Z-Wave Long Range PHY and MAC Layer Specification

Release 3.9.0

Z-Wave Alliance

Aug 20, 2025

Table of contents

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

1 Preamble

1.1 Description

Z-Wave Long Range PHY and MAC layer specification

Reviewed by the Z-Wave Alliance Core Stack Working Group (CSWG) and approved by the Z-Wave Alliance Technical Committee.

1.2 Disclaimer

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE ALLIANCE, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

1.3 Revision Record

Table 1.1: Revision History

Doc. Rev	Date	By	Pages Affected	Brief Description of Changes
0.3	2020.09.01	CSWG	ALL	PHY section completed and some MAC sections started
0.5	2020.09.23	CSWG	ALL	PHY and MAC sections ready for 0.5 review
0.7	2020.09.28	CSWG	ALL	Review of 0.5 completed and document moved to 0.7
0.7.1	2020.12.14	CSWG	6.5.1.5.5 Table 6.15 6.5.1.2 5.3.3 6.5.1.5.4 5.3 & 6.3 6.5.1.2.1	Changed retransmit wording to allow reception of ack during random backoff Added NodeID 4001-4004 as valid ID's for virtual nodes Allow acknowledge to be send without CCA Fixed bit order of SOF Added description of retransmission on secondary channel Fixed PHY bit order description Added **shall** about Tx power Control
0.7.2	2021.02.26	CSWG	6.3.6.3	Fixed wrong bit size of destination Node ID in wakeup beam description
0.8	2021.03.01	CSWG	None	Updated to revision 0.8 after initial TC review
0.8.1	2021.03.16	CSWG	5.2.2 5.2.4	Updated with comments from TC review Removed references to US region Changed EVM to Offset EVM
0.9	2021.03.29	CSWG	Frontpage	Cleanup for IPR review
1.0	2021.08.27	ZWA Board		Approved for Publication
1.5	2023.06.20	CSWG	Table 5.5	Corrected symbol mapping.
1.8	2023.07.03	CSWG	Frontpage	Approved by CSWG and TC, updated for members review.
2.0	2023.08.17	ZWA Board		Approved for Publica on
2.5.0	2024.10.21	CSWG	All	Migration from Word to GitHub
2.5.1	2024.10.22	CSWG	5.2.5.10	Add specification for Side-Lobe suppression in Long Range
2.5.2	2024.10.22	CSWG	5.2.5.6	Fix incorrect receiver blocking test description
2.7.0	2025/03/21	CSWG	n/a	Ready for the TC review.
2.9.0	2025/05/30	TC	n/a	Approved for IPR review.
3.0.0	tbd	ZWA Board	n/a	Approved for publication.
3.1.0	2025/07/22	WePower	6.3.1.7 6.5.1.2.1	Added description for noise level value for WOEEN Added description for Dynamic TX Power Algorithm for WOEEN
3.7.0	2025/07/28	CSWG	n/a	Ready for the TC review.
3.9.0	2025/08/20	TC	n/a	Approved by TC

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

1.4 Abbreviations

Table 1.2: Abbreviations

Abbreviation	Explanation
ACK	Acknowledgement
ADP	Application Data Primitives
AE	Application Entity
AIS	Application Interface Sublayer
AL	Always Listening
CC	Channel Configuration
CCA	Clear Channel Assessment
CP	Consumer Premises
CRC	Cyclic Redundancy Check
DSSS	Direct Sequence Spread Spectrum
Dst	Destination
EOF	End of Frame delimiter
EIRP	Effective Isotropic Radiated Power
EMC	Electromagnetic Compatibility
ERP	Effective 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
EHR	End Header
EVM	Error Vector Magnitude
ID	Identification
IDB	Inter-Domain Bridge
ISI	Inter-Symbol Interference
ISM	Industrial, Scientific and Medical
Kcps	(Kilo Chips Per Second)
Ksps	Kilo Symbols Per Second
LBT	Listen Before Talk
LLC	Logical Link Control
LR1	Long range Rate type 1 (100 kbit/s, 800 kcps)
LR2	Long range Rate type 2 (XX kbit/s, YY kcps)
LR3	Long range Rate type 3 (UU kbit/s, ZZ kcps)
LFR	Long range Radio Frequencies
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
MSK	Minimum Shift Keying
NPDU	Network layer Protocol Data Unit
NRZ	Non Return to Zero
OQPSK	Offset Quadrature Phase Shift Key
OSI	Open System Interconnect

continues on next page

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

Table 1.2 – continued from previous page

Abbreviation	Explanation
PD	PHY Data
PD-SAP	PHY Data – Service Access Point
PDU	Protocol Data Unit
PHR	PHY Header
PHY	Physical layer
PLME	Physical Layer Management Entity
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
RSSI	Receive Signal Strength Indication
RX	Receive/Receiver
SAP	Service Access Point
Src	Source
SDU	Service Data Unit
SOF	Start of Frame delimiter
SHR	Start Header
SNR	Signal to Noise Ratio
TRX	Transceiver
TX	Transmit/Transmitter
WOEEN	Wake On Event End Node

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

2 DEFINITIONS

This specification defines the following terms:

Alien domain: Any group of nodes that is not compliant with this recommendation or G.9959 [G9959] connected to the same or a different medium (wired or wireless). These domains can be used as backbones to this network domain 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.

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.

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.

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.

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.

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.

Domain: A collection of nodes compliant with this recommendation comprising the domain master and all those nodes that are registered with the same domain master. In the context of this Recommendation,

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

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

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

HomeID: Information unit used as a domain ID in G.9959 [G9959].

Inclusion: The process 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.

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

ISM band: Frequency bands for industrial, scientific and medical use, allocated by the Z-Wave alliance.

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.

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.

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.

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.

Network: One or more, potentially overlapping, domains.

Node: Any device that contains an transceiver in compliance with this recomandation. . The term 'alien node' means a a device with a transiever not compliant with this recomendation.

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

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.

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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.

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

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.

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).

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

3 INTRODUCTION

3.1 Purpose

The purpose of this document is to describe the PHY and MAC layer of the Z-Wave long range protocol

3.2 Audience and Prerequisites

The audience for this document is Z-Wave alliance members

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

4 Z-WAVE LONG RANGE PROTOCOL STACK OVERVIEW AND REFERENCE MODEL

4.1 Generic Description

Z-Wave Long Range provides an extended range version of the Z-Wave technology, targeting deployments over a kilometer radius, suitable in both indoors and outdoors areas.

4.2 Basic Principles of Z-Wave Long Range Networking

The following are the basic principles of the Z-Wave Long Range 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 4000 nodes (including the domain master).
   - Each node in the domain is identified by a NodeID that is unique inside the domain. A NodeID is a 12-bit short address. The first node in a Z-wave Long Range network hands out the HomeID and unique NodeIDs to all other nodes added to 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 can only communicate with the domain master.
3. Nodes of different Z-Wave Long Range domains:
   - Nodes in different domains can communicate via inter-domain bridges (IDB). The IDB function is a bridging function associated with the domain master 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 specification. In addition, inter-domain bridges communications are also beyond the scope of this specification.

The main scope of this specification is limited to the PHY and MAC of Z-Wave Long Range radio communication transceivers.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

4.3 Z-Wave Long Range protocol stack overview

Similar to Z-Wave protocol, the Z-Wave Long Range protocol stack is defined in terms of layers. Each layer is responsible for one part of the operation and offers services to the higher layers through two service access points (SAP), knows as a data service entity and a management service entity. The 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 Long Range node implements the PHY layer, which contains the RF TRX along with its low-level control mechanism, a data link layer that provides access to the physical channel for all types of transfers, a network layer that controls a network formation and maintenance , and a combined application layer that collapses the OSI stack layers transport, session, and presentation.

Figure 4.1 shows the Z-Wave Long Range protocol stack overview

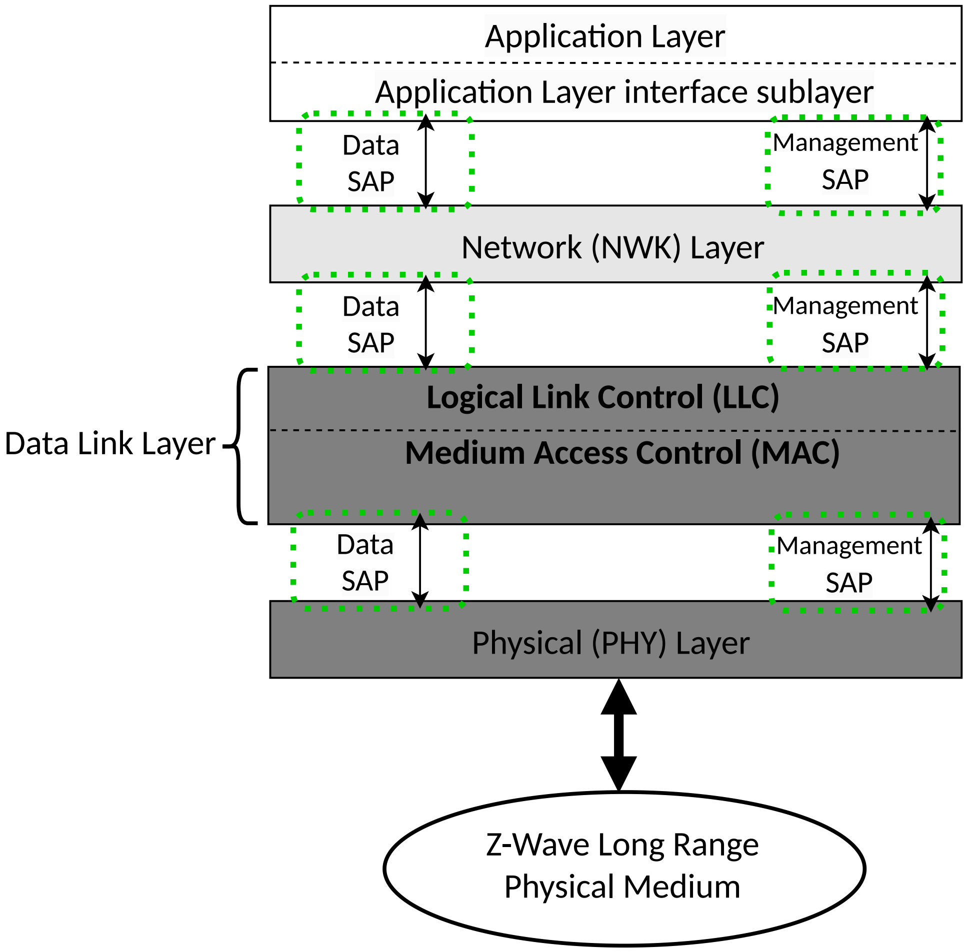

Figure 4.1: Z-Wave Long Range protocol stack layers

This specification document defines only the PHY and Data Link layers. Upper layers are outside the scope of this specification.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

4.3.1 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 can 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:

- the physical layer data service accessed through the PD-SAP; and
- 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.

Section 5 contains the full specification of the Z-Wave Long Range PHY layer.

4.3.2 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.

Section 6 contains the full specification of the Z-Wave Long Range MAC layer.

4.3.3 Logical Link layer (LLC)

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

4.4 Z-Wave Long Range TRX Reference models

4.4.1 Protocol reference model of a transceiver

The protocol reference model of a transceiver is presented in Figure 4.2. 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 (Section 4.4.3).

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.

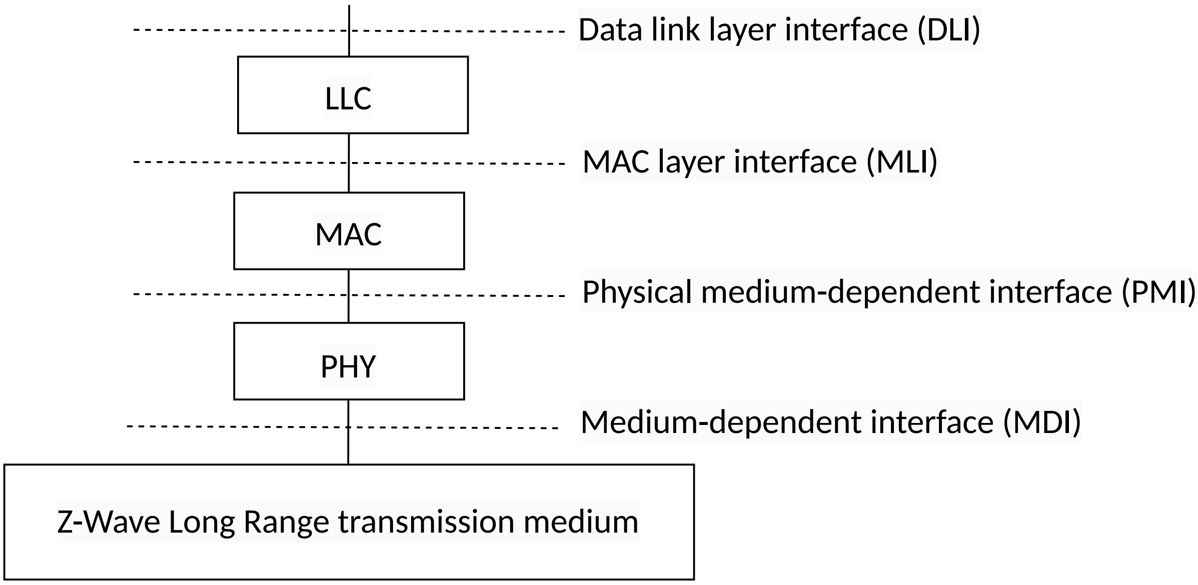

Figure 4.2: Protocol reference model of Z-Wave Long Range TRX

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

The medium access control layer (MAC) 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

4.4.2 Functional description of the interface

This section contains the functional description of the TRX interfaces (reference points) based on the protocol reference model presented in Figure 4.2. The interfaces shown in Figure 4.3 are defined in this specification.

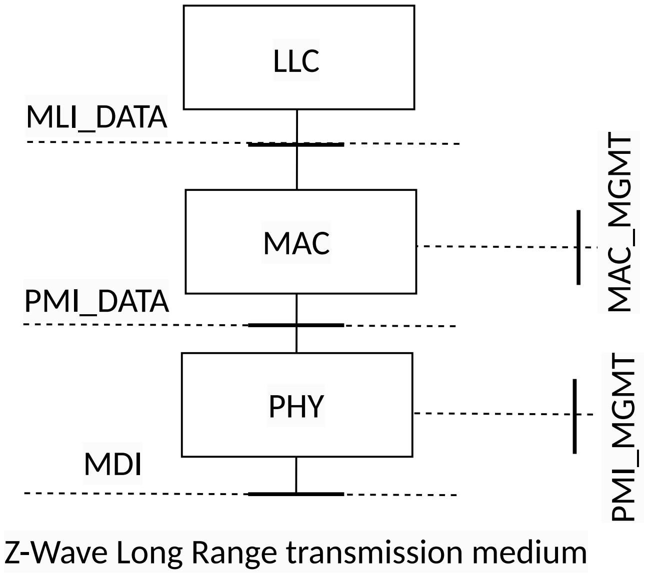

Figure 4.3: Z-Wave Long Range TRX reference points related to PHY/MAC

The reference model in Figure 4.3 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.

4.4.3 Functional model of a transceiver

The functional model of a Z-Wave Long Range TRX is presented in Figure 4.4.

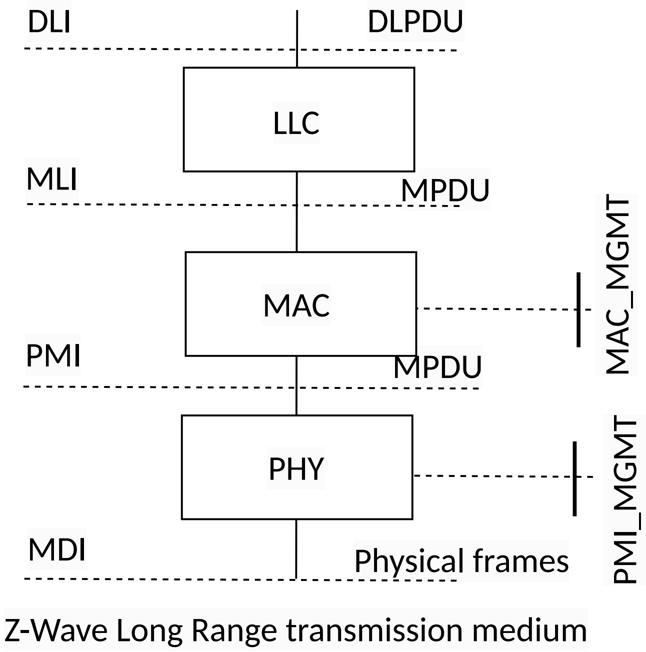

Figure 4.4: Functional model of a Z-Wave Long Range TRX

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

The detail description of the PHY later is presented in Section 5. The detail description of the MAC layer is presented in Section 6. The MAC layer interface (MLI) may deviate 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 the upper layers to perform security encapsulation, segmentation or IP header compression operation, based on the information carried out in MPDU header. The detail of the LLC layer is out of scope for this recomnendation.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

4.5 Operation modes

Similar to the Z-Wave nodes, the Z-Wave Long Range nodes may operate in two different receiving modes: always listening (AL) and frequently listening (FL). The long-range 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

4.6 Concept of service primitives

This clause provides a brief overview of the concept of service primitives (operations) that is applied to describe the Z-Wave Long Range protocol stack layers interaction. Refer to [b ITU T X.210] for more detailed information about the concept of service primitive. The services of a layer are the capabilities it offers to the user in the next higher layer or sublayer by building its functions on the services of the next lower layer. This concept is illustrated in Figure 4.5, showing the service hierarchy and the relationship of the two correspondent N-users and their associated N-layer (or sublayer) peer protocol entities.

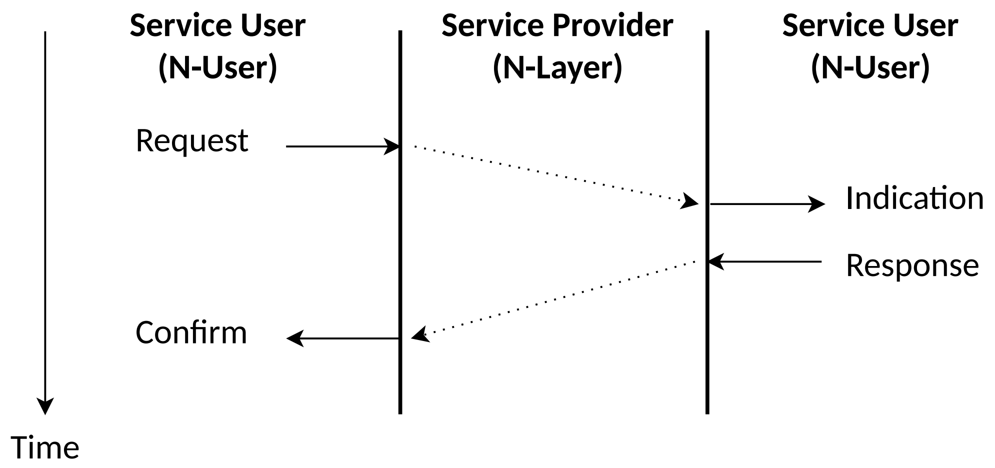

Figure 4.5: Service primitives

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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5 Z-WAVE LONG RANGE PHY SPECIFICATION

5.1 General

The PHY layer defines modulation schemes, data rates, synchronization methods and a frame format for use in high-power, wide-bandwidth control networks. If a device claims to adhere to the following PHY specification, it must also support all the requirements stated in Section 7 of [G9959].

5.1.1 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 transceiver,
- transmission and reception,
- clear channel assessment,
- frequency selection, and
- link quality assessment for received frames

The RF transceiver **shall** be able to operate in a one, two, three, four or five channel configuration in license-free RF bands (the RF channels as defined in Section 7 of [G9959] and what is defined in Section 5.2).

The PHY **shall** provide two services: (1) the PHY data service accessed via the PHY data (PD) service access point (PD-SAP) and (2) the PHY management service accessed via the physical layer management entity (PLME) service access point (PLME-SAP). The PHY data service enables the transmission and reception of PHY protocol data units (PPDUs) over the physical radio channel. See Section 5.4 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 5.27. Attributes have a general prefix of “phy”, e.g., *phyCurrentTxChannel*, and are listed in Table 5.28.

5.1.2 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 Section 5.3.

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. The SHR and PSDU together form the PHY protocol data unit (PPDU).

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

5.2 Transceiver front-end specifications

5.2.1 LRF profiles

An LRF (Z-Wave Long Range Radio Frequency) 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 LRF profiles is specified in Table 5.1. A transceiver **shall** support up to 5 radio channels, each characterized by an RF profile ([G9959] Section 7) and an LRF profile. Each channel **shall** have a unique RF or LRF profile assigned to it. Depending on the actual region, LRF Profiles may allow communication at one or more data rates and one or more channels.

Table 5.1: LRF profiles

LRF Profile	Region	Channel Configurations	Centre frequency	LR1
0	n/a		n/a	
1	Region	1, 2, 3	F_{LR1}	√
2			F_{LR2}	√

5.2.2 Data rates

The PHY **shall** comply with the data rate and accuracy requirements listed in Table 5.2.

Table 5.2: Data rate, symbol rate and accuracy

Data rate	Bit rate	Symbol rate	Accuracy
LR1	100 kbit/s	25 ksps	± 27 ppm

5.2.3 Channel configurations

A compliant node **shall** operate in one of the channel configurations listed in Table 5.3.

Table 5.3: Channel configurations

Channel Configuration	Number of channels	Data rate LR1
1	1	Ch A
2	1	Ch B
3	2	Ch A / Ch B

Table 5.1 and Table 5.3 **shall** be used in combination.

Channel configuration 1 and 2 are valid for controlling devices, channel configuration 3 is for non- controlling devices.

Example 1:

A controlling node may use configuration 2 which provide one communication channel. The following LRF profile is available:

- LRF profile 2 at the frequency f_{LR2} (Ch B) supporting data rate LR1.

Example 2:

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

A non-controlling node must use configuration 3 which provides two alternative communication channels. The following LRF profiles are available:

- LRF profile 1 at the frequency f_{LR1} (Ch A) supporting data rate LR1.
- LRF profile 2 at the frequency f_{LR2} (Ch B) supporting data rate LR1.

5.2.4 Modulation and encoding

The Z-Wave Long Range PHY **shall** employ a 16-ary quasi-orthogonal modulation type using offset quadrature phase shift keying (O-QPSK) with direct sequence spread spectrum (DSSS) for RF modulation at the data rate LR1, Table 5.2.

The modulator aggregates data bits into symbols, Table 5.5, and the symbols are mapped to a sequence of chips to be transmitted, Table 5.6.

The chip sequence is modulated onto the LRF carrier using O-QPSK with a half-sine pulse shaping. Phase modulation must be used to generate the modulated signal. If IQ modulation is used, the even indexed chip sequences must be modulated onto the I carrier (in-phase carrier) and odd-indexed chip sequences must be modulated onto the Q carrier (quadrature-phase carrier), the IQ modulation principle is shown in Figure 5.1. The modulation can also be obtained using MSK modulation techniques.

The coding principle to be used can be illustrated as shown Figure 5.1

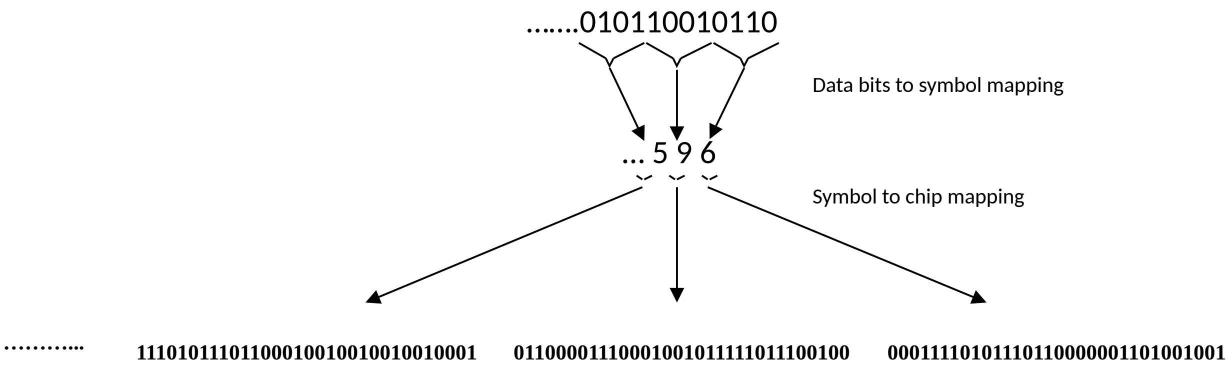

Figure 5.1: Coding principle of LRF profiles

The modulation and the coding format are summarized in Table 5.4

Table 5.4: Modulation and coding format

Data rate	Modulation	Coding	Chip rate
LR1	O-QPSK	DSS	800 kcps

The mapping of data bits to symbols is given in Table 5.5:

Table 5.5: Data bit to symbol mapping

Data bits				**Symbols**
d_3	d_2	d_1	d_0	**Symbol Number**
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

The mapping of symbols to chip sequences is given in Table 5.6:

Table 5.6: Symbol to chip sequence mapping

		Chip sequences (least significant chip to be transmitted first) Chip #																															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
S	0	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1
y	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	0	0	1
m	2	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1
b	3	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1
o	4	1	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1	0
l	5	1	1	1	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1
	6	0	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1
S	7	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0
y	8	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0
m	9	0	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0
b	10	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	0
o	11	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0
l	12	1	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1
#	13	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	0	0
	14	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0
	15	1	1	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	0	1	1	0	0	0	0	1

The O-QPSK modulation to be used can be illustrated as shown below in Figure 5.2:

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

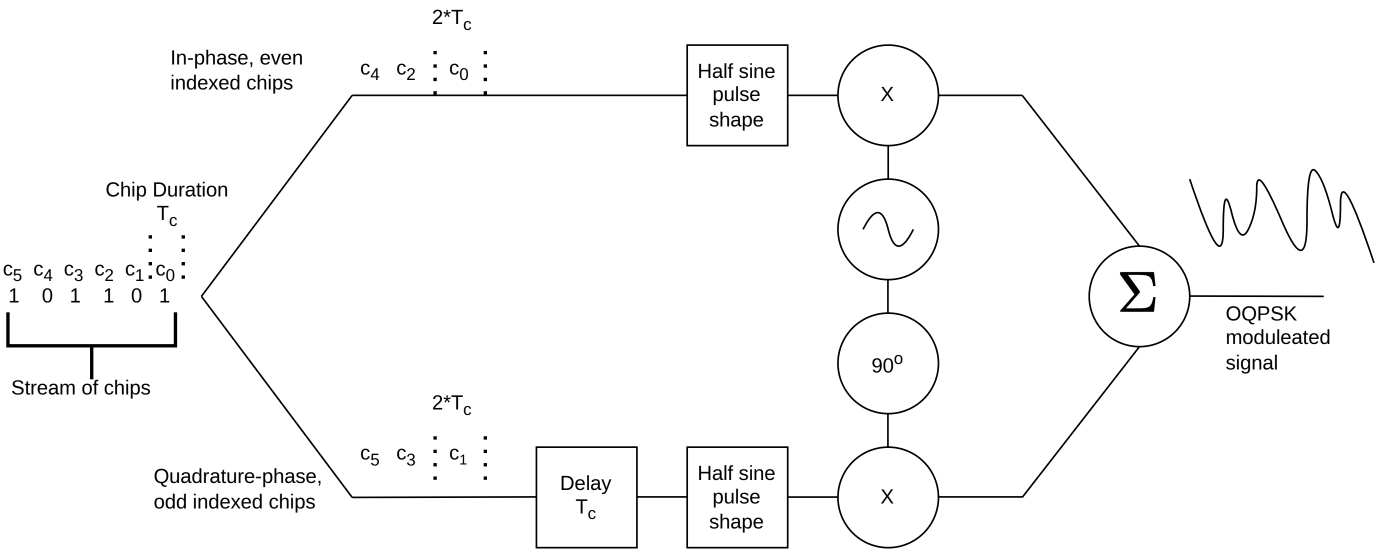

Figure 5.2: O-QPSK modulation principle

The half sine pulse shaping is defined as follows:

$$p(t) = \sin(\pi * \frac{t}{2 * T_c}), 0 \leq t \leq 2 * T_c$$
$$p(t) = 0, t \leq 0 \; or \; t \geq 2 * T_c$$

T_c is identical to (1 / “chip rate”). The chip rate is given in Table 5.4. The accuracy of T_c is identical to the accuracy of the bit rates and symbol rates stated in Table 5.2.

The precision of the O-QPSK modulation must be to such a degree, that when measuring the Error Vector Magnitude (Offset EVM) for 1000 transmitted symbols / chips, the Offset EVM value must be below 20%.

5.2.5 Transmitter and receiver requirements

Unless stated otherwise, all LRF 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.

5.2.5.1 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

5.2.5.2 Transmit power adjustments (conducted)

The transmit power of the transmitter **shall** be adjustable between a minimum power and a maximum power in steps of 2 dB or less. The minimum power **shall** be -10 dBm or less and the maximum power **shall** be 10 dBm or more.

5.2.5.3 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 5.7.

The minimum receiver sensitivity for each data rate **shall** be as specified in Table 5.8.

Table 5.7: Test conditions

Term	Definition	Conditions
Standard Test Frame	PHY frame used for testing sensitivity	PHY frame with at least ten 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 5.8: Minimum receiver sensitivity

Rate	Minimum receiver sensitivity
LR1	–102 dBm

5.2.5.4 Clear channel assessment

The PHY **shall** be able to perform a clear channel assessment (CCA) with a threshold of –60 dBm or a threshold that complies with local RF regulations. 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.

5.2.5.5 Receiver spurious requirement

A transceiver **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 5.3. The measurement bandwidth **shall** be 100 kHz.

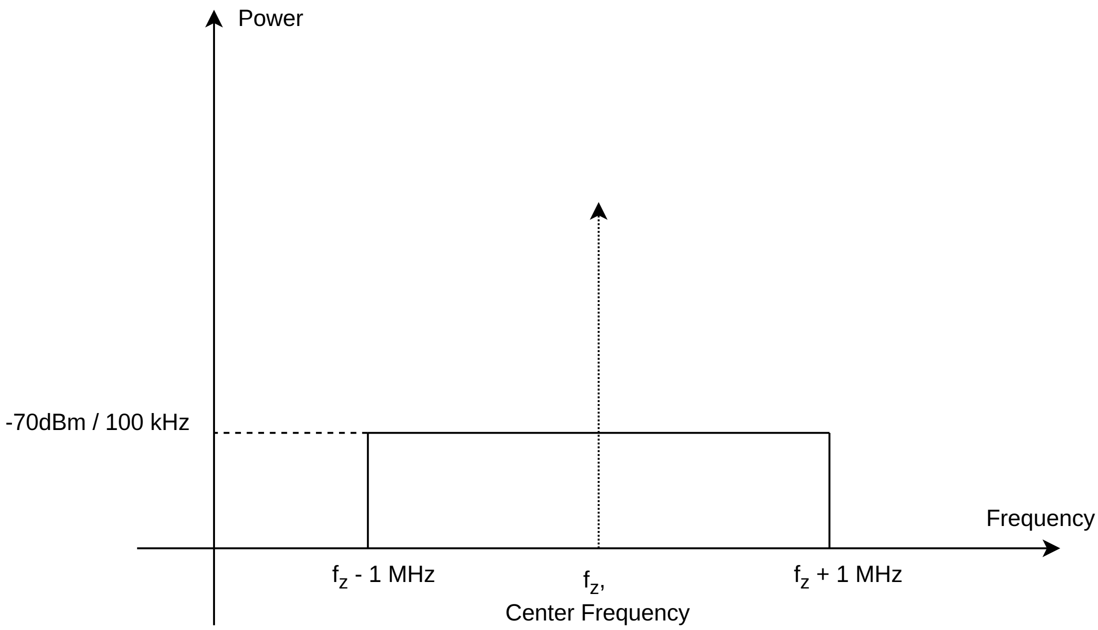

Figure 5.3: Receiver spurious limit

5.2.5.6 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.

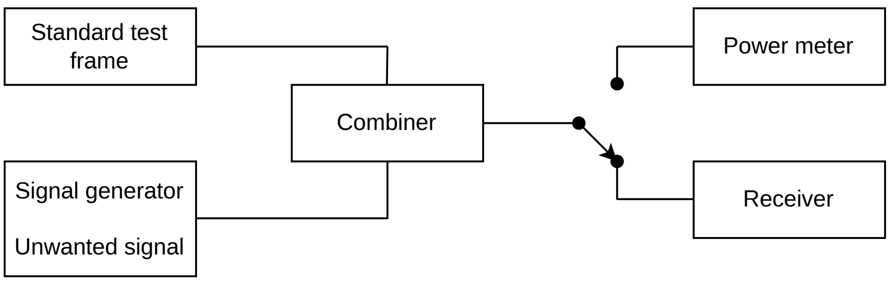

Figure 5.4: Receiver blocking test definition

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 of -89 dBm.
- The blocking test signal **shall** be a carrier transmitted at a specific offset frequency as defined in Table 5.9. The blocking test signal power **shall** be increased until the receiver experiences a FER that corresponds to the sensitivity level.

Limits:

Table 5.9: Blocking limits

Frequency offset	Limits
± 2 MHz	-40 dBm
± 5 MHz	-35 dBm
± 10 MHz	-30 dBm

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

5.2.5.7 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 5.7 while receiving at an input power level greater than or equal to 0 dBm to sustain "zero" distance between two or more devices.

5.2.5.8 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 5.27).

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

5.2.5.9 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 5.27).

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

5.2.5.10 Side-lobe suppression

Suppression of side-lobes in the DSSS O-QPSK modulated signal of a transmitter prevents false preamble detection in the demodulator of a nearby receiver.

Side-lobe suppression applies to all LRF profiles specified in Table 5.1.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

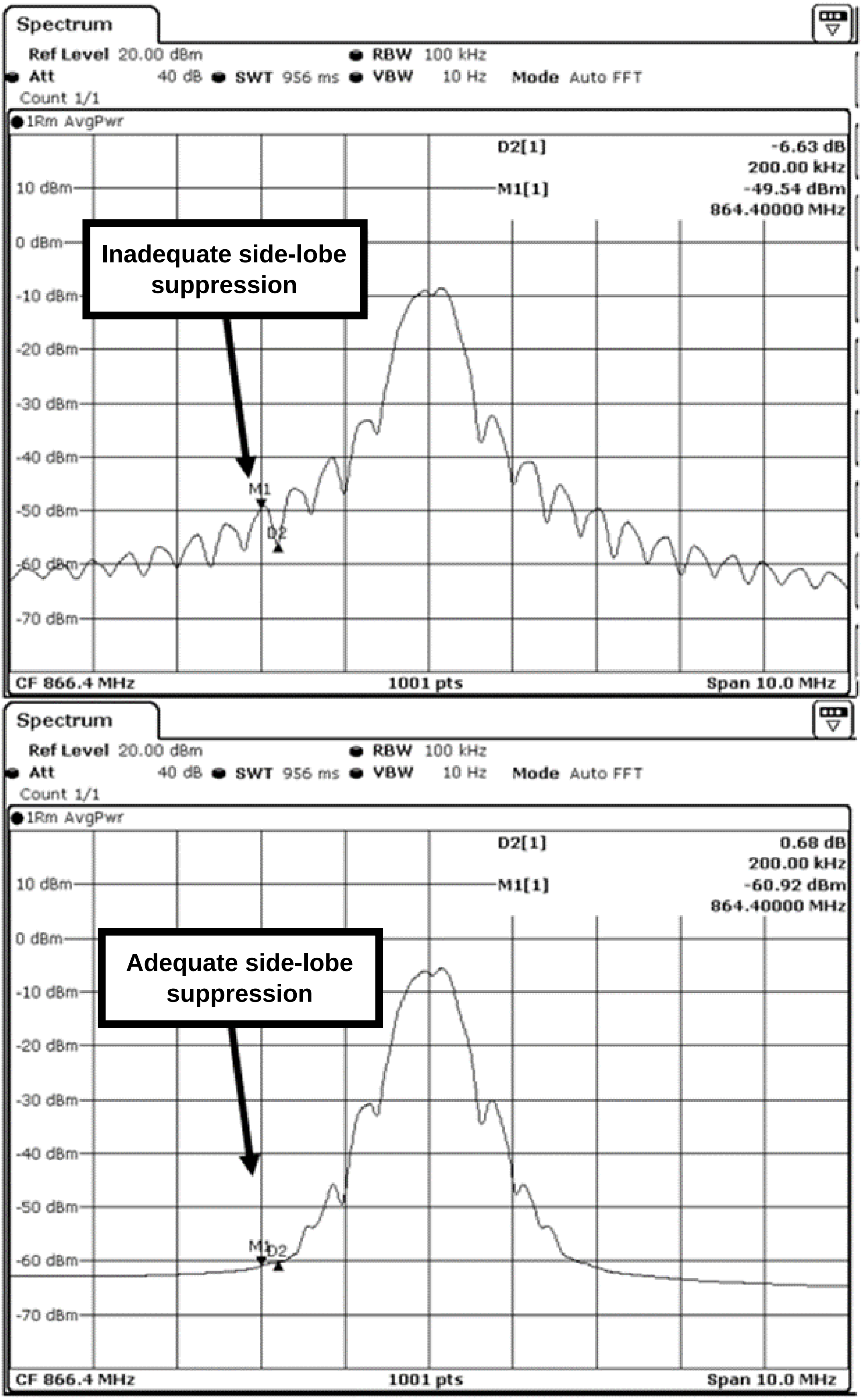

Figure 5.5: Side-lobe Suppression

Method of measurement:

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

– Enable transmission of random data encoded according to Table 5.5 and Table 5.6 and modulated according to Figure 5.2.

– Measure the power difference in dB (P_{Diff1} and P_{Diff2}) with a spectrum analyzer between the sets of frequency locations:

$P_{Diff1} = P_{FC\text{-}2MHz*x} - P_{FC\text{-}(2MHz*x\text{-}200kHz)}$

$P_{Diff2} = P_{FC+2MHz*x} - P_{FC+(2MHz*x\text{-}200kHz)}$

FC = Centre Frequency of LRF.

For x = 1 and x=4.

P_{Diff1} and P_{Diff2} must be <= 1dB.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5.3 PPDU format

Bit to symbol conversion: The general PPDU frame structure is outlined in Figure 5.6. The frame format is depicted in the order in which it is converted to symbols by the PHY. All binary data in the PPDU **shall** be encoded using the encoding described in Section 5.2.4. The binary data in the PPDU **shall** be converted like this. The 4 LSBs (b3, b2, b1, b0) of each octet in the PPDU **shall** map into one data symbol, and the 4 MSBs (b7, b6, b5, b4) of each octet in the PPDU **shall** map into the next data symbol. Each octet of the PPDU is handled as described in Section 5.2.4, beginning with the Preamble field and ending with the last data octet of the PSDU. Within each octet, the least significant symbol (b3, b2, b1, b0) is handled first and the most significant symbol (b7, b6, b5, b4) is handled second.

Bit and byte order in PPDU: 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).

5.3.1 General PHY frame format

The PPDU **shall** be formatted as illustrated in Figure 5.6.

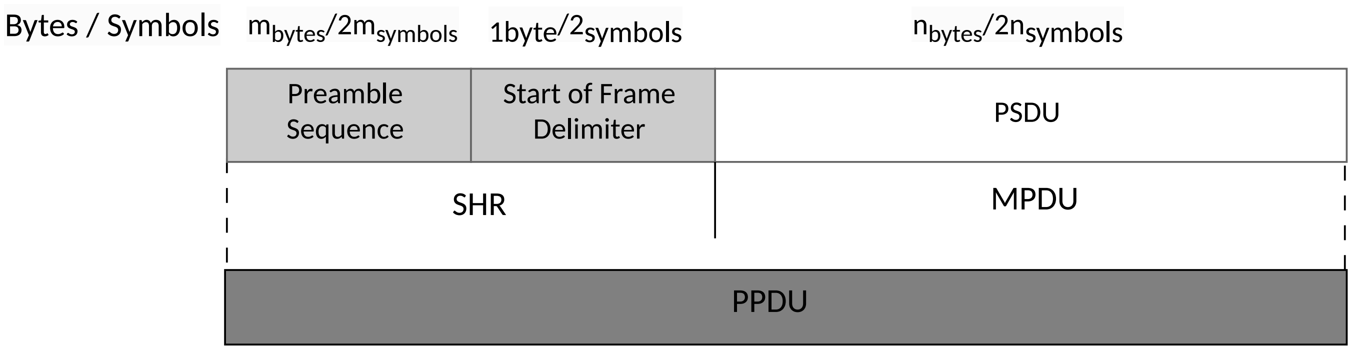

Figure 5.6: PPDU format

5.3.2 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 "00000000". According to Table 5.5 this equivalates to the transmission of two Symbol#0 for each preamble byte to transmit. Figure 5.7 shows the logical bit waveform of the encoded preamble pattern for the data rate LR1.

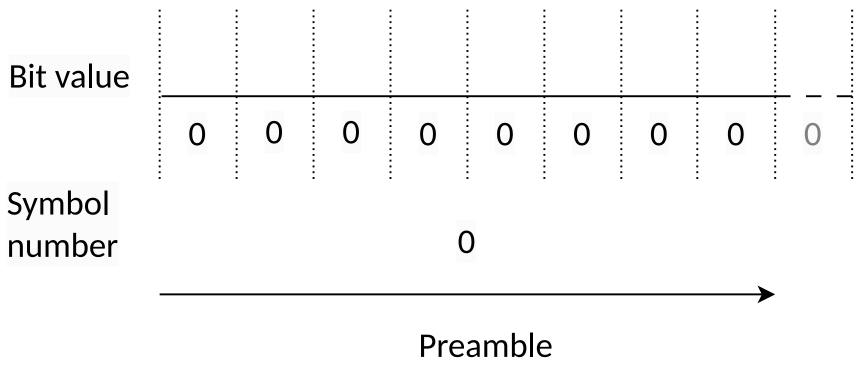

Figure 5.7: Preamble pattern encoded to symbols

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

The preamble length **shall** comply with Table 5.10. The values allow a receiver to scan all channels and obtain synchronization at an arbitrary channel during the duration of the preamble (the RF channels as defined in [G.9959] Section 7 and what is defined in this recommendation).

Table 5.10: Required Preamble length

Channel Configuration	Rate	Required Preamble length in bytes/symbols		
		Singlecast/ broadcast	Multicast	Beam
1 or 2 or 3	LR1	40 bytes/ 80 symbols	40 bytes/ 80 symbols	8 bytes/ 16 symbols

5.3.3 Start of frame field

The start of frame (SOF) is an 8-bit field terminating the preamble field and the start of the PSDU. The SOF **shall** be formatted as the logical binary pattern "01011110" as illustrated in Table 5.11. According to Table 5.10, the SOF will be transmitted as two symbols.

Table 5.11: Format of the SOF field

Bit #	7	6	5	4	3	2	1	0
Bit Value	0	1	0	1	1	1	1	0
Symbol #	5				14			

5.3.4 PSDU field

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5.4 PHY service specifications

The PHY layer **shall** provide two services, accessed via two service access points (SAPs): the PHY data, 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 5.8 shows the components and interfaces of the PHY.

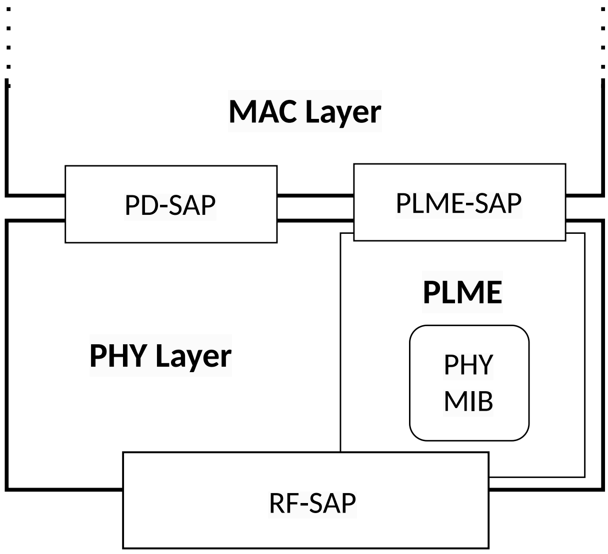

Figure 5.8: Phy reference model

5.4.1 PHY data service

The PD-SAP supports the transport of MPDUs between peer MAC entities. Table 5.12 lists the primitives supported by the PD-SAP.

Table 5.12: PD-SAP primitives

PD-SAP primitive	Request	Confirm	Indication
PD-DATA	Section 5.4.1.1	Section 5.4.1.2	Section 5.4.1.3

5.4.1.1 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.

5.4.1.1.1 Semantics of the PHY data request primitive

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

PD-DATA.request (

psduChannel,

psduRate,

psduLength,

psdu

)

Table 5.13 specifies the parameters for the PD-DATA.request primitive.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

Table 5.13: PD-DATA.request parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B according to the applied LRF profile (see Table 5.1 and Table 5.3)	Channel to use
psduRate	Enumeration	Data rate LR1 according to the applied LRF profile (see Table 5.1 and Table 5.3)	Data rate to use
psduLength	Byte	Less than or equal to *aPhyMaxFrameSizeRx*, where LRx is LR1	Number of bytes to transmit
psdu	Byte Array	-	PSDU to transmit

5.4.1.1.2 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.

5.4.1.1.3 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 transmitter 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 transceiver 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.

5.4.1.2 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.

5.4.1.2.1 Semantics of the PHY data confirm primitive

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

PD-DATA.confirm (

status

)

Table 5.14 specifies the parameters for the PD-DATA.confirm primitive.

Table 5.14: PD-DATA.confirm parameters

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5.4.1.2.2 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.

5.4.1.2.3 Effects on receipt

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

5.4.1.3 PD-DATA.indication

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

5.4.1.3.1 Semantics of the PHY data indication primitive

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

PD-DATA.indication (

psduByte

)

Table 5.15 specifies the parameters for the PD-DATA.indication primitive.

Table 5.15: PD-DATA.indication parameters

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

5.4.1.3.2 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5.4.1.3.3 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 FCS before issuing an MD-DATA.indication to higher layers.

5.4.2 PHY management service

The PLME-SAP allows the transport of management commands between the MLME and the PLME. Table 5.16 lists the primitives supported by the PLME-SAP.

Table 5.16: PLME-SAP primitives

PLME-SAP Primitive	Request	Confirm	Indication	Response
PLME-SOF	-	-	Section 5.4.2.1	-
PLME-GET-CCA	Section 5.4.2.2	Section 5.4.2.3	-	-
PLME-GET	Section 5.4.2.4	Section 5.4.2.5	-	-
PLME-SET-TRX-MODE	Section 5.4.2.6	Section 5.4.2.7	-	-
PLME-SET	Section 5.4.2.8	Section 5.4.2.9	-	-

5.4.2.1 PLME-SOF.indication

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

5.4.2.1.1 Semantics for the service primitive

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

PLME-SOF.indication (

psduChannel,

psduRate

)

Table 5.17 specifies the parameters for the PD-DATA.indication primitive.

Table 5.17: PLME-SOF.indication parameters

Name	Type	Valid range	Description
psduChannel	Enumeration	Channel A, B	Channel from which the PSDU was received
psduRate	Enumeration	Data rate LR1	Data rate of the received PSDU

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

5.4.2.1.2 When generated

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

5.4.2.1.3 Effect on receipt

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

5.4.2.2 PLME-GET-CCA.request

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

5.4.2.2.1 Semantics for the service primitive

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

PLME-GET-CCA.request (

channel

)

Table 5.18 specifies the parameters for the PLME-GET-CCA.request primitive.

Table 5.18: PLME-GET-CCA.request parameters

Name	Type	Valid range	Description
Channel	Enumeration	Channel A or B according to the applied LRF profile (see Table 5.1 and Table 5.3)	The physical channel on which the CCA is to be performed

5.4.2.2.2 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.

5.4.2.2.3 Effect on receipt

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

5.4.2.3 PLME-GET-CCA.confirm

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

5.4.2.3.1 Semantics for the service primitive

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

PLME-GET-CCA.confirm (

status

)

Table 5.19 specifies the parameters for the PLME-GET-CCA.confirm primitive.

Table 5.19: 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

5.4.2.3.2 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 transceiver is not in RX mode (and thus, unable to perform a CCA).

5.4.2.3.3 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.

5.4.2.4 PLME-GET.request

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

5.4.2.4.1 Semantics for the service primitive

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

PLME-GET.request (

PhyMibAttribute

)

Table 5.20 specifies the parameters for the PLME-GET.request primitive.

Table 5.20: PLME-GET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See Table 5.28	Identifier of the PHY MIB attribute

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

5.4.2.4.2 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.

5.4.2.4.3 Effect on receipt

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

5.4.2.5 PLME-GET.confirm

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

5.4.2.5.1 Semantics for the service primitive

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

PLME-GET.confirm (

status,

PhyMibAttribute,

PhyMibAttributeValue

)

Table 5.21 specifies the parameters for the PLME-GET.confirm primitive.

Table 5.21: 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 5.28	Identifier of the PHY MIB attribute
PhyMibAttribute Value	Various	Attribute specific	Value of the specified MIB attribute

5.4.2.5.2 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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.

5.4.2.5.3 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.

5.4.2.6 PLME-SET-TRX-MODE.request

The PLME-SET-TRX-MODE.request primitive requests that the PHY entity changes the operating mode of the transceiver. The transceiver may be set to one of the modes outlined in Table 5.22.

5.4.2.6.1 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 5.22 specifies the parameters for the PLME-SET-TRX-MODE.request primitive.

Table 5.22: PLME-SET-TRX_MODE.request parameters

Name	Type	Valid range	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

5.4.2.6.2 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 transceiver.

5.4.2.6.3 Effect on receipt

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

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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

5.4.2.7 PLME-SET-TRX-MODE.confirm

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

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

If the transceiver 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 transceiver 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 transceiver 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.

5.4.2.7.1 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 5.23 specifies the parameters for the PLME-SET-TRX-MODE.confirm primitive.

Table 5.23: PLME-SET-TRX-MODE.confirm parameters

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

5.4.2.7.2 When generated

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

5.4.2.7.3 Effect on receipt

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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.

5.4.2.8 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.

5.4.2.8.1 Semantics for the service primitive

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

PLME-SET.request (

PhyMibAttribute,

PhyMibAttributeValue

)

Table 5.24 specifies the parameters for the PLME-SET.request primitive.

Table 5.24: PLME_SET.request parameters

Name	Type	Valid range	Description
PhyMibAttribute	Enumeration	See Section 5.4.2.1	Identifier of the PHY MIB attribute
PhyMibAttributeValue	Various	Attribute specific	Value of the PHY MIB attribute

5.4.2.8.2 When generated

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

5.4.2.8.3 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

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

5.4.2.9 PLME-SET.confirm

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

5.4.2.9.1 Semantics for the service primitive

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

PLME-SET.confirm (

status,

PhyMibAttribute

)

Table 5.25 specifies the parameters for the PLME-SET.confirm primitive.

Table 5.25: 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 5.28	The identifier of the PHY MIB attribute

5.4.2.9.2 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.

5.4.2.9.3 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 5.25.

5.4.2.9.4 PHY enumerations description

Table 5.26 shows PHY enumeration values defined for the PHY layer.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

Table 5.26: PHY enumerations description

Enumeration	Description
BUSY	CCA operation detected a busy channel
BUSY_RX	Transceiver is receiving and cannot change its mode (unless forced)
BUSY_TX	Transceiver is transmitting and cannot change its mode (unless forced)
CCA_CLEAR	CCA operation detected a clear channel.
FORCE_TRX_OFF	Transceiver **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 transceiver is disabled
TX_ON	Command or Status indicating that the transmitter is enabled
UNSUPPORTED_ATTRIBUTE	The specified MIB attribute is not supported

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

5.5 PHY constants and MIB attributes

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

5.5.1 PHY constants

The PHY **shall** comply with the constants defined in Table 5.27.

Table 5.27: PHY constants

Constants	Description	Value
aPhyMaxFrameSizeLR1	Maximum PSDU size for data rate LR1	192 bytes
aPhyTurnaroundTimeTXRX	TX-to-RX maximum turnaround time (see Section 5.2.5.8).	1 ms
aPhyTurnaroundTimerRXTX	RX-to-TX minimum turnaround time (see Section 5.2.5.9).	1 ms

5.5.2 PHY MIB attributes

The PHY management information base (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 5.28.

Table 5.28: PHY MIB attributes

Attribute	Type	Range	Description
phyCurrentTxChannel	Enumeration	A, B (see Table 5.3)	TX channel to use
phyMapChannelA	Enumeration	Available LRF profiles (see Table 5.1)	Apply LRF profile to channel A
phyMapChannelB	Enumeration	Available LRF profiles (see Table 5.1)	Apply LRF profile to channel B
phyTransmitPower	Enumeration	Valid output power levels (see Section 5.2.5.2)	Transmit power to use

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6 Z-WAVE LONG RANGE MAC LAYER SPECIFICATION

6.1 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 and does not use streaming. 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.

6.1.1 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
- node identification
- collision avoidance algorithm
- backoff algorithm
- automatic retransmission in case of transmission errors
- channel selection

The MAC layer provides two services: the MAC data service, accessed through the MAC layer data service access point (MD-SAP) and the MAC management service interfacing with the MAC layer management entity (MLME) service access point (MLME-SAP). The MAC data service enables reliable transmission and reception of MAC protocol data units (MPDUs) across the PHY data 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 "aMacLR", e.g., *aMacLRMaxMSDUSize*, and are listed in Table 6.32 and Table 6.33. Attributes have a general prefix of "macLR", e.g., *macLRHomeID*, and are listed in Table 6.34.

6.1.2 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 12 bit short address. A primary node hands out the HomeID and unique NodeIDs to all other nodes included in the domain.

6.1.3 Functional overview

The MAC features of an Z-Wave Long Range network include data transfer model, frame structure, robustness and power consumption.

6.1.3.1 MPDU formats

Several MPDU formats are defined.

Figure 6.1 shows the structure of the general MPDU. The MAC service data unit (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 MAC Protocol Data Unit (MPDU).

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

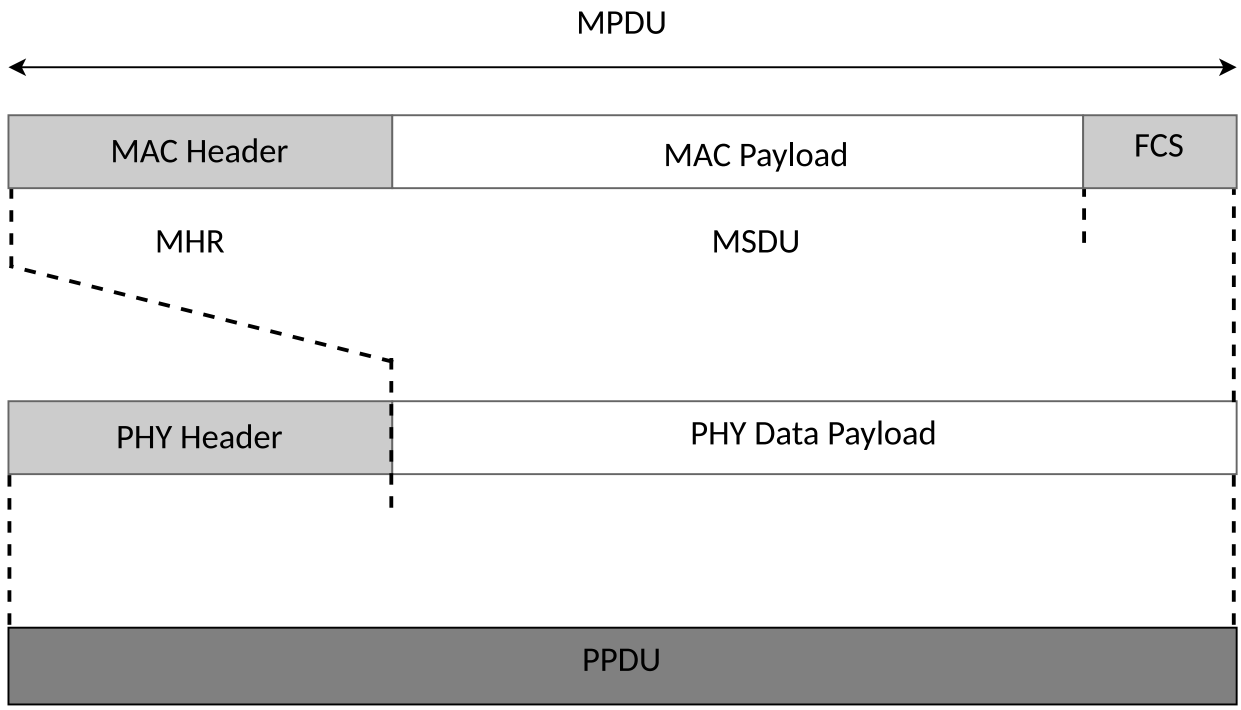

Figure 6.1: Generic MPDU format

The MPDU is passed to the PHY layer as a PHY service data unit (PSDU).

6.1.3.1.1 Singlecast MPDU

The singlecast MPDU uses the Generic MPDU format

6.1.3.1.2 Acknowledgment MPDU

The acknowledgment MPDU uses the Generic MPDU format. The MAC service data unit (MSDU) may have a length of zero bytes when used for acknowledgment.

6.1.3.2 Network Robustness

The Z-Wave Long Range MAC layer employs various mechanisms to ensure robustness in data transmission. These mechanisms of the MAC layer are back off algorithm, frame acknowledgment, data verification and frame retransmission.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.1.3.2.1 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.

6.1.3.2.2 Acknowledgment

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

6.1.3.2.3 Retransmissions

If the source node does not receive an acknowledgment, and has requested one, it **shall** assume that the transmission was unsuccessful and in response it **shall** retry the MPDU transmission up to *aMacLRMaxFrameRetries* times.

In order to avoid collisions, each transmitter **shall** delay a retransmission by a random delay. See Section 6.5.1

6.1.3.2.4 Data Validation

A 16-bit non-correcting frame check sequence (FCS) mechanism is employed to detect bit errors.

6.1.3.2.5 Channel selection

When a node is using channel configuration 3 the MAC layer is responsible for choosing the active channel used for transmission.

6.1.3.3 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 Always Listening mode is used for listening during wake-up polling.

Other battery-power nodes may require a more responsive behavior than can be achieved via periodic wake-up. Such nodes may use the PHY Frequently Listening mode for incoming messages.

6.1.3.3.1 Communication with a Frequently Listening node

Battery-powered devices may need to be reachable at any time. Nodes that are listening at regular intervals are said to operate in FL (Frequently Listening) 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.2 MAC Layer Service Specification

The MAC layer provides an interface to higher layers, typically the network layer, and the PHY layer. The MAC layer management entity (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 management information base (MAC MIB). Figure 6.2 depicts the components and interfaces of the MAC layer.

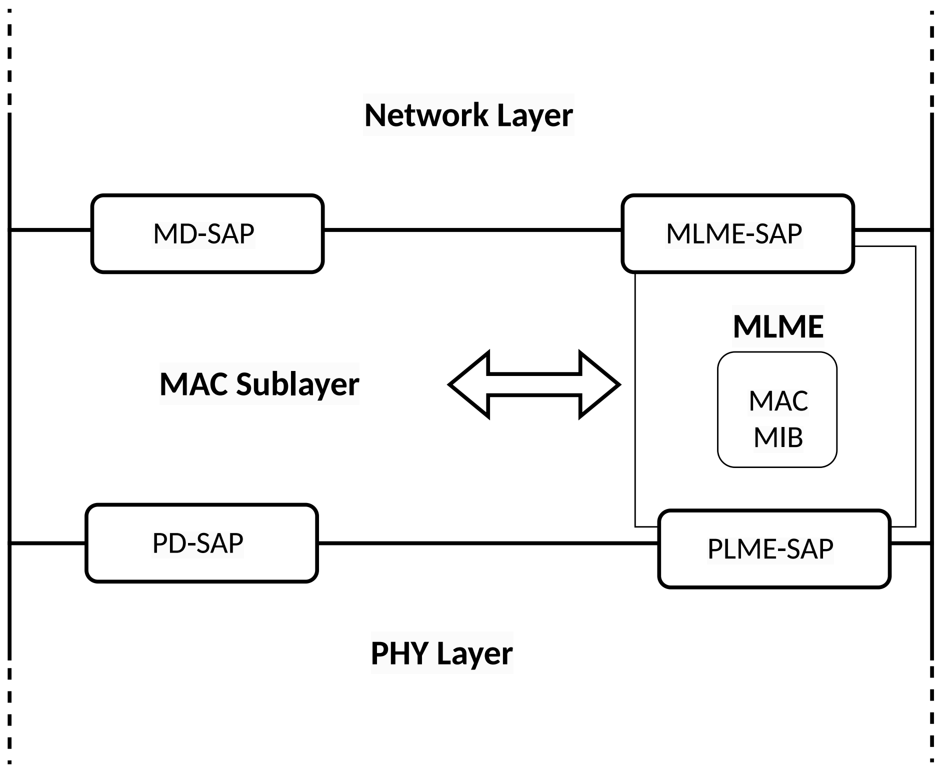

Figure 6.2: MAC layer reference model

The MAC layer **shall** provide two services to the network layer, accessed through two service access points (SAPs):

- The MAC data service, accessed through the MD-SAP, and
- The MAC management service, accessed through the MLME-SAP.

6.2.1 MAC enumerations description

This clause explains the meaning of the enumerations used in the primitives defined in the MAC entity specification. Table 6.1 shows a description of the MAC enumeration values.

Table 6.1: MAC enumerations description

Enumeration	Description
SUCCESS	The requested operation was completed successfully.
DISABLE_TRX_FAILURE	The attempt to disable the transceiver has failed.
FRAME_TOO_LONG	The frame has a length that is greater than *aMacLRMaxMSDUSize.*
INVALID_PARAMETER	A parameter in the primitive is out of the valid range.
NO_ACK	No acknowledgment was received after *aMacLRMaxFrameRetries*
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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.2.2 MAC Data Service

The MD-SAP supports the transport of network layer protocol data units (NPDU) between peer network layer entities. Table 6.2 lists the primitives supported by the MD-SAP. The primitives are discussed in the clauses referenced in the table.

Table 6.2: MD-SAP primitives

MD-SAP Primitive	Request	Confirm	Indicate
MD-DATA	Section 6.2.2.1	Section 6.2.2.2	Section 6.2.2.3

6.2.2.1 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.

6.2.2.1.1 Semantics of the service primitive

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

Table 6.3 specifies the parameters for the MD-DATA.request primitive:

MD-DATA.request (

SrcHomeID,

SrcNodeID,

DstNodeID,

sduLength,

msdu,

SequenceNumber,

TxType,

TxOptions,

BeamOption,

ChannelOption,

TxPower,

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

)

Table 6.3: MD-DATA.request parameters

Name	Type	Valid range	Description
SrcHomeID	Byte Array	0x00000000..0xFFFFFFFF	Domain identifier to use
SrcNodeID	Word (12 bit)	0x000..0xFE8	Source node identifier to use.
DstNodeID	Word (12 bit)	0x000..0xFFF	Destination node identifier to use.
msduLength	Byte	Less than or equal to *aMacLRMaxMSDUSize*	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	0x00,0x01	Transmission options for this MSDU. The control word is formed as a bit wise OR of one or more of the following values: 0x01 = Acknowledged transmission.
TxOptions	Byte	0x01,0x03	Transmission type to use. 0x01 = Singlecast or broadcast transmission 0x03 = Acknowledged transmission
BeamOption	Byte	0x00, 0x01	Battery support options for this MSDU. 0x00: No beam 0x01: fragmented beam
ChannelOption	Enumeration	Channel A, B or Primary channel or Secondary channel	Channel to use
TxPower	Byte	-127..127	Tx Power in dBm

6.2.2.1.2 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.

6.2.2.1.3 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 clear channel access (CCA, see Section 6.5.1.1). If the PHY PLME-GET-CCA primitive returns TRUE the MAC entity enables the transmitter 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 transmitter 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 acknowledgment for at least *aMacLRMinAckWaitDuration* symbols. If the MAC entity does not receive an acknowledgment within this time, it **shall** retransmit the MPDU one or more times as defined by *aMacLRMaxFrameRetries*. If the MAC entity does still not receive an acknowledgment, 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

If the MPDU could not be transmitted due to a busy channel (see Section 6.5.1.1) 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 *aMacLRMaxMSDUSize* 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 MAC Header (MHR) see Section 6.1.3.1.

6.2.2.2 MD-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.

6.2.2.2.1 Semantics of the PHY data confirm primitive

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

PD-DATA.confirm (

Status

)

Table 6.4 specifies the parameters for the PD-DATA.confirm primitive:

Table 6.4: MD-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

6.2.2.2.2 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.

6.2.2.2.3 Effects on receipt

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

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.2.2.3 MD-DATA.indication

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

6.2.2.3.1 Semantics of the PHY data indication primitive

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

MD-DATA.indication (

FrameType,

SrcHomeID,

SrcNodeID,

DstNodeID,

msduLength,

msdu,

SequenceNumber,

ChannelOption,

RxPower,

)

Table 6.5 specifies the parameters for the MD-DATA.indication primitive.

Frame Type indicates if the data delivered to the network layer is a data frame or a beam fragment. Refer to Section 6.3.6 for details on beam fragments.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

Table 6.5: MD-DATA.indication parameters

Name	Type	Valid range	Description
FrameType	Enumeration	0x01..0x	Code indicating the frame type: 0x00: Uninitialized type 0x01: Singlecast frame 0x03: Ack frame 0x09: Beam fragment
SrcHomeID	Byte Array	0x00000000..0xFFFFFFFF The value 0x00000000 is reserved for uninitialized 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	Word (12 bit)	0x000..0xFE8	NodeID of the originating node 0x000: Uninitialized NodeID 0x01..0xFE8: NodeID 0xE9..0xFF: Reserved
DstNodeID (Not applicable for multicast frame type)	Word (12 bit)	0x000..0xFFF	NodeID of the destination node 0x000: Reserved 0x01..0xFE8: NodeID 0xFE9..0xFFE: Reserved 0xFFF: Broadcast NodeID
msduChannel	Enumeration	Channel A or B	The channel the MSDU was received from
msduLength	Byte	Less than or equal to *aMacLRMaxMSDUSize*	The number of received bytes
msdu	Byte Array	n/a	The received MSDU
SequenceNumber	Byte	n/a	The unique number of this frame. Use same value for retransmissions
RxPower	Byte + (signed)	-127 .. 127	RSSI in dBm measured on the incoming frame

In case of a beam frame, the SrcHomeID field **shall** be formatted as specified in Section 6.3.6.

6.2.2.3.2 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.

6.2.2.3.3 Effects on receipt

The network layer is notified of the arrival of data.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

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.

6.2.2.4 Data service sequence chart

Figure 6.3 MAC data service sequence chart illustrates the sequence of messages necessary for a successful data transfer between two nodes.

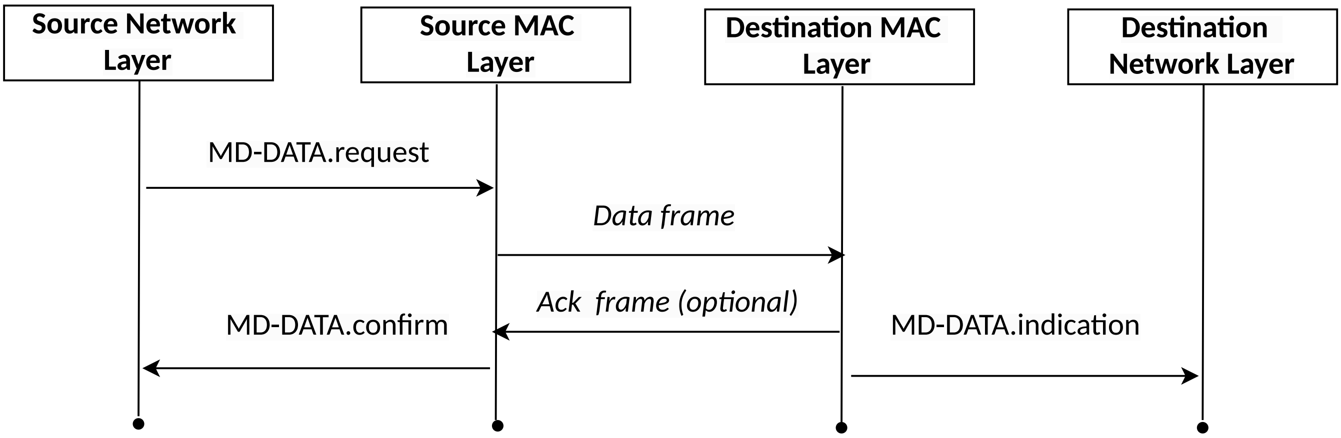

Figure 6.3: MAC data service sequence chart

6.2.3 MAC management service

The MLME-SAP allows the transport of management commands between the next higher layer and the MLME. Table 6.6 summarizes the primitives supported by the MLME through the MLME SAP interface.

Table 6.6: MLME-SAP primitives

MLME-SAP Primitive	Request	Confirm	Indication	Response
MLME-GET	Section 6.2.3.1	Section 6.2.3.2	n/a	n/a
MLME-SET	Section 6.2.3.3	Section 6.2.3.4	n/a	n/a
MLME-RESET	Section 6.2.3.5	Section 6.2.3.6	n/a	n/a

6.2.3.1 MLME_GET.request

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

6.2.3.1.1 Semantics for the service primitive

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

MLME-GET.request (

MacMibAttribute

)

Table 6.7 specifies the parameters for the MLME-GET.request primitive.

Table 6.7: PLME-SET-TRX_MODE.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table 6.34	The identifier of the MAC MIB attribute to read.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.2.3.1.2 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.

6.2.3.1.3 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.

6.2.3.2 MLME-GET.confirm

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

6.2.3.2.1 Semantics for the service primitive

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

MLME-GET.confirm (

status,

MacMibAttribute,

MacMibAttributeValue

)

Table 6.8 specifies the parameters for the MLME-GET.confirm primitive.

Table 6.8: 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 6.34 UNSUPPORTED_ATTRIBUTE	The identifier of the MAC MIB attribute that was read.
MacMibAttribute Value	Various	Attribute specific, See Table 6.34	The value of the indicated MAC MIB attribute that was read.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.2.3.2.2 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.

6.2.3.2.3 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.

6.2.3.3 MLME-SET.request

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

6.2.3.3.1 Semantics for the service primitive

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

MLME-SET.request (

MacMibAttribute,

MacMibAttributeValue

)

Table 6.9 specifies the parameters for the MLME-SET.request primitive.

Table 6.9: MLME-SET.request parameters

Name	Type	Valid range	Description
MacMibAttribute	Integer	See Table 6.34	The identifier of the MAC MIB attribute to write.
MacMibAttribute Value	Various	Attribute specific, See Table 6.34	The value to write to the indicated MAC MIB attribute.

6.2.3.3.2 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.

6.2.3.3.3 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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

6.2.3.4 MLME-SET.confirm

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

6.2.3.4.1 Semantics for the service primitive

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

MLME-SET.confirm (

status,

MacMibAttribute

)

Table 6.10 specifies the parameters for the MLME-SET.confirm primitive.

Table 6.10: 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	See Table 6.34	The identifier of the MAC MIB attribute that was written.

6.2.3.4.2 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 MC MIB attribute is updated, the MLME **shall** advertise a status of SUCCESS.

6.2.3.4.3 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 6.10

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.2.3.5 MLME-RESET.request

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

6.2.3.5.1 Semantics for the service primitive

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

MLME-RESET.request (

SetDefaultMIB

)

Table 6.11 specifies the parameter for the MLME-RESET.request primitive.

Table 6.11: 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

6.2.3.5.2 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.

6.2.3.5.3 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.

6.2.3.6 MLME-RESET.confirm

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

6.2.3.6.1 Semantics for the service primitive

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

MLME-RESET.confirm (

status

)

Table 6.12 specifies the parameter for the MLME-RESET.confirm primitive.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

Table 6.12: MLME-RESET.confirm parameters

Name	Type	Valid range	Description
status	Enumeration	SUCCESS, DISABLE_TRX_FAILURE	MAC entity successfully reset MAC entity reset operation failed

6.2.3.6.2 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.

6.2.3.6.3 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 6.12.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.3 MPDU Formats

This clause specifies the format of the MAC Protocol Data Unit (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 processed by the PHY. 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).

6.3.1 General MPDU format

The general MPDU format comprises the fields MHR, Data payload and MFR. The general MPDU **shall** be formatted as illustrated in Figure 6.4.

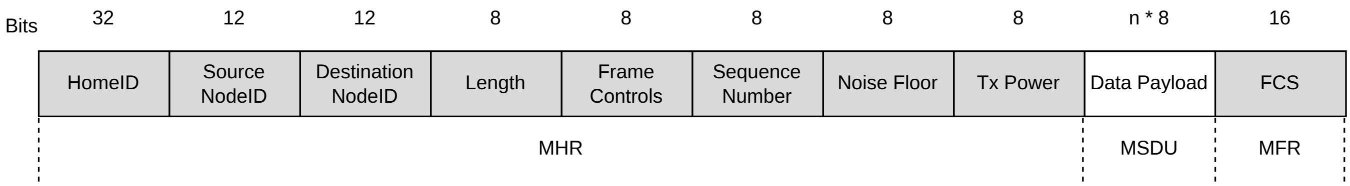

Figure 6.4: General MPDU Format

6.3.1.1 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.

Table 6.13: 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)							

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

The MAC layer **shall** support configuration of a promiscuous mode; forwarding all MPDUs to higher layers.

6.3.1.2 Source NodeID

The Source NodeID is 12 bit 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.

Table 6.14: Source NodeID

Byte \ Bit	11	10	9	8	7	6	5	4	3	2	1	0
4	Source NodeID (11-4)											
5	Source NodeID (3-0)											

The source NodeID **shall** comply with Table 6.15

Table 6.15: Source NodeID Values

NodeID	Usage
0x000	Uninitialized node
0x000 - 0x0FF	Reserved
0x100 - 0xFA0	Unique NodeID
0xFA1 - 0xFA5	Virtual Nodes
0xFA6 - 0xFFF	Reserved

6.3.1.3 Destination NodeID

The Destination NodeID is 12 bit in length and **shall** identify one ore more nodes in a given domain. Together with the HomeID, the destination NodeID identifies the node(s) that **shall** receive the MPDU.

Table 6.16: Destination NodeID

Byte \ Bit	11	10	9	8	7	6	5	4	3	2	1	0
4	Source NodeID (11-4)						Destination NodeID (11-8)					
5	Destination NodeID (7-0)											

The destination NodeID **shall** comply with Table 6.17

Table 6.17: Destination nodeID values

NodeID	Usage
0x000	Uninitialized node
0x000 - 0x0FF	Reserved
0x100 - 0xFA0	Unique NodeID
0xFA1 - 0xFA5	Virtual Nodes
0xFA6 - 0xFFE	Reserved
0xFFF	Broadcast

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.3.1.4 Length

The Length field is 1 byte in length and **shall** indicate the length of the MPDU in bytes including the FCS.

Table 6.18: Length

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

The length is limited to *aMacLRMaxMSDUSize*. The actual values can be found in Table 6.33. A receiving node **shall not** read more bytes than the maximum length allowed.

6.3.1.5 Frame Control

The Frame Control field is 8 bits in length and contains information defining the frame type and other control flags. The frame control field **shall** be formatted as illustrated in Table 6.19.

Table 6.19: Frame Control

Byte \ Bit	7	6	5	4	3	2	1	0
8	Ack Req	Extended	Reserved			Header Type		

6.3.1.5.1 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 acknowledgment is needed.

A receiving node **shall** return an Ack MPDU in response to the Acknowledgment Request.

6.3.1.5.2 Extend subfield

The Extended subfield is 1 bit in length and set to one if the frame header contains a header extension. If set to 0 the header does not contain a extended header. See Section 6.3.5 for details about the extended header.

6.3.1.5.3 Header Type subfield

The header type subfield defines the frame header type as described in Table 6.20.

Table 6.20: Frame Type

Header Type value	Description
0x1	Singlecast MPDU
0x2	Reserved
0x3	Acknowledgement MPDU
0x4 - 0x07	Reserved

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

A broadcast MPDU is a singlecast MPDU (header type 0x1) carrying destination NodeID = 0xFFF; see Section 6.3.1.3.

6.3.1.5.4 Reserved

All reserved fields **shall** be transmitted as 0 and and ignored by the receiver.

6.3.1.6 Sequence Number

The Sequence Number is an 8-bit field of the MPDU Header (MHR). The sequence number **shall** be formatted as illustrated in Table 6.21

Table 6.21: Sequence Number

Byte \ Bit	7	6	5	4	3	2	1	0
9	Sequence Number							

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 acknowledgment MPDU if acknowledgment is requested.

A transmitting node **shall** validate the received sequence number in an acknowledgment MPDU.

6.3.1.7 Noise Floor

The Noise Floor is a 8 bit signed field that indicates the radio noise level measured on the channel the frame is transmitted on. The Noise Floor field **shall** be formatted as illustrated in Table 6.22.

Table 6.22: Noise Floor

Byte \ Bit	7	6	5	4	3	2	1	0
10	Noise Floor							

The Noise floor **shall** be a RSSI value in dBm measured as a running average when there is no Z-Wave Long Range traffic being received on the channel.

The noise level **shall** comply with the values in Table 6.23

Table 6.23: Noise Level Values

Noise Level values	Description
-120..30	RSSI value in dBm
127	RSSI not available
-127..-126	Reserved
30..126	Reserved

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

In self-powered mode, the noise level value for a Wake On Event End Node (WOEEN) **may** be 127, as the node might skip the noise floor measurement in this mode.

6.3.1.8 Tx Power

The Tx Power is a 8 bit signed field that specifies the transmit power used to transmit this frame. The Tx Power field **shall** be formatted as illustrated in Table 6.24.

Table 6.24: Tx Power

Byte \ Bit	7	6	5	4	3	2	1	0
11	Tx Power							

The Tx Power **shall** be in dBm and measured as output from the radio not taking the antenna gain/loss into account..

The Tx Power **shall** comply with the values in Table 6.25.

Table 6.25: Tx Power Values

Noise Level values	Description
-100..35	Tx power in dBm
-127..-101	Reserved
36..127	Reserved

6.3.1.9 Data Payload

The Data Payload field has a variable length. A receiving node may derive the length of the Data Payload field from the MPDU Length field.

6.3.1.10 FCS

A 16-bit non-correcting Cyclic Redundancy Code (CRC) **shall** be used for validating the MPDU integrity.

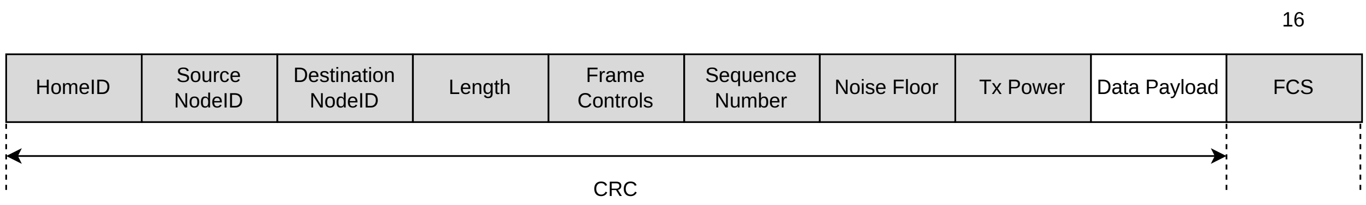

Figure 6.5: CRC calculation

The CRC-16 generator polynomial **shall** be:

P(x) = x16+x12+x5+1, 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.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

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.

6.3.2 Singlecast MPDU format

The singlecast MPDU format **shall** use the general MPDU frame format as defined in Figure 6.4. These fields in the general MPDU **shall** be set according to the rules outlined in the following clauses:

6.3.2.1 Destination NodeID

The destination NodeID must be set to a value identifying either a uninitialized node, a virtual node or a unique node in the network. See Section 6.3.1.3 for details.

6.3.2.2 Frame Control

6.3.2.2.1 Header Type subfield

The Header Type Subfield must be set to the Singlecast MPDU frame type. Refer to Section 6.3.1.5.3 for details.

6.3.2.3 Data Payload

The data payload of a singlecast frame **shall** contain at lease 1 byte of data.

6.3.3 Acknowledgement MPDU format

The Acknowledgement MPDU format **shall** use the general MPDU frame format as defined in Figure 6.4. The acknowledgement MDPU has one more field added apart from the fields in the general MPDU.

The Acknowledgement MPDU must be returned on the Long Range channel where the Singlecast frame triggering the Acknowledgement was received.

The Acknowledgement MPDU **shall** only be send as a reply to a singlecast MPDU with the Ack Req field set to 1.

32	12	12	8	8	8	8	8	8	n * 8	16
HomeID	Source NodeID	Destination NodeID	Length	Frame Controls	Sequence Number	Noise Floor	Tx Power	Receive RSSI	Data Payload	FCS

Figure 6.6: Acknowledgement MPDU

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

These fields in the Acknowledgement MPDU **shall** be set according to the rules outlined in the following clauses:

6.3.3.1 Destination NodeID

The destination NodeID **shall** be set to the Source NodeID of the Singlecast MPDU with the Ack Req field set.

6.3.3.2 Frame Control

6.3.3.2.1 Ack Req subfield

The Ack Req subfiels **shall** be set to 0 for a Acknowledgement MPDU.

6.3.3.2.2 Header type subfield

The header type subfield **shall** be set to the Acknowledgement header type. See Section 6.3.1.5.3 for details.

6.3.3.3 Sequence Number

The Sequence Number **shall** be set to the Sequence Number of the Singlecast MPDU with the Ack Req field set.

6.3.3.4 Received RSSI

The Receive RSSI is a 8 bit signed field that indicates the signal strength measured while receive the frame.. The Received RSSI field **shall** be formatted as illustrated in Table 6.26.

Table 6.26: Received RSSI

Byte \ Bit	7	6	5	4	3	2	1	0
12	Received RSSI							

The Received RSSI **shall** be a RSSI value in dBm measured as an average of at least 1 sample during reception of the Singlecast MPDU.

The Received RSSI **shall** comply with the values in Table 6.27.

Table 6.27: Received RSSI Values

Noise Level values	Description
-120..30	RSSI value in dBm
127	RSSI not available
-127..-126	Reserved
31..126	Reserved

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

6.3.3.5 Data Payload

The Data payload may contain any data for the acknowledgement MPDU.

6.3.4 Broadcast MPDU format

The broadcast MPDU format **shall** use the general MPDU frame format as defined in Figure 6.4. These fields in the general MPDU **shall** be set according to the rules outlined in the following clauses:

6.3.4.1 Destination NodeID

The destination NodeID must be set to the broadcast nodeID value. See Section 6.3.1.3 for details.

6.3.4.2 Frame Control

6.3.4.2.1 Ack Req subfield

The Ack Req subfiels **shall** be set to 0 for a Broadcast MPDU.

6.3.4.2.2 Header type subfield

The header type subfield **shall** be set to the Singlecast header type. See Section 6.3.1.5.3 for details.

6.3.4.3 Data Payload

The data payload of a Broadcast MPDU **shall** contain at lease 1 byte of data.

6.3.5 MPDU header extension format

The extended MPDU header format is an extension to the general MPDU frame format as defined in Figure 6.4. These fields in the general MPDU **shall** be set according to the rules outlined in the following clauses:

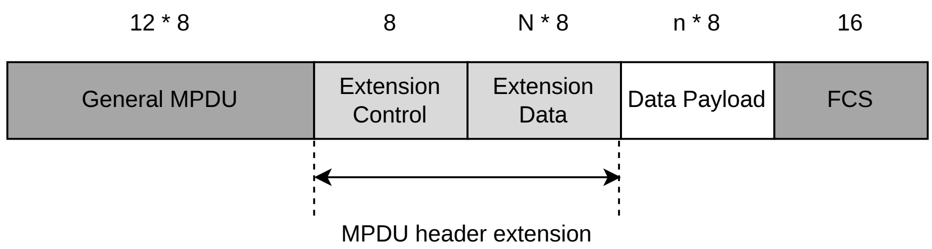

Figure 6.7: MPDU header extension

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.3.5.1 Frame Control

The Frame Control field is 8 bits in length and contains information defining the frame type and other control flags. The frame control field **shall** be formatted as illustrated in Table 6.28.

Table 6.28: Frame Control

Byte \ Bit	7	6	5	4	3	2	1	0
12	Reserved	Extension type			Discard unknown	Extension length		

6.3.5.1.1 Extension type

The Extension type subfield is 3 bit in length and specifies the type of data contained in the extension data field.

Table 6.29: Extension type Values

Extension Type	Description
0x0..0x7	Reserved

6.3.5.1.2 Discard unknown

The Discard unknown subfield is 1 bit in length and set to 1 when the transmitting node wants the receiving node to discard the frame if the Extension type is unknows to the receiving node.

If the Discard unknown subfield is set to 0 and the extension type is unknown to the receiver then the receiving node should discard the extension and treat the frame as a frame without extension.

6.3.5.1.3 Extension length

The extension length is 3 bit in length and contains the number of bytes in the Extension Data field. The length **shall** be from 0 to 7 bytes.

6.3.6 Beam Frame format

Beam frames may be used to awaken battery powered nodes operating in frequently listening (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 four 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 Section 6.3.6.1) this is a beam frame and the following **three** bytes carry beaming relevant information.

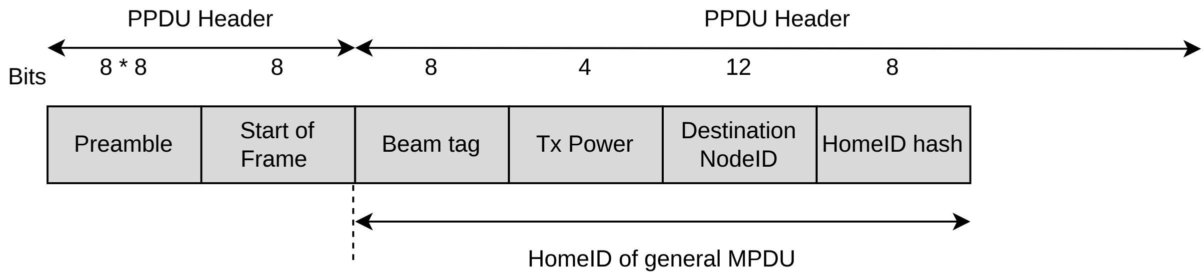

Figure 6.8: Beam MPDU format

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

A node **shall** listen long enough to detect a beam frame during worst case conditions around the SOF. 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.

6.3.6.1 Beam Tag

The bean tag is a 8 bit field identifying the MPDU as a beam frame. The beam tag **shall** comply with the values in Table 6.30

Table 6.30: Beam Tag Values

Beam Tag Values	Description
0x00..0x54	Frame is in general MPDU format
0x55	Beam frame
0x56..0xFF	Frame is in general MPDU format

HomeIDs in the range 0x55000000..0x55FFFFFF **shall not** be assigned to any domain as this would collide with the beam tag.

6.3.6.2 Tx Power

The Tx Power is a 4 bit field specifying the Rx Power used to transmit the beam frame. The Tx Power **shall** comply with the values in Table 6.31.

Table 6.31: Tx Power values

Tx Power Values	Description
0	-6dBm
1	-2dBm
2	+2dBm
3	+6dBm
4	+10dBm
5	+13dBm
6	+16dBm
7	+19dBm
8	+21dBm
9	+23dBm
10	+25dBm
11	+26dBm
12	+27dBm
13	+28dBm
14	+29dBm
15	+30dBm

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.3.6.3 Destination nodeID

The Destination NodeID is a 12 bit field identifying the destination of the beam frame. If the value is 0xFFF 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. See Section 6.3.1.3 for valid values.

6.3.6.4 HomeID hash

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 **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.

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++;
        }
        return HomeIdHash;
}
```

6.3.7 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.

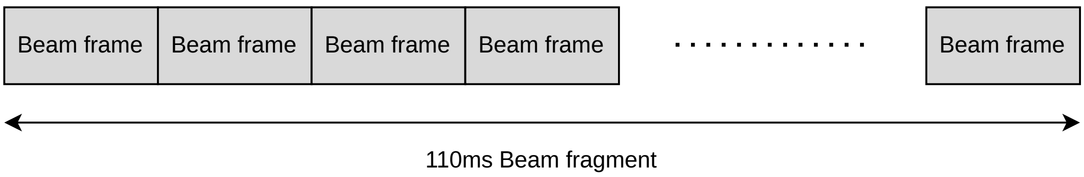

Figure 6.9: Beam fragment format

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. A receiver **shall** be able to monitor both channel A and B for beam frames.

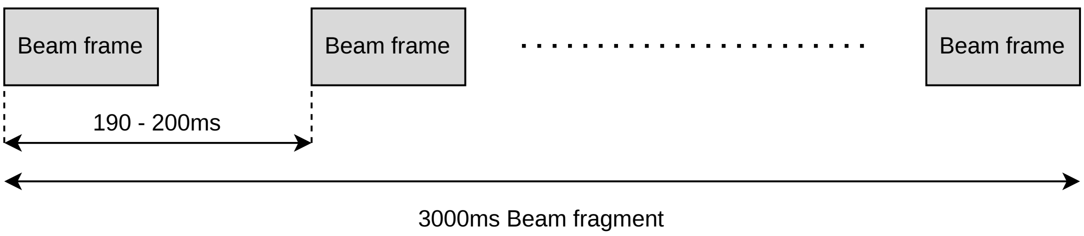

Figure 6.10: Fragmented beam format

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

Proper TX scheduling allows a sending node to reach sleeping nodes that use a range of wakeup intervals from 100ms to 1000ms. The chosen wakeup interval should be a trade off between battery lifetime and response time. When choosing a wakeup interval it should be set to a value that ensure that if the first wakeup is in a Tx pause period then the next wakeup **shall** be in the Tx period of the fragmented beam,

A fragmented beam may address any NodeID. A full fragmented beam **shall** span 3000 ms. A singlecast frame **shall** follow the fragmented beam. A receiving node **shall** interrupt the transmission of a fragmented beam by acknowledging a singlecast 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 not** send an Ack MPDU.

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.

6.3.7.1 Broadcast beaming

A receiving node receiving a beam with a broadcast address (0xFFF) and a HomeID hash matching its own HomeID, **shall** stay in receive and expect a MPDU from the originator if the *macLRenableFLBroadcast* is set to 1. If *macLRenableFLBroadcast* is set to 0 the MAC **shall not** expect to receive an MPDU and the receiver should be powered down.

A receiving node receiving a beam with a broadcast address (0xFFF) and a HomeID hash matching its own HomeID **shall** acknowledging the beam fragment. A receiving FL node detecting a positive match for the HomeID Hash field of the Beam frame **shall** return the Ack MPDU after a random wait in the range defined in *aMacLRBroadcastFLAckWait*. A receiving FL node not detecting a positive match for the HomeID Hash field **shall not** send an Ack MPDU.

A transmitting node transmitting a broadcast fragmented beam may stop sending the fragmented beam if it has received ACK frames from all expected receivers of the broadcast fragmented beam.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.4 MAC constants and MIB attributes

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

6.4.1 MAC constants

The constants used by the MAC layer are presented in Table 6.32 and Table 6.33. These constants are hardware dependent and cannot be changed during operation.

Table 6.32: General MAC Constants

Constants	Description	Value
aMacLRMaxMSDUSize	The maximum singlecast/broadcast MSDU size	190 bytes
aMacLRMinAckWaitDuration	The minimum Ack wait duration in milliseconds	*aPhyTurnaroundTimeRxTx* + (*aMacLRTransferAckTimeTX* * (1/100kbps))
aMacLRMaxFrameRetries	The number of retries after a transmission failure	2
aMacLRMaxFrameRetriesSecondary	The number of retries on the secondary channel when running channel configuration 3	1
aMacLRMinCCARetryDuration	The minimum duration of clear channel access assessment. See Section 6.5.1.1 for details.	110 ms

Table 6.33: MAC Constants for MPDU transfer

Constants	Description	Value
aMacLRBroadcastFLAckWait	Random time to wait before returning a beam ack when receiving a broadcast addressed beam	0..10ms
aMacLRTransferAckTimeTX	The number of symbols of an Ack MPDU; including preamble	448 bits
aMacLRTypicalFrameLengthTX(ch)	The number of symbols of a singlecast MPDU with a data payload of 4 bytes	472 bits
aMacLRcMinRetransmitDelay	Random backoff **shall** be higher than this	10 ms
aMacLRMaxRetransmitDelay	Random backoff **shall** be lower than this value	40 ms

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.4.2 MIB attributes

The MAC management information base (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 6.34.

Table 6.34: MAC MIB attributes

Attribute	Type	Range	Description	Default
macLRCCARetryDuration	Integer	aMacLRMinCCARetryDuration - infinite	The duration of clear channel access assessment.	~ 110 ms
macLRHomeID	Byte Array	0x00000000 .. 0xFFFFFFFF	The HomeID is the unique domain identifier.	(random)
macLRNodeID	12 bit	0x000..0xFFF	The NodeID is the address of the individual nodes in a domain.	0x000
macLRPromiscuousMode	Boolean	TRUE or FALSE	This indicates whether the MAC layer is in a promiscuous (receive all) mode. A value of TRUE indicates that the MAC layer accepts all frames received from the PHY.	FALSE
macLRRxOnWhenIdle	Boolean	TRUE or FALSE	This indicates whether the MAC layer is to enable its receiver at any time.	TRUE
macLRenableFLBroadcast	Boolean	0 or 1	0 Reception of broadcast beam frames are disabled 1 Reception of broadcast beam frames is enabled	0

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.5 MAC Functional description

6.5.1 Transmission, Reception and Acknowledgement

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

6.5.1.1 Clear Channel Assessment

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.

Note that the timing and method for doing a single CCA is defined by RF regulatory rules for the specific frequency band and country where the implementation is to be used.

If the MAC layer finds the channel busy for a period of *macLRCCARetryDuration* 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 Section 6.2.2.2).

6.5.1.2 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. If transmitting an Acknowledgement MPDU on the same channel as the Singlecast MPDU was received on the CCA should not be performed before transmitting the acknowledgement.

The source HomeID and source NodeID field **shall** identify the sending node and the destination NodeID **shall** identify the destination node.

6.5.1.2.1 Dynamic Tx Power

To ensure a power efficient implementation and avoid unnecessary network disturbance, the MAC layer **shall** implement an algorithm for reducing Tx power to the necessary power to reach the destination. The algorithm **shall** ensure that the minimum Tx power is used and at the same time maintain a link budget between the source and destination that ensures a robust and error free communication.

The algorithm **may** be disabled for a Wake On Event End Node (WOEEN) when it is in self-powered mode.

6.5.1.3 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 transceiver task requests from the network layer. A transceiver task **shall** be defined as a transmission request, a reception request, or a clear channel access detection. On completion of each transceiver 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 acknowledgment request subfield of the frame control field is set to 1, the MAC layer **shall** send an acknowledgment frame.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved This document may only be copied and distributed internally.

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

6.5.1.3.1 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 6.32.

6.5.1.4 Backup channel handling (Channel configuration 3)

When a node is running channel configuration 3, supporting both channel A and B, the MAC layer is responsible for keeping track of what channel is the active channel that is preferred for transmission. A node running channel configuration 3 will have the consept of a Primary channel and a Secondarys channel Both channels **shall** be scanned for incoming frames but only the primary channel should be used for transmissions.

Selecting the Primary channel is based on reception of frames. When a node receives a frame on channel X and the MAC layer has validated the frame, and has a match on HomeID and NodeID then the node **shall** set its Primary channel to channel X.

6.5.1.5 Use of Acknowledgement

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

Sequence number checking **shall** be applied to acknowledgment handling. Refer to Section 6.3.1.6.

6.5.1.5.1 No Acknowledgement

An MPDU transmitted with its acknowledgment 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 6.11 shows the scenario for transmitting a single MPDU without requiring an acknowledgment.

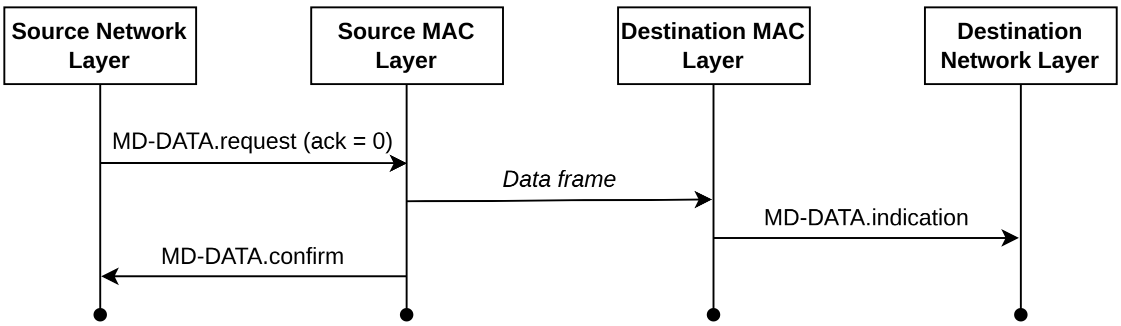

Figure 6.11: Successful transmission, no acknowledgement

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

6.5.1.5.2 Acknowledgement

A singlecast MPDU transmitted with the acknowledgment 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 acknowledgment MPDU. Only the singlecast MPDU may be sent with the acknowledgment request subfield set to 1. For other frame types the acknowledgment request subfield **shall** be ignored by the intended recipient.

The transmission of an acknowledgment MPDU **shall not** commence before aPhyTurnaroundTimeRXTX symbols have elapsed after the reception of the last symbol of the frame. Refer to Section 5.2.5.9.

The sequence diagram in Figure 6.12 shows the transmission of an acknowledged singlecast MPDU.

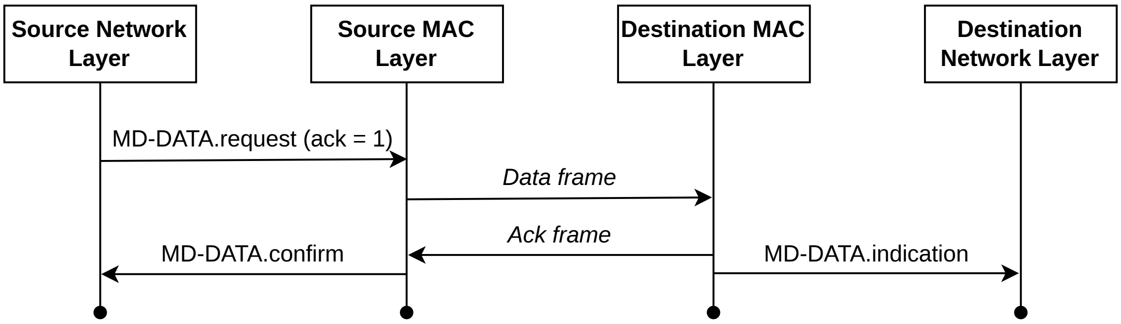

Figure 6.12: Successful transmission, acknowledgement

6.5.1.5.3 Retransmissions (Channel configuration 1 and 2)

A node that sends a singlecast MPDU with its acknowledgment request subfield set to 1 **shall** wait for a minimum of *aMacLRMinAckWaitDuration* for the corresponding Ack MPDU to be received. If an Ack MPDU is received within *aMacLRMinAckWaitDuration* and contains the correct HomeID, source NodeID and a matching sequence number, the transmission is considered successful, and no further action **shall** be taken by the originator. If an acknowledgment MPDU is not received within *aMacLRMinAckWaitDuration* the originator **shall** start the random backoff periode (see Section 6.5.1.5.5), and repeat the process of transmitting the MPDU and waiting for the Ack MPDU up to *aMacLRMaxFrameRetries* times.

If an Ack MPDU is still not received after *aMacLRMaxFrameRetries* 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 Section 5.4.1.2).

6.5.1.5.4 Retransmissions (Channel configuration 3)

A node that sends a singlecast MPDU with its acknowledgment request subfield set to 1 **shall** wait for a minimum of *aMacLRMinAckWaitDuration* for the corresponding Ack MPDU to be received. If an Ack MPDU is received within *aMacLRMinAckWaitDuration* and contains the correct HomeID, source NodeID and a matching sequence number, the transmission is considered successful, and no further action **shall** be taken by the originator. If an acknowledgment MPDU is not received within *aMacLRMinAckWaitDuration* the originator **shall** start the random backoff period (see Section 6.5.1.5.5), and repeat the process of transmitting the MPDU and waiting for the Ack MPDU up to *aMacLRMaxFrameRetries* times.

If an Ack MPDU is still not received after *aMacLRMaxFrameRetries* retransmissions, the MAC layer **shall** switch to the Secondary channel and and repeat the process of transmitting the MPDU and waiting for the Ack MPDU up to *aMacLRMaxFrameRetriesSecondary* times.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

If an Ack MPDU is still not received after *aMacLRMaxFrameRetries* + *aMacLRMaxFrameRetriesSecondary* 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 Section 5.4.1.2).

6.5.1.5.5 Random backoff

If a singlecast MPDU with its acknowledgment request subfield set to 1 or the corresponding acknowledgment 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 *aMacLRMinRetransmitDelay… aMacLRMaxRetransmitDelay* (Refer to Table 6.33).

If an Ack MPDU is received within the random backoff period and contains the correct HomeID, source NodeID and a matching sequence number, the transmission is considered successful.

6.5.1.6 Idle mode

If the MLME is requested to set *macLRRxOnWhenIdle* to TRUE the PHY **shall** enter RX mode and stay in RX mode when a MPDU has been transmitted (always listening). 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 *macLRRxOnWhenIdle* 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.

6.5.2 Transmission Scenarios

Due to the imperfect nature of the radio medium, a transmitted MPDU does not always reach its intended destination. Figure 6.13 to Figure 6.15 illustrates 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 *aMacLRMinAckWaitDuration* 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.

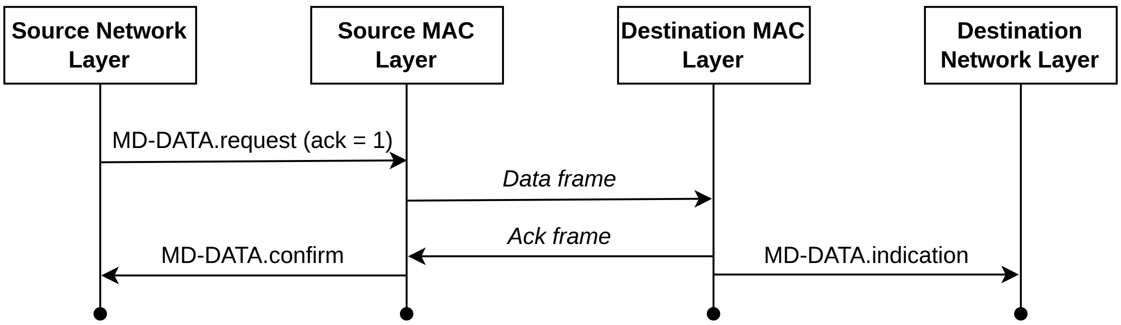

Figure 6.13: Successful transmission scenario

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

- **Lost MPDU.** The originating MAC layer transmits the MPDU to the recipient via the PHY data service. The originator MAC layer waits for *aMacLRMinAckWaitDuration* symbols. The destination MAC layer does not receive the MPDU 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 *aMacLRMaxFrameRetries* times. If transmissions fail a total of (1 + *aMacLRMaxFrameRetries*) times, the originator MAC layer issues a failure confirmation to the network layer.

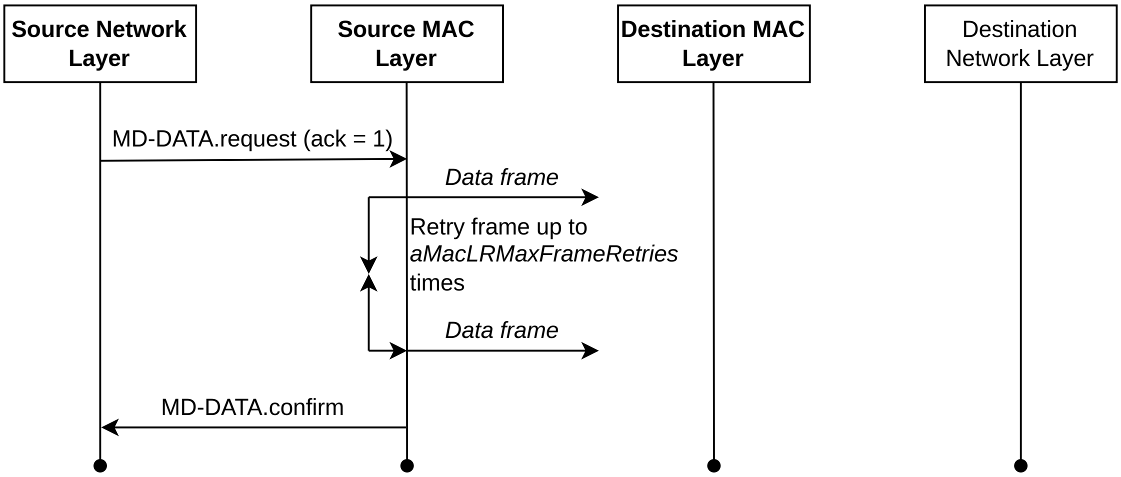

Figure 6.14: Lost frame transmission scenario

- **LostAck MPDU.** The originating MAC layer transmits the MPDU to the recipient via the PHY data service. The originating MAC layer waits for *aMacLRMinAckWaitDuration* 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 + *aMacLRMaxFrameRetries*) times, the MAC layer issues a failure confirmation to the network layer.

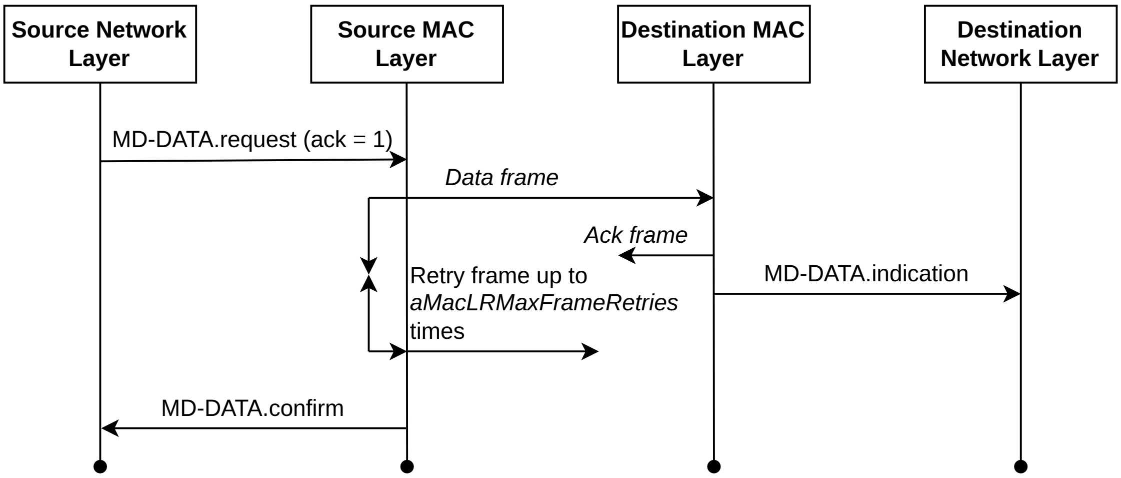

Figure 6.15: Lost acknowledgement transmission scenario

© 2025 Z-Wave Alliance, Inc. All Rights Reserved

References

[G9959] G.9959. ITU-T G.9959 Short range narrowband digital radiocommunication transceivers – PHY & MAC layer specifications.

© 2025 Z-Wave Alliance, Inc. All Rights Reserved
This document may only be copied and distributed internally.