.2 Relaxed transmission timing for long payloads

In order not to congest the network, large data transfers shall leave transmit opportunities for other nodes in the network. If sending a command longer than *aSarRelaxedTimingMacFrames* MSDUs, a node shall implement a delay between every transmitted MSDU. The minimum required time delay and number of MSDUs before a delay shall be inserted depends on the actual data rate as follows:

- Data rate R2: At least *aSarRelaxedTimingDelayR2* ms if sending more than *aSarRelaxed-TimingMacFrames* MSDUs back-to-back;
- Data rate R3: At least *aSarRelaxedTimingDelayR3* ms if sending more than *aSarRelaxedTimingMacFrames* MSDUs back-to-back.

10.1.2.3 Varying PHY PDU size

The available MPDU payload size depends on the actual bit rate. The SAR adaptation layer shall support all MSDU sizes offered by the actual MAC/PHY layer. The payload length available to SAR segments may be shorter than the MSDU sizes specified in Table 8-18 if additional headers, e.g., a routing header, are prepended by lower adaptation layers; refer to clause 9.2.

10.1.2.4 Segment re-transmission

The segment reassembly process may end up missing a number of segments in the receive buffer, e.g., due to a checksum error. The receiving node shall be able to request the retransmission of individual segments to complement the segments already received. The algorithm shall be able to handle the case where retransmitted segments are also lost. If segments are received via broadcast or multicast transmission, the receiving node shall not request the retransmission of segments.

10.1.2.5 Robust error handling

In case no segments are received in response to a retransmission request, a receiving node shall discard all segments previously received.

If a sending node repeatedly fails to deliver a segment to a destination node, the node shall abort transmission of all remaining segments.

10.1.3 The transport service command class

The SAR adaptation layer shall use the transport service command class to transfer long datagrams between two nodes. The datagram shall start with two fields carrying a command class and command, refer to clause 9.

10.1.3.1 First segment command

The first segment command initiates the transfer of a datagram, as shown in Figure 10-2.

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_TRANSPORT_SERVICE</td></tr>
<tr><td colspan="5">Command = COMMAND_FIRST_SEGMENT</td><td colspan="3">datagram_size_1 [10..8]</td></tr>
<tr><td colspan="8">datagram_size_2 [7..0]</td></tr>
<tr><td colspan="4">Session ID</td><td>Ext</td><td colspan="3">Reserved</td></tr>
<tr><td colspan="8">Header Extension Length (OPTIONAL)</td></tr>
<tr><td colspan="8">Header Extension (OPTIONAL)</td></tr>
<tr><td colspan="8">Payload 1</td></tr>
<tr><td colspan="8">…</td></tr>
<tr><td colspan="8">Payload N</td></tr>
<tr><td colspan="8">FCS 1 [15..8]</td></tr>
<tr><td colspan="8">FCS 2 [7..0]</td></tr>
</table>

Figure 10-2 – Frame format of first segment command

10.1.3.1.1 Command (5 bits)

The command field is reduced to a 5-bit field in order to limit the overhead introduced by the segment header. The LS 3 bits shall be masked out before decoding the command code when detecting the transport service command class.

The command code shall be the 5-bit value *aSarFirstSegmentCommand*.

10.1.3.1.2 Datagram size (11 bits)

Indicates the overall size of the datagram carried in the segments. The unit is 1 byte. Bit 0 is the least significant bit (LSB). The value shall not exceed *aLlcMaxDlpduSize* bytes.

10.1.3.1.3 Reserved (3 bits)

This field shall be set to 0 by a sending node and shall be ignored by a receiving node.

10.1.3.1.4 Ext (1 bit)

This field shall be used to indicate the presence of the header extension length and header extension fields. The field shall be set to 1 if there is a header extension. A receiving node shall ignore unsupported header extensions and process the command, skipping unsupported extensions.

10.1.3.1.5 Session ID (4 bits)

This field identifies the session. The session ID shall be incremented when a new datagram transmission is initiated. A receiver shall verify the session ID of all received segments. Segments shall be discarded if an unknown session ID is detected.

10.1.3.1.6 Header extension length (8 bits)

This field is only present if the Ext bit is set to 1.

A sending node may add header extension fields to the command header. In this case, the Ext bit shall be set to 1 and the following fields shall be present in the command header:

- Header extension length
- Header extension

10.1.3.1.7 Header extension (variable size)

This field is only present if the Ext bit is set to 1. Refer to clause 10.1.3.1.4.

10.1.3.1.8 Payload (variable size)

The payload field of a first segment shall carry the first segment of a datagram.

A receiving node shall derive the size of the payload field from the size of entire command, subtracting all header fields as well as the checksum field.

10.1.3.1.9 FCS (16 bits)

This field carries a CRC-16 FCS. The FCS shall be calculated using the CRC-16-CCITT mechanism with the following parameters:

- Polynomium code shall be 0x1021 (*aSarFcsPolynomium*), representing the polynomium $x^{16} + x^{12} + x^5 + x^0$;
- CRC register shall be initialized to *aSarFcsInitValue* by the TX as well as receiver;
- The calculated CRC-16 value shall cover all fields of the first segment command except for the FCS fields;
- The calculated CRC-16 value shall not include padding after the frame, neither zeros nor any other value;
- The CRC-16 value shall be calculated one byte at a time in the transmission order, starting with the command class field;
- The most significant bit (MSB) of a byte shall be transmitted and received first, and likewise be the first to be processed in a serialized implementation;
- Bit 0 of the CRC-16 value corresponds to the input register of a serialized implementation of the CRC-16 calculation. Bit 0 is the LSB of the 16-bit word.

A conceptual serial example is provided in Figure 10-3 below on how the polynomium $x^{16} + x^{12} + x^5 + x^0$ may be implemented.

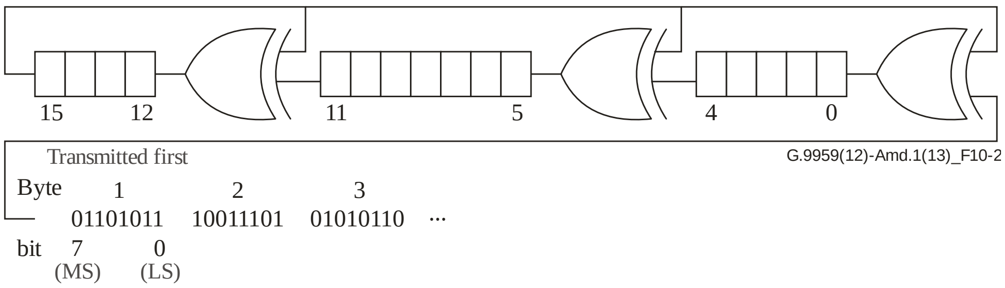

Figure 10-3 – CRC-16 implementation example

The following test vector shown in Figure 10-4 may be used for ensuring compliance with this Recommendation.

Test frame	CRC
C2 A2 15 0D 03 03 02 0B 01	2C 66

Figure 10-4 – CRC-16 implementation test vector

10.1.3.2 Subsequent segment command

One or more subsequent segment commands follow the first segment command and carry the remaining part of the datagram. (See frame format shown in Figure 10-5.)

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Command Class = COMMAND_CLASS_TRANSPORT_SERVICE</td></tr>
<tr><td colspan="5">Command = COMMAND_SUBSEQUENT_SEGMENT</td><td colspan="3">datagram_size_1 [10..8]</td></tr>
<tr><td colspan="8">datagram_size_2 [7..0]</td></tr>
<tr><td colspan="4">Session ID</td><td>Ext</td><td colspan="3">datagram_offset_1 [10..8]</td></tr>
<tr><td colspan="8">datagram_offset_2 [7..0]</td></tr>
<tr><td colspan="8">Header extension length (OPTIONAL)</td></tr>
<tr><td colspan="8">Header extension (OPTIONAL)</td></tr>
<tr><td colspan="8">Payload 1</td></tr>
<tr><td colspan="8">...</td></tr>
<tr><td colspan="8">Payload N</td></tr>
<tr><td colspan="8">FCS 1 [15..8]</td></tr>
<tr><td colspan="8">FCS 2 [7..0]</td></tr>
</table>

Figure 10-5 – Frame format of subsequent segment command

Receiving the last segment of a singlecast datagram, the receiving node shall return either a segment request command or a segment complete command. The receiving node shall not respond to a broadcasted datagram.

10.1.3.2.1 Command (5 bits)

The command code shall be the 5-bit value *aSarSubsequentSegmentCommand*.

10.1.3.2.2 Datagram size (11 bits)

Refer to clause 10.1.3.1.2.

10.1.3.2.3 Ext (1 bit)

Refer to clause 10.1.3.1.4.

10.1.3.2.4 Session ID (4 bits)

Refer to clause 10.1.3.1.5.

10.1.3.2.5 Datagram offset (11 bits)

This field indicates the offset to apply to this segment when reassembling the datagram in the receive buffer. The unit is 1 byte. Bit 0 is the LSB.

10.1.3.2.6 Header extension length (8 bits)

This field is only present if the Ext bit is set to 1.

Refer to clause 10.1.3.1.6.

10.1.3.2.7 Header extension (variable size)

This field is only present if the Ext bit is set to 1.

Refer to clause 10.1.3.1.7.

10.1.3.2.8 Payload (variable size)

The payload field of a subsequent segment may carry any portion of a datagram.

A receiving node shall derive the size of the payload field from the size of entire command, subtracting all header fields as well as the checksum field.

10.1.3.2.9 Frame check sequence (16 bits)

Refer to clause 10.1.3.1.9.

10.1.3.3 Segment request command

The segment request command may be used by a receiving node to request a missing segment. (See frame format shown in Figure 10-6.) Multiple missing segments should be requested one at a time.

This message shall not be used in response to segments received via broadcast or multicast.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_SEGMENT_REQUEST					*reserved*		
Session ID				*reserved*	datagram_offset_1 [10..8]		
datagram_offset_2 [7..0]							

Figure 10-6 – Frame format of segment request command

10.1.3.3.1 Command (5 bits)

The command code shall be the 5-bit value *aSarSegmentRequestCommand*.

10.1.3.3.2 Reserved

This field shall be set to 0 by a sending node and shall be ignored by a receiving node.

10.1.3.3.3 Session ID (4 bits)

Refer to clause 10.1.3.1.5.

10.1.3.3.4 Datagram offset

Refer to clause 10.1.3.2.5.

10.1.3.4 Segment complete command

The Segment complete command shall be returned from the receiving node when all segments have been correctly received. (See frame format shown in Figure 10-7.)

This message shall not be used in response to segments received via broadcast or multicast.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_SEGMENT_COMPLETE					*reserved*		
Session ID				*reserved*			

Figure 10-7 – Frame format of segment complete command

10.1.3.4.1 Command (5 bits)

The command code shall be the 5-bit value *aSarSegmentCompleteCommand*.

10.1.3.4.2 Reserved

This field shall be set to 0 by a sending node and shall be ignored by a receiving node.

10.1.3.4.3 Session ID (4 bits)

Refer to clause 10.1.3.1.5.

10.1.3.5 Segment wait command

The segment wait command may be returned in response to a first segment command or a subsequent segment command by a node which is already receiving segments from another node. (See frame format shown in Figure 10-8.)

This message shall not be used in response to segments received via broadcast or multicast.

7	6	5	4	3	2	1	0
Command class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_SEGMENT_WAIT					*Reserved*		
Pending segments							

Figure 10-8 – Frame format of segment wait command

10.1.3.5.1 Command (5 bits)

The command code shall be the 5-bit value *aSarSegmentWaitCommand*.

10.1.3.5.2 Reserved

This field shall be set to 0 by a sending node and shall be ignored by a receiving node.

10.1.3.5.3 Pending segments (8 bits)

The value indicates the number of segments not yet received by the receiving node. Since the delivery time of any segment depends on path length (direct vs. 5 hop routing), the value does not represent an absolute time. Furthermore, the node currently sending may initiate another transfer immediately after the completion of the actual segmented datagram.

A sending node shall stop sending segments for a minimum period of (100 ms × [number of pending segments]) if receiving a segment wait command. When resuming transmission, the sending node shall restart segment transmission with the first segment command.

10.1.3.6 Managing segment transfer and handling exceptions

The following clauses present the mechanisms for handling segments and exceptions such as missing segments.

10.1.3.6.1 Avoiding concurrent peer-to-peer transfers

The SAR adaptation layer shall not allow other multi-MSDU datagrams to be transmitted to any destination node before the ongoing transmission of one multi-MSDU datagram has been completed. Doing so could delay the datagram transmission, causing security timers to timeout. While the SAR adaptation layer is transmitting a datagram to a given destination, the SAR adaptation layer may transmit a single-MSDU command to another destination node.

A receiving node may return a segment wait command in response to a singlecast segment if the node is already receiving another segmented datagram.

10.1.3.6.2 Completing the transfer (receiving node)

A receiving node shall return a segment complete command if all segments have been received. A segment request command shall be returned if the receiving has timed out waiting for segments.

The receiving node shall wait the following time after the reception of a segment before requesting missing segments:

- Data rate R2: at least *aSarTransferTimeoutR2*
- Data rate R3: at least *aSarTransferTimeoutR3*

If the datagram is broadcasted, the node shall drop already received segments and shall not send a segment request command.

10.1.3.6.3 Completing the transfer (sending node)

When a sending node has completed transmission of a datagram, the sending node shall wait for a segment complete message from the receiving node. Depending on the transmission speed, the sending node shall wait the following time before sending another datagram if no segment complete message is received:

- Data rate R2: at least *aSarSegmentCompleteTimeoutR2*
- Data rate R3: at least *aSarSegmentCompleteTimeoutR3*

As retransmission is provided by the MAC layer, the sending node shall not retransmit the last segment in case of a timeout.

10.1.3.6.4 Tie-breaking

In a special case, two nodes may start sending segments at the same time. The following algorithm shall be used to avoid a deadlock situation where both nodes consistently send Wait messages to the other node.

A receiving node which is already transmitting segments should perform a tie-breaking check before dropping an incoming segment. If the tie-breaking check is positive, the receiving node should accept the incoming segment and abort the ongoing transmission. The tie-break check is positive if the following three conditions are all met:

1. The receiving node is currently transmitting a datagram.
2. The recipient of the datagram being transmitted is also the originator of the received segment
3. The receiving node has a lower NodeID than the originator.

10.1.4 SAR adaptation layer constants

This clause specifies the constants relating to the SAR adaptation layer.

10.1.4.1 SAR constants

The SAR shall comply with the constants defined in Table 10-1.

Table 10-1 – SAR constants

Constants	Description	Value
aSarRelaxedTimingMacFrames	Use relaxed transmission timing if sending SAR datagrams longer than *aSarRelaxedTimingFrameCount* MPDUs.	2 frames
aSarRelaxedTimingDelayR2	Delay between each frame at data rate R2 when using relaxed transmission timing	35 ms
aSarRelaxedTimingDelayR3	Delay between each frame at data rate R3 when using relaxed transmission timing	15 ms
aSarTransferTimeoutR2	Waiting time before requesting a missing segment at data rate R2	800 ms
aSarTransferTimeoutR3	Waiting time before requesting a missing segment at data rate R3	400 ms
aSarSegmentCompleteTimeoutR2	Waiting time before sending another datagram at data rate R2	1000 ms

Table 10-1 – SAR constants

Constants	Description	Value
aSarSegmentCompleteTimeoutR3	Waiting time before sending another datagram at data rate R3	500 ms
aSarTransportServiceCc	Transport service command class identifier	0x55
aSarFirstSegmentCommand	Transport service first segment identifier	0x18
aSarSubsequentSegmentCommand	Transport service subsequent segment identifier	0x1C
aSarSegmentRequestCommand	Transport service segment request identifier	0x19
aSarSegmentCompleteCommand	Transport service segment complete identifier	0x1D
aSarSegmentWaitCommand	Transport service segment wait identifier	0x1E
aSarFcsPolynomium	FCS polynomium defined as $x^{16} + x^{12} + x^5 + x^0$	0x1021
aSarFcsInitValue	FCS init value	0x1D0F

Annex A

Non-radio related aspects of the Z-Wave PHY and MAC Specifications

(This annex forms an integral part of this Recommendation.)

A.1 Introduction

A.1.1 General

Z-Wave is a low-power, low-cost wireless technology enabling consumer-grade products with networked features. Examples include remote controlled light dimmers, networked temperature sensors, electronic door locks and AV systems.

This annex presents the physical (PHY) layer to be implemented by a device operating in a Z-Wave network.

Low-power wireless radio networks suffer from frequent frame drops due to fading effects, spurious noise and reflections. Z-Wave provides low-level retransmissions to recover from such situations. Network layers may employ mesh routing to extend networks beyond what is possible in direct range. The Z-Wave routing protocol is one such example.

A.1.2 Technology overview

NOTE – The information in this clause is for informational purposes only and contains features that are out of scope of this Recommendation.

The Z-Wave framework incorporates its own network, transport, and application layers. These should not be confused with the OSI layers or the network and transport layers defined by the IETF for IP transport. The Z-Wave framework supports IP transport as well as IP routing protocols.

Nodes receive a unique NodeID during network inclusion. A network may hold up to 232 nodes. A HomeID is used to identify a specific domain. HomeID and NodeID are auto-configured with no user intervention.

The Z-Wave network layer defines the Z-Wave routing protocol which is a source routing protocol. The Z-Wave routing protocol allows an originator to reach a destination node via one or more repeaters. A reactive discovery algorithm allows originators to determine new source routes when required.

Z-Wave uses a special frame format for multicast messages. The Z-Wave multicast frame header carries a complete destination bit map in the frame header.

The Z-Wave application layer defines a wide range of device and command classes for various devices such as lamps, door locks and temperature sensors.

A.2 General description

The Z-Wave PHY layer defines modulation schemes, data rates, synchronization methods and a frame format for use in low-power, low-bandwidth control networks.

The Z-Wave MAC layer defines a half duplex protocol for acknowledged wireless communication in a low-cost control network. The MAC layer targets "soft" real-time applications which are not time critical in nature. The MAC layer supports on-demand communication to battery operated nodes.

The PHY layer is responsible for the following tasks:

– assignment of RF profiles to physical channels;
– activation and deactivation of the radio TRX;
– transmission and reception;

– clear channel assessment;
– frequency selection;
– link quality assessment for received frames.

A Z-Wave TRX can operate in a one, two or three channel configuration in license-free RF bands.

The MAC is responsible for handling the following:

– HomeID assignment;
– NodeID assignment;
– collision avoidance algorithm;
– backoff algorithm;
– automatic retransmission in case of transmission errors;
– support for battery operation.

MAC frames carry one small header in order to conserve bandwidth. While presented as one header, a few fields are used by higher layers. These fields are carried transparently and ignored by the MAC layer.

The Z-Wave routing protocol is out of scope of this Recommendation.

A.2.1 Network topology

A.2.1.1 Network components

Z-Wave nodes can be divided into two subgroups: normal nodes; and low-power nodes. The normal nodes are typically in receive mode at all times. The low-power nodes spend the majority of the operating life in power down and wake up at regular short intervals to minimize power consumption.

A.2.1.2 Network topology

NOTE – The network formation is handled by the network layer, which is out of the scope of this Recommendation. However, this clause provides a brief overview of how the supported network topology is formed.

In a mesh network topology, each device is capable of communicating with devices within direct range. A node can send and receive messages and can also relay messages for its neighbours. Introducing routing provides for a redundant and more reliable network. An example of a mesh network is shown in Figure A.1.

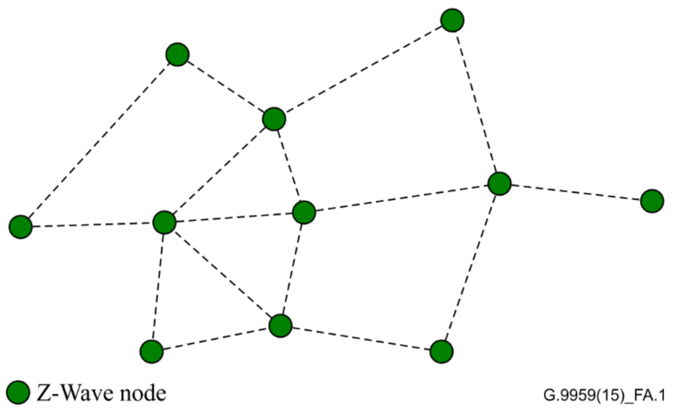

Figure A.1 – Mesh network topology

Routes may be established on a proactive basis or via on-demand discovery. If messages are lost due to fading effects and/or interference, the node may retransmit messages and if still failing, use alternative routes. New alternative routes may be discovered if needed.

A.2.1.3 Bootstrapping

A unique 32-bit identifier called the HomeID is used to identify individual domains.

NodeIDs are unique within a given domain. A NodeID is an 8-bit short address. The first node hands out the HomeID and unique NodeIDs to all other nodes added to the domain.

A.2.2 Network architecture

The Z-Wave architecture is defined in terms of layers. Each layer is responsible for one part of the operation and offers services to the higher layers: a data entity provides a data transmission service; and a management entity provides other services related to the actual layer. The data and management entities define the logical links between the layers.

A Z-Wave node implements the PHY layer, which contains the RF TRX along with its low-level control mechanism, a MAC layer that provides access to the physical channel for all types of transfers, a network layer controlling message routing, and a combined application layer that collapses the OSI stack layers transport, session, and presentation.

Figure A.2 shows these layers in a graphical representation.

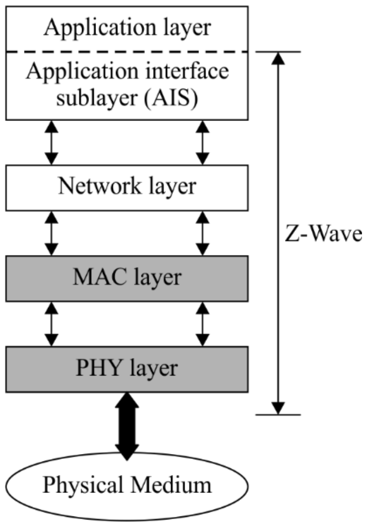

Figure A.2 – Z-Wave layers

This annex specifies the PHY and MAC layers. Upper layers are outside the scope of this Recommendation.

A.2.2.1 PHY layer

The features of the PHY are activation and deactivation of the RF TRX, frequency selection, and transmitting as well as receiving frames. The RF receiver is able to perform a clear channel assessment. The RF TRX operates in a one, two, or three-channel configuration located in the license-free ISM frequency bands.

The PHY provides two services: (1) the PD service accessed through the PD-SAP; and (2) the PHY management service interfacing with the physical layer management entity service access point (PLME-SAP). The PD service enables the transmission and reception of PPDUs across the physical radio channel.

Clause A.3 contains the full specifications of the PHY layer.

A.2.2.2 MAC layer

The features of the MAC layer are channel access, frame validation, acknowledged frame delivery, and retransmissions.

The MAC layer provides two services: the MAC data service, accessed through the MD-SAP, and the MAC management service interfacing with the MAC layer management entity service access point (MLME-SAP). The MAC data service enables the transmission and reception of MPDUs across the PD service.

Clause A.4 contains the full specifications of the MAC layer.

A.2.3 Functional overview

A brief overview of the general functions of a Z-Wave network is given in the following clauses and includes information on the data transfer model, the frame structure, robustness and power consumption considerations.

A.2.3.1 MPDU formats

A number of MAC PDU formats are defined.

Figure A.3 shows the structure of the general MPDU. The MAC service data unit (MSDU) carries the payload data from the network layer. The MSDU is prepended with a MAC header (MHR) and appended with a MAC footer (MFR). The MHR, MSDU, and MFR together constitute the MAC protocol data unit (PDU).

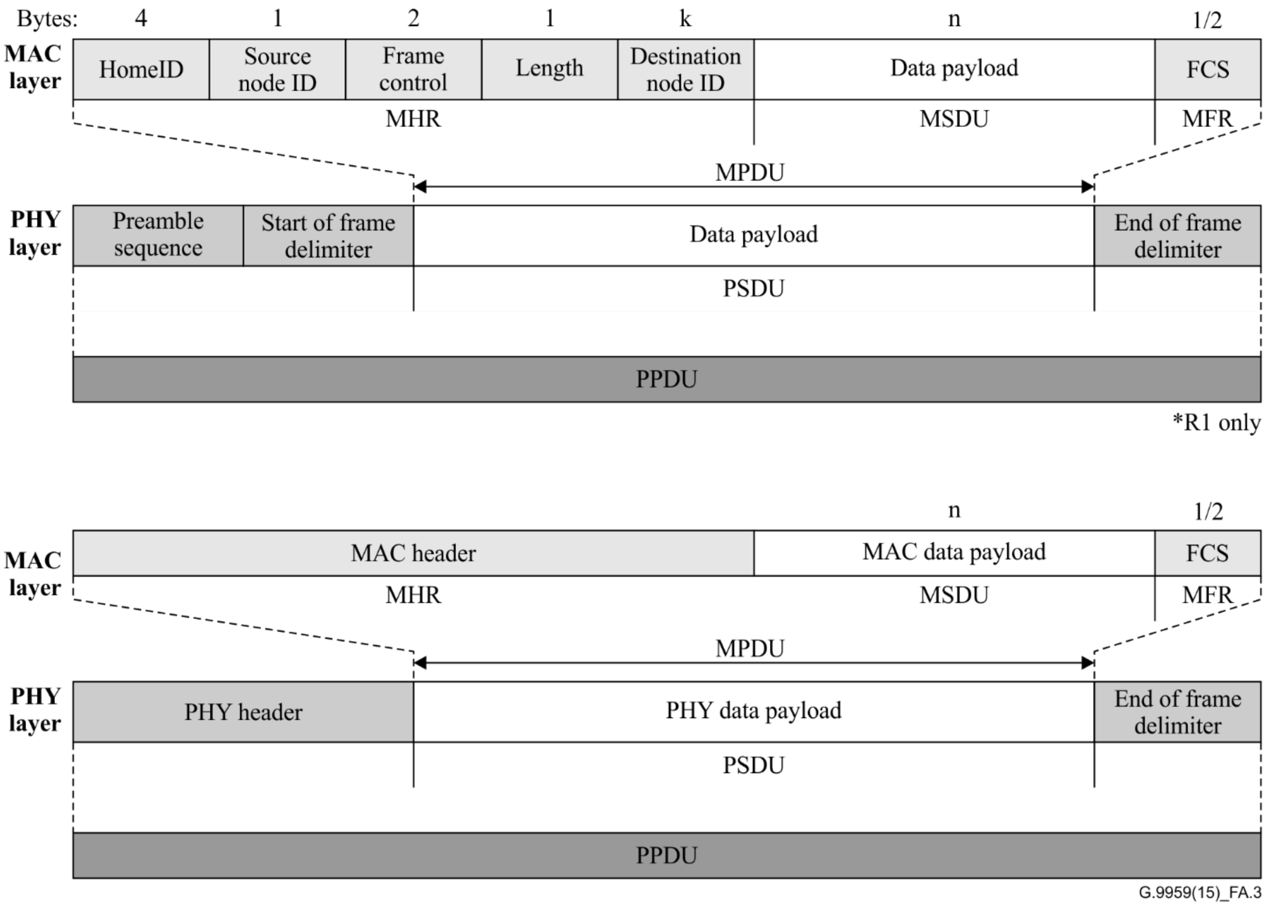

Figure A.3 – General MPDU structure

The MPDU is passed to the PHY layer as a PHY layer data unit (PSDU). The PSDU is prefixed with a start header (SHR) and an end header (EHR).

A.2.3.1.1 Singlecast MPDU

The singlecast MPDU uses the general frame structure (Figure A.3).

A.2.3.1.2 ACK MPDU

The ACK MPDU uses the general MPDU structure (Figure A.3). The MSDU may have a length of zero bytes.

A.2.3.1.3 Multicast MPDU

A dedicated Z-Wave multicast header explicitly identifies each target node via a dedicated multicast addressing bitmask.

A.2.3.2 Network robustness

The Z-Wave protocol employs various mechanisms to ensure robustness in the data transmission. These mechanisms of the MAC layer are backoff algorithm, frame ACK, data verification, and frame retransmission.

A.2.3.2.1 Clear channel assessment

A node shall query the availability of the channel from the PHY layer before transmitting. If the channel is found to be idle, the node may transmit its data. If the channel is found busy, the node shall wait for idle channel before transmitting.

A.2.3.2.2 Acknowledgment

A successful reception and validation of an MPDU may be confirmed with an ACK MPDU. If the Dst node receives an MPDU containing bit errors, the message shall not be acknowledged. An ACK request may be carried in the frame that needs ACK.

A.2.3.2.3 Retransmission

If two or more nodes transmit MPDUs simultaneously a collision may occur and the MPDUs may not reach their destinations.

If the source node does not receive an ACK, it assumes that the transmission was unsuccessful and retries the MPDU transmission up to *aMacMaxFrameRetries* times.

In order to avoid another collision each TX shall delay a retransmission by a random delay.

A.2.3.2.4 Multi-hop routing

The MAC layer defines MPDU header fields for support of multi-hop routing. Routing is outside the scope of this Recommendation.

A.2.3.2.5 Data validation

An 8-bit non-correcting FCS mechanism is employed to detect bit errors for data rates R1 and R2. A 16-bit non-correcting FCS mechanism is employed to detect bit errors for data rate R3. Refer to clause A.3.1.3 for details on data rates.

A.2.3.3 Power consumption considerations

One category of battery-powered nodes spend most of their operational life in a sleep mode. Such nodes may periodically wake up and poll other nodes to get pending messages. The PHY layer AL mode is used for listening during wake-up polling.

Other low-power nodes may require a more responsive behaviour than can be achieved via periodic wake-up. Such nodes may use the PHY FL mode for incoming messages.

A.2.3.3.1 Communication with a FLN

Battery-powered devices may need to be reachable at any time. Nodes that are listening at regular intervals are said to operate in FL mode. The PHY layer provides an extended preamble sequence that allows an FL node to operate at a very low duty cycle while still being reachable.

A.2.4 Concept of primitives

This clause provides a brief overview of the concept of service primitives (operations). Refer to [b-ITU-T X.210] for more detailed information.

The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure A.4, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

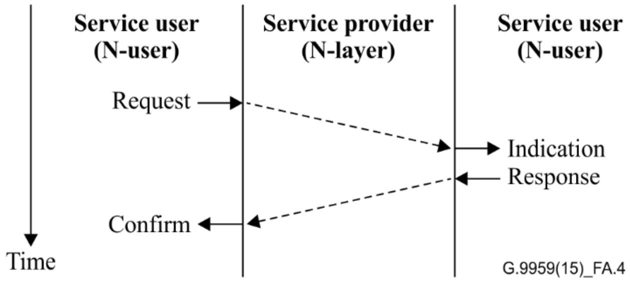

Figure A.4 – Service primitives

The services are specified by describing the information flow between the N-user and the N-layer. This information flow is modelled 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 SAP associated with an N-user. Service primitives convey the required information by providing a particular service. These service primitives are an abstraction because they specify only the provided service rather than the means by which it is provided. This definition is independent of any other interface implementation.

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 four generic types:

- Request: the request primitive is passed from the N-user to the N-layer to request that a service is initiated;
- Indication: the indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event;
- Response: the response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive;
- Confirm: the confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

A.3 Physical (PHY) layer specification

This clause specifies the PHY. The PHY is responsible for the following tasks:

- Assignment of an RF profile to a physical channel;
- Activation and deactivation of the radio TRX;
- Data transmission and reception;
- clear channel assessment;
- Frequency selection;
- Link quality for received frames.

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., *aPhyMaxFrameSizeRx*, and are listed in Table A.27. Attributes have a general prefix of "phy", e.g., *phyCurrentChannel*, and are listed in Table A.28.

A.3.1 TRX front end specifications

This clause specifies requirements of the PHY.

The Z-Wave specification shall conform to established regulations in Europe, United States, and other world regions.

A.3.1.1 RF profiles

An RF profile defines one or more data rates for use in a given radio channel. The definition of specific regional frequencies is outside the scope of this Recommendation.

The list of RF profiles is specified in Table A.1. A TRX shall support up to 3 radio channels, each characterized by an RF profile. Each channel shall have a unique RF profile assigned to it. Depending on the actual region, RF Profiles may allow communication at one or more data rates and one or more channels.

Table A.1 – RF profiles

RF profile	Region	Channel configurations	Centre frequency	R3	R2	R1	FL (Note)
0	n/a		n/a				
1	European Union	1,2	f_{EU1}	√			
2			f_{EU2}		√	√	
3			f_{EU2}		√		√
4	United States	1,2	f_{US1}	√			
5			f_{US2}		√	√	
6			f_{US2}		√		√
7	Hong Kong	1,2	f_{HK1}	√			
8			f_{HK1}		√	√	
9			f_{HK1}		√		√
10	Australia and New Zealand	1,2	f_{ANZ1}	√			
11			f_{ANZ2}		√	√	
12			f_{ANZ2}		√		√
13	Malaysia	1,2	f_{MY1}	√			
14			f_{MY1}		√	√	

Table A.1 – RF profiles

RF profile	Region	Channel configurations	Centre frequency	R3	R2	R1	FL (Note)
15			f_{MY1}		√		√
16	India	1,2	f_{IN1}	√			
17			f_{IN1}		√	√	
18			f_{IN1}		√		√
19	Japan	3	f_{JP1}	√			√
20			f_{JP2}	√			√
21			f_{JP3}	√			√
22	Israel	1,2	f_{IL1}	√			
23			f_{IL1}		√	√	
24			f_{IL1}		√		√
25	Korea	3	f_{KR1}	√			√
26			f_{KR2}	√			√
27			f_{KR3}	√			√
28	Russia	1,2	f_{RU1}	√			
29			f_{RU1}		√	√	
30			f_{RU1}		√		√
31	China	1,2	f_{CN1}	√			
32			f_{CN1}		√	√	
33			f_{CN1}		√		√
NOTE – FL mode is only applicable for the indicated RF profiles. AL mode may be used with any RF profile.							

A.3.1.2 Transmit frequency error

Frequency error is defined as the difference between the measured transmitted centre frequency and the actual regional centre frequency.

The frequency error shall not exceed ±27 ppm.

A.3.1.3 Data rates

The PHY shall support the data rates and the accuracy as listed in the table below.

Table A.2 – Data rate and data rate accuracy

Data rate	Bit rate	Symbol rate	Accuracy
R1	9.6 kbit/s	19.2 kbaud	±27 ppm
R2	40 kbit/s	40 kbaud	±27 ppm
R3	100 kbit/s	100 kbaud	±27 ppm

A.3.1.4 Channel configuration

A compliant node shall operate in one of the three possible configurations as listed in the table below.

Table A.3 – Channel configurations

Configuration	Num channels	Data rate		
		R3	R2	R1
1	1	–	Ch B	Ch B
2	2	Ch A	Ch B	Ch B
3	3	Ch A Ch B Ch C	–	–

Tables A.1 and A.3 shall be used in combination.

Example:

A node intended for the EU region may use configuration 2 which provides two alternative communication channels. The following RF profiles are available:

- RF profile 1 at the frequency f_{EU1} (Ch A) supporting data rate R3 in AL mode only;
- RF profile 2 at the frequency f_{EU2} (Ch B) supporting data rates R1 and R2 in AL mode only;
- RF profile 3 at the frequency f_{EU2} (Ch B) supporting data rate R2 in AL and FL mode.

This imaginary node runs on battery but has to support low-latency communication. Thus, FL mode is preferred. When listening or transmitting to other FL nodes, the node uses RF Profile 3 (Ch B). When sending to AL nodes, the node uses RF Profile 1 (Ch A) or RF Profile 2 (Ch B).

A.3.1.5 Modulation and encoding

The PHY shall employ FSK for RF modulation at data rates R1 and R2. The PHY shall employ GFSK for RF modulation at data rates R3.

Manchester code shall be used for data symbol encoding at data rate R1 and non-return-to-zero (NRZ) shall be used for data symbol encoding at data rates R2 and R3.

The modulation and the coding format are summarized in Table A.4.

Table A.4 – Modulation and coding format

Data Rate	Modulation	Coding	Frequency offset	Separation	Symbols
R1	FSK	Manchester	20 kHz	40 kHz ±20%	Binary
R2	FSK	NRZ	0 kHz	40 kHz ±20%	Binary
R3	GFSK, BT = 0.6	NRZ	0 kHz	58 kHz ±20%	Binary

The mapping of NRZ symbols to the physical medium is given in Table A.5.

Table A.5 – NRZ symbol mapping

Symbol	Frequency
0	$f_{\text{center frequency}}$ + separation/2
1	$f_{\text{center frequency}}$ – separation/2

The mapping of Manchester symbols to the physical medium is given in Table A.6 below.

Table A.6 – Manchester symbol mapping

Symbol	Frequency
0	Transition from ($f_{\text{center frequency}} + f_{\text{offset}}$ – separation/2) to ($f_{\text{center frequency}} + f_{\text{offset}}$ + separation/2)
1	Transition from ($f_{\text{center frequency}} + f_{\text{offset}}$ + separation/2) to ($f_{\text{center frequency}} + f_{\text{offset}}$ – separation/2)

Table A.6 refers to a frequency offset, f_{offset}. The frequency offset is specified in Table A.4.

A.3.1.6 RF power measurement

Unless stated otherwise, all RF power measurements, either transmit or receive, shall be made at the antenna connector. The measurements shall be made with equipment that is either matched to the impedance of the antenna connector or corrected for any mismatch. For devices without an antenna connector, the measurements shall be interpreted as EIRP (i.e., a 0 dBi gain antenna); and any radiated measurements shall be corrected to compensate for the antenna gain in the implementation.

A.3.1.7 Transmit power adjustments (conducted)

The TX output power level shall conform to local regulations. The output power level allowed by local regulations is denoted as "nominal". Actual absolute power levels are out of scope of this Recommendation.

The TX shall be able to adjust its output power in steps of 2 dB in the range between the nominal transmit power down to (nominal output power – 10 dB).

The TX shall further be able to reduce its output power to at least (nominal output power – 20 dB).

A.3.1.8 Receiver sensitivity

The receiver shall be of receiving a standard test frame at a minimum power level.

The minimum receiver sensitivity for each data rate shall be as specified in Table A.8.

The standard test frame and test conditions shall be as specified in Table A.7.

Table A.7 – Standard test frame

Term	Definition	Conditions
Standard Test Frame	PHY frame used for testing sensitivity	PHY frame with at least four bytes of random payload data
FER	Average frame loss	Average measured over standard test frames
Receiver sensitivity	Threshold input signal power that yields a specific FER	FER < 1%. Power measured at antenna terminals Interference not present

Table A.8 – Minimum receiver sensitivity

Bit rate	Minimum receiver sensitivity
R1	–95 dBm
R2	–92 dBm
R3	–89 dBm

A.3.1.9 Clear channel assessment

The PHY shall be able to perform a CCA with a threshold of –80 dBm. If the RF channel is found to be idle, the PHY may transmit its data.

In a given deployment, a LBT operation based on CCA shall comply with actual regional RF regulatory requirements, e.g., listening period and sensitivity level.

A.3.1.10 Receiver spurious requirement

A TRX shall limit its RF emissions when in RX mode. Emissions near the centre frequency may affect the ability of other nearby devices to receive weak signals.

A receiver shall not emit more than –70 dBm within ±1 MHz from the centre frequency as shown in Figure A.5. The measurement bandwidth shall be 100 kHz.

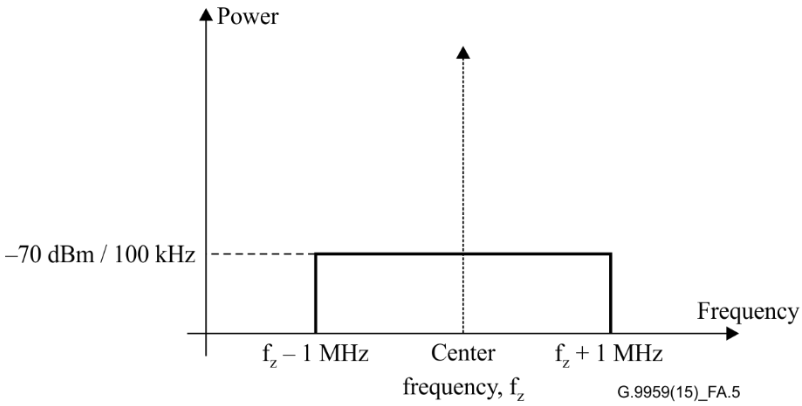

Figure A.5 – Receiver spurious limit

A.3.1.11 Receiver blocking

Blocking is a measure of the capability of the receiver to receive the intended modulated signal without experiencing degradation due to the presence of another unwanted input signal.

A conforming implementation shall be able to pass a blocking test as described below for all bit rates.

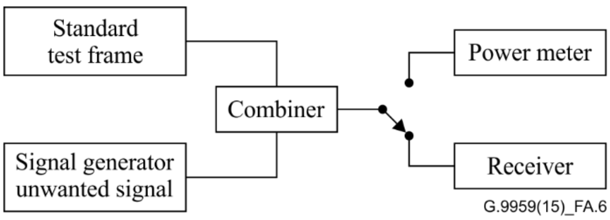

Figure A.6 – Receiver blocking test definitions

Method of measurement:

– Standard test frames shall be transmitted at the nominal frequency using the modulation specified for the actual data rate. Its power is adjusted down to a power level which is 3 dB higher than the sensitivity level defined in Table A.8;

– The blocking test signal shall be a carrier transmitted at a specific offset frequency as defined in Table A.9. The blocking test signal power shall be increased until the receiver experiences an FER that corresponds to the sensitivity level.

Limits:

Table A.9 – Blocking limits

Frequency offset	Limits
±1 MHz	–44 dBm
±2 MHz	–34 dBm
±5 MHz	–27 dBm
±10 MHz	–25 dBm

A.3.1.12 Receiver saturation

The receiver saturation power level is the maximum power level, in decibels relative to 1 mW, present at the input of the receiver. A receiver shall meet the FER criterion in Table A.7 while receiving at an input level greater than or equal to 0 dBm to sustain "zero" distance between two or more devices.

A.3.1.13 TX-to-RX turnaround time

The TX-to-RX turnaround time shall be measured from the trailing edge of the last transmitted symbol to the leading edge of the first symbol of the received preamble.

The TX-to-RX turnaround time shall be less than *aPhyTurnaroundTimeTXRX* (see Table A.27).

Latency estimations shall be calculated on the 99th percentile of all latency measurements.

A.3.1.14 RX-to-TX turnaround time

The RX-to-TX turnaround time shall be measured from the trailing edge of the last received symbol to the leading edge of the first transmitted preamble symbol.

The RX-to-TX turnaround time shall be more than *aPhyTurnaroundTimeRXTX* (see Table A.27).

Latency estimations shall be calculated on the 99th percentile of all latency measurements.

A.3.2 PPDU format

This clause specifies the format of the PPDU.

The general PPDU shall be formatted as outlined in Figure A.7. The frame format is depicted in the order in which it is transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from $k – 1$ (leftmost and most significant) down to 0 (rightmost and least significant), where the length of the field is k bits. Bytes within each multi-byte field are numbered from 1 (leftmost and most significant) up to $n – 1$ (rightmost and least significant), where the length of the field is n bytes. Bits within each byte are numbered from 7 (leftmost and most significant) down to 0 (rightmost and least significant).

A.3.2.1 General PHY frame format

The PPDU shall be formatted as illustrated in Figure A.7:

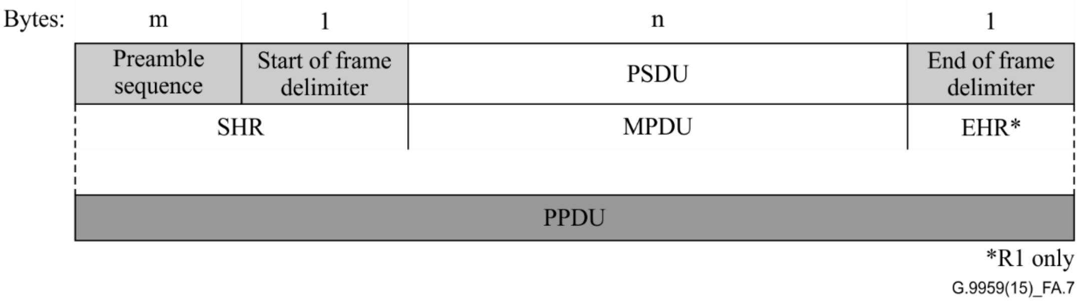

Figure A.7 – PPDU format

A.3.2.1.1 Preamble field

The preamble field allows a receiver to obtain symbol synchronization. The preamble field shall be composed of a sequence of bytes containing the binary pattern "01010101". Figure A.8 shows the logical bit waveform of the Manchester encoded preamble for data rate R1.

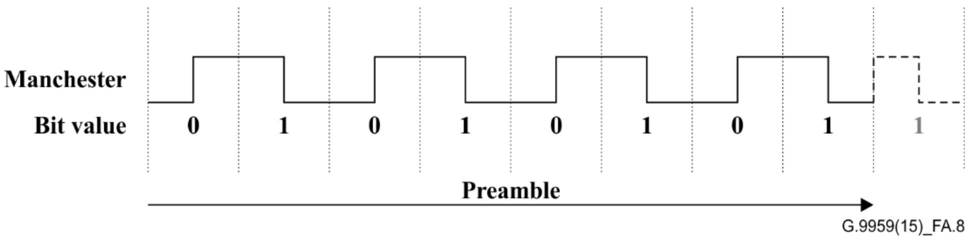

Figure A.8 – Manchester encoded preamble pattern (R1)

Figure A.9 shows the logical bit waveform of the NRZ encoded preamble pattern for data rates R2 and R3.

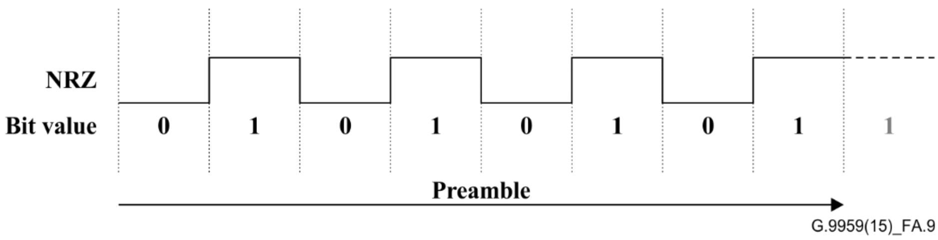

Figure A.9 – NRZ encoded preamble pattern (R2, R3)

The preamble length shall comply with Table A.10. The values allow a receiver to scan all channels and obtain synchronization at an arbitrary channel during the duration of the preamble.

Table A.10 – Minimum preamble length

Channel configuration	Rate	Minimum preamble length in bytes		
		Singlecast/ broadcast	Multicast	Beam
1	R1 R2 R3	10 10 n/a	10 20 n/a	n/a 20 n/a
2	R1 R2 R3	10 10 40	10 20 40	n/a 20 n/a
3	R1 R2 R3	n/a n/a 24	n/a n/a 24	n/a n/a 8

A.3.2.1.2 SOF field

The SOF delimiter is an 8-bit field terminating the preamble field and the start of the PSDU. The SOF shall be formatted as the logical bit pattern illustrated in Table A.11.

Table A.11 – Format of the SOF field

Bit number	7	6	5	4	3	2	1	0
Value	1	1	1	1	0	0	0	0

A.3.2.1.3 PSDU field

The PSDU field has a variable length and carries the data of the PHY frame.

A.3.2.1.4 EOF delimiter field

The EOF delimiter field shall be sent only when transmitting at data rate R1. The field shall carry a sequence of 8 Manchester code violations each denoted E. Each violation, E, shall be a symbol without transition. Refer to Figure A.10.

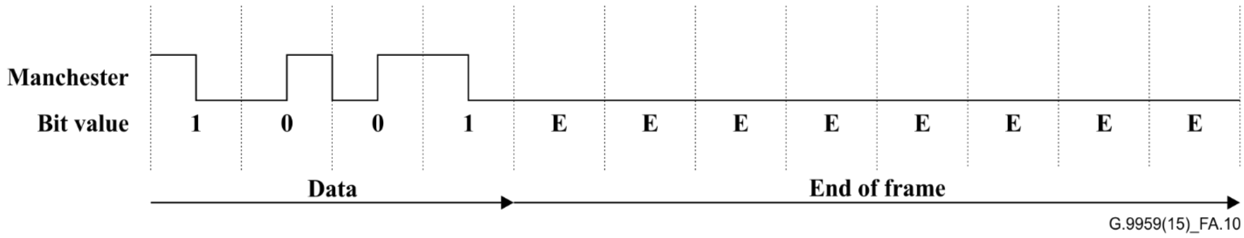

Figure A.10 – EOF delimiter pattern (R1 only)

A.3.3 PHY service specifications

The PHY layer shall provide two services, accessed via two SAPs: the PD, accessed via the PHY data SAP (PD-SAP), and the PHY management service, accessed via the PHY layer management entity SAP (PLME-SAP). The PLME is responsible for maintaining a database of managed objects pertaining to the PHY. This database is referred to as the PHY management information base (MIB). Figure A.11 shows the components and interfaces of the PHY.

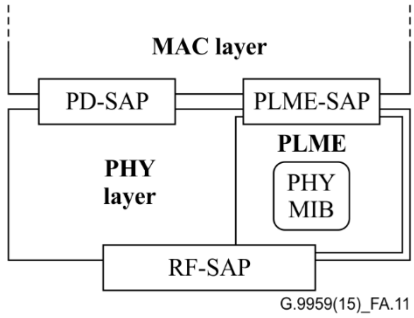

Figure A.11 – PHY reference model

A.3.3.1 PD service

The PD-SAP supports the transport of MPDUs between peer MAC entities. Table A.12 lists the primitives supported by the PD-SAP.

Table A.12 – PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	A.3.3.1.1	A.3.3.1.2	A.3.3.1.3

A.3.3.1.1 PD-DATA.request

The PD-DATA.request primitive requests the transfer of an MPDU (i.e., the PSDU) from the MAC entity to the local PHY entity.

A.3.3.1.1.1 Semantics of the PD request primitive

The semantics of the PD-DATA.request primitive shall be as follows:

```
PD-DATA.request    (
                   psduChannel,
                   psduRate,
                   psduLength,
                   psdu
                   )
```

Table A.13 specifies the parameters for the PD-DATA.request primitive:

Table A.13 – PD-DATA.request parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B, C according to the applied RF profile (see Tables A.1 and A.3)	The physical channel to use
psduRate	Enumeration	Data rate R1, R2, R3 according to the applied RF profile (see Tables A.1 and A.3)	The data rate to use
psduLength	Byte	≤ *aPhyMaxFrameSizeRx*, where Rx is R1, R2, or R3 depending on actual data rate	The number of bytes to transmit
psdu	Byte Array	–	The PSDU to transmit

A.3.3.1.1.2 When generated

The PD-DATA.request primitive is generated by the MAC entity and issued to the PHY entity to request the transmission of an MPDU.

A.3.3.1.1.3 Effects on receipt

The receipt of the PD-DATA.request primitive by the PHY entity shall cause the transmission of the supplied PSDU. Provided the TX is enabled (TX_ON mode), the PHY shall construct a PPDU, containing the supplied PSDU, and then transmit the PPDU. When the PHY entity has completed the transmission, it shall 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 mode) or if the TRX is disabled (TRX_OFF mode), the PHY entity shall issue the PD-DATA.confirm primitive with a status of RX_ON or TRX_OFF, respectively.

A.3.3.1.2 PD-DATA.confirm

The PD-DATA.confirm primitive confirms the end of the transmission of an MPDU (i.e., PSDU) from the MAC entity to the physical media.

A.3.3.1.2.1 Semantics of the PD confirm primitive

The semantics of the PD-DATA.confirm primitive shall be as follows:

```
PD-DATA.confirm   (
                  status
                  )
```

Table A.14 specifies the parameters for the PD-DATA.confirm primitive:

Table A.14 – PD-DATA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF	Transmission was successful Error: Currently receiving Error: Transmitter is disabled

A.3.3.1.2.2 When generated

The PD-DATA.confirm primitive is generated by the PHY entity and issued to the MAC entity in response to a PD-DATA.request primitive. The PD-DATA.confirm primitive shall return a status of either SUCCESS, indicating that the request to transmit was successful, or an error code of RX_ON or TRX_OFF.

A.3.3.1.2.3 Effects on receipt

The PD-DATA.confirm primitive allows the MAC entity to take proper action when the transmission has been completed.

A.3.3.1.3 PD-DATA.indication

The PD-DATA.indication primitive indicates the transfer of an MPDU (i.e., PSDU) from the PHY to the local MAC entity.

A.3.3.1.3.1 Semantics of the PD indication primitive

The semantics of the PD-DATA.indication primitive shall be as follows:

PD-DATA.indication (
psduByte
)

Table A.15 specifies the parameters for the PD-DATA.indication primitive.

Table A.15 – PD-DATA.indication parameters

Name	Type	Valid range	Description
psduByte	Byte	–	One byte of the PSDU to be transmitted by the PHY entity

A.3.3.1.3.2 When generated

The PD-DATA.indication primitive is generated by the PHY entity and issued to the MAC entity to transfer a received PSDU.

A.3.3.1.3.3 Effect on receipt

On receipt of the PD-DATA.indication primitive, the MAC entity is notified of the arrival of a MPDU data. The MAC layer shall monitor incoming bytes until a complete MPDU has been received. The MAC layer shall use the Length field to determine the length of the MPDU. The MAC layer shall verify the FCS before issuing an MD-DATA.indication to higher layers.

A.3.3.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME and the PLME. Table A.16 lists the primitives supported by the PLME-SAP.

Table A.16 – PLME-SAP primitives

PLME-SAP primitive	Request	Confirm	Indication	Response
PLME-SOF	–	–	A.3.3.2.1	–
PLME-GET-CCA	A.3.3.2.2	A.3.3.2.3	–	–
PLME-GET	A.3.3.2.4	A.3.3.2.5	–	–
PLME-SET-TRX-MODE	A.3.3.2.6	A.3.3.2.7	–	–
PLME-SET	A.3.3.2.8	A.3.3.2.9	–	–

A.3.3.2.1 PLME-SOF.indication

The PLME-SOF.indication primitive indicates the reception of a SOF delimiter from the PHY to the MAC entity.

A.3.3.2.1.1 Semantics for the service primitive

The semantics of the PLME-SOF.indication primitive shall be as follows:

PLME-SOF.indication (
psduChannel,
psduRate
)

Table A.17 specifies the parameters for the PD-DATA.indication primitive.

Table A.17 – PLME-SOF.indication parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B, C	The channel from which the PSDU was received
psduRate	Enumeration	Data rate R1, R2, R3	The bit rate of the received PSDU

A.3.3.2.1.2 When generated

The PLME-SOF.indication primitive is generated by the PLME and issued to the PLME whenever a SOF delimiter is detected by the PHY.

A.3.3.2.1.3 Effect on receipt

The MAC entity is notified of the reception of a SOF delimiter. This information may be used by the MAC entity for preparing frame reception and inhibiting transmissions.

A.3.3.2.2 PLME-GET-CCA.request

The PLME-GET-CCA.request primitive requests a CCA for a specified channel.

A.3.3.2.2.1 Semantics for the service primitive

The semantics of the PLME-GET-CCA.request primitive shall be as follows:

```
PLME-GET-CCA.request    (
                        channel
                        )
```

The table below specifies the parameters for the PLME-GET-CCA.request primitive.

Table A-18 – PLME-GET-CCA.request parameters

Name	Type	Valid range	Description
channel	Enumeration	Channel A, B, or C according to the applied RF profile (see Tables A.1 and A.3)	The physical channel on which the CCA shall be performed

A.3.3.2.2.2 When generated

The PLME-GET-CCA.request primitive is generated by the MLME and issued to the PLME to request information regarding the specified channel.

A.3.3.2.2.3 Effect on receipt

On receipt of the PLME-GET-CCA.request primitive, the PLME should perform a CCA for the specified channel. When the operation is completed the PLME shall issue a PLME-GET-CCA.confirm advertising the status.

A.3.3.2.3 PLME-GET-CCA.confirm

The PLME-GET-CCA.confirm primitive reports the result of a CCA request.

A.3.3.2.3.1 Semantics for the service primitive

The semantics of the PLME-GET-CCA.confirm primitive shall be as follows:

```
PLME-GET-CCA.confirm    (
                        status
                        )
```

Table A.19 specifies the parameters for the PLME-GET-CCA.confirm primitive.

Table A.19 – PLME-GET-CCA.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	CCA_CLEAR, CCA_NOT_CLEAR, CCA_RX_OFF	The result of the CCA request

A.3.3.2.3.2 When generated

The PLME-GET-CCA.confirm primitive shall be generated by the PLME in response to a PLME-GET-CCA.request primitive. The PLME-GET-CCA.confirm primitive may return a status of CCA_CLEAR, CCA_NOT_CLEAR or RX_OFF. The CCA_RX_OFF status shall be returned if the TRX is not in RX mode (and thus, unable to perform a CCA).

A.3.3.2.3.3 Effect on receipt

The MLME is notified of the result of the CCA operation. This information may be used by the MAC entity for channel availability evaluation or for deciding whether to transmit now.

A.3.3.2.4 PLME-GET.request

The PLME-GET.request primitive requests the value of the specified PHY MIB attribute.

A.3.3.2.4.1 Semantics for the service primitive

The semantics of the PLME-GET.request primitive shall be as follows:

```
PLME-GET.request    (
                    PhyMibAttribute
                    )
```

Table A.20 specifies the parameters for the PLME-GET.request primitive.

Table A.20 – PLME-GET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See Table A.28	The identifier of the PHY MIB attribute to get

A.3.3.2.4.2 When generated

The PLME-GET.request primitive shall be generated by the MLME and issued to the PLME to request information from the PHY MIB.

A.3.3.2.4.3 Effect on receipt

On receipt of the PLME-GET.request primitive, the PLME should retrieve the value of the specified PHY MIB attribute.

A.3.3.2.5 PLME-GET.confirm

The PLME-GET.confirm primitive reports the result of an information request from the PHY MIB.

A.3.3.2.5.1 Semantics for the service primitive

The semantics of the PLME-GET.confirm primitive shall be as follows:

```
PLME-GET.confirm    (
                    status,
                    PhyMibAttribute,
```

PhyMibAttributeValue
)

Table A.21 specifies the parameters for the PLME-GET.confirm primitive.

Table A.21 – PLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the PHY MIB attribute request
PhyMibAttribute	Enumeration	See clause A.3.4.2	The identifier of the PHY MIB attribute
PhyMibAttributeValue	Various	Attribute specific	The value of the specified PHY MIB attribute

A.3.3.2.5.2 When generated

The PLME-GET.confirm primitive shall be generated by the PLME in response to a PLME-GET.request primitive.

If a non-existent PHY MIB attribute is requested, the PLME shall issue the PLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested PHY MIB attribute is found, the PLME shall issue the PLME-GET.confirm primitive with a status of SUCCESS as well as the MIB attribute identifier and its value.

A.3.3.2.5.3 Effect on receipt

On receipt of the PLME-GET.confirm primitive, the MLME is notified of the result of the PHY MIB attribute request. If the request was successful, MLME may use the returned MIB attribute value.

A.3.3.2.6 PLME-SET-TRX-MODE.request

The PLME-SET-TRX-MODE.request primitive requests that the PHY entity changes the operating mode of the TRX. The TRX may be set to one of the modes outlined in Table A.22.

A.3.3.2.6.1 Semantics for the service primitive

The semantics of the PLME-SET-TRX-MODE.request primitive shall be as follows:

PLME-SET-TRX-MODE.request (
mode
)

Table A.22 specifies the parameters for the PLME-SET-TRX-MODE.request primitive.

Table A.22 – PLME-SET-TRX-MODE.request parameters

Name	Type	Valid values	Description
mode	Enumeration	RX_ON, TRX_OFF, FORCE_TRX_OFF, TX_ON	Enable receiver if possible Disable receiver and transmitter if possible Force receiver and transmitter off Enable transmitter if possible

A.3.3.2.6.2 When generated

The PLME-SET-TRX-MODE.request primitive may be generated by the MLME and issued to the PLME to change the operational mode of the TRX.

A.3.3.2.6.3 Effect on receipt

On receipt of the PLME-SET-TRX-MODE.request primitive, the PHY should change the TRX operation mode.

If the PHY is busy receiving or transmitting, the PHY shall ignore the mode request.

If this primitive is issued with FORCE_TRX_OFF, the PHY shall set the TRX mode to TRX_OFF irrespective of the current mode.

A.3.3.2.7 PLME-SET-TRX-MODE.confirm

The PLME-SET-TRX-MODE.confirm primitive shall report the operating mode of the TRX in response to a PLME-SET-TRX-MODE.request.

If the TRX operation mode is changed, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with a status of SUCCESS.

If the TRX is requested to change to the current operation mode, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with a status advertising the current mode, i.e., RX_ON, TRX_OFF, or TX_ON.

If the TRX is requested to change to the RX_ON or TRX_OFF mode and the PHY is busy transmitting, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with the status BUSY_TX.

If the TRX is requested to change to the TX_ON or TRX_OFF mode and the PHY is busy receiving, the PHY shall issue the PLME-SET-TRX-MODE.confirm primitive with the status BUSY_RX.

A.3.3.2.7.1 Semantics for the service primitive

The semantics of the PLME-SET-TRX-MODE.confirm primitive shall be as follows:

```
PLME-SET-TRX-MODE.confirm    (
                              status
                              )
```

Table A.23 specifies the parameters for the PLME-SET-TRX-MODE.confirm primitive.

Table A.23 – PLME-SET-TRX-MODE.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, RX_ON, TRX_OFF, TX_ON, BUSY_RX, or BUSY_TX	Change request was successful Receiver is enabled Receiver and transmitter are disabled Transmitter is enabled Receiver is enabled and busy Transmitter is enabled and busy

A.3.3.2.7.2 When generated

The PLME-SET-TRX-MODE.confirm primitive is generated by the PLME and issued to the MLME.

A.3.3.2.7.3 Effect on receipt

The MLME is notified of the operating mode of the TRX.

The PLME-SET-TRX-MODE.confirm primitive may advertise the status BUSY_RX or BUSY_TX. This indicates that the request for a new operation mode was ignored.

A.3.3.2.8 PLME-SET.request

The PLME-SET.request primitive may be issued to request that the specified PHY MIB attribute is set to the specified value.

A.3.3.2.8.1 Semantics for the service primitive

The semantics of the PLME-SET.request primitive shall be as follows:

```
PLME-SET.request   (
                   PhyMibAttribute,
                   PhyMibAttributeValue
                   )
```

Table A.24 specifies the parameters for the PLME-SET.request primitive.

Table A.24 – PLME-SET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See clause A.3.4.2	The identifier of the PHY MIB attribute
PhyMibAttributeValue	Various	Attribute specific	The value of the PHY MIB attribute

A.3.3.2.8.2 When generated

The PLME-SET.request primitive is generated by the MLME and issued to the PLME to set the specified PHY MIB attribute.

A.3.3.2.8.3 Effect on receipt

On receipt of the PLME-SET.request primitive, the PLME should set the specified PHY MIB attribute to the specified value.

If a non-existent PHY MIB attribute is specified, the PLME shall not change any MIB attribute.

If the specified value is invalid for the specified PHY MIB attribute, the PLME shall not change the PHY MIB attribute value.

If the specified PHY MIB attribute is changed, the change should take effect immediately.

A.3.3.2.9 PLME-SET.confirm

The PLME-SET.confirm primitive shall report the result of a request MIB attribute change.

A.3.3.2.9.1 Semantics for the service primitive

The semantics of the PLME-SET.confirm primitive shall be as follows:

```
PLME-SET.confirm   (
                   status,
                   PhyMibAttribute
                   )
```

Table A.25 specifies the parameters for the PLME-SET.confirm primitive.

Table A.25 – PLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER	The status of the requested MIB attribute change
PhyMibAttribute	Enumeration	See clause A.3.4.2	The identifier of the PHY MIB attribute

A.3.3.2.9.2 When generated

The PLME-SET.confirm primitive shall be generated by the PLME and issued to the MLME in response to a PLME-SET.request primitive.

If a non-existent PHY MIB attribute is specified, the PLME shall advertise a status of UNSUPPORTED_ATTRIBUTE.

If the specified value is invalid for the specified PHY MIB attribute, the PLME shall advertise a status of INVALID_PARAMETER.

If the specified PHY MIB attribute is updated, the PLME shall advertise a status of SUCCESS.

A.3.3.2.9.3 Effect on receipt

The MLME is notified of the result of request PHY MIB attribute change. The MLME should verify that the status parameter advertises the value SUCCESS. Refer to Table A.25.

A.3.3.3 PHY enumerations description

Table A.26 shows PHY enumeration values defined in the PHY specification.

Table A.26 – PHY enumerations description

Enumeration	Description
BUSY	The CCA operation detected a busy channel
BUSY_RX	The transceiver is receiving and cannot change its mode (unless forced)
BUSY_TX	The transceiver is transmitting and cannot change its mode (unless forced)
CCA_CLEAR	The CCA operation detected a clear channel
FORCE_TRX_OFF	The transceiver shall be disabled, even if it is currently receiving or transmitting data
INVALID_PARAMETER	The specified parameter is out of range for the actual MIB attribute
CCA_NO_CLEAR	The CCA operation detected a busy channel
RX_ON	The receiver is enabled (used for command as well as status)
CCA_RX_OFF	The CCA request failed: The receiver is disabled
SUCCESS	The operation was successful
TRX_OFF	The transceiver is disabled (used for command as well as status)
TX_ON	The transmitter is enabled (used for command as well as status)
UNSUPPORTED_ATTRIBUTE	The specified MIB attribute is not supported

A.3.4 PHY constants and MIB attributes

This clause specifies the constants and attributes relating to the PHY layer.

A.3.4.1 PHY constants

The PHY shall comply with the constants defined Table A.27.

Table A.27 – PHY constants

Constants	Description	Value
aPhyMaxFrameSizeR1	The maximum PSDU size to support at data rate R1	64 byte
aPhyMaxFrameSizeR2	The maximum PSDU size to support at data rate R2	64 byte
aPhyMaxFrameSizeR3	The maximum PSDU size to support at data rate R3	170 byte
aPhyTurnaroundTimeTXRX	TX-to-RX maximum turnaround time (see clause A.3.1.13)	1 ms
aPhyTurnaroundTimerRXTX	RX-to-TX minimum turnaround time (see clause A.3.1.14)	1 ms

A.3.4.2 PHY MIB attributes

The PHY management information base (MIB) comprises the attributes required to manage the PHY. 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 MIB are presented in Table A.28.

Table A.28 – PHY MIB attributes

Attribute	Type	Range	Description
phyCurrentTxChannel	Enumeration	A, B, C (Refer to Table A.3)	The TX channel to use
phyMapChannelA	Enumeration	Available RF profiles (Refer to Table A.1)	Apply RF profile to channel A
phyMapChannelB	Enumeration	Available RF profiles (Refer to Table A.1)	Apply RF profile to channel B
phyMapChannelC	Enumeration	Available RF profiles (Refer to Table A.1)	Apply RF profile to channel C
phyTransmitPower	Enumeration	Valid output power levels (Refer to clause A.3.1.7)	The transmit power to use

A.4 Medium access (MAC) layer specification

The MAC layer handles all access to the physical layer and is responsible for the following tasks:

- Frame ACK;
- Retransmission;
- Providing a reliable link between two peer MAC entities.

Constants and attributes that are specified and maintained by the MAC layer are written in the text of this clause in italics. Constants have a general prefix of "a", e.g., *aMacMaxFrameRetries*, and are listed in Tables A.47 and A.48. Attributes have a general prefix of "mac", e.g., *macHomeID*, and are listed in Table A.47.

A.4.1 MAC layer service specification

The MAC layer provides an interface between the network layer and the PHY layer. The MLME provides the service interfaces through which layer management functions may be invoked. The MLME is responsible for maintaining a database of managed objects pertaining to the MAC layer. This database is referred to as the MAC layer MAC MIB. Figure A.12 depicts the components and interfaces of the MAC layer.

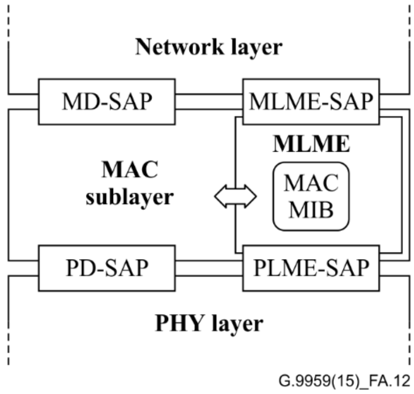

Figure A.12 – MAC layer reference model

The MAC layer provides two services to the network layer, accessed through two SAPs:

– The MD service, accessed through the MD-SAP;

– The MAC management service, accessed through the MLME-SAP.

A.4.1.1 MD service

The MD-SAP supports the transport of NPDU between peer network layer entities. Table A.29 lists the primitives supported by the MD-SAP. The primitives are discussed in the clauses referenced in the table.

Table A.29 – MD-SAP primitives

MD-SAP primitive	Request	Confirm	Indication
MD-DATA	A.4.1.1.1	A.4.1.1.2	A.4.1.1.3

A.4.1.1.1 MD-DATA.request

The MD-DATA.request primitive requests the transfer of an NPDU (i.e., MSDU) from the network layer to the PHY entity.

A.4.1.1.1.1 Semantics of the service primitive

The semantics of the MD-DATA.request primitive shall be as follows:

```
MD-DATA.request   (
                  SrcHomeID,
                  SrcNodeID,
                  DstNodeID,
                  msduLength,
```

msdu,
SequenceNumber,
TxType,
TxOptions,
BeamOption,
RateOption,
ChannelOption,
MulticastOption
)

Table A.30 specifies the parameters for the MD-DATA.request primitive:

Table A.30 – MD-Data.request parameters

Name	Type	Valid range	Description
SrcHomeID	Byte Array	0x00000000..0xFFFFFFFF	Domain identifier to use
SrcNodeID	Byte	0x00..0xE8	Source node identifier to use: 0x00: Uninitialized Z-Wave Node ID 0x01 – 0xE8: NodeID 0xE9 – 0xFE: Reserved
DstNodeID	Byte	0x00..0xFF	Destination node identifier to use: 0x00: Reserved 0x01 – 0xE8: NodeID 0xE9 – 0xFE: Reserved 0xFF: Broadcast NodeID
msduLength	Byte	≤ *aMacMaxMSDUSizeX*	Number of bytes to transmit
msdu	Byte Array	–	MSDU to transmit
SequenceNumber	Byte	–	Unique number of this frame. The same value shall be used for retransmissions
TxType	Byte	0x01..0x03	Transmission type to use: 0x01 = Singlecast or broadcast transmission 0x02 = Multicast transmission 0x03 = Acknowledged transmission
TxOptions	Byte	0x01, 0x02, 0x04	Transmission options for this MSDU The control word is formed as a bit wise OR of one or more of the following values: 0x01 = Routed frame transmission 0x02 = Acknowledged transmission 0x04 = Low power transmission (Note 1)
BeamOption	Byte	0x00, 0x02..0x04	Battery support options for this MSDU: 0x00: No beam 0x02: Short Continuous beam 0x03: Long Continuous beam 0x04: Fragmented beam
ChannelOption	Enumeration	Channel A,B, C (Note 2)	Channel to use
RateOption	Enumeration	Data rate R1, R2, R3 according to the applied	Data rate to use

Table A.30 – MD-Data.request parameters

Name	Type	Valid range	Description
		RF profile (see Tables A.1, A.2 and A.3)	
MulticastOption	Byte + Bitmap (30 bytes)	–	If TxType is "Multicast transmission", this bitmap indicates the intended recipients
NOTE 1 – Bit 2 (0x04) controls which transmission power level the MAC layer requests from the PHY layer. NOTE 2 – Channels A, B, and C are preconfigured to RF profiles from the list presented in clause A.3.1.1 via MAC MIB attributes defined in Table A.47. Channel configuration MAC MIB profiles map directly to PHY MIB profiles.			

A.4.1.1.1.2 When generated

The MD-DATA.request primitive is generated by a local network layer entity when a data NPDU (i.e., MSDU) is to be transferred to one or more peer network layer entities.

The TxOptions.LowPower is an option targeting the PHY layer. The option shall make the MAC request the PHY power level defined by the MAC constant aMacTxPhyPowerLevelLow instead of aMacTxPhyPowerLevelNormal.

A.4.1.1.1.3 Effects on receipt

The receipt of the MD-DATA.request primitive by the MAC entity shall cause the transmission of the supplied MSDU.

The MAC entity builds an MPDU to transmit from the supplied parameters. The TxOptions parameters indicate optional parameters on how the MAC entity transmits the supplied MSDU.

The MAC entity checks for a CCA (see clause A.4.4.1.1). If the PHY PLME-GET-CCA primitive returns TRUE, the MAC entity enables the TX by issuing the PLME-SET-TRX-MODE.request primitive with a mode of TX_ON to the PHY. On receipt of the PLME-SET-TRX-MODE.confirm primitive with a status of either SUCCESS or TX_ON the constructed MPDU is then transmitted by issuing the PD-DATA.request primitive. Finally, on receipt of the PD-DATA.confirm primitive, the MAC entity disables the TX by issuing the PLME-SET-TRX-MODE.request primitive with a mode of RX_ON to the PHY.

If the TxOptions parameter specifies that acknowledged transmission is required, the MAC entity shall enable its receiver immediately following the transmission of the MPDU and wait for an ACK for at least *aMacMinAckWaitDuration* symbols. If the MAC entity does not receive an ACK within this time, it shall retransmit the MPDU one or more times as defined by *aMacMaxFrameRetries*. If the MAC entity still does not receive an ACK, it shall discard the MSDU and issue the MD-DATA.confirm primitive with a status of NO_ACK.

If the MPDU was successfully transmitted, the MAC entity shall issue the MD-DATA.confirm primitive with a status of SUCCESS.

If the MPDU could not be transmitted due to a busy channel (see clause A.4.4.1.1), the MAC entity shall issue the MD-DATA.confirm primitive with a status of NO_CCA.

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

If the MSDU length is longer than *aMacMaxMSDUSizeX* the MAC entity shall issue the MD-DATA.confirm primitive with a status of FRAME_TOO_LONG.

The MD-Data.request parameters are used to construct the MHR see clause A.4.2.

A.4.1.1.2 MD-DATA.confirm

The MD-DATA.confirm primitive confirms the transmission of an MSDU.

A.4.1.1.2.1 Semantics of the service primitive

The semantics of the MD-DATA.confirm primitive shall be as follows:

```
MD-DATA.confirm  (
                 status
                 )
```

Table A.31 specifies the parameters for the MD-DATA.confirm primitive.

Table A.31 – MD-Data.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, NO_ACK, INVALID_PARAMETER, NO_CCA, FRAME_TOO_LONG	Transmission was successful Transmission was not acknowledged Transmission aborted; invalid parameter Transmission aborted; no clear channel Transmission aborted; frame too long

A.4.1.1.2.2 When generated

The MD-DATA.confirm primitive is generated by the MAC entity in response to an MD-DATA.request primitive. The MD-DATA.confirm primitive shall return a status as defined in Table A.31.

The primitive may be delayed if the MAC has to wait for an ACK. If the frame is transmitted using beaming, the delay may be more than one second.

A.4.1.1.2.3 Effects on receipt

The MD-DATA.confirm primitive allows higher layers to take proper action whether the transmission was successful or unsuccessful.

A.4.1.1.3 MD-DATA.indication

The MD-DATA.indication primitive indicates the reception of a data NPDU (i.e., MSDU) from the MAC entity to higher layer entities.

A.4.1.1.3.1 Semantics of the service primitive

The semantics of the MD-DATA.indication primitive shall be as follows:

```
MD-DATA.indication    (
                      FrmType,
                      SrcHomeID,
                      SrcNodeID,
                      DstNodeID,
                      msduLength,
                      msdu,
                      SequenceNumber,
                      ChannelOption,
                      RateOption,
                      MulticastOption
                      )
```

Table A.32 specifies the parameters for the MD-DATA.indication primitive.

FrmType indicates if the data delivered to the network layer is a data frame or a beam fragment.

Table A.32 – MD-DATA.indication parameters

Name	Type	Valid range	Description
FrmType	Enumeration	0x01..0xFF	Code indicating the frame type: 0x00: Uninitialized type 0x01: Singlecast frame 0x02: Multicast frame 0x03: ACK frame 0x04..0x07: Not to be used (Note) 0x08: Routed frame 0x09: Beam fragment 0x0A..0xFF: Reserved
SrcHomeID	Byte Array	0x00000000..0xFFFFFFFF The value 0x00000000 is reserved for un-initialized legacy nodes. HomeID 0x00000000 shall not be used for any domain	The 4-byte domain identifier of the node from which the MSDU was received
SrcNodeID	Byte	0x00..0xFF The value 0x00 is reserved for un-initialized nodes.	NodeID of the originating node. 0x00: Uninitialized NodeID 0x01..0xE8: NodeID 0xE9..0xFF: Reserved
DstNodeID (Not applicable for multicast frame type)	Byte	0x00..0xFF	NodeID of the destination node 0x00: Uninitialized NodeID 0x01..0xE8: NodeID 0xE9..0xFE: Reserved 0xFF: Broadcast NodeID
msduChannel	Enumeration	Channel A, B, C	The physical channel from which the MSDU was received
msduRate	Enumeration	Data rate R1, R2, R3	The bit rate of the received MSDU
msduLength	Byte	≤ *aMacMaxMSDUSizeX*	The number of received bytes
msdu	Byte Array	–	The received MSDU
SequenceNumber	Byte	–	The unique number of this frame. Use same value for retransmissions
MulticastOption	Byte + Bitmap (30 bytes)	–	If TxType is "Multicast transmission", this bitmap indicates the intended recipients
NOTE – These fields shall not be used due to backwards compatibility.			

In case of a beam frame, the SrcHomeID field shall be formatted as specified in clause A.4.2.10.

A.4.1.1.3.2 When generated

The MD-DATA.indication primitive is generated by the MAC entity on receipt of a frame from the PHY layer. If the frame checksum is valid, the frame shall be forwarded to the Z-Wave network layer.

A.4.1.1.3.3 Effects on receipt

The network layer is notified of the arrival of data.

Also, beam fragments shall be forwarded to the network layer, which may then forward the beam fragment to higher layers. Higher layers of an FLN node may decide to re-enable sleep mode if the NodeID of a beam fragment is intended for another node.

A.4.1.1.4 Data service message sequence charts

Figure A.13 illustrates the sequence of messages necessary for a successful data transfer between two nodes.

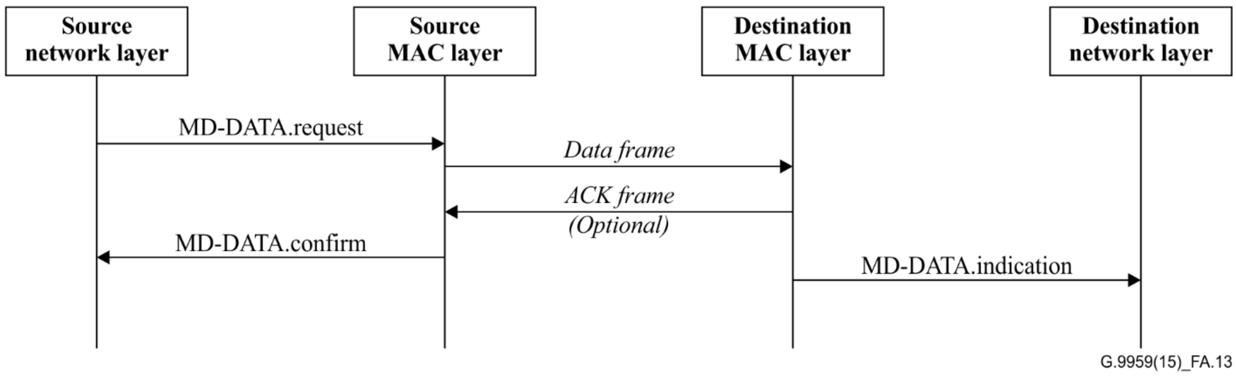

Figure A.13 – MAC data service message sequence chart

A.4.1.2 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table A.33 summarizes the primitives supported by the MLME through the MLME-SAP interface.

The primitives are discussed in the clauses referenced in the table.

Table A.33 – MLME-SAP primitives

MLME-SAP Primitive	Request	Confirm	Indication	Response
MLME-GET	A.4.1.2.1	A.4.1.2.2	–	–
MLME-SET	A.4.1.2.3	A.4.1.2.4	–	–
MLME-RESET	A.4.1.2.5	A.4.1.2.6	–	–

A.4.1.2.1 MLME-GET.request

The MLME-GET.request primitive requests information about a given MAC MIB attribute.

A.4.1.2.1.1 Semantics of the service primitive

The semantics of the MLME-GET.request primitive shall be as follows:

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

Table A.33a specifies the parameters for the MLME-GET.request primitive.

Table A.33a – MLME-GET.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table A.49	The identifier of the MAC MIB attribute to read

A.4.1.2.1.2 When generated

The MLME-GET.request primitive is generated by the next higher layer and issued to the MLME to obtain information from the MAC MIB.

A.4.1.2.1.3 Effects on receipt

On receipt of the MLME-GET.request primitive, the MLME attempts to retrieve the requested MAC MIB attribute from its database. If the identifier of the MAC MIB attribute is not found in the database, the MLME shall issue the MLME-GET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE.

If the requested MAC MIB attribute is successfully retrieved, the MLME shall issue the MLME-GET.confirm primitive with a status of SUCCESS and the MAC MIB attribute value.

A.4.1.2.2 MLME-GET.confirm

The MLME-GET.confirm primitive reports the result of a MAC MIB attribute request.

A.4.1.2.2.1 Semantics of the service primitive

The semantics of the MLME-GET.confirm primitive shall be as follows:

```
MLME-GET.confirm    (
                    status,
                    MacMibAttribute,
                    MacMibAttributeValue
                    )
```

Table A.34 specifies the parameters for the MLME-GET.confirm primitive.

Table A.34 – MLME-GET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The result of the request for MAC MIB attribute information
MacMibAttribute	Integer	See Table A.49	The identifier of the MAC MIB attribute that was read
MacMibAttributeValue	Various	Attribute specific, See Table A.49	The value of the indicated MAC MIB attribute that was read

A.4.1.2.2.2 When generated

The MLME-GET.confirm primitive is generated by the MLME and issued to the 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 MAC MIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE.

A.4.1.2.2.3 Effects on receipt

The MLME-GET.confirm primitive reports the result of the MAC MIB attribute request. If the request was successful, the requester may use the returned MIB attribute value.

A.4.1.2.3 MLME-SET.request primitives

The MLME-SET.request primitive may be used to request that the specified MAC MIB attribute is set to the specified value.

A.4.1.2.3.1 Semantics of the service primitive

The semantics of the MLME-SET.request primitive shall be as follows:

```
MLME-SET.request    (
                    MacMibAttribute,
                    MacMibAttributeValue
                    )
```

Table A.35 specifies the parameters for the MLME-SET.request primitive.

Table A.35 – MLME-SET.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table A.49	The identifier of the MAC MIB attribute to write
MacMibAttributeValue	Various	Attribute specific, See Table A.49	The value to write to the indicated MAC MIB attribute

A.4.1.2.3.2 When generated

The MLME-SET.request primitive is generated by the next higher layer and issued to the MLME to set the specified MAC MIB attribute.

A.4.1.2.3.3 Effects on receipt

On receipt of the MLME-SET.request primitive, the MLME should set the specified MAC MIB attribute to the specified value.

If a non-existent MAC MIB attribute is specified, the MLME shall not change any MIB attribute.

If the specified value is invalid for the specified MAC MIB attribute, the MLME shall not change the MAC MIB attribute value.

If the specified MAC MIB attribute is changed, the change should take effect immediately.

A.4.1.2.4 MLME-SET.confirm

The MLME-SET.confirm primitive shall report the result of the requested MAC MIB attribute change.

A.4.1.2.4.1 Semantics of the service primitive

The semantics of the MLME-SET.confirm primitive shall be as follows:

```
MLME-SET.confirm    (
                    status,
                    MacMibAttribute
                    )
```

Table A.36 specifies the parameters for the MLME-SET.confirm primitive.

Table A.36 – MLME-SET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, INVALID_PARAMETER	Attribute successfully changed Attribute is not supported Attribute value is invalid
MacMibAttribute	Integer	Table A.49	The identifier of the MAC MIB attribute that was written.

A.4.1.2.4.2 When generated

The MLME-SET.confirm primitive shall be generated by the MLME and issued to the next higher layer in response to an MLME-SET.request primitive.

If a non-existent MAC MIB attribute is specified, the MLME shall advertise a status of UNSUPPORTED_ATTRIBUTE.

If the specified value is invalid for the specified MAC MIB attribute, the MLME shall advertise a status of INVALID_PARAMETER.

If the specified MAC MIB attribute is updated, the MLME shall advertise a status of SUCCESS.

A.4.1.2.4.3 Effects on receipt

The next higher layer is notified of the result of the MAC MIB attribute change request. The next higher layer should verify that the status parameter advertises the status value SUCCESS. Refer to Table A.36.

A.4.1.2.5 MLME-RESET.request

The MLME-RESET.request primitive allows the next higher layer to request that the MLME performs a reset operation.

A.4.1.2.5.1 Semantics of the service primitive

The semantics of the MLME-RESET.request primitive shall be as follows:

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

Table A.37 specifies the parameter for the MLME-RESET.request primitive.

Table A.37 – MLME-RESET.request parameters

Name	Type	Valid range	Description
SetDefaultMIB	Boolean	TRUE FALSE	Reset MAC entity and all MIB attributes Reset MAC entity state machine only

A.4.1.2.5.2 When generated

The MLME-RESET.request primitive is generated by the next higher layer and issued to the MLME to request a reset of the MAC entity to its initial conditions. The MLME-RESET.request primitive is issued when a node is excluded from a domain.

A.4.1.2.5.3 Effects on receipt

On receipt of the MLME-RESET.request primitive, the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF. On receipt of the PLME-SET-TRX-STATE.confirm primitive, the MAC entity is then set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultMIB parameter is set to TRUE, the MAC MIB attributes are set to their default values.

If the PLME-SET-TRX-STATE.confirm primitive is successful, the MLME shall issue the MLME-RESET.confirm primitive with the status of SUCCESS. Otherwise, the MLME shall issue the MLME-RESET.confirm primitive with the status of DISABLE_TRX_FAILURE.

A.4.1.2.6 MLME-RESET.confirm

The MLME-RESET.confirm primitive reports the results of the reset operation.

A.4.1.2.6.1 Semantics of the service primitive

The semantics of the MLME-RESET.confirm primitive shall be as follows:

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

Table A.38 specifies the parameter for the MLME-RESET.confirm primitive.

Table A.38 – MLME-RESET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS DISABLE_TRX_FAILURE	MAC entity successfully reset MAC entity reset operation failed

A.4.1.2.6.2 When generated

The MLME-RESET.confirm primitive is generated by the MLME and issued to the next higher layer in response to an MLME-RESET.request primitive and following the receipt of the PLME-SET-TRX-STATE.confirm primitive.

A.4.1.2.6.3 Effects on receipt

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of the request to reset the MAC entity. This primitive shall return a status as defined in Table A.38.

A.4.1.3 MAC enumeration description

This clause explains the meaning of the enumerations used in the primitives defined in the MAC entity specification. Table A.39 shows a description of the MAC enumeration values.

Table A.39 – MAC enumerations description

Enumeration	Description
SUCCESS	The requested operation was completed successfully
DISABLE_TRX_FAILURE	The attempt to disable the TRX has failed
FRAME_TOO_LONG	The frame has a length that is greater than *aMacMaxMSDUSizeX*
INVALID_PARAMETER	A parameter in the primitive is out of the valid range
NO_ACK	No ACK was received after*aMacMaxFrameRetries*
NO_CCA	No clear channel access was possible after a period of *.macCCARetryDuration*
UNSUPPORTED_ATTRIBUTE	A SET/GET request was issued with the identifier of a MAC MIB attribute that is not supported

A.4.2 MPDU formats

This clause specifies the format of the MPDU. Each MPDU consists of the following basic components:

1. An MHR, which comprises address, frame control, and length information;
2. A MAC data payload, of variable length, which contains information specific to the frame type;
3. An MFR, which contains a FCS.

The MPDU is defined as a sequence of fields. All MPDU formats in this clause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from $k - 1$ (leftmost and most significant) down to 0 (rightmost and least significant), where the length of the field is k bits. Bytes within each multi-byte field are numbered from 1 (leftmost and most significant) up to $n - 1$ (rightmost and least significant), where the length of the field is n bytes. Bits within each byte are numbered from 7 (leftmost and most significant) down to 0 (rightmost and least significant).

The general MPDU format comprises the fields MHR, data payload, and MFR. The general MPDU shall be formatted as illustrated in Figure A.14.

The MPDU formats for channel configurations 1, 2, and channel configuration 3 are explained in the following clauses.

Singlecast MPDUs shall use the formats outlined in Figures A.14 and A.15.

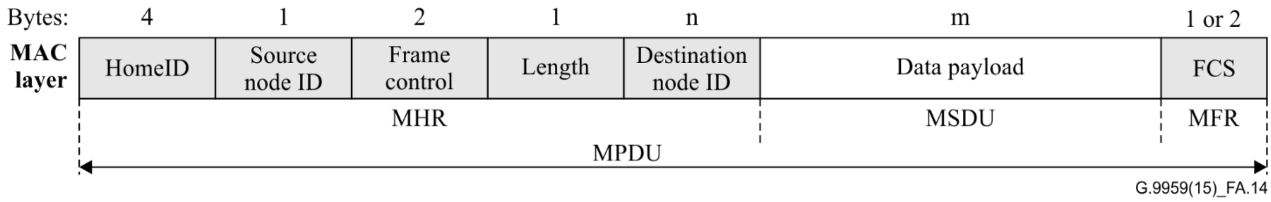

Figure A.14 – General MPDU format (Channel configurations 1, 2 only)

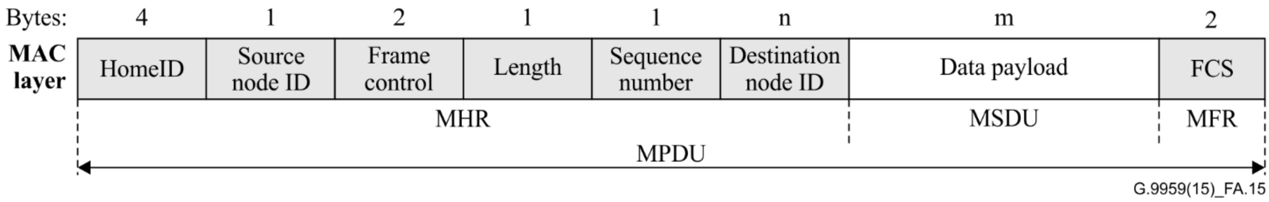

Figure A.15 – General MPDU format (Channel configuration 3 only)

The multicast MPDU shall use the formats outlined in Figures A.16 and A.17.

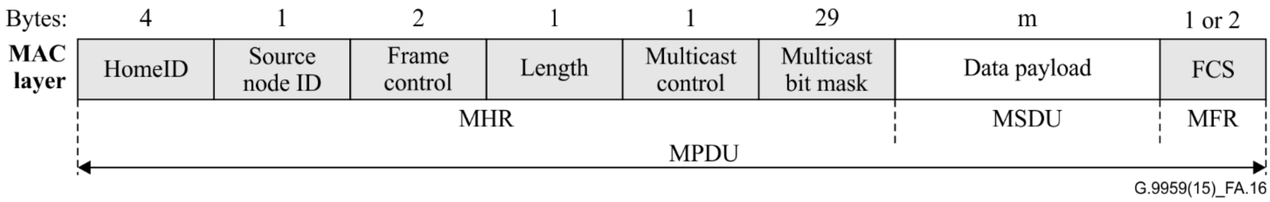

Figure A.16 – Multicast MPDU format (Channel configurations 1, 2 only)

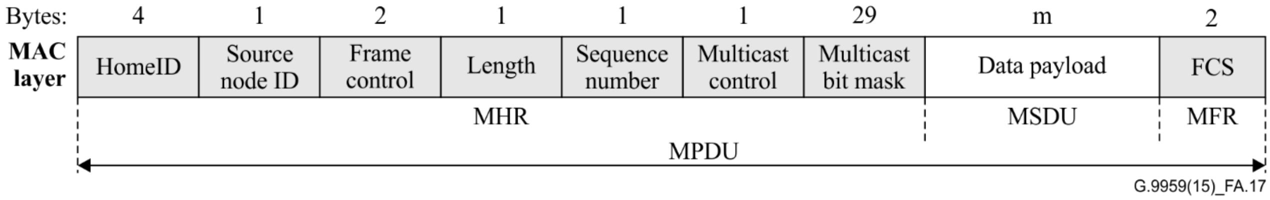

Figure A.17 – Multicast MPDU format (Channel configuration 3)

Refer to clause A.4.2.5.1 for details on transmission and processing of multicast frames.

A.4.2.1 HomeID

The HomeID identifier field is 4 bytes in length and specifies the unique domain identifier. All nodes in a domain shall have the same HomeID.

Byte \ Bit	7	6	5	4	3	2	1	0
0	HomeID (0) MSB							
1	HomeID (1)							
2	HomeID (2)							
3	HomeID (3) LSB							

G.9959(15)_FA.18

Figure A.18 – Format of a HomeID field

The MAC layer shall support configuration of a promiscuous mode, forwarding all MPDUs to higher layers. Refer to clause A.4.4.1.3.

A.4.2.2 Source NodeID

The source NodeID is 8 bits in length and shall be a unique identifier of a node in a given domain. Together with the HomeID, the source NodeID identifies the node that originated the frame.

Byte \ Bit	7	6	5	4	3	2	1	0
4	Source NodeID							

G.9959(15)_FA.19

Figure A.19 – Source NodeID field

The source NodeID shall comply with Table A.40.

Table A.40 – Source NodeID values

NodeID	Usage
0x00	Uninitialized NodeID
0x01..0xE8	NodeID
0xE9..0xFE	Reserved

A.4.2.3 Frame control

The Frame Control field is 16 bits in length and contains information defining the frame type and other control flags. The frame control field shall be formatted as illustrated in Figures A.20 and A.21.

Byte \ Bit	7	6	5	4	3	2	1	0
5	Routed	ACK Req.	Low power	Speed modified	Header type			
6	Reserved	Beaming Info.		Reserved	Sequence number			

G.9959(15)_FA.20

Figure A.20 –Frame Control field (Channel configurations 1, 2 only)

Byte \ Bit	7	6	5	4	3	2	1	0
5	ACK Req.	Low power	Reserved		Header type			
6	Reserved	Beaming Info.			Reserved			

G.9959(15)_FA.21

Figure A.21 – Format of a frame control field (Channel configuration 3)

A.4.2.3.1 Routed subfield (Channel configurations 1, 2 only)

The Routed subfield is 1 bit in length and shall be set to 0 when the frame is not routed and set to 1 when the frame is routed. The use of this field is out of scope of this Recommendation.

A.4.2.3.2 ACK Req subfield

The ACK Req subfield is 1 bit in length and set to 1 when the source node wants the destination node to acknowledge the frame, and the bit is set to 0 when no ACK is needed.

A receiving node shall return an ACK MPDU in response to the ACK request.

A.4.2.3.3 Low power subfield

The Low Power subfield is 1 bit in length and is set by the source node. The bit informs a destination node that the actual transmission was using low power.

A receiving node shall return an ACK MPDU in low power in response to the low power bit.

A.4.2.3.4 Speed modified subfield (Channel configurations 1, 2 only)

The Speed modified subfield is 1 bit in length. It shall be set to 1 if an MPDU is sent at a lower speed than supported by the source and destination. The field shall not be used for routed and multicast MPDUs. The field shall be set to 0 if the MPDU is sent at the highest speed supported by the source and destination.

A.4.2.3.5 Header type subfield

The header type subfield defines the frame header type.

Table A.41 – Header type

Header type value	Description
0x1	Singlecast MPDU
0x2	Multicast MPDU
0x3	ACK MPDU
0x4..0x7	Not to be used (Note)
0x8	Routed MPDU (Channel configuration 3 only)
0x9..0xB	Not to be used (Note)
0xC..0xF	*Reserved*
NOTE – These values shall not be transmitted due to backward compatibility. Received frames shall be forwarded to higher layers.	

A broadcast MPDU is a singlecast MPDU (header type 0x1) carrying a destination NodeID = 0xFF, see clause A.4.2.6.

A receiving node shall forward routed MPDUs to higher layers. The transmission of routed MPDUs is out of scope of this Recommendation.

A.4.2.3.6 Beaming Information

The Beaming Information subfield shall advertise the capability of a sending FL node to be awakened by a beam if it is asleep. The Beaming Information subfield shall be interpreted in combination with the actual channel configuration.

Table A.42 – Beaming information, channel configuration 1, 2

Channel configuration	Beam information value	Description
1,2	Frame control[6..5] = "00"	No beam
1,2	Frame control[6..5] = "01"	Short continuous beam
1,2	Frame control[6..5] = "10"	Long continuous beam
1,2	Frame control[6..5] = "11"	Reserved

Table A.43 – Beaming information, channel configuration 3

Channel configuration	Beam information value	Description
3	Frame control[6..4] = "000"	Reserved
3	Frame control[6..4] = "001"	Reserved
3	Frame control[6..4] = "010"	Reserved
3	Frame control[6..4] = "011"	Reserved
3	Frame control[6..4] = "100"	Fragmented beam
3	Frame control[6..4] = "101"	Reserved
3	Frame control[6..4] = "110"	Reserved
3	Frame control[6..4] = "111"	Reserved

A.4.2.3.7 Sequence Number (Channel configurations 1, 2 only)

The Sequence Number is a 4-bit sub-field of the frame control field. The Sequence Number is provided by higher layers when transmitting. The MAC layer of a transmitting node shall forward the sequence number value transparently to the PHY. The MAC layer of a receiving node shall forward the sequence number value transparently to higher layers.

The MAC layer shall use the same sequence number for the initial transmission and for all retransmissions of a given MPDU. The transmitted sequence number shall be in the range 0x1..0xf. The value 0xf shall be followed by the value 0x1.

A receiving node shall accept any sequence number value in the range 0x0..0xf. The receiving node shall return the same value in an ACK MPDU if ACK is requested. A transmitting node shall validate the received sequence number in an ACK MPDU. In order to support legacy implementations, a transmitting node which receives an ACK MPDU shall accept the sequence number value zero irrespective of the sequence number transmitted.

A.4.2.4 Length

The Length field is 1 byte in length and shall indicate the length of the MPDU in bytes.

Byte \ Bit	7	6	5	4	3	2	1	0
7	Length							

G.9959(15)_FA.22

Figure A.22 – Length field

The length is limited to *aMacMaxMSDUSizeX*. The actual values can be found in Tables A.47 and A.48. A receiving node shall not read more bytes than the maximum length allowed for the actual data rate.

A.4.2.5 Sequence number (Channel configuration 3 only)

The sequence number is an 8-bit field of the MPDU Header (MHR). The MAC layer of a transmitting node shall forward the sequence number value transparently to the PHY. The MAC layer of a receiving node shall forward sequence number value transparently to higher layers.

The MAC layer shall use the same sequence number for the initial transmission and for all retransmissions of a given MPDU. The transmitted sequence number shall be in the range 0x00..0xff. The value 0xff shall be followed by the value 0x00.

A receiving node shall accept any sequence number value in the range 0x00..0xff. The receiving node shall return the same value in an ACK MPDU if ACK is requested. A transmitting node shall validate the received sequence number in an ACK MPDU.

A.4.2.6 Destination NodeID

The destination NodeID shall specify a destination node in the same domain identified by the HomeID.

The destination NodeID value shall comply with Table A.44:

Table A.44 – Destination NodeID

NodeID	Node type
0x00	Uninitialized NodeID
0x01..0xE8	NodeID
0xE9..0xFE	Reserved
0xFF	Broadcast NodeID

The destination NodeID value 0xFF may be used for broadcasting an MPDU to all nodes within direct range. The MAC layer of a receiving node shall always forward a broadcast MPDU to higher layers.

The MAC layer shall support the configuration of a promiscuous receiver mode, forwarding all MPDUs to higher layers irrespective of the actual destination NodeID.

A.4.2.6.1 Multicast destination fields

A multicast MPDU shall carry a Multicast Control field and a multi-byte multicast bit mask as shown in Figure A.23.

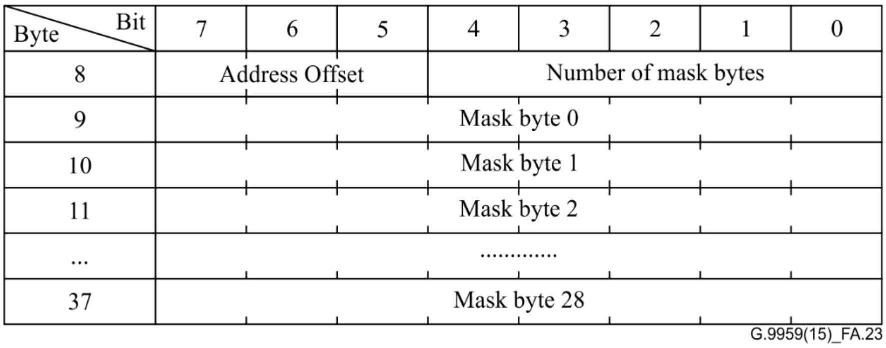

Figure A.23 – Multicast destination address format

A receiving node shall not acknowledge a multicast MPDU. A sending node shall set the ACK Req bit of a multicast MPDU to zero. Likewise, a sending node shall set the Address Offset field to zero and the number of mask bytes field shall be set to 29.

A receiving node shall support the values of the Address Offset field outlined in Table A.45.

The address offset field is a 3-bit field. The encoding of the field is outlined in Table A.45.

Table A.45 – Multicast address offset encoding

Address offset value	Bit address offset
0	0
1	32
2	64
3	96
4	128
5	160
6	192
7	224

Each bit of a mask byte represents a node. Mask byte 0 represents the NodeIDs 1 to 8.

As an example a mask byte 0 value of 0xC5 (in binary: 11000101) covers nodeIDs 1, 3, 7, and 8 while a mask byte 1 value of 0xC5 (in binary: 11000101) covers nodeIDs 9, 11, 15, and 16.

The destination NodeID can be calculated using the following formula.

$$\text{Destination NodeID} = \text{bit address offset} + \text{mask byte number} \times 8 + \text{mask bit number} + 1$$

A.4.2.7 Data payload

The data payload field has a variable length. An ACK MPDU may carry zero or more data payload bytes. A receiving node may derive the length of the data payload field from the MPDU Length field.

A.4.2.8 FCS

An 8-bit non-correcting FCS shall be used for validating the MPDU integrity for data rates R1 and R2. The FCS shall be calculated from the HomeID field to the Data Payload field, both included, as shown in Figure A.24.

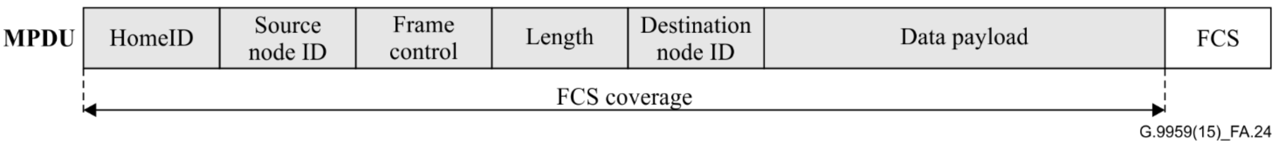

Figure A.24 – FCS coverage

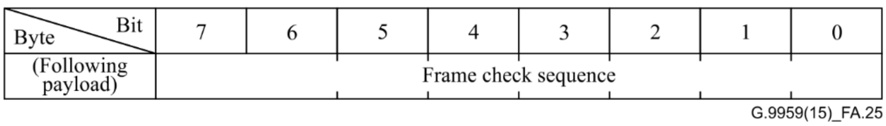

Figure A.25 – FCS field

The FCS shall be calculated as an odd (XOR'ed) checksum as shown in the following algorithm.

```
BYTE GenerateCheckSum(BYTE *Data, BYTE Length)
{
  BYTE CheckSum = 0xFF;
  for (; Length > 0; Length--)
  {
    CheckSum ^= *Data++;
  }
  return CheckSum;
}
```

A.4.2.9 CRC

A 16-bit non-correcting CRC shall be used for validating the MPDU integrity for data rate R3.

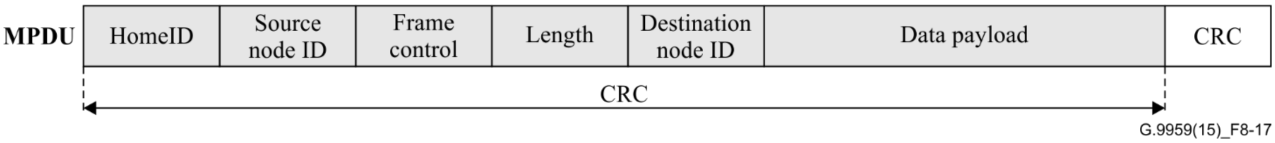

Figure A.26 – CRC coverage

The CRC-16 generator polynomial shall be:

$$P(x) = x^{16} + x^{12} + x^5 + 1, \text{ also known as CRC-CCITT.}$$

The CRC-16 shall be calculated over the whole frame, except for the preamble, SOF, and the CRC-16 fields.

The CRC-16 generator shall be initialized to 1D0Fh before applying the first frame byte of a frame. Additional bits shall not be appended the frame data.

A.4.2.10 Beam frame format

Beam frames may be used to awaken battery powered nodes operating in FL mode. Beam frames are used for several beam types. Beam frames are transmitted back-to-back to ensure an FL node can detect a beam within a very short time window before returning to sleep.

Each beam frame carries a preamble sequence and an SOF field, just like the start of any other PHY PDU. The SOF is followed by two or three bytes, replacing the HomeID field found in a general MPDU. A receiving node may distinguish a general MPDU from a beam frame by inspecting the MS byte of the HomeID. If this byte carries a beam tag (refer to Table A.46) this is a beam frame and the following one or two bytes carry beaming relevant information.

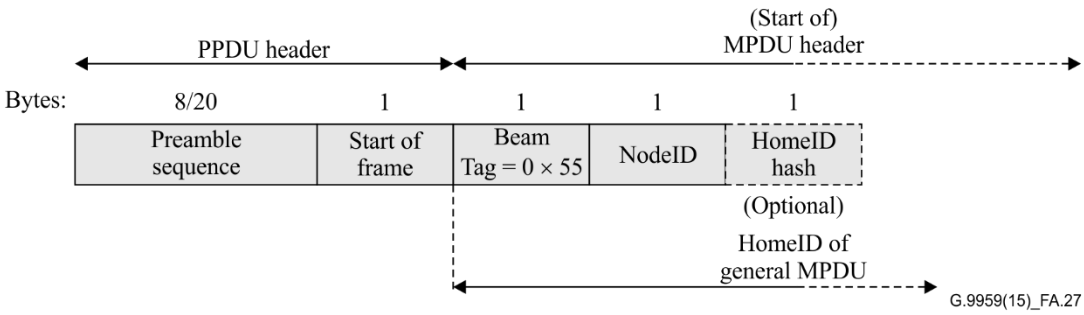

Figure A.27 – Beam frame (Tag = 0x55)

The preamble sequence length depends on the actual RF profile used. A node shall listen long enough to detect a beam frame during worst case conditions around the SOF, beam tag, and beam information. In a worst-case situation, a node in FL mode starts listening just when it is too late to achieve lock to the preamble sequence. In this case, the node has to wait until another preamble sequence starts. Only after achieving lock to the preamble, the SOF and beam tag fields may be correctly decoded.

Each beam frame shall carry the beam tag and NodeID fields. The NodeID field should be followed by the optional HomeID Hash field.

Table A.46 – Beam frame fields

Field name	Value	Description
Beam tag	0x55	HomeID MSB reserved for beam frame transmission HomeIDs in the range 0x55000000..0x55FFFFFF shall not be assigned to any domain The beam tag value 0x55 shall advertise the presence of a NodeID field and an optional HomeID Hash field
	0x54 (reserved)	HomeID MSB reserved for beam frame transmission HomeIDs in the range 0x54000000..0x54FFFFFF shall not be assigned to any domain The beam tag value 0x54 shall be reserved for future use
NodeID	1..232, 255	Destination NodeID. If the value is 255 or it matches the NodeID of an FL node, the node shall also inspect the HomeID Hash field. If no match is found, the FL node should return to sleep.
HomeID Hash (optional)	Hashed 8-bit version of the HomeID Ignore if 0x0A, 0x4A or 0x55.	Hash value for preventing NodeID collisions in dense network environments with overlapping domains. Refer to text following this table.

It is recommended that any node implements support for the HomeID Hash field presented in Figure A.27 and Table A.46.

The HomeID hash value shall be calculated as shown in the following algorithm.

```
BYTE GenerateHomeIdHash(BYTE *HomeId)
{
  BYTE HomeIdHash = 0xFF;
  for (Length = 4; Length > 0; Length--)
  {
    HomeIdHash ^= *HomeId++;
  }
  switch (HomeIdHash)
  {
    Case 0x0A:
    Case 0x4A:
    Case 0x55:
      HomeIdHash++;
  }
  return HomeIdHash;
}
```

A sending node should include a HomeID Hash field in the beam frame (Tag 0x55) to assist FL nodes in filtering out beam frames belonging to other domains.

A FL node receiving one of the HomeID hash values 0x0A, 0x4A, or 0x55 shall accept the value as a potential match to the actual HomeID and therefore it shall stay awake to receive the MPDU that follows if there is a match for the NodeID. Likewise, an FL node shall stay awake to receive the MPDU that follows if it detects a match to the hashed version of its own HomeID and if there is also a match for the actual NodeID. In case of no match for HomeID hash and/or NodeID, the node may return to sleep immediately.

A.4.2.11 Fragmented beam format

A beam fragment shall comprise a number of beam frames. The beam fragment duration shall be in the range 110-115 ms. Beam frames shall be sent back-to-back to ensure that the beam fragment can be detected by a node waking up at any moment during the duration of the beam fragment.

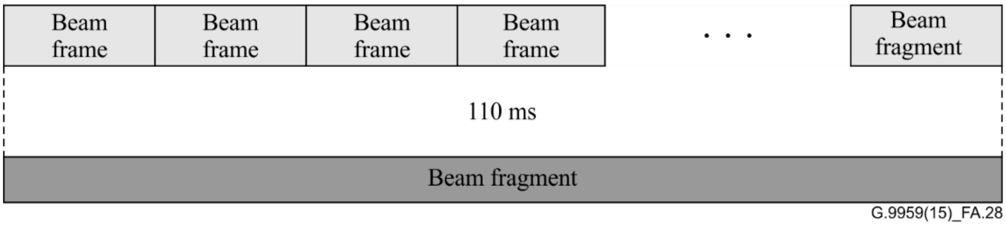

Figure A.28 – Format of the beam fragment

A fragmented beam shall comprise a number of beam fragments. The next beam fragment shall begin in the range 190-200 ms measured from the beginning of the previous beam fragment. Fragmented beams shall be sent at data rate R3. A receiver shall be able to monitor multiple channels for beam frames.

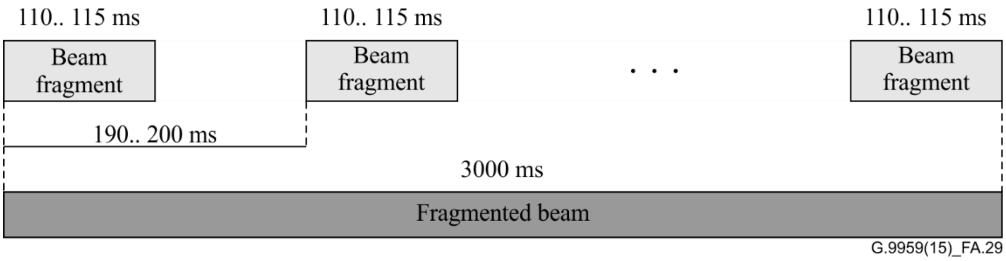

Figure A.29 – Format of the fragmented beam

Proper TX scheduling allows a sending node to reach sleeping nodes that use a range of wakeup intervals 100 ms, 150 ms, 250 ms, 500 ms, 700 ms, 900 ms. The chosen wakeup interval should be a trade-off between battery lifetime and response time.

A fragmented beam may address any NodeID and also the broadcast NodeID 0xFF. A full fragmented beam shall span 3000 ms. A singlecast frame shall follow the fragmented beam. A receiving node may interrupt the transmission of a fragmented beam by acknowledging a singlecast beam fragment. A receiving node shall not acknowledge a broadcast beam fragment. A receiving FL node detecting a positive match for the HomeID Hash field of the beam frame may return the ACK MPDU immediately. A receiving FL node not detecting a positive match for the HomeID Hash field shall apply a random delay of 0..50 ms to the ACK MPDU to prevent collisions between FL nodes having the same NodeID in different domains.

In response to an ACK MPDU, the originating node shall transmit the MPDU to the receiving node if the Src HomeID of the ACK MPDU matches the HomeID of the beaming node. If the HomeID or NodeID does not match, the originating node shall ignore the ACK MPDU.

A.4.2.12 (intentionally left blank)

A.4.2.13 Continuous beam format

A continuous beam is a series of beam frames spanning a fixed period of time. The beam frames shall be sent back-to-back to prevent other TXs from interrupting the continuous beam. The continuous beam shall be sent at data rate R2.

A continuous beam may address any NodeID, and also the broadcast NodeID (0xFF).

There are two types of continuous beams: long and short. A long continuous beam may last for a maximum of 1160 ms, and a short one may last for a maximum of 300 ms. The recommended duration is 1100 ms for a long continuous beam and 275 ms for a short continuous beam. A continuous beam shall always be followed by a singlecast frame.

A.4.2.14 Beam formats on channel configurations 1 and 2

FL nodes operating in channel configuration 1 or 2 shall be able to receive and transmit the continuous beam format at data rate R2.

A.4.2.15 Beam formats on channel configuration 3

FL nodes operating in channel configuration 3 shall be able to receive and transmit the fragmented beam format at data rate 3.

A.4.3 MAC constants and MIB attributes

This clause specifies the constants and attributes used by the MAC.

A.4.3.1 MAC constants

The constants used by the MAC layer are presented in Table A.47. These constants are hardware dependent and cannot be changed during operation.

Table A.47 – MAC constants in general

Constants	Description	Value
aMacMaxMSDUSizeR1	The maximum singlecast/broadcast MSDU size for data rate R1	54 bytes
aMacMaxMSDUSizeMultiR1	The maximum multicast MSDU size for data rate R1	25 bytes
aMacMaxMSDUSizeR2	The maximum singlecast/broadcast MSDU size for data rate R2	54 bytes
aMacMaxMSDUSizeMultiR2	The maximum multicast MSDU size for data rate R2	25 bytes
aMacMaxMSDUSizeR3	The maximum singlecast/broadcast MSDU size for data rate R3	158 bytes
aMacMaxMSDUSizeMultiR3	The maximum multicast MSDU size for data rate R3	129 bytes
aMacMinAckWaitDuration	The minimum ACK wait duration	aPhyTurnaroundTimeRxTx + (aMacTransferAckTimeTX * (1/data rate))
aMacMaxFrameRetries	The number of retries after a transmission failure	2
aMacMinCCARetryDuration	The minimum duration of clear channel access assessment	1100 ms
aMacTxPhyPowerLevelLow	The value used for phyTransmitPower to transmit at low power	Implementation dependent
aMacTxPhyPowerLevelNormal	The value used for phyTransmitPower to transmit at normal power	Implementation dependent

The term *aMacMaxMSDUSizeX* is used in text that applies to all data rates (*aMacMaxMSDUSizeR1, aMacMaxMSDUSizeR2, aMacMaxMSDUSizeR3, aMacMaxMSDUSizeMultiR1, aMacMaxMSDUSizeMultiR2, aMacMaxMSDUSizeMultiR3).*

Retransmitted frames may cause prolonged transmission times as the retransmission uses longer preambles in certain transmission modes.

Table A.48 outlines the constant values to use for message transfer at data rates R1, R2, R3.

Table A.48 – MAC constants for MPDU transfer

Constants	Description	R1	R2	R3	
				Channel Cfg 1, 2	**Channel Cfg 3**
aMacTransferAckTimeTX (ch)	The number of symbols of an ACK MPDU, including preamble	168 bits	248 bits	416 bits	296 bits
aMacTypicalFrameLengthTX (ch)	The number of symbols of a singlecast MPDU with a data payload of 4 bytes	200 bits	280 bits	448 bits	328 bits
aMacMinRetransmitDelay	Random backoff shall be higher than this value	10 ms			
aMacMaxRetransmitDelay	Random backoff shall be lower than this value	40 ms			

The MAC determines if the PHY is configured for channel configurations 1, 2, or channel configuration 3 operation by evaluating the RF profile mappings configured via the *macMapPhyChannel** MAC MIB parameters and the RF profile table defined for the PHY.

A.4.3.2 MAC MIB attributes

The MAC MIB comprises the attributes required to manage the MAC layer. Each of these attributes can be read or written using the MLME-GET.request and MLME-SET.request primitives, respectively. The attributes contained in the MAC MIB are presented in Table A.49.

Table A.49 – MAC MIB attributes

Attribute	Type	Range	Description	Default
macCCARetryDuration	Integer	*aMacMinCCARetryDuration – infinite*	The duration of clear channel access assessment	~ 1100 ms
macHomeID	Byte Array	0x00000000.. 0xFFFFFFFF	The HomeID is the unique domain identifier	(random)
macNodeID	Byte	0x00..0xFF	The NodeID is the address of the individual nodes in a domain	0x00
macPromiscuousMode	Boolean	TRUE or FALSE	This indicates whether the MAC layer is in a promiscuous (receive all) mode. A value of TRUE indicates that the	FALSE

Table A.49 – MAC MIB attributes

Attribute	Type	Range	Description	Default
			MAC layer accepts all frames received from the PHY	
macRxOnWhenIdle	Boolean	TRUE or FALSE	This indicates whether the MAC layer is to enable its receiver at any time	FALSE
macMapPhyChannelA	Integer	Available RF profiles (Table A.1)	Apply RF profile to channel A. Parameter maps directly to the PHY MIB parameter *phyMapChannelA*	–
macMapPhyChannelB	Integer	Available RF profiles (Table A.1)	Apply RF profile to channel B. Parameter maps directly to the PHY MIB parameter *phyMapChannelB*	–
macMapPhyChannelC	Integer	Available RF profiles (Table A.1)	Apply RF profile to channel C. Parameter maps directly to the PHY MIB parameter *phyMapChannelC*	–

A.4.4 MAC functional description

This clause provides a detailed description of the MAC functionality. Throughout this clause, the receipt of an MPDU is defined as the successful receipt of an incoming MPDU from the RF media to the PHY and the successful verification of the FCS by the MAC layer, as described in clause A.4.2.8.

A.4.4.1 Transmission, reception and ACK

This clause describes the fundamental procedures for transmission, reception, and ACK.

A.4.4.1.1 Clear channel access

The MAC layer shall request the channel status from the PHY layer.

A PLME-GET-CCA.request message is used to evaluate the channel. A PLME-GET-CCA.confirm message returns the current channel status.

If the MAC layer finds the channel busy for a period of *macCCARetryDuration* times the transmission has failed. This shall be indicated to the network layer via the MD-DATA.confirm primitive with a status of NO_CCA (see clause A.4.1.1.2).

A.4.4.1.2 Transmission

To avoid RF collisions the MAC layer shall perform a CCA before transmitting. If the channel is found idle, the MPDU may be transmitted. The Src HomeID and Src NodeID field shall identify the sending node and the Dst NodeID shall identify the Dst node.

A.4.4.1.3 Reception and rejection

Each node may choose whether the MAC layer is to enable its receiver during idle periods. During these idle periods, the MAC layer shall still service TRX task requests from the network layer. A TRX task shall be defined as a transmission request, a reception request, or a clear channel access detection. On completion of each TRX task, the MAC layer shall request that the PHY enables or disables its receiver, depending on whether *macRxOnWhenIdle* is set to TRUE or FALSE, respectively.

Due to the broadcast nature of radiocommunications, a node is able to receive and decode transmissions from all nodes that are operating on the same channel(s). The MAC layer shall be able to filter incoming frames and present only the frames that are of interest to the upper layers.

In promiscuous mode, the MAC layer shall pass all MPDUs directly to the network layer. If the MAC layer is not in promiscuous mode (i.e., *macPromiscuousMode* is set to FALSE), it shall only accept MPDUs and issue an MD-DATA.indication to the network layer if the MPDU header contains the HomeID and NodeID of the receiving node. MPDUs shall also be accepted if addressed to the broadcast address or if the NodeID is included in a multicast header.

If the frame type subfield indicates a singlecast frame and the ACK request subfield of the frame control field is set to 1, the MAC layer shall send an ACK frame.

The MAC layer shall be able to receive beam fragments and forward these to higher layers.

A.4.4.1.3.1 RX filtering

An MPDU shall be discarded if the received MPDU has an invalid FCS value.

An MPDU shall be discarded if it has a length field less than 9 or greater than the maximum size values indicated in Tables A.47 and A.48.

A.4.4.1.4 Use of ACK

A singlecast MPDU may be sent with the ACK request subfield of the frame control field set to 1. Any multicast or broadcast frame shall be sent with the ACK request subfield set to 0.

Sequence number checking shall be applied to ACK handling. Refer to clause A.4.2.3.7.

A.4.4.1.4.1 No ACK

An MPDU transmitted with its ACK request subfield set to 0 shall not be acknowledged by its intended recipient. The originating node shall assume that the transmission was successful. The sequence diagram in Figure A.30 shows the scenario for transmitting a single MPDU without requiring an ACK.

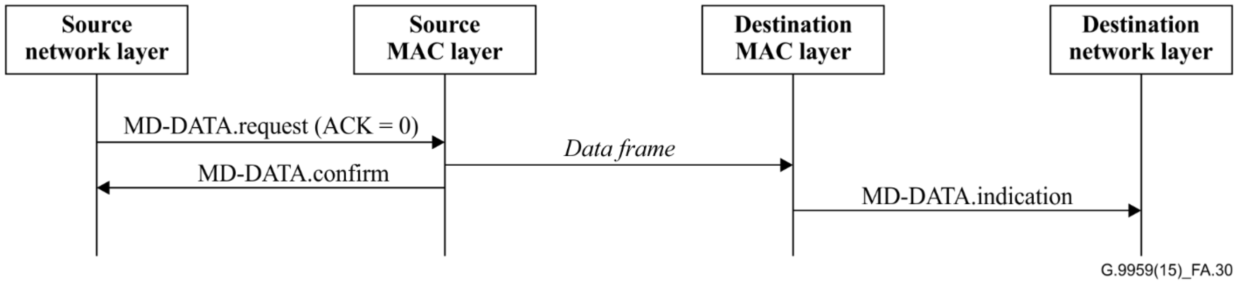

Figure A.30 – Successful transmission an ACK

A.4.4.1.4.2 ACK

A singlecast MPDU transmitted with the ACK request subfield of its MPDU control field set to 1 shall be acknowledged by the recipient. If the intended recipient correctly receives the MPDU, it shall return an ACK MPDU. Only the singlecast MPDU may be sent with the ACK request subfield set to 1. For other frame types the ACK request subfield shall be ignored by the intended recipient.

The transmission of an ACK MPDU shall not commence before *aPhyTurnaroundTimeRXTX* symbols have elapsed after the reception of the last symbol of the frame. Refer to clause A.3.1.14.

The sequence diagram in Figure A.31 shows the transmission of an acknowledged singlecast MPDU.

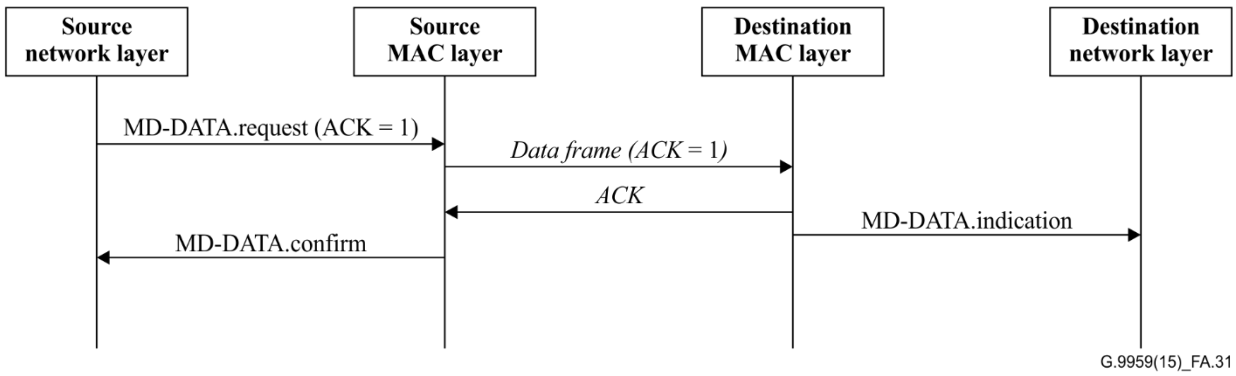

Figure A.31 – Successful transmission with ACK

A.4.4.1.4.3 Retransmission

A node that sends a singlecast MPDU with its ACK request subfield set to 1 shall wait for a minimum of *aMacMinAckWaitDuration* symbols for the corresponding ACK MPDU to be received. If an ACK MPDU is received within *aMacMinAckWaitDuration* symbols and contains the correct HomeID and Src NodeID, the transmission is considered successful, and no further action shall be taken by the originator. If an ACK MPDU is not received within *aMacMinAckWaitDuration* symbols the transmission attempt has failed. The originator shall repeat the process of transmitting the MPDU and waiting for the ACK MPDU up to *aMacMaxFrameRetries* times. Before retransmitting the node shall wait for a random backoff period (see clause A.4.4.1.4.4).

If an ACK MPDU is still not received after *aMacMaxFrameRetries* retransmissions, the MAC layer shall assume the transmission has failed and notify the network layer of the failure. This shall be done via the MD-DATA.confirm primitive with a status of NO_ACK (see clause A.4.1.1.2).

A.4.4.1.4.4 Random backoff

If a singlecast MPDU with its ACK request subfield set to 1 or the corresponding ACK MPDU is lost or corrupted, the singlecast MPDU shall be retransmitted. The MAC layer collision avoidance mechanism prevents nodes from retransmitting at the same time. The random delay shall be calculated as a period in the interval *aMacMinRetransmitDelay… aMacMaxRetransmitDelay;* Refer to Table A.48.

A.4.4.1.5 Idle mode

If the MLME is requested to set *macRxOnWhenIdle* to TRUE the PHY shall enter RX mode and stay in RX mode when a MPDU has been transmitted (AL). This is achieved when the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of RX_ON.

If the MLME is requested to set *macRxOnWhenIdle* to FALSE, the PHY shall disable its receiver when a MPDU has been transmitted. This is achieved by the MLME issuing the PLME-SET-TRX-STATE.request primitive with a state of TRX_OFF.

A.4.4.2 Transmission scenarios

Due to the imperfect nature of the radio medium, a transmitted MPDU does not always reach its intended destination. Figures A.32 to A.34 illustrate three different data transmission scenarios:

– ***Successful transmission***. The originating MAC layer transmits the MPDU to the recipient via the PD service. The originating MAC layer waits for *aMacMinAckWaitDuration* symbols. The Dst MAC layer receives the MPDU, returns an ACK MPDU, and passes the MPDU to the next higher layer. The originating MAC layer receives the ACK MPDU. The data transfer is now complete, and the originating MAC layer issues a success confirmation to the network layer.

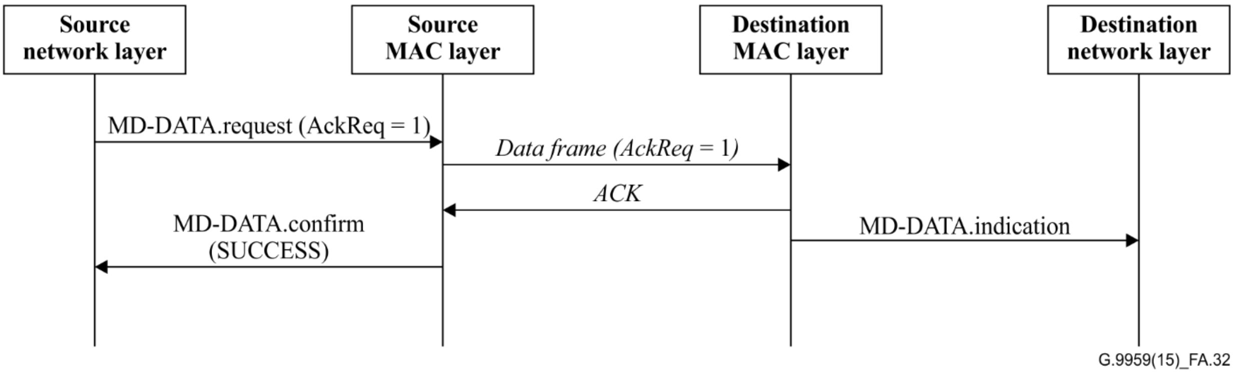

Figure A.32 – Successful transmission scenario

– ***Lost MPDU***. The originating MAC layer transmits the MPDU to the recipient via the PD service. The originator MAC layer waits for *aMacMinAckWaitDuration* symbols. The destination MAC layer does not receive the MPDU, and therefore does not return an ACK MPDU. The timer of the originator MAC layer expires. The transmission has failed and the originator retransmits the MPDU. This sequence is repeated up to *aMacMaxFrameRetries* times. If transmissions fail a total of (1 + *aMacMaxFrameRetries*) times, the originator MAC layer issues a failure confirmation to the network layer.

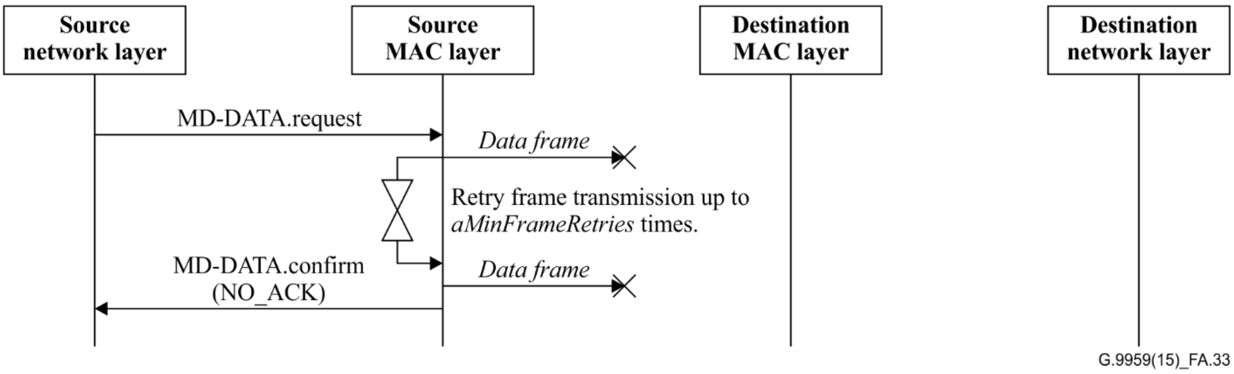

Figure A.33 – Lost MPDU scenario

– ***LostAck MPDU***. The originating MAC layer transmits the MPDU to the recipient via the PD service. The originating MAC layer waits for *aMacMinAckWaitDuration* symbols. The destination MAC layer receives the MPDU, returns an ACK MPDU back to the originator, and passes the MPDU to the network layer. The originating MAC layer does not receive the ACK MPDU and its timer expires. The transmission has failed, and the originator retransmits the MPDU. If transmissions fail a total of (1 + *aMacMaxFrameRetries*) times, the MAC layer issues a failure confirmation to the network layer.

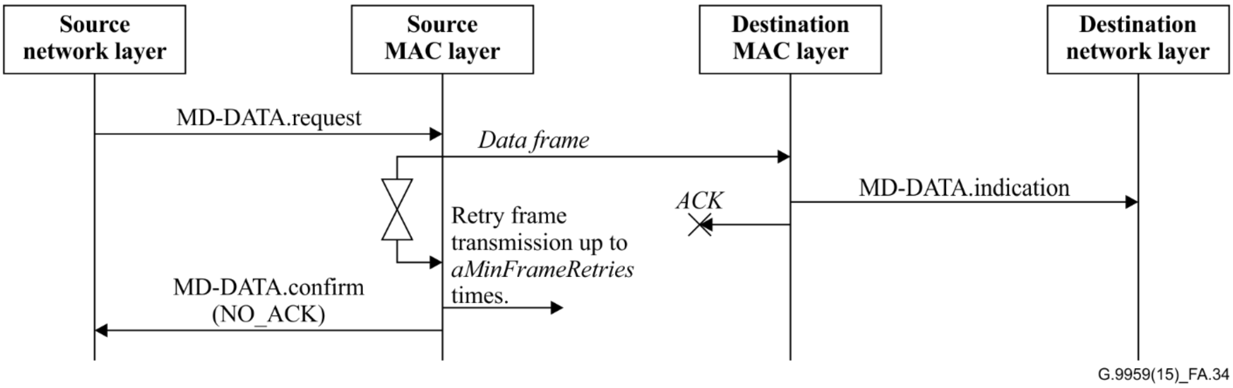

Figure A.34 – Lost ACK MPDU scenario

Bibliography

[b-ITU-T X.210] Recommendation ITU-T X.210 (1993) | ISO/IEC 10731: 1994, *Information technology – Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2015