ISO/IEC 29145-2

Edition 1.0 2014-03

INTERNATIONAL STANDARD

Information technology – Wireless beacon-enabled energy efficient mesh network (WiBEEM) standard for wireless home network services – Part 2: MAC layer

ISO/IEC 29145-2:2014-03(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

ISO/IEC 29145-2

Edition 1.0 2014-03

INTERNATIONAL STANDARD

Information technology – Wireless beacon-enabled energy efficient mesh network (WiBEEM) standard for wireless home network services – Part 2: MAC layer

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE U

ICS 35.200

ISBN 978-2-8322-1449-7

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INFORMATION TECHNOLOGY – WIRELESS BEACON-ENABLED ENERGY EFFICIENT MESH NETWORK (WIBEEM) STANDARD FOR WIRELESS HOME NETWORK SERVICES –

Part 2: MAC layer

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 29145-2 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 29145 series, under the general title *Information technology – Wireless beacon-enabled energy efficient mesh network (WiBEEM) for wireless home network services*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

This International Standard specifies the WiBEEM (Wireless Beacon-enabled Energy Efficient Mesh network) protocol, which provides low-power-consuming mesh network functions by enabling the “beacon mode operation”. WiBEEM is based on IEEE 802.15.4 standard with additional upper layer protocols and a specific usage of the MAC layer protocol. Through the novel use of beacons, WiBEEM technology achieves longer battery life, larger network support, quicker response, enhanced mobility and dynamic reconfiguration of the network topology compared with other protocols such as ZigBee.

In the beacon mode, Beacon information propagates over the entire mesh network nodes during the BOP (Beacon-Only Period) of the superframe structure without any beacon conflicts by utilising a smart beacon scheduling technique in the BOP. It also provides location information about moving devices without spending extra time running a positioning and locating algorithm by using RSSI (Received Signal Strength Indication). These features allow the WiBEEM protocol to be widely used for wireless home network services in the ubiquitous network era.

One of the key features of the WiBEEM protocol is that it has a special time interval called BOP (Beacon-Only Period) in the superframe structure that allows more than two beacons to be transmitted. This unique time period is located at the beginning of the Superframe. Because the BOP does not use the CSMA/CA mechanism, the network will not work properly in the beacon mode unless an appropriate algorithm is applied. This algorithm needs to manage and control multiple beacons in a single superframe. The solution is the Beacon Scheduling method applied in the BOP to avoid collisions among beacons, providing synchronisation among all the nodes of the entire mesh network.

For the network layer, the NAA (Next Address Available) mechanism, which is a short address allocation algorithm, has been adopted to provide an efficient way of utilising the complete 16-bit address space. The NAA algorithm does not limit the maximum number of children nodes that a node of a mesh network can have. Since the number of children nodes is unlimited, the NAA mechanism allows the WiBEEM protocol to be used not only for home network services, but also for community services. WiBEEM can be used where high network expandability through efficient use of short address spaces, device mobility and end-to-end QoS are required.

This part of the standard ISO/IEC 29145 specifies the Medium Access Control (MAC) layer of the WiBEEM protocol.

INFORMATION TECHNOLOGY – WIRELESS BEACON-ENABLED ENERGY EFFICIENT MESH NETWORK (WIBEEM) STANDARD FOR WIRELESS HOME NETWORK SERVICES –

Part 2: MAC layer

1 Scope

This part of ISO/IEC 29145 specifies the MAC of the WiBEEM (Wireless Beacon-enabled Energy Efficient Mesh network) protocol for wireless home network services that supports a low power-consuming wireless mesh network as well as device mobility and QoS.

2 Normative reference

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 29145-1:2014, *Information technology – Wireless beacon-enabled energy efficient mesh network (WiBEEM) for wireless home network services – Part 1: PHY layer*

IEEE 802.15.4:2003, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 29145-1, as well as the following apply.

3.1.1
association
service used to establish the membership of a device in a wireless mesh network

3.1.2
co-ordinator
wireless device configured to provide synchronisation services through the transmission of beacons

Note 1 to entry: If a co-ordinator is the principal controller of a wireless mesh network, it is called the WMC (WiBEEM Mesh Co-ordinator).

3.1.3
device
entity containing an implementation of the WiBEEM applications, NWK, MAC and physical interface to the wireless medium

3.1.4
disassociation
service that removes an existing association

3.1.5
frame
data format of aggregated bits from a medium access control (MAC) layer entity transmitted in a specified sequence

3.1.6
logical channel
abstract representation of a communications link independent of the physical implementation

Note 1 to entry: Defines "logical channel" as it is used this International Standard.

3.1.7
mobile device
device that uses network communications while in motion

3.1.8
orphaned device
device that has lost its parent node with its associated mesh network

3.1.9
packet
format of aggregated bits transmitted in a specified sequence across the physical medium

3.1.10
payload data
contents of a data message that is being transmitted

3.1.11
personal operating space
space of typically about 10 m around a person or an object, no matter whether this person or object is stationary or in motion

3.1.12
protocol data unit
unit of data exchanged between two peer entities

3.1.13
service data unit
information delivered as a unit through a service access point (SAP)

3.2 Abbreviations

The following acronyms and abbreviations are used in this standard. They are commonly used in other industry publications.

BI	Beacon Interval
BN	Beacon Number
BO	Beacon Order
BOP	Beacon Only Period
BSN	Beacon Sequence Number
BTTSL	Beacon Transmit Time Slot Length
CAP	Contention Access Period

CFP	Contention-Free Period
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access With Collision Avoidance
DSN	Data Sequence Number
ED	Energy Detection
ID	Identifier
IFS	Interframe Space or Spacing
LIFS	Long Interframe Spacing
LQ	Link Quality
LQI	Link Quality Indication
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
MFR	MAC Footer
MHR	MAC Header
MIB	MAC Information Base
MLME	MAC Layer Management Entity
MLME-SAP	MAC Layer Management Entity-Service Access Point
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NAA	Next Address Available
PDU	Protocol Data Unit
PHY	Physical Layer
PIB	PAN Information Base
PLME	Physical Layer Management Entity
PLME-SAP	Physical Layer Management Entity-Service Access Point
POS	Personal Operating Space
PQP	Prioritised QoS Period
QoS	Quality of Service
RAP	Reservation-based Access Period
RSSI	Received Signal Strength Indication
RX	Receive or Receiver
SAP	Service Access Point
SD	Superframe Duration
SDL	Specification and Description Language
SDU	Service Data Unit
SO	Superframe Order
WED	WiBEEM End Device
WiBEEM	Wireless Beacon-enabled Energy Efficient Mesh network
WMC	WiBEEM Mesh Co-ordinator
WRC	WiBEEM Routable Co-ordinator

3.3 Conventions

All the italicized words used in this standard represent relevant constants defined and stored in the MIB (Management Information Base) of each layer.

4 Conformance

A wireless device that claims conformance to this standard shall implement all the primitives that are specified in 6.2 and the MAC frame formats in 6.3. Each WiBEEM device shall be able to act as a WMC, a WRC and a WED. When operating in the role of a WMC it shall act as specified in 5.3.2 of ISO/IEC 29145-1:2014, when operating in the role of a WRC, it shall act as specified in 5.3.3 of ISO/IEC 29145-1:2014, and when operating in the role of a WED, it shall act as specified in 5.3.3 of ISO/IEC 29145-1:2014.

5 Overview of the WiBEEM technology

Clause 5 of ISO/IEC 29145-1:2014 presents an overview of the WiBEEM technology and the functionalities of the WiBEEM devices.

6 MAC layer specifications

6.1 General

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

- generating network beacons if the device is a co-ordinator;
- synchronising to network beacons;
- supporting the mesh network association and disassociation;
- supporting device security;
- employing the CSMA/CA mechanism for channel access;
- handling and maintaining the RAP mechanism;
- providing a reliable link between two peer MAC entities;
- providing multi-rate operation between PLME and MLME.

Constants and attributes that are specified and maintained by the MAC layer are written in the text of this clause in italics. Constants have a general prefix of “a”. Attributes have a general prefix of “mac”.

6.2 MAC layer service specifications

6.2.1 Service overview

The MAC layer services provide an interface between the MAC layer and the PHY layer. The MAC layer conceptually includes a management entity called MLME. This entity shall provide the service interfaces through which layer management functions may be invoked. MLME is also responsible for maintaining a database of managed objects pertaining to the MAC layer. This database is referred to as the MIB representing the MAC layer information base. The MAC layer provides two services, accessed through two SAPs:

- MAC data service, accessed through the MAC layer data SAP (MLDE-SAP), and
- MAC management service, accessed through the MLME-SAP.

These two services provide the interface between the MAC and the PHY layers, via the PLDE-SAP and PLME-SAP interfaces shown in Figure 1. In addition to these external interfaces, an implicit interface also exists between the two layers.

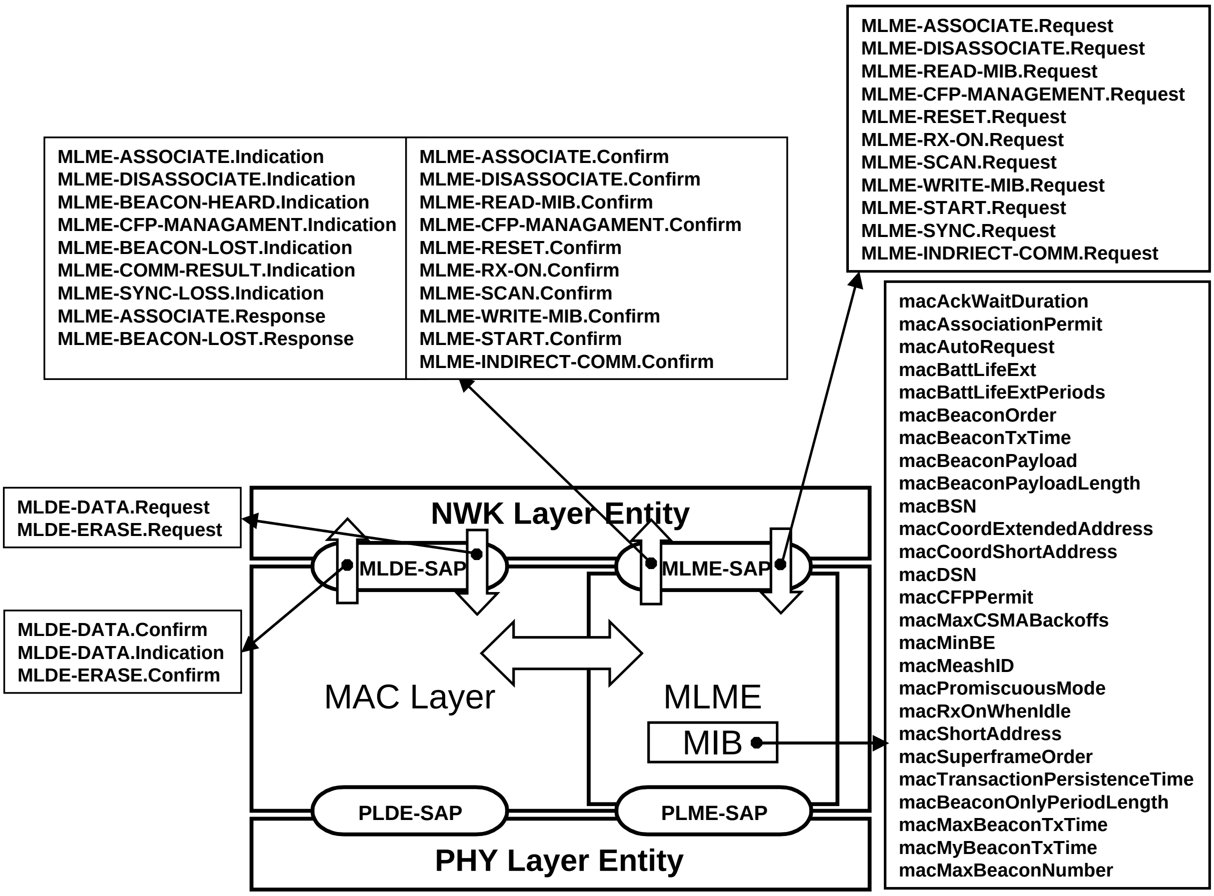


Figure 1 – MAC layer structure

6.2.2 MAC data service

6.2.2.1 Overview

The MLDE-SAP supports the transport of MAC protocol data units (MPDUs) between peer MAC entities.

6.2.2.2 MLDE-DATA.request

6.2.2.2.1 Function

The MLDE-DATA.request primitive requests the transfer of a data MSDU from a local MAC entity to a single peer MAC entity.

6.2.2.2.2 Semantics of the service primitive

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

```
MLDE-DATA.request (
                        SrcAddrMode,
                        SrcMeshId,
                        SrcAddr,
                        DstAddrMode,
                        DstMeshId,
                        DstAddr,
                        msduLength,
                        msdu,
                        msduHandle,
```

TxOptions
)

Table 1 specifies the parameters for the MLDE-DATA.request primitive.

Table 1 – MLDE-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00 to 0x03	The source addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 6.3.2). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcMeshId	Integer	0x0000 to 0xffff	The 16-bit Mesh identifier of the entity to which the MSDU is being transferred.
SrcAddre	Device address	As specified by theSrcAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
DstAddrMode	Integer	0x00 to 0x03	The destination addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted, see 6.3.2). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstMeshId	Integer	0x0000 to 0xffff	The 16-bit mesh network identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	As specified by the DstAddrMode parameter	The individual device address of the entity to which the MSDU is being transferred.
msduLength	integer	≤ aMaxMAC-FrameSize	The number of octets contained in the MSDU to be transmitted by the MAC layer entity.
msdu	Set of octets	–	The set of octets forming the MSDU to be transmitted by the MAC layer entity.
msduHandle	integer	0x00 to 0xff	The handle associated with the MSDU to be transmitted by the MAC layer entity.
TxOptions	Bitmap	3-bit field	The 3 bits (b_0, b_1, b_2) indicate the transmission options for this MSDU. 0x01 = acknowledged transmission 0x02 = RAP transmission 0x04 = Indirect transmission 0x08 = security enabled transmission

6.2.2.2.3 Appropriate usage

The MLDE-DATA.request primitive is generated by a local MAC entity when an MSDU is to be transferred to a peer MAC entity.

6.2.2.2.4 Effect on receipt

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

6.2.2.3 MLDE-DATA.confirm

6.2.2.3.1 Function

The MLDE-DATA.confirm primitive reports the results of a request to transfer a data MSDU from a local MAC entity to a single peer MAC entity.

6.2.2.3.2 Semantics of the service primitive

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

```
MLDE-DATA.confirm          (
                            msduHandle,
                            Status
                            )
```

Table 2 specifies the parameters for the MLDE-DATA.confirm primitive.

Table 2 – MLDE-DATA.confirm parameters

Name	Type	Valid range	Description
MsduHandle	Integer	0x00 to 0xff	The handle associated with the MSDU being confirmed.
Status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_RAP, NO_ACK, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the last MSDU transmission.

6.2.2.3.3 When generated

The MLDE-DATA.confirm primitive is generated by the MAC layer entity in response to an MLDE-DATA.request primitive. The MLDE-DATA.confirm primitive returns a status of either SUCCESS, indicating that the request to transmit was successful, or the appropriate error code including TRANSACTION_ OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_RAP, NO_ACK, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER.

6.2.2.3.4 Appropriate usage

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

6.2.2.4 MLDE-DATA.indication

6.2.2.4.1 Function

The MLDE-DATA.indication primitive indicates the transfer of an MSDU from the MAC layer to the local MAC entity.

6.2.2.4.2 Semantics of the service primitive

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

```
MLDE-DATA.indication  (
                        SrcAddrMode,
                        SrcMeshID,
                        SrcAddr,
                        DstAddrMode,
                        DstMeshID
                        DstAddr,
                        msduLength,
                        msdu,
                        mpduLinkQuality,
                        SecurityUse,
                        ACLEntry
                        )
```

Table 3 specifies the parameters for the MLDE-DATA.indication primitive.

Table 3 – MLDE-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Integer	0x00 to 0x03	The source addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcMeshID	Integer	0x0000 to 0xffff	The 16-bit Mesh identifier of the entity from which the MSDU was received.
SrcAddre	Device address	Specified by the SrcAddrMode parameter	The individual device address of the entity from which the MSDU was received.
DstAddrMode	Integer	0x00 to 0x03	The destination addressing mode for this primitive corresponding to the received MPDU. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short device address. 0x03 = 64-bit extended device address.
DstMeshID	Integer	0x0000 to 0xffff	The 16-bit Mesh identifier of the entity to which the MSDU is being transferred.
DstAddr	Device address	Specified by the DstAddr	The individual device address of the entity to which the MSDU is being transferred.
msduLength	integer	≤ aMaxMAC-FrameSize	The number of octets contained in the MSDU being indicated by the MAC layer entity.
msdu	Set of octets	-	The set of octets forming the MSDU being indicated by the MAC layer entity.
mpduLinkQuality	integer	0x00 to 0xff	LQI value measured during reception of the MPDU. Lower values represent lower LQI.
SecurityUse	Boolean	TRUE or FALSE	The security level purportedly used by the received data frame.
ACLEntry	integer	0x00 to 0x08	Parameter Value of ACL *macSecurityMode*

6.2.2.4.3 When generated

The MLDE-DATA.indication primitive is generated by the MAC layer and issued to the SSCS on receipt of a data frame at the local MAC layer entity.

6.2.2.4.4 Appropriate usage

On receipt of the MLDE-DATA.indication primitive, the MAC entity is notified of the arrival of data at the device.

6.2.2.5 MLDE-ERASE.request

6.2.2.5.1 Function

The MLDE-ERASE.request primitive allows the next higher layer to erase an MSDU from the transaction queue.

6.2.2.5.2 Semantics of the service primitive

The semantics of the MLDE-ERASE.request primitive are as follows:

```
MLDE-ERASE.request  (
                     msduHandle
                     )
```

Table 4 specifies the parameters for the MLDE-ERASE.request primitive.

Table 4 – MLDE-ERASE.request parameters

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

6.2.2.5.3 Appropriate usage

The MLDE-ERASE.request primitive is generated by the next higher layer whenever a MSDU is to be erased from the transaction queue.

6.2.2.5.4 Effect on receipt

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

6.2.2.6 MLDE-ERASE.confirm

6.2.2.6.1 Function

The MLDE-ERASE.confirm primitive allows the MAC layer to notify the next higher layer of the success of its request to erase an MSDU from the transaction queue.

6.2.2.6.2 Semantics of the service primitive

The semantics of the MLDE-ERASE.confirm primitive are as follows:

```
MLDE-ERASE.confirm (
                     msduHandle,
                     Status
                    )
```

Table 5 specifies the parameters for the **MLDE-ERASE**.confirm primitive.

Table 5 – MLDE-ERASE.confirm parameters

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

6.2.2.6.3 When generated

The MLDE-ERASE.confirm primitive is generated by the MAC layer entity in response to an MLDE-ERASE.request primitive. The MLDE-ERASE.confirm primitive returns a status of either SUCCESS, indicating that the erase request was successful, or INVALID_HANDLE, indicating an error.

6.2.2.6.4 Appropriate usage

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

6.2.2.7 Data service message sequence chart

Figure 2 illustrates a sequence of messages necessary for a successful data transfer between two devices.

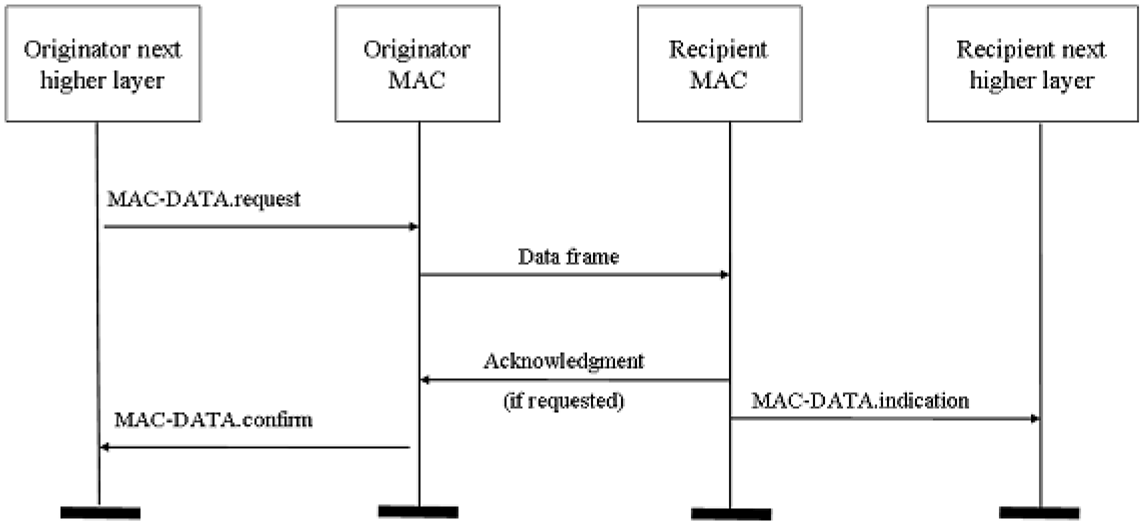


Figure 2 – Message sequence chart describing the MAC data service

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 summarises the primitives supported by the MLME through the MLME-SAP interface. Primitives marked with an (opt) are optional.

Table 6 – Summary of the primitives accessed through the MLME-SAP

Name	Request	Indication	Response	Confirm
MLME-ASSOCIATE	O	O (opt)	O (opt)	O
MLME-DISASSOCIATE	O	O		O
MLME-BEACON-HEARD		O		
MLME-READ-MIB	O			O
MLME-RAP-MANAGEMENT	O (opt)	O (opt)		O (opt)
MLME-BEACON-LOST		O (opt)	O (opt)	
MLME-RESET	O			O
MLME-RX-ON	O			O
MLME-SCAN	O			O
MLME-COMM-RESULT		O		
MLME-WRITE-MIB	O			O
MLME-START	O (opt)			O (opt)
MLME-SYNC	O			
MLME-SYNC-LOSS		O		
MLME-INDIRECT-COMM	O			O

6.2.4 Association primitives

6.2.4.1 Overview

MLME-SAP association primitives define how a device becomes associated with a WMC or WRC. All devices shall provide an interface for the request and confirm association primitives. The indication and response association primitives are optional for a WED.

6.2.4.2 MLME-ASSOCIATE.request

6.2.4.2.1 Function

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

6.2.4.2.2 Semantics of the service primitive

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

```
MLME-ASSOCIATE.request    (
                           LogicalChannel,
                           Co-ordAddrMode,
                           Co-ordMeshId,
                           Co-ordAddress,
                           CapabilityInformation,
                           SecurityEnable
                           )
```

Table 7 specifies the parameters for the MLME-ASSOCIATE.request primitive.

Table 7 – MLME-ASSOCIATE.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY	The logical channel on which to attempt association.
Co-ordAddrMode	Integer	0x02 to 0x03	The WRC addressing mode for this primitive and subsequent MPDU. This value can take one of the following values: 2 = 16-bit short address. 3 = 64-bit extended address.
Co-ordMeshId	Integer	0x0000 to 0xffff	The identifier of the mesh network with which to associate.
Co-ordAddress	Device address	As specified by the Co-ordAddrMode parameter.	The address of the WRC with which to associate.
CapabilityInformation	Bitmap	–	Specifies the operational capabilities of the associating device.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.4.2.3 When generated

This primitive is generated by the next higher layer of an unassociated device and issued to its MLME to request an association with a WRC. If the device wishes to associate with a WRC on a beacon enabled mesh network, the MLME may optionally track the beacon of that WRC prior to issuing this primitive.

6.2.4.2.4 Effect on receipt

On receipt of this primitive, the MLME of an unassociated device first updates *phyCurrentChannel* with the value of the LogicalChannel parameter by issuing the PLME-SET.request primitive and then update *mac-MeshId* with the value of the Co-ordMeshId parameter. The MLME then generates an association request command.

6.2.4.3 MLME-ASSOCIATE.indication

6.2.4.3.1 Function

The MLME-ASSOCIATE.indication primitive is used to indicate the reception of an association request command.

6.2.4.3.2 Semantics of the service primitive

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

```
MLME-ASSOCIATE.indication      (
                                DeviceAddress,
                                CapabilityInformation,
                                SecurityUse,
                                ACLEntry
                               )
```

Table 8 specifies the parameters for the MLME-ASSOCIATE.indication primitive.

Table 8 – MLME-ASSOCIATE.indication parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	extended address(64 bit)	The address of the device requesting association.
CapabilityInformation	Bitmap		The operational capabilities of the device requesting association.
SecurityUse	Bitmap	TRUE or FALSE	An indication of whether the received MAC command frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Bitmap	0x00 to 0x08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame.

6.2.4.3.3 When generated

This primitive is generated by the MLME of the WRC and issued to its next higher layer to indicate the reception of an association request command.

6.2.4.3.4 Effect on receipt

When the next higher layer of a WRC receives this primitive, the WRC determines whether to accept or reject the unassociated device using an algorithm outside the scope of this standard. The next higher layer of the WRC then issues the MLME-ASSOCIATE.response primitive to its MLME.

6.2.4.4 MLME-ASSOCIATE.response

6.2.4.4.1 Function

This primitive is used to initiate a response to an MLME-ASSOCIATE.indication primitive.

6.2.4.4.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-ASSOCIATE.response          (
                                    DeviceAddress,
                                    AssocShortAddress,
                                    Status,
                                    SecurityEnable
                                  )
```

Table 9 specifies the parameters for the MLME-ASSOCIATE.response primitive.

Table 9 – MLME-ASSOCIATE.response parameters

Name	Type	Valid range	Description
DeviceAddress	Device address	64-bit Extended address	The address of the device requesting association.
AssocShortAddress	Integer	0x0000 to 0xffff	The 16-bit short address of a device allocated by the co-ordinator on successful association. This parameter is set to 0xffff if the association was unsuccessful.
Status	Enumeration	–	The status of the association attempt.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.4.4.3 When generated

This primitive is generated by the next higher layer of a WRC and issued to its MLME in order to respond to the MLME-ASSOCIATE.indication primitive.

6.2.4.4.4 Effect on receipt

When the MLME of a WRC receives this primitive, it generates an association response command. The command is sent to the device requesting association using indirect transmission, i.e. the command frame is added to the list of pending transactions stored on the WRC.

6.2.4.5 MLME-ASSOCIATE.confirm

6.2.4.5.1 Function

This primitive is used to inform the next higher layer of the initiating device whether its request to associate was successful or unsuccessful.

6.2.4.5.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-ASSOCIATE.confirm      (
                            AssocShortAddress,
                            Status
                            )
```

Table 10 specifies the parameters for the MLME-ASSOCIATE.confirm primitive.

Table 10 – MLME-ASSOCIATE.confirm parameters

Name	Type	Valid range	Description
AssocShortAddress	Integer	0x0000 to 0xffff	The short device address allocated by the co-ordinator on successful association. This parameter will be equal to 0xffff if the association attempt was unsuccessful.
Status	Enumeration	The value of the status field of the associate response command SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK or INVALID_PARAMETER.	The status of the association attempt.

6.2.4.5.3 When generated

This primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-ASSOCIATE.request primitive. If the request was successful, the status parameter will indicate a successful association, as contained in the status field of the association response command. Otherwise, the status parameter indicates either an error code from the received association response command or an error code of CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_ CHECK or INVALID_PARAMETER.

6.2.4.5.4 Effect on receipt

On receipt of this primitive, the next higher layer of the initiating device is notified of the result of its request to associate with a co-ordinator. If the association attempt was successful, the status parameter will indicate a successful association, as contained in the status field of the associate response command, and the device will be provided with a short address. If this short address is in the range of 0x0000 to 0xfffd, it may be used for communication in the mesh network. If the short address is equal to 0xfffe, the device will use its extended 64-bit address for communication in the mesh network. If the association attempt was unsuccessful, the address will be equal to 0xffff and the status parameter will indicate the error.

6.2.4.6 Association message sequence charts

Figure 3 illustrates the sequence of messages necessary for a device to successfully associate with a mesh network.

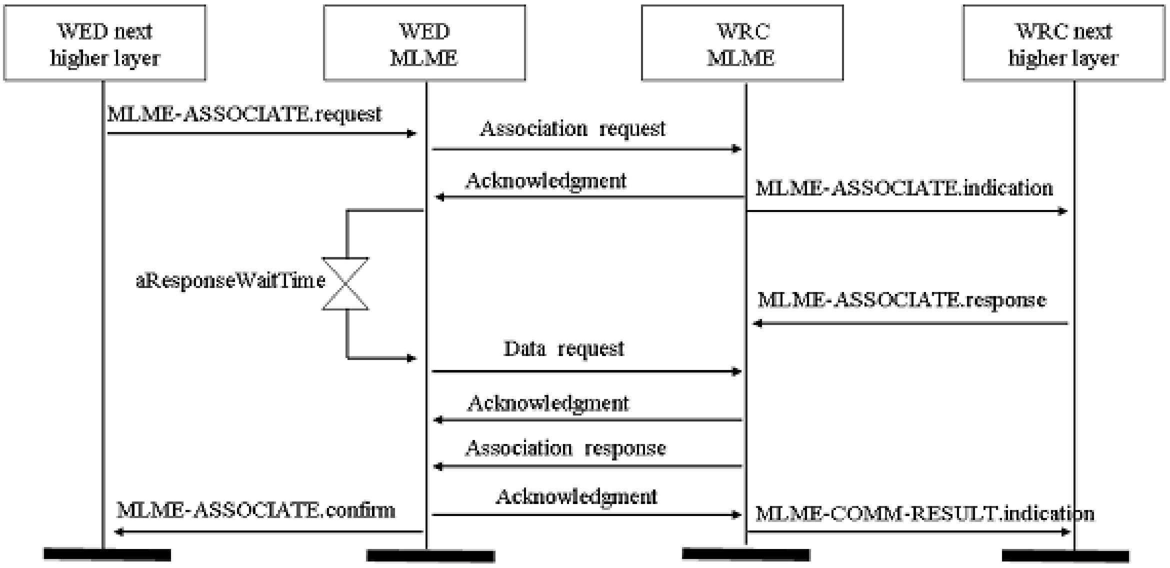


Figure 3 – Message sequence chart for association

6.2.5 Disassociation primitives

6.2.5.1 Overview

These primitives define how a device can disassociate from a mesh network. All devices shall provide an interface for these primitives.

6.2.5.2 MLME-DISASSOCIATE.request

6.2.5.2.1 Function

This primitive is used by an associated device to notify the WRC of its intent to leave the mesh network. It is also used by the WRC to instruct an associated device to leave the mesh network.

6.2.5.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-DISASSOCIATE.request     (
                              DeviceAddress,
                              DisassociateReason,
                              SecurityEnable
                               )
```

Table 11 specifies the parameters for the MLME-DISASSOCIATE.request primitive.

Table 11 – MLME-DISASSOCIATE.request parameters

Name	Type	Valid range	Description
Device Address	Device address	An extended 64-bit, IEEE address.	The address of the device to which to send the disassociation notification command.
DisassociateReason	Integer	0x00 to 0xff	The reason for the disassociation.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.5.2.3 When generated

This primitive is generated by the next higher layer of an associated device and issued to its MLME to request disassociation from the mesh network. It is also generated by the next higher layer of the WRC and issued to its MLME to instruct an associated device to leave the mesh network.

6.2.5.2.4 Effect on receipt

On receipt of this primitive, the MLME generates a disassociation notification command. If the DeviceAddress parameter is equal to *macCo-ordExtendedAddress*, the command will be sent to its WRC in the CAP for a beacon enabled mesh network.

6.2.5.3 MLME-DISASSOCIATE.indication

6.2.5.3.1 Function

This primitive is used to indicate the reception of a disassociation notification command.

6.2.5.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-DISASSOCIATE.indication    (
                                 DeviceAddress,
                                 DisassociateReason,
                                 SecurityUse,
                                 ACLEntry
                                )
```

Table 12 specifies the parameters for the MLME-DISASSOCIATE.indication primitive.

Table 12 – MLME-DISASSOCIATE.indication parameters

Name	Type	Valid range	Description
Device Address	Device address	An extended 64-bit, IEEE address.	The address of the device requesting disassociation.
DisassociateReason	Integer	0x00 to 0xff	The reason for the disassociation.
SecurityUse	Bitmap	TRUE or FALSE	This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Bitmap	0x00 to 0x08	The *macSecurityMode* parameter value from the ACL entry associated with the sender of the data frame.

6.2.5.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer on receipt of a disassociation notification command.

6.2.5.3.4 Effect on receipt

The next higher layer is notified of the reason for the disassociation.

6.2.5.4 MLME-DISASSOCIATE.confirm

6.2.5.4.1 Function

This primitive reports the results of an MLME-DISASSOCIATE.request primitive

6.2.5.4.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-DISASSOCIATE.confirm          (
                                   Status
                                   )
```

Table 13 specifies the parameters for the MLME-DISASSOCIATE.confirm primitive.

Table 13 – MLME-DISASSOCIATE.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the disassociation attempt.

6.2.5.4.3 When generated

This primitive is generated by the initiating MLME and issued to its next higher layer in response to an MLME-DISASSOCIATE.request primitive. This primitive returns a status of either SUCCESS, indicating that the disassociation request was successful, or an error code of TRANSACTION_OVERFLOW, TRANSACTION_ EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK or INVALID_PARAMETER.

6.2.5.5 Disassociation message sequence charts

Figure 4 illustrates the sequence of messages necessary for successful disassociation from a mesh network. The originating device may either be a device or the WRC to which the device has associated.

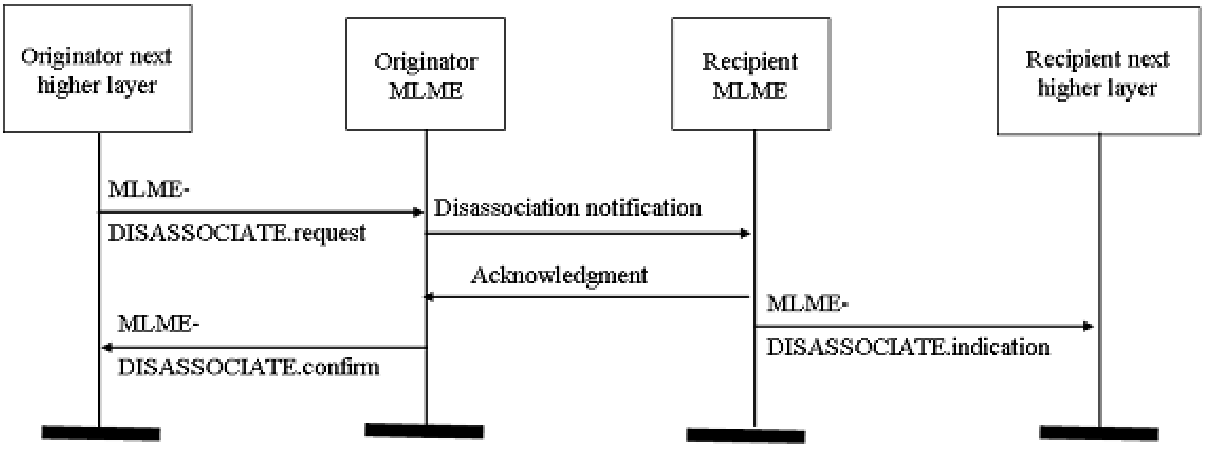

Originator next higher layer
Originator MLME
Recipient MLME
Recipient next higher layer
MLME-DISASSOCIATE.request
Disassociation notification
Acknowledgment
MLME-DISASSOCIATE.indication
MLME-DISASSOCIATE.confirm

Figure 4 – Message sequence chart for disassociation

6.2.6 Beacon notification primitive

6.2.6.1 General

This primitive defines how a device may be notified when a beacon is received during normal operating conditions. All devices shall provide an interface for this primitive.

6.2.6.2 MLME-BEACON-HEARD.indication

6.2.6.2.1 Function

This primitive is used to send parameters contained within a beacon frame received by the MAC layer to the next higher layer. The primitive also sends a measure of the link quality and the time the beacon frame was received.

6.2.6.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-BEACON-HEARD.indication    (
                                BSN,
                                PANDescriptor,
                                PendAddrSpec,
                                AddrList,
                                sduLength,
                                sdu
                                )
```

Table 14 specifies the parameters for the MLME-BEACON-HEARD.indication primitive, while Table 15 describes the elements of the PANDescriptor type.

Table 14 – MLME-BEACON-NOTIFY.indication parameters

Name	Type	Valid range	Description
BSN	Integer	0x00 to 0xff	The beacon sequence number.
PANDescriptor	PANDescriptor value	-	The PAN descriptor for the received beacon.
PendAddrSpec	Bitmap		The beacon pending address specification.
AddrList	List of device addresses	–	The list of addresses of the devices for which the beacon source has data.
sduLength	Integer	0 to aMaxBeaconPayloadLength	The number of octets contained in the beacon payload of the beacon frame received by the MAC sublayer.
sdu	Set of octets	–	The set of octets comprising the beacon payload to be transferred from the MAC sublayer entity to the next higher layer.

Table 15 – Elements of mesh descriptor

Name	Type	Valid range	Description
Co-ordAddrMode	Integer	0x02 to 0x03	The co-ordinator addressing mode corresponding to the received beacon frame. This value can take one of the following values: 2 = 16-bit short address, 3 = 64-bit extended address
Co-ordMeshId	Integer	0x0000 to 0xffff	The Mesh identifier of the co-ordinator as specified in the received beacon frame.
Co-ordAddres	Device address	As specified by the Co-ordAddrMode parameter.	The address of the co-ordinator as specified in the received beacon frame.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY.	The current logical channel occupied by the network.
SuperframeSpec	Bitmap		The superframe specification as specified in the received beacon frame.
RAPPermit	Boolean	TRUE or FALSE	TRUE if the beacon is from a WMC which is accepting RAP requests.
LinkQuality	Integer	0x00 to 0xff	The link quality at which the network beacon was received. Lower values represent lower link quality.
TimeStamp	Integer	0x000000 to 0xffffff	The time at which the beacon frame was received, in symbols. This value is equal to the timestamp taken when the beacon frame was received. The precision of this value is a minimum of 20-bit, with the lowest four bit being the least significant.
SecurityUse	Boolean	True or FALSE	An indication of whether the received beacon frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0x00 to 0x08	The macSecurityMode parameter value from the ACL entry associated with the sender of the data frame.
SecurityFailure	Boolean	TRUE or FALSE	The parameter is set to TRUE if there was an error in the security processing of the frame. Otherwise, the parameter is set to FALSE.
BTTS	Integer	0x00 to 0xff	The time slot in which a device transmits its beacon.
maxBeaconTxSize	Integer	0x00 to 0xff	The maximum size of the beacons transmitted in a superframe. including BeaconTxMargin.
BTTSL	BYTE	0x00 to 0xff	The length of the beacon transmission time slot. This value can be calculated by adding the size of the beacon and the BeaconTxMargin.
Depth	BYTE	0x00 to 0xff	The depth of the device from the WMC.
NAA	BYTE	0x0000 to 0xffff	The short address that is delivered to the entire mesh nodes using the beacon payload in such a way that a newly joining WiBEEM device shall decide to use as its short address. This address is determined by the NWK layer.
Profile ID	BYTE	0x00 to 0xff	The number that specifies its corresponding application. A different application program is running based on this value.
PQPL	BYTE	0x00 to 0xff	The length of PQP. This field is generated when the value of PQP enable in the superframe specification of beacon frame is 1. This field represents the value when the mesh network decides to use the QoS capability.

6.2.6.2.3 When generated

This primitive is generated by the MLME and issued to its next higher layer upon receipt of a beacon frame either when *macAutoRequest* is set to FALSE or the beacon frame contains one or more octets of payload.

6.2.6.2.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of the arrival of a beacon frame at the MAC layer.

6.2.7 Primitives for reading PIB attributes

6.2.7.1 General

These primitives define how to read values from the PIB. All devices shall provide an interface for these primitives.

6.2.7.2 MLME-READ-MIB.request

6.2.7.2.1 Function

This primitive requests information about a given MIB attribute.

6.2.7.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-READ-MIB.request    (
                              MIBAttribute
                         )
```

Table 16 specifies the parameters for the MLME-GET.request primitive.

Table 16 – MLME-READ-MIB.request parameters

Name	Type	Valid range	Description
MIBAttribute	Integer	–	The identifier of the MIB attribute to read.

6.2.7.2.3 When generated

This primitive is generated by the next higher layer and issued to its MLME to obtain information from the MIB.

6.2.7.2.4 Effect on receipt

On receipt of this primitive, the MLME attempts to retrieve the requested MIB attribute from its database. If the identifier of the MIB attribute is not found in the database, the MLME will issue the MLM-READ-MIB.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the requested MIB attribute is successfully retrieved, the MLME will issue the MLME-READ-MIB.confirm primitive with a status of SUCCESS.

6.2.7.3 MLME-READ-MIB.confirm

6.2.7.3.1 Function

This primitive reports the results of an information request from the MAC PIB.

6.2.7.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-READ-MIB.confirm  (
                       Status,
                       MIBAttribute,
                       MIBAttributeValue
                       )
```

Table 17 specifies the parameters for the MLME-READ-MIB.confirm primitive.

Table 17 – MLME-READ-MIB.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE	The result of the request for MIB attribute information.
MIBAttribute	Integer	–	The identifier of the MIB attribute that was read.
MIBAttributeValue	Various	–	The identifier of the MIB attribute that was read.

6.2.7.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer in response to an MLME-READ-MIB.request primitive. This primitive returns a status of either SUCCESS, indicating that the request to read a MAC PIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE.

6.2.7.3.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of the results of its request to read a MIB attribute. If the request to read a MIB attribute was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.2.8 RAP management primitives

6.2.8.1 General

These primitives define how RAP (Reservation-based Access Period) are requested and maintained. Devices wishing to use these primitives and RAP in general will already be tracking the beacons of their WMC.

6.2.8.2 MLME-RAP-MANAGEMENT.request

6.2.8.2.1 Function

This primitive allows a device to send a request to the WMC to allocate a new RAP or to deallocate an existing RAP.

6.2.8.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-RAP-MANAGEMENT.request (
                            RAPCharacteristics,
                            SecurityEnable
                            )
```

Table 18 specifies the parameters for the MLME- RAP-MANAGEMENT.request primitive.

Table 18 – MLME- RAP-MANAGEMENT.request parameters

Name	Type	Valid range	Description
RAPCharacteristics	RAP characteristics	–	The characteristics of the RAP request.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.8.2.3 When generated

This primitive is generated by the next higher layer and issued to its MLME to request the allocation of a new RAP or to request the deallocation of an existing RAP. The RAP characteristics parameter specifies whether the request is for the allocation of a new RAP or for the deallocation of an existing RAP. If the characteristics type field of the RAP characteristics parameter is equal to 1, the remaining fields in the RAP characteristics will specify the desired characteristics of the new RAP, i.e., its length and direction. If the characteristics type field of the RAP characteristics parameter is equal to 0, the remaining fields in the RAP characteristics will specify the length and direction of the RAP that the device wishes to reallocate.

6.2.8.2.4 Effect on receipt

On receipt of this primitive, the MLME of a device attempts to generate a RAP request command with the information contained in this primitive and, if successful, sends it to the WMC.

6.2.8.3 MLME-RAP-MANAGEMENT.confirm

6.2.8.3.1 Function

This primitive reports the results of a request to allocate a new RAP or deallocate an existing RAP.

6.2.8.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-RAP-MANAGEMENT.confirm     (
                                 RAPCharacteristics,
                                 Status
                                 )
```

Table 19 specifies the parameters for the MLME-RAP-MANAGEMENT.confirm primitive.

Table 19 – MLME-RAP-MANAGEMENT.confirm parameters

Name	Type	Valid range	Description
RAPCharacteristics	RAP Characteristics	–	The characteristics of RAP request.
Status	Enumeration	SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY FAILED_SECURITY_CHECK, INVALID_PARAMETER.	The status of the RAP request.

6.2.8.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer in response to a previously issued MLME-RAP-MANAGEMENT.request primitive. If the request to allocate or deallocate a RAP was successful, this primitive will return a status of SUCCESS and the characteristics type field of the RAP characteristics parameter will have the value of 1 or 0, respectively. If the WMC denied the request, the primitive will return a status of DENIED. Otherwise, the status parameter indicates an error code of NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_ CHECK or INVALID_PARAMETER.

6.2.8.3.4 Effect on receipt

On receipt of this primitive the next higher layer is notified of the result of its request to allocate or deallocate a RAP. If the request was successful, the status parameter will indicate a successful RAP operation. Otherwise, the status parameter will indicate the error.

6.2.8.4 MLME- RAP-MANAGEMENT.indication

6.2.8.4.1 Function

This primitive indicates that a RAP has been allocated or that a previously allocated RAP has been deallocated.

6.2.8.4.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-RAP-MANAGEMENT.indication      (
                                    DevAddress,
                                    RAPCharacteristics,
                                    SecurityUse,
                                    ACLEntry
                                    )
```

Table 20 specifies the parameters for the MLME- RAP-MANAGEMENT.indication primitive.

Table 20 – MLME-RAP-MANAGEMENT.indication parameters

Name	Type	Valid range	Description
Device Address	Device address	0x0000 to 0xfffd	The short address of the device that has been allocated or deallocated a RAP.
RAPCharacteristics	RAP characteristics	–	The characteristics of the RAP.
SecurityUse	Boolean	TRUE or FALSE	An indication whether the received frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	integer	0x00 to 0x08	The macSecurityMode parameter value from the ACL entry associated with the sender of the data frame.

6.2.8.4.3 When generated

This primitive is generated by the MLME of the WMC to its next higher layer whenever a RAP is allocated or deallocated following the reception of a RAP request command by the MLME. The MLME of the WMC also generates this primitive when a RAP deallocation is initiated by the WMC itself. The characteristics type field in the RAP characteristics parameter will be equal to 1 if a RAP has been allocated or 0 if a RAP has been deallocated.

6.2.8.4.4 Effect on receipt

On receipt of this primitive the next higher layer is notified of the allocation or deallocation of a RAP.

6.2.8.5 RAP management message sequence charts

Figure 5 and Figure 6 illustrate the sequence of messages necessary for successful RAP management. The first depicts the message flow for the case in which the device initiates the RAP allocation. The second depicts the message flow for the two cases for which a RAP deallocation occurs, firstly by a device a) and secondly by the WMC b), see Figure 6.

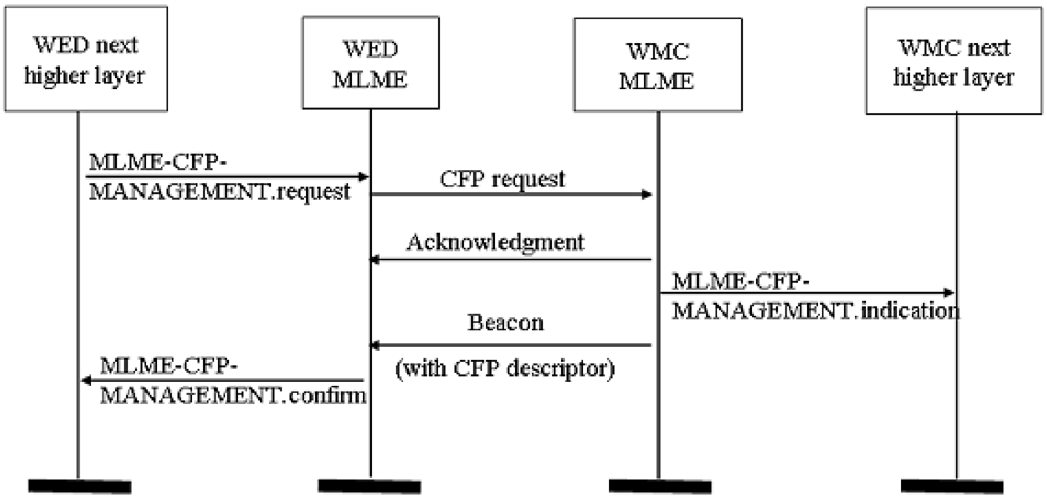


Figure 5 – Message sequence chart for RAP allocation initiated by a device

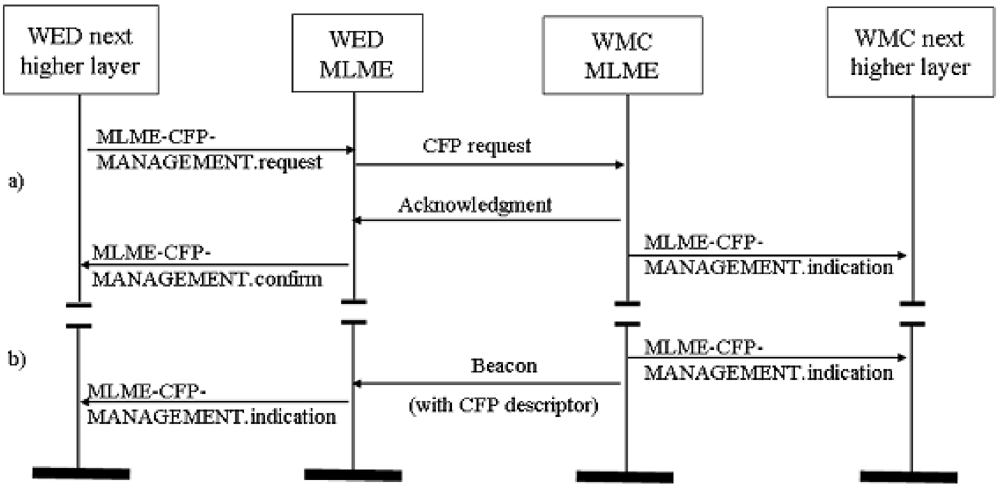

WED next higher layer
WED MLME
WMC MLME
WMC next higher layer
a)
MLME-CFP-MANAGEMENT.request
CFP request
Acknowledgment
MLME-CFP-MANAGEMENT.confirm
MLME-CFP-MANAGEMENT.indication
b)
MLME-CFP-MANAGEMENT.indication
Beacon
(with CFP descriptor)
MLME-CFP-MANAGEMENT.indication

Figure 6 – Message sequence chart for RAP deallocation initiated by a device and the WMC

6.2.9 Primitives for orphan notification

6.2.9.1 Overview

MLME-SAP orphan notification primitives define how a WRC can issue a notification of an orphaned device. These orphan notification primitives are optional for a WED.

6.2.9.2 MLME-BEACON-LOST.indication

6.2.9.2.1 Function

The MLME-ORPHAN.indication primitive allows the MLME of a WRC to notify the next higher layer of the presence of an orphaned device.

6.2.9.2.2 Semantics of the service primitive

The semantics of the MLME-BEACON-LOST.indication primitive are as follows:

```
MLME-BEACON-LOST.indication     (
                                   OrphanAddress,
                                   SecurityUse,
                                   ACLEntry
                                )
```

Table 21 specifies the parameters for the MLME-BEACON-LOST.indication primitive.

Table 21 – MLME-BEACON-LOST.indication parameters

Name	Type	Valid range	Description
OrphanAddress	Device Address	Extended 64 bit IEEE address	The address of the orphaned device
SecurityUse	Boolean	TRUE or FALSE	An indication of whether the received MAC command frame is using security. This value is set to TRUE if the security enable subfield was set to 1 or FALSE if the security enabled subfield was set to 0.
ACLEntry	Integer	0x00 to 0x08	The macSecurityMode parameter value from the ACL entry associated with the sender of the data frame. This value is set to 0x08 if the sender of the data frame was not found in the ACL.

6.2.9.2.3 When generated

The MLME-ORPHAN.indication primitive is generated by the MLME of a WRC and issued to its next higher layer on receipt of an orphan notification command.

6.2.9.2.4 Effect on receipt

The effect on receipt of the MLME-ORPHAN.indication primitive is that the next higher layer is notified of the orphaned device. The next higher layer then determines whether the device was previously associated and issues the MLME-ORPHAN.response primitive to the MLME with its decision.

6.2.9.3 MLME-BEACON-LOST.response

6.2.9.3.1 Function

The MLME-ORPHAN.response primitive allows the next higher layer of a WRC to respond to the MLME-ORPHAN.indication primitive.

6.2.9.3.2 Semantics of the primitives

The semantics of MLME-BEACON-LOST.response primitive are as follows:

```
MLME-BEACON-LOST.response  (

                              OrphanAddress,
                              ShortAddress,
```

AssociatedMember,
SecurityEnable
)

Table 22 specifies the parameters for the of MLME-BEACON-LOST.response primitive.

Table 22 – MLME-BEACON-LOST.indication parameters

Name	Type	Valid range	Description
OrphanAddress	Device address	Extended 64 bit IEEE address	The address of the orphaned device.
ShortAddress	Integer	0x0000 to 0xffff	The short address allocated to the orphaned device if it is associated with this WRC. The special short address 0xfffe indicates that no short address was allocated, and the device will use its 64-bit extended address in all communications. If the device was not associated with this WRC, this field will contain the value 0xffff and shall be ignored on receipt.
AssociatedMember	Boolean	TRUE or FALSE	TRUE if the orphaned device is associated with this WRC, FALSE otherwise.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.9.3.3 When generated

The MLME-ORPHAN.response primitive is generated by the next higher layer and issued to its MLME when it reaches a decision about whether the orphaned device indicated in the MLME-ORPHAN.indication primitive is associated.

6.2.9.3.4 Effect on receipt

If the AssociatedMember parameter is set to TRUE, the orphaned device is associated with the WRC. In this case, the MLME generates and sends the WRC realignment command to the orphaned device containing the value of the ShortAddress field. This command is sent in the CAP if the WRC is on a beacon-enabled mesh network or immediately otherwise. If the AssociatedMember parameter is set to FALSE, the orphaned device is not associated with the WRC and this primitive will be ignored.

6.2.9.4 Beacon lost (orphan notification) message sequence chart

Figure 7 illustrates the sequence of messages necessary for a WRC to issue a notification of an orphaned device.

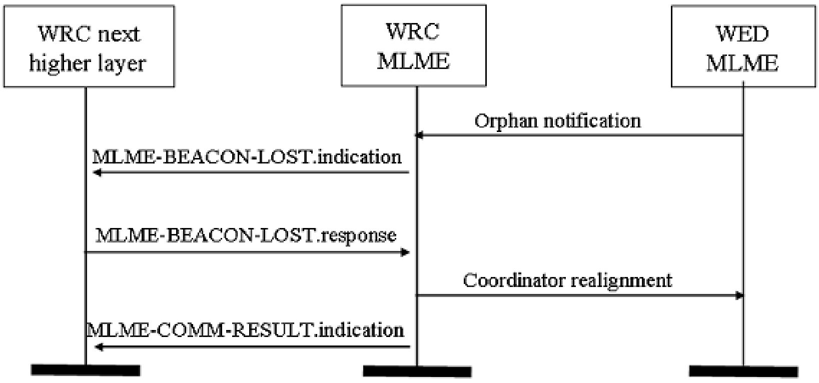

WRC next higher layer
WRC MLME
WED MLME
Orphan notification
MLME-BEACON-LOST.indication
MLME-BEACON-LOST.response
Coordinator realignment
MLME-COMM-RESULT.indication

Figure 7 – Beacon lost (orphan notification) message sequence chart

6.2.10 Primitives for resetting the MAC layer

6.2.10.1 General

MLME-SAP reset primitives specify how to reset the MAC sublayer to its default values. All devices shall provide an interface for these reset primitives.

6.2.10.2 MLME-RESET.request

6.2.10.2.1 Function

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

6.2.10.2.2 Semantics of the service primitive

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

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

Table 23 specifies the parameter for the MLME-RESET.request primitive.

Table 23 – MLME-RESET.request parameters

Name	Type	Valid range	Description
SetDefaultMIB	Boolean	TRUE or FALSE	If TRUE, the MAC sublayer is reset and all MAC MIB attributes are set to their default values. If FALSE, the MAC sublayer is reset but all MAC MIB attributes retain their values prior to the generation of the MLME-RESET.request primitive.

6.2.10.2.3 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 sublayer to its initial conditions. The MLME-RESET.request primitive is issued prior to the use of the MLME-START.request or the MLME-ASSOCIATE.request primitives.

6.2.10.2.4 Effect on receipt

On receipt of the MLME-RESET.request primitive, the MLME issues the PLME-SET-TRX-STATE.request primitive with a state of FORCE_TRX_OFF. On receipt of the PLME-SET-TRXSTATE.confirm primitive, the MAC sublayer is then set to its initial conditions, clearing all internal variables to their default values. If the SetDefaultPIB parameter is set to TRUE, the MAC PIB attributes are set to their default values. The MLME-RESET.confirm primitive with a status of SUCCESS is issued on completion.

6.2.10.3 MLME-RESET.confirm

6.2.10.3.1 Function

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

6.2.10.3.2 Semantics of the service primitive

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

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

Table 24 specifies the parameter for the MLME-RESET.confirm primitive.

Table 24 – MLME-RESET.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS or DISABLE_TRX_FAILURE	The result of the reset operation.

6.2.10.3.3 When generated

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

6.2.10.3.4 Effect on receipt

On receipt of the MLME-RESET.confirm primitive, the next higher layer is notified of its request to reset the MAC sublayer. This primitive returns a status of SUCCESS indicating that the request to reset the MAC sublayer was successful.

6.2.11 Primitives for specifying the receiver enable time

6.2.11.1 Overview

MLME-SAP receiver state primitives define how a device can enable or disable the receiver at a given time. These receiver state primitives are optional.

6.2.11.2 MLME-RX-ON.request

6.2.11.2.1 Function

The MLME-RX-ENABLE.request primitive allows the next higher layer to request that the receiver is either enabled for a finite period of time or disabled.

6.2.11.2.2 Semantics of the service primitive

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

```
MLME-RX-ON.request        (
                          DeferPermit,
                          RxOnTime,
                          RxOnDuration
                          )
```

Table 25 specifies the parameters for the MLME-RX-ON.request primitive.

Table 25 – MLME-RX-ON.request parameters

Name	Type	Valid range	Description
DeferPermit	Boolean	TRUE or FALSE	TRUE if the receiver enable can be deferred until during the next superframe if the requested time has already passed. FALSE if the receiver enable is only to be attempted in the current superframe. This parameter is ignored for non-beacon enabled PANs.
RxOnTime	Integer	0x000000 to 0xffffff	The number of symbols from the start of the superframe before the receiver is to be enabled. The precision of this value is a minimum of 20 bit, with the lowest four bit being the least significant. This parameter is ignored for non-beacon enabled PANs.
RxOnDuration	Integer	0x000000 to 0xffffff	The number of symbols for which the receiver is to be enabled.

6.2.11.2.3 When generated

The MLME-RX-ENABLE.request primitive is generated by the next higher layer and issued to the MLME to enable the receiver for a fixed duration, at a time relative to the start of the current or next superframe on a beacon-enabled PAN. This primitive may also be generated to cancel a previously generated request to enable the receiver. The receiver is enabled or disabled exactly once per primitive request.

6.2.11.2.4 Effect on receipt

The MLME will treat the request to enable or disable the receiver as secondary to other responsibilities of the device (e.g., CFP, WRC beacon tracking, beacon transmissions). When the primitive is issued to enable the receiver, the device will enable its receiver until either the device has a conflicting responsibility or the time specified by RxOnDuration has expired. In the case of a conflicting responsibility, the device will interrupt the receive operation. After the completion of the interrupting operation, the RxOnDuration will be checked to determine whether the time has expired. If so, the operation is complete. If not, the receiver is re-enabled until either the device has another conflicting responsibility or the time specified by RxOnDuration has expired. When the primitive is issued to disable the receiver, the device will disable its receiver unless the device has a conflicting responsibility.

6.2.11.3 MLME-RX-ON.confirm

6.2.11.3.1 Function

The MLME-RX-ENABLE.confirm primitive reports the results of the attempt to enable or disable the receiver.

6.2.11.3.2 Semantics of the service primitive

The semantics of the MLME-RX-ON.confirm primitive are as follows:

```
MLME-RX-ON.confirm        (
                          Status
                          )
```

Table 26 specifies the parameter for the MLME-RX-ON.confirm primitive.

Table 26 – MLME-RX-ON.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, TX_ACTIVE, OUT_OF_CAP, or INVALID_PARAMETER	The result of the receiver enable request.

6.2.11.3.3 When generated

The MLME-RX-ENABLE.confirm primitive is generated by the MLME and issued to its next higher layer in response to an MLME-RX-ENABLE.request primitive.

6.2.11.3.4 Effect on receipt

On receipt of the MLME-RX-ENABLE.confirm primitive, the next higher layer is notified of its request to enable or disable the receiver. This primitive returns a status of either SUCCESS, if the request to enable or disable the receiver was successful, or the appropriate error code.

6.2.12 Primitives for channel scanning

6.2.12.1 Overview

MLME-SAP scan primitives define how a device can determine the energy usage or the presence or absence of PANs in a communications' channel. All devices shall provide an interface for these scan primitives.

6.2.12.2 MLME-SCAN.request

6.2.12.2.1 Function

The MLME-SCAN.request primitive is used to initiate a channel scan over a given list of channels. A device can use a channel scan to measure the energy on the channel, search for the WRC with which it is associated, or search for all WRCs transmitting beacon frames within the POS of the scanning device.

6.2.12.2.2 Semantics of the service primitive

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

```
MLME-SCAN.request    (
                     ScanType,
                     ScanChannels,
                     ScanDuration
                     )
```

Table 27 specifies the parameters for the MLME-SCAN.request primitive.

Table 27 – MLME-SCAN.request parameters

Name	Type	Valid range	Description
ScanType	Integer	0x00 to 0x03	Indicates the type of scan performed: 0x00 = energy detection scan (WRC only), 0x01 = active scan (WRC only), 0x02 = passive scan, 0x03 = orphan scan.
ScanChannels	Bitmap	32bit field	The five most significant bits (b_{27}, ... , b_{31}) are reserved. The 27 least significant bits (b_0, b_1, ..., b_{26}) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 valid channels.
ScanDuration	Integer	0 to 14	A value used to calculate the length of time to spend scanning each channel for energy detection, active and passive scans. This parameter is ignored for orphan scans. The time spent scanning each channel is (*aBaseSuperframeDuration* * (2^n + 1)) symbols, where *n* is the value of the ScanDuration parameter.

6.2.12.2.3 When generated

The MLME-SCAN.request primitive is generated by the next higher layer and issued to its MLME to initiate a channel scan to search for activity within the POS of the device. This primitive can be used to perform an ED scan to determine channel usage, an active or passive scan to locate beacon frames containing any PAN identifier, or an orphan scan to locate a PAN to which the device is currently associated. All devices shall be capable of performing passive scans and orphan scans. ED scans and active scans are optional for a WED.

6.2.12.2.4 Effect on receipt

If the MLME receives the MLME-SCAN.request primitive while performing a previously initiated scan operation, it issues the MLME-SCAN.confirm primitive with a status of SCAN_IN_PROGRESS. Otherwise, the MLME initiates a scan in all channels specified in the ScanChannels parameter.

6.2.12.3 MLME-SCAN.confirm

6.2.12.3.1 Function

The MLME-SCAN.confirm primitive reports the result of the channel scan request.

6.2.12.3.2 Semantics of the service primitive

The semantics of the MLME-SCAN.confirm primitive are as follows:

```
MLME-SCAN.confirm (
                    Status,
                    ScanType,
                    ChannelPage,
                    UnscannedChannels,
                    ResultListSize,
                    EnergyDetectList,
                    PANDescriptorList
                    )
```

Table 28 specifies the parameters for the MLME-SCAN.confirm primitive.

Table 28 – MLME-SCAN.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, NO_BEACON, or INVALID_PARAMETER	The status of the scan request.
ScanType	Integer	0x00 to 0x03	Indicates if the type of scan performed is: 0x00 = energy detection scan (WRC only), 0x01 = active scan (WRC only), 0x02 = passive scan, 0x03 = orphan scan.
UnscannedChannels	Bitmap	32 bit field	Indicates which channels given in the request were not scanned (1 = not scanned, 0 = scanned or not requested). This parameter is only valid for passive or active scans.
ResultListSize	Integer	Implementation specific	The number of elements returned in the appropriate result lists. This value is 0 for the result of an orphan scan.
EnergyDetectList	List of integers	0x00 to 0xff for each integer	The list of energy measurements, one for each channel searched during an energy detection scan. This parameter is null for active, passive and orphan scans.
PANDescriptorList	List of PANDescriptor values		The list of PAN descriptors, one for each beacon found during an active or passive scan. This parameter is null for energy detection and orphan scans.

6.2.12.3.3 When generated

The MLME-SCAN.confirm primitive is generated by the MLME and issued to its next higher layer when the channel scan initiated with the MLME-SCAN.request primitive has completed. If the MLMESCAN.request primitive requested an active, passive, or orphan scan, the EnergyDetectList parameter will be null. If the MLME-SCAN.request primitive requested an ED or orphan scan, the PANDescriptorList parameter will be null; this is also the case if the MLME-SCAN.request primitive requested an active or passive scan with *macAutoRequest* set to FALSE. If the MLME-SCAN.request primitive requested an orphan scan, the ResultListSize parameter will be set to zero.

6.2.12.3.4 Effect on receipt

On receipt of the MLME-SCAN.confirm primitive, the next higher layer is notified of the results of the scan procedure. If the requested scan was successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.2.13 Communication status primitive

6.2.13.1 Overview

The MLME-SAP communication status primitive defines how the MLME communicates to the next higher layer about transmission status, when the transmission was instigated by a response primitive, and about security errors on incoming packets. All devices shall provide an interface for this communication status primitive.

6.2.13.2 MLME-COMM-RESULT.indication

6.2.13.2.1 Function

This primitive allows the MLME to indicate a communications status.

6.2.13.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-COMM-RESULT.indication    (
                                MeshID,
                                SrcAddrMode,
                                SrcAddr,
                                DstAddrMode,
                                DstAddr,
                                Status
                                )
```

Table 29 specifies the parameters for the MLME-COMM- RESULT.indication primitive.

Table 29 – MLME-COMM-RESULT.indication parameters

Name	Type	Valid range	Description
MeshID	integer	0x0000 to 0xffff	The 16-bit PAN identifier of the device from which the frame was received or to which the frame was being sent.
SrcAddrMode	integer	0x00 to 0x03	The source addressing mode for this primitive. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
SrcAddr	Device address	SrcAddrMode parameters	The individual device address of the entity from which the frame causing the error originated.
DstAddrMode	Integer	0x00 to 0x03	The destination addressing mode for this primitive. This value can take one of the following values: 0x00 = no address (addressing fields omitted). 0x01 = reserved. 0x02 = 16-bit short address. 0x03 = 64-bit extended address.
DstAddr	Device address	DstAddrMode parameters	The individual device address of the device for which the frame was intended.
Status	Enumeration	SUCCESS. TRANSACTION_OVERFLOW TRANSACTION_EXPIRED CHANNEL_ACCESS_FAILURE NO_ACK UNAVAILABLE_KEY FRAME_TOO_LONG FAILED_SECURITY_CHECK or INVALID_PARAMETER	The communications status.

6.2.13.2.3 When generated

This primitive is generated by the MLME and issued to its next higher layer either following a transmission instigated through response primitive or on receipt of a frame which generates an error in its secure processing.

6.2.13.2.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of the communication status of a transmission or that an error has occurred during the secure processing of incoming frame.

6.2.14 Primitives for writing MAC PIB attributes

6.2.14.1 Overview

These primitives define how MAC PIB attributes may be written. All devices shall provide an interface for these primitives.

6.2.14.2 MLME-WRITE-MIB.request

6.2.14.2.1 Function

This primitive attempts to write the given value to the indicated MAC PIB attribute.

6.2.14.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-WRITE-MIB.request  (
                         MIBAttribute,
                         MIBAttributeValue
                         )
```

Table 30 specifies the parameters for MLME-WRITE-MIB.request primitive.

Table 30 – MLME-WRITE-MIB.request parameters

Name	Type	Valid range	Description
MIBAttribute	Integer	–	The identifier of the MAC PIB attribute to write.
MIBAttributeValue	Various	–	The value to write to the indicated MAC PIB attribute.

6.2.14.2.3 When generated

This primitive is generated by the next higher layer and issued to its MLME to write the indicated MAC PIB attribute.

6.2.14.2.4 Effect on receipt

On receipt of this primitive, the MLME attempts to write the given value to the indicated MAC PIB attribute in its database. If the PIBAttribute parameter specifies an attribute that is not found in the database, the MLME will issue the MLME-SET.confirm primitive with a status of UNSUPPORTED_ATTRIBUTE. If the PIBAttributeValue parameter specifies a value that is out of the valid range for the given attribute, the MLME will issue the MLME-SET.confirm primitive with a status of INVALID_PARAMETER. If the requested MAC PIB attribute is successfully written, the MLME will issue the MLME-SET.confirm primitive with a status of SUCCESS.

6.2.14.3 MLME-WRITE-MIB.confirm

6.2.14.3.1 Function

This primitive reports the results of an attempt to write a value to a MAC PIB attribute.

6.2.14.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-WRITE-MIB.confirm (
                         Status,
                         MIBAttribute
                        )
```

Table 31 specifies the parameters for the MLME-WRITE-MIB.confirm primitive.

Table 31 – MLME-WRITE-MIB.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE, or INVALID_PARAMETER	The result of the request to write the MAC PIB attribute.
MIBAttribute	Integer	–	The identifier of the MAC PIB attribute that was written.

6.2.14.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer in response to an MLME-SET.request primitive. This primitive returns a status of either SUCCESS, indicating that the requested value was written to the indicated MAC PIB attribute, or an error code of UNSUPPORTED_ATTRIBUTE or INVALID_PARAMETER.

Effect on receipt of this primitive, the next higher layer is notified of the result of its request to set the value of a MAC PIB attribute. If the requested value was written to the indicated MAC PIB attribute, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.2.15 Primitives for updating the superframe configuration

6.2.15.1 Overview

These primitives define how an WRC can request to start using a new superframe configuration in order to either initiate a PAN, begin transmitting beacons on an already existing PAN, facilitating device discovery, or to stop transmitting beacons. These primitives are optional for a WED.

6.2.15.2 MLME-START.request

6.2.15.2.1 Function

This primitive makes a request for the device to start using a new superframe configuration.

6.2.15.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-START.request   (
                      MeshId,
                      LogicalChannel,
                      BeaconOrder,
                      SuperframeOrder,
                      PANWRC,
                      BatteryLifeExtension,
                      Co-ordRealignment,
                      SecurityEnable,
                      My_BTTS,
                      BTTSL,
                      MaxBeaconTxNumber,
                      Profile ID
                      )
```

Table 32 specifies the parameters for the MLME-START.request primitive.

Table 32 – MLME-START.request parameters

Name	Type	Valid range	Description
MeshId	integer	0x0000-0xffff	The mesh identifier to be used by the beacon.
LogicalChannel	integer	logical channel range	The logical channel on which to start transmitting beacons.
BeaconOrder	integer	0 to 15	This specifies how often the beacon is to be transmitted. The BeaconOrder, *BO*, and the beacon interval, *BI*, are related as follows: for 0≤BO≤14, BI = aBaseSuperframeduration * 2^{BO} symbols. If *BO*= 15, the WRC will not transmit a beacon and the SuperframeOrder parameter value is ignored.
SuperframeOrder	integer	0 to BO or 15	This specifies the length of the active portion of the superframe, including the beacon frame. The SuperframeOrder, *SO*, and the superframe duration, *SD*, are related as follows: for 0≤SO≤BO≤14, SD=aBaseSuperFrameduration *2^{so} symbols. If SO = 15, the superframe will not be active after the beacon.
PANWRC	Boolean	TRUE or FALSE	If this value is TRUE, the device will become the PAN WRC of a new PAN. If this value is FALSE, the device will begin transmitting beacons on the PAN with which it is associated.
BatteryLifeExtension	Boolean	TURE or FALSE	If this value is TRUE, the receiver of the beaconing device is disabled *mac-BattLifeExtPeriods* full backoff periods after the inter-frame space (IFS) period of the beacon frame. If this value is FALSE, the receiver of the beaconing device remains enabled for the entire CAP.
Co-ordRealignment	Boolean	TRUE or FALSE	TRUE if a WRC realignment command is to be transmitted prior to changing the superframe configuration or FALSE otherwise.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for beacon transmissions or FALSE otherwise.
My BTTS	Integer	0x00 to 0xff	Beacon Transfer offset parameter with BOP
BTTSL	BYTE	0x00 to 0xff	Beacon size + TxBeaconMargin. Calculated BOP sized.
MaxBeaconTxNumber	BYTE	0x00 to 0xff	Maximum Beacon Number with BOP
Profile ID	BYTE	0x00 to 0xff	Application identifier

6.2.15.2.3 When generated

This primitive is generated by the next higher layer and issued to its MLME to request that a device start using a new superframe configuration.

6.2.15.2.4 Effect on receipt

If this primitive is received when *macShortAddress* is set to 0xffff, the MLME will issue the MLME-START.confirm primitive with a status of NO_SHORT_ADDRESS. On receipt of this primitive, the MLME sets *macBeaconOrder* to the value of the BeaconOrder parameter. If *macBeaconOrder* is equal to 15, the MLME will also set *macSuperframeOrder* to 15. In this case, this primitive configures a beaconless PAN. If *macBeaconOrder* is less than 15, the MLME will set *macSuperframe-Order* to the value of the SuperframeOrder parameter. In the case where the PANWRC parameter is set to TRUE, the MLME updates *macMeshId* with the value of the MeshId parameter and *phyCurrentChannel* with the value of the LogicalChannel parameter, by issuing the PLME-SET.request primitive. In the case where the PANWRC parameter is set to FALSE, the MLME ignores the MeshId and LogicalChannel parameters.

6.2.15.3 MLME-START.confirm

6.2.15.3.1 Function

This primitive reports the results of the attempt to start using a new superframe configuration.

6.2.15.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-START.confirm    (
                       Status
                      )
```

Table 33 specifies the parameters for the MLME-START.confirm primitive.

Table 33 – MLME-START.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, NO_SHORT_ADDRESS, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The result of the attempt to start using an updated superframe configuration.

6.2.15.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer in response to an MLMESTART.request primitive. This primitive returns a status of either SUCCESS, indicating that the MAC sublayer has started using the new superframe configuration, or an error code of NO_SHORT_ADDRESS, UNAVAILABLE_KEY, FRAME_TOO_LONG, FAILED_SECURITY_CHECK or INVALID_PARAMETER.

6.2.15.3.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of the result of its request to start using a new superframe configuration. If the MAC sublayer has been successful, the status parameter will be set to SUCCESS. Otherwise, the status parameter indicates the error.

6.2.15.4 Message sequence chart for updating the superframe configuration

Figure 8 illustrates the sequence of messages necessary for initiating beacon transmissions in a WRC.

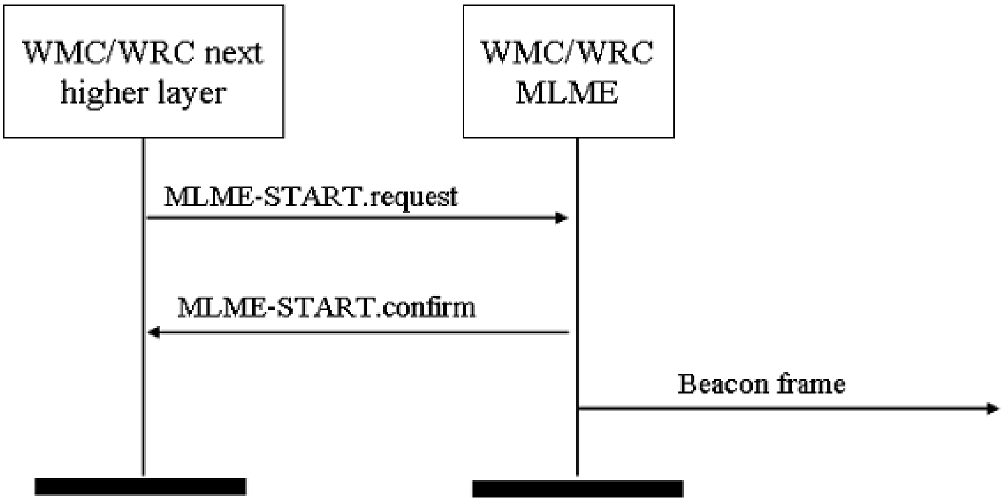


Figure 8 – Message sequence chart for updating the superframe configuration

6.2.16 Primitives for synchronising with a WRC

6.2.16.1 Overview

These primitives define how synchronisation with a WRC may be achieved and how a loss of synchronisation is communicated to the next higher layer.

All devices shall provide an interface for these primitives.

6.2.16.2 MLME-SYNC.request

6.2.16.2.1 Function

This primitive requests to synchronise with the WRC by acquiring and, if specified, tracking its beacons.

6.2.16.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-SYNC.request     (
                       LogicalChannel,
                       TrackBeacon
                       )
```

Table 34 specifies the parameters for the MLME-SYNC.request primitive.

Table 34 – MLME-SYNC.request parameters

Name	Type	Valid range	Description
LogicalChannel	Integer	PHY supported used logical channel range	The logical channel on which to attempt WRC synchronisation.
TrackBeacon	Boolean	TRUE or FALSE	TRUE if the MLME is to synchronise with the next beacon and attempt to track all future beacons. FALSE if the MLME is to synchronise with only the next beacon.

6.2.16.2.3 When generated

This primitive is generated by the next higher layer of a device on a beacon enabled PAN and issued to its MLME to synchronise with the WRC.

6.2.16.2.4 Effect on receipt

If this primitive is received by the MLME on a beacon enabled PAN, it will first set *phyCurrentChannel* equal to the LogicalChannel parameter by issuing the PLME-SET.request primitive to the PHY layer. The MLME then enables its receiver and search for the current network beacon. If the TrackBeacon parameter is equal to TRUE, the MLME will track the beacon, i.e. enable its receiver just before the expected time of each beacon so that the beacon frame can be processed. If the TrackBeacon parameter is equal to FALSE, the MLME will locate the beacon but not continue to track it. If this primitive is received by the MLME while it is currently tracking the beacon, the MLME will not discard the primitive but rather treat it as a new synchronisation request. If the beacon could not be located either on its initial search or during tracking, the MLME will issue the MLME-SYNC-LOSS.indication primitive with a loss reason of BEACON_LOST.

6.2.16.3 MLME-SYNC-LOSS.indication

6.2.16.3.1 Function

This primitive indicates the loss of synchronisation with a WRC.

6.2.16.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-SYNC-LOSS.indication    (
                              LossReason
                              )
```

Table 35 specifies the parameters for the MLME-SYNC-LOSS.indication primitive.

Table 35 – MLME-SYNC-LOSS.indication parameters

Name	Type	Valid range	Description
LossReason	Enumeration	REALIGNMENT or BEACON_LOST	The reason that synchronisation was lost.

6.2.16.3.3 When generated

This primitive is generated by the MLME of a device and issued to its next higher layer in the event of a loss of synchronisation with the WRC. It is also generated by the MLME of a PAN WRC and issued to its next higher layer in the event of a mesh ID conflict.

6.2.16.3.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of a loss of synchronisation.

6.2.16.4 Message sequence chart for synchronising with a WRC

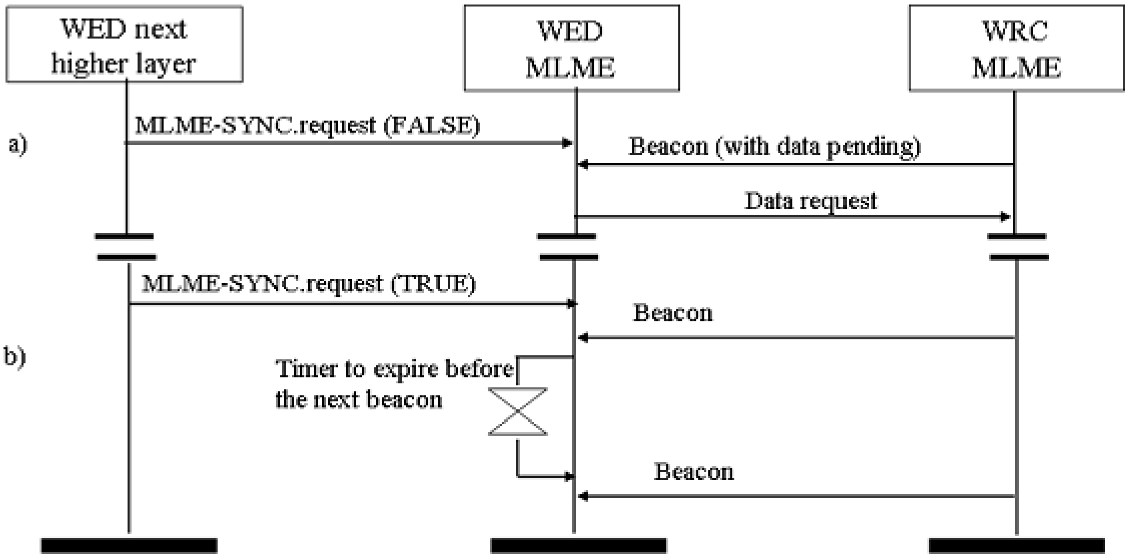

WED next higher layer
WED MLME
WRC MLME
a)
MLME-SYNC.request (FALSE)
Beacon (with data pending)
Data request
MLME-SYNC.request (TRUE)
Beacon
b)
Timer to expire before the next beacon
Beacon

Figure 9 – Message sequence chart for synchronising to a WRC in a beacon-enabled WPAN

Figure 9 illustrates the sequence of messages necessary for a device to synchronise with a WRC. In a), a single synchronisation request is issued. The MLME then searches for a beacon and, if found, determines whether the WRC has any data pending for the device. If so, the data is requested. In b), a track synchronisation request is issued. The MLME then searches for a beacon and, if found, attempts to keep track of it using a timer which expires just before the expected time of the next beacon. The MLME also checks for any data pending in the WRC for the device whenever a beacon frame is received.

6.2.17 Primitives for requesting data from a WRC

6.2.17.1 Overview

These primitives define how to request data from a WRC. All devices shall provide an interface for these primitives.

6.2.17.2 MLME-INDIRECT-COMM.request

6.2.17.2.1 Function

This primitive prompts the device to request data from the WRC.

6.2.17.2.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-INDIRECT-COMM.request        (
                                   Co-ordAddrMode,
                                   Co-ordMeshId,
                                   Co-ordAddress,
                                   SecurityEnable
                                   )
```

Table 36 specifies the parameter for the MLME-INDIRECT-COMM.request primitive.

Table 36 – MLME-INDIRECT-COMM.request parameters

Name	Type	Valid range	Description
Co-ordAddrMode	Integer	0x02 to 0x03	The addressing mode of the WRC to which the poll is intended. This parameter can take one of the following values: 2 = 16-bit short address, 3 = 64-bit extended address.
Co-ordMeshId	Integer	0x0000 to 0xfffe	The PAN identifier of the WRC to which the poll is intended.
Co-ordAddress	Device address	Co-ordAddrMode parameters.	The address of the WRC to which the poll is intended.
SecurityEnable	Boolean	TRUE or FALSE	TRUE if security is enabled for this transfer or FALSE otherwise.

6.2.17.2.3 When generated

This primitive is generated by the next higher layer and issued to its MLME when data is to be requested from a WRC.

6.2.17.2.4 Effect on receipt

On receipt of this primitive the MLME generates and sends a data request command. If the poll is directed to the PAN WRC, the data request command is generated without any destination address information present. Otherwise, the data request command is generated with the destination address information comprised of the Co-ordMeshId and Co-ordAddress parameters.

6.2.17.3 MLME-INDIRECT-COMM.confirm

6.2.17.3.1 Function

This primitive reports the results of a request to poll the WRC for data.

6.2.17.3.2 Semantics of the service primitive

The semantics of this primitive is as follows:

```
MLME-INDIRECT-COMM.confirm      (
                                 Status
                                 )
```

Table 37 specifies the parameters for the MLME-INDIRECT-COMM.confirm primitive.

Table 37 – MLME-INDIRECT-COMM.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK, or INVALID_PARAMETER	The status of the data request.

6.2.17.3.3 When generated

This primitive is generated by the MLME and issued to its next higher layer in response to an MLMEPOLL.request primitive. If the request was successful, the status parameter will be equal to SUCCESS indicating a successful poll for data. Otherwise, the status parameter indicates an error code of CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, UNAVAILABLE_KEY, FAILED_SECURITY_CHECK or INVALID_PARAMETER.

6.2.17.3.4 Effect on receipt

On receipt of this primitive, the next higher layer is notified of the status of the procedure to request data from the WRC.

6.2.17.3.5 Message sequence chart for requesting data from a WRC

Figure 10 illustrates the sequence of messages necessary for a device to request data from a WRC. In both cases, a poll request is issued to the MLME, which then sends a data request command to the WRC. In a), the corresponding acknowledgement has the frame pending (FP) subfield set to 0 and the MLME issues the poll request confirmation immediately. In b), the corresponding acknowledgement has the frame pending (FP) subfield set to 1 and the MLME enables the receiver in anticipation of the data frame from the WRC. On receipt of this data frame, the MLME issues a poll request confirmation followed by a data indication containing the data of the received frame.

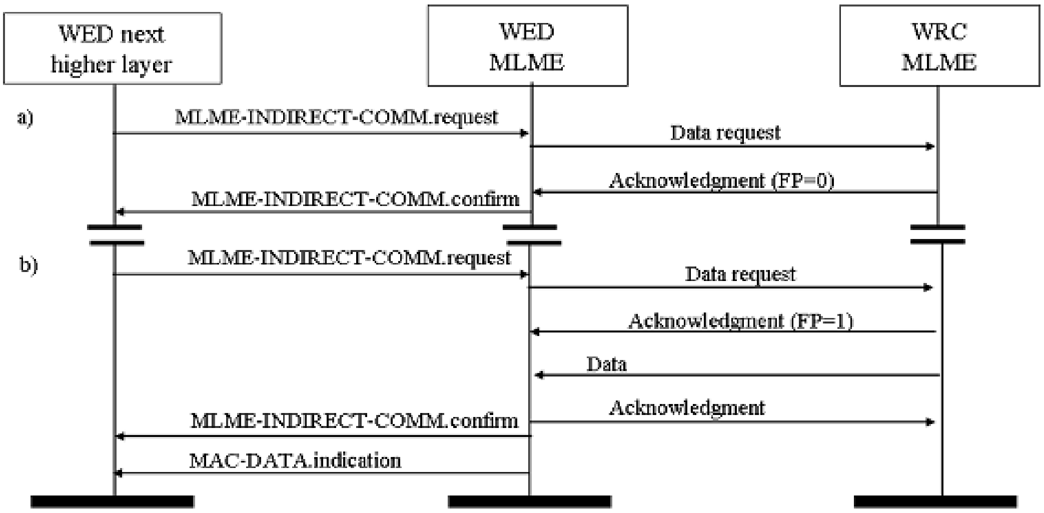


Figure 10 – Message sequence chart for requesting data from the WRC

6.3 MAC frame formats

6.3.1 General

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

a) A MHR, which comprises frame control, sequence number, address information and security related information.

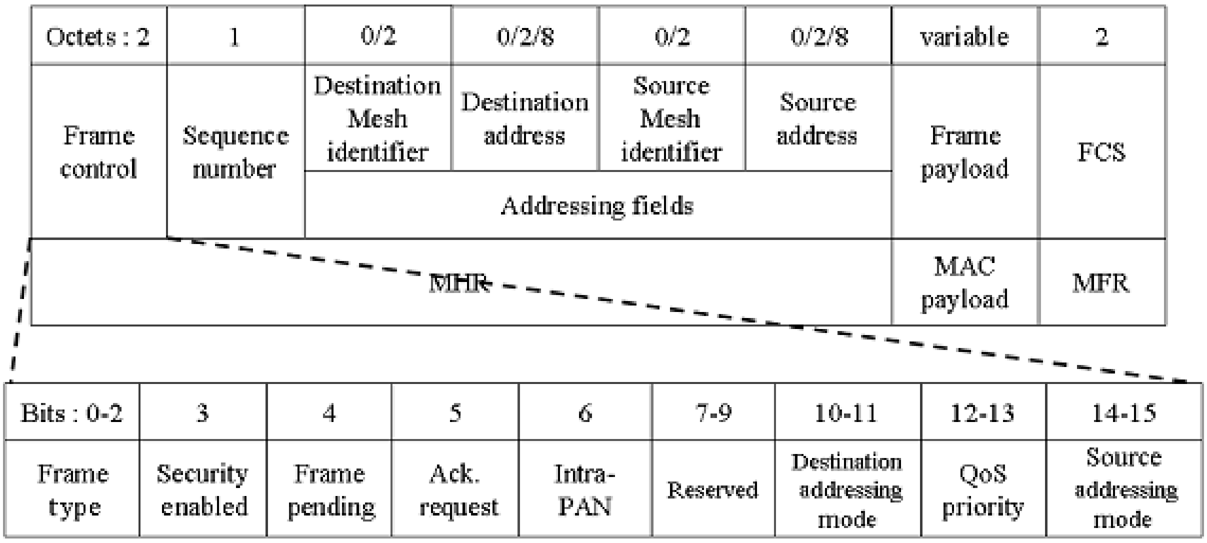
b) A MAC payload, of variable length, which contains information specific to the frame type. Acknowledgement frames do not contain a payload.

c) A MFR, which contains a FCS.

The frames in the MAC layer are described as a sequence of fields in a specific order. All frame formats in this subclause are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from 0 (leftmost and least significant) to *k* – 1 (rightmost and most significant), where the length of the field is *k* bit. Fields that are longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits. For every MAC frame, all reserved bits shall be set to zero upon transmission and shall be ignored upon receipt.

6.3.2 General MAC frame format

The MAC frame format is composed of a MHR, a MAC payload and a MFR. The fields of the MHR appear in a fixed order. However, the addressing fields may not be included in all frames. The general MAC frame shall be formatted as illustrated in Figure 11.



Octets : 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination Mesh identifier	Destination address	Source Mesh identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Bits : 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Destination addressing mode	QoS priority	Source addressing mode

Figure 11 – General MAC frame format

The frame control field is 2 octets in length and contains information defining the frame type, addressing fields and other control flags. The frame control field shall be formatted as illustrated in Figure 11 The frame type subfield is 3 bit in length and shall be set to one of the non-reserved values. The security enabled subfield is 1 bit in length, and it shall be set to one if the frame is protected by the MAC sublayer and shall be set to zero otherwise. The auxiliary security header field of the MHR shall be present only if the security enabled subfield is set to one.

The frame pending subfield is 1 bit in length and shall be set to one if the device sending the frame has more data for the recipient. This subfield shall be set to zero otherwise. The frame pending subfield shall be used only in beacon frames or frames transmitted either during the CAP by devices operating on a beacon-enabled mesh or at any time by devices operating on a non-beacon-enabled mesh network. At all other times, it shall be set to zero on transmission and ignored on reception. The acknowledgement request subfield is 1 bit in length and specifies whether an acknowledgement is required from the recipient device on receipt of a data or MAC command frame. If this subfield is set to one, the recipient device shall send an acknowledgement frame only if, upon reception, the frame passes the third level of filtering. If this subfield is set to zero, the recipient device shall not send an acknowledgement frame.

The mesh ID compression subfield is 1 bit in length and specifies whether the MAC frame is to be sent containing only one of the mesh identifier fields when both source and destination addresses are present. If this subfield is set to one, and both the source and destination addresses are present, the frame shall contain only the destination mesh identifier field, and the source mesh identifier field shall be assumed equal to that of the destination. If this subfield is set to zero and both the source and destination addresses are present, the frame shall contain both the source mesh identifier and destination mesh identifier fields. If only one of the addresses is present, this subfield shall be set to zero and the frame shall contain the mesh identifier field corresponding to the address. If neither address is present, this subfield shall be set to zero, and the frame shall not contain either mesh identifier field.

The destination addressing mode subfield is 2 bit in length and shall be set to one of the non-reserved values. If this subfield is equal to zero and the frame type subfield does not specify that this frame is an acknowledgement or beacon frame, the source addressing mode subfield shall be nonzero, implying that the frame is directed to the mesh co-ordinator with the mesh identifier as specified in the source mesh identifier field. The frame version subfield is 2 bit in length and specifies the version number corresponding to the frame. This subfield shall be set to 0x00 to indicate a frame compatible with IEEE 802.15.4-2003 and 0x01 to indicate an IEEE 802.15.4 frame. All other subfield values shall be reserved for future use.

The source addressing mode subfield is 2 bit in length and shall be set to one of the non-reserved values. If this subfield is equal to zero and the frame type subfield does not specify that this frame is an acknowledgement frame, the destination addressing mode subfield shall be nonzero, implying that the frame has originated from the mesh co-ordinator with the mesh identifier as specified in the destination mesh identifier field.

The sequence number field is 1 octet in length and specifies the sequence identifier for the frame. For a beacon frame, the sequence number field shall specify a BSN. For a data, acknowledgement, or MAC command frame, the sequence number field shall specify a DSN that is used to match an acknowledgement frame to the data or MAC command frame.

The destination mesh identifier field, when present, is 2 octets in length and specifies the unique mesh identifier of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast mesh identifier, which shall be accepted as a valid mesh identifier by all devices currently listening to the channel. This field shall be included in the MAC frame only if the destination addressing mode subfield of the frame control field is nonzero.

The destination address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the destination addressing mode subfield of the frame control field, and specifies the address of the intended recipient of the frame. A 16-bit value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid 16-bit short address by all devices currently listening to the channel. This field shall be included in the MAC frame only if the destination addressing mode subfield of the frame control field is nonzero.

The source mesh identifier field, when present, is 2 octets in length and specifies the unique mesh identifier of the originator of the frame. This field shall be included in the MAC frame only if the source addressing mode and mesh ID compression subfields of the frame control field are nonzero and equal to zero, respectively. The mesh identifier of a device is initially determined during association on a mesh network, but may change following a mesh identifier conflict resolution.

The source address field, when present, is either 2 octets or 8 octets in length, according to the value specified in the source addressing mode subfield of the frame control field, and specifies the address of the originator of the frame. This field shall be included in the MAC frame only if the source addressing mode subfield of the frame control field is nonzero.

The auxiliary security header field has a variable length and specifies information required for security processing, including how the frame is actually protected (security level) and which keying material from the MAC security PIB is used. This field shall be present only if the security enabled subfield is set to one.

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

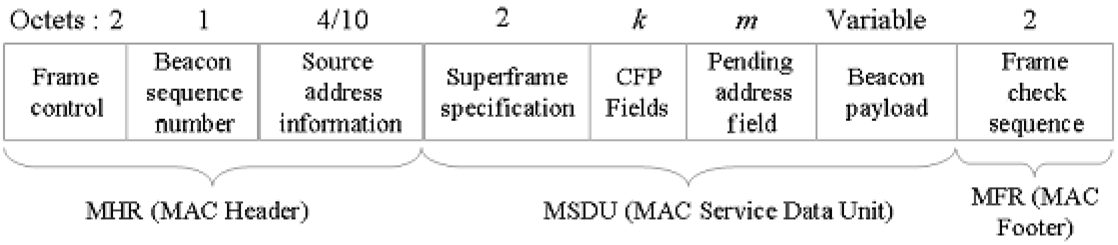
The FCS field is 2 octets in length and contains a 16-bit ITU-T CRC. The FCS is calculated over the MHR and MAC payload parts of the frame. The FCS shall be calculated using the following standard generator polynomial of degree 16:

$$G(x) = x^{16} + x^{12} + x^{6} + 1$$

6.3.3 Beacon frame format

6.3.3.1 Overview

The beacon frame shall be formatted as illustrated in Figure 12.



Octets : 2	1	4/10	2	*k*	*m*	Variable	2
Frame control	Beacon sequence number	Source address information	Superframe specification	CFP Fields	Pending address field	Beacon payload	Frame check sequence

Figure 12 – Beacon frame format

6.3.3.2 Beacon frame MHR fields

The MHR for a beacon frame shall contain the frame control field, the sequence number field, the source mesh identifier field and the source address field. In the frame control field, the frame type subfield shall contain the value that indicates a beacon frame, and the source addressing mode subfield shall be set as appropriate for the address of the co-ordinator transmitting the beacon frame. If protection is used for the beacon, the security enabled subfield shall be set to one. The frame version subfield shall be set to one only if the security enabled subfield is set to one. If a broadcast data or command frame is pending, the frame pending subfield shall be set to one. All other subfields shall be set to zero and ignored on reception. The sequence number field shall contain the current value of *macBSN*. The addressing fields shall comprise only the source address fields. The source PAN identifier and source address fields shall contain the PAN identifier and address, respectively, of the device transmitting the beacon. The auxiliary security header field, if present, shall contain the information required for security processing of the beacon frame. The superframe specification field is 16 bit in length.

6.3.4 MAC command frame format

6.3.4.1 Overview

Table 38 shows the WiBEEM MAC command frame. WMC/WRC should transmit/receive all types of command frames. WED could use the part of command frame types when necessary. In the beacon mode, command frames should be transmitted or received only in CAP.

Table 38 – MAC command frame

Command identifier	Command name
0x01	Association request
0x02	Association response
0x03	Disassociation notification
0x04	Data request
0x05	Mesh ID conflict notification
0x06	Orphan notification
0x07	Beacon request
0x08	Co-ordinator realignment
0x09	CFP request
0x0A	Rate reconfiguration request
0x0B	Rate reconfiguration response

6.3.4.2 Association request command

6.3.4.2.1 General

Association request command is used for a WED to request association to the WRC. The association request command should be transmitted only when a free device intends to request an association to the specified mesh network. A free device finds the other devices by the scan procedure and then, requests the association. All devices should use association request command, but WED cannot receive the command. Table 39 shows the format of the association request command.

Table 39 – Association request command

Octets: 17/23	1	1
MHR fields	Command frame identifier	Capability information

6.3.4.2.2 MHR field

The fields of the MHR of the general MAC frame format shall be specified as indicated in this subclause.

The source addressing mode subfield of the frame control field shall be set to 3 (64 bit extended addressing). The destination addressing mode subfield shall be set to the same mode as indicated in the beacon frame to which the association request command refers.

If security is used for the association request command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security-enabled subfield shall be set to 0.

The frame pending subfield of the frame control field shall be set to 0 and ignored upon reception, and the acknowledgement request subfield shall be set to 1.

The destination MeshID identifier field shall contain the identifier of the PAN to which to associate. The destination address field shall contain the address from the beacon frame that was transmitted by the WMC to which the association request command is being sent. The source MeshID identifier field shall contain the broadcast MeshID identifier (i.e., 0xffff). The source address field shall contain the value of *aExtendedAddress*.

6.3.4.2.3 Capability information field

The capability information field shall be formatted as illustrated in Table 40.

Table 40 – Capability information field format

Octets: 17/23	1	2	3	4	5	6	7
MHR fields	Command frame identifier	Capability information	Receive on when idle	Mobility	Reserved	Security capability	Allocate address

The alternate mesh co-ordinator subfield is 1 bit in length and shall be set to 1 if the device is capable of becoming a WMC. Otherwise, the alternate mesh co-ordinator subfield shall be set to 0.

The device type subfield is 1 bit in length and shall be set to 1 if the device is a WMC/WRC. Otherwise, the device type subfield shall be set to 0 to indicate a WED.

The power source subfield is 1 bit in length and shall be set to 1 if the device is receiving power from the alternating current mains. Otherwise, the power source subfield shall be set to 0.

The receiver is on when idle subfield is 1 bit in length and shall be set to 1 if the device does not disable its receiver to conserve power during idle periods. Otherwise, the receiver idle subfield shall be set to 0. The mobility subfield denotes the mobility characteristic of the device. If a device is a mobile device, the subfield is set to 1, otherwise 0. The security capability subfield is 1 bit in length and shall be set to 1 if the device is capable of sending and receiving secured MAC frames using the security suite. Otherwise the security capability subfield shall be set to 0. The allocate address subfield is one bit in length and shall be set to 1 if the device wishes the WMC/WRC to allocate a short address as a result of the association procedure. If this subfield is set to 0, the special short address of 0xfffe shall be allocated to the device and returned through the association response command. In this case, the device shall communicate on the mesh using only its 64 bit extended address.

6.3.4.3 Association response command

6.3.4.3.1 General

The association response command allows the WMC/WRC to communicate the results of an association attempt back to the device requesting association.

This command shall only be sent by the WMC/WRC to a device that is at the same time trying to be associated.

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

The association response command shall be formatted as illustrated in Table 41.

Table 41 – Association response command

Octets: 17/23	1	2	1
MHR fields	Command frame identifier	Short address	Association statues

6.3.4.3.2 MHR field

The fields of the MHR of the general MAC frame format shall be specified as indicated in this subclause. The destination addressing mode and source addressing mode subfields of the frame control field shall each be set to 3 (i.e., 64 bit extended addressing).

If security is used for the association response command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The destination and source mesh identifier fields shall contain the value of *macMeshId*. The destination address field shall contain the extended address of the device requesting association. The source address field shall contain the value of *aExtendedAddress*.

6.3.4.3.3 Short address field

The short address field is 16 bit in length.

If the WMC/WRC was not able to associate this device to its mesh, this field shall be set to 0xffff, and the association status field shall contain the reason for the failure. If the WMC/WRC was able to associate the device to its mesh, this field shall contain the short address that the device may use in its communications on the mesh until it is disassociated.

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

6.3.4.3.4 Association status field

Association status field has 8 bit length as shown in Table 42.

Table 42 – Association status field

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

6.3.4.4 Disassociation notification command

6.3.4.4.1 General

Either the WMC/WRC or an associated device may send the disassociate notification command. All devices shall implement this command.

The disassociation notification command shall be formatted as illustrated in Table 43.

Table 43 – Disassociation notification command format

Octets: 17	1	2
MHR fields	Command frame identifier	Disassociation reason

6.3.4.4.2 MHR field

The fields of the MHR of the general MAC frame format shall be specified as indicated in this subclause.

The destination addressing mode and source addressing mode subfields of the frame control field shall both be set to 3 (i.e., 64 bit extended addressing).

If security is used for the disassociation notification command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to the destination address. Otherwise, the security enabled subfield shall be set to 0.

The destination and source mesh identifier fields shall contain the value of *macMeshId*. If the WMC/WRC wants an associated device to leave the mesh, then the destination address field shall contain the extended address of the device being removed from the mesh. If an associated device wants to leave the mesh, then the destination address field shall contain the value of *macCo-ordExtendedAddress.* The source address field shall contain the value of *aExtendedAddress*.

6.3.4.4.3 Disassociation reason field

The disassociation reason field is 8 bit in length and shall contain one of the nonreserved values listed in Table 44.

Table 44 – Disassociation reason code

Disassociate reason	Description
0x00	Reserved
0x01	The co-ordinator wishes the device to leave the PAN
0x02	The device wishes to leave the PAN
0x03 to 0x7f	Reserved
0x80 to 0xff	Reserved for MAC primitive enumeration values

6.3.4.5 Data request command

6.3.4.5.1 General

The data request command is sent by a device to request data from a WMC/WRC.

On a beacon-enabled mesh network, this command shall be sent by a device when *macAutoRequest* is equal to TRUE and a beacon frame indicating that data are pending for that device is received from its WMC/WRC. The WMC/WRC indicates pending data in its beacon frame by adding the address of the recipient of the data to the address list field.

This command shall also be sent when instructed to do so by the next higher layer on reception of the MLME-INDIRECT-COMM.request primitive. In addition, a device may send this command to the co-ordinator *aResponseWaitTime* symbols after the acknowledgement to a request command, such as an association or a CFP request.

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

The data request command shall be formatted as illustrated in Table 45.

Table 45 – Data request command format

Octets: 7/11/13/17	1
MHR fields	Command frame identifier

6.3.4.5.2 MHR field

The fields of the MHR of the general MAC frame format shall be specified as indicated in this subclause.

The destination addressing mode subfield of the frame control field shall be set to 0 (i.e., destination addressing information not present) if the data request command is to be sent to the WMC/WRC or set otherwise according to the WMC/WRC to which the data request command is directed. The source addressing mode subfield shall be set to 3 (i.e., 64 bit extended address) if the value of *macShortAddress* is equal to either 0xfffe or 0xffff or set to 2 (i.e., 16 bit short addressing) otherwise.

If security is used for the data request command, the security enabled subfield shall be set to 1, and the frame shall be processed according to the method defined by the security suite corresponding to *macCo-ord-ExtendedAddress*. Otherwise, the security enabled subfield shall be set to 0.

If the destination addressing mode subfield of the frame control field is set to 2, the destination mesh identifier and destination address fields shall contain the value of *macMeshId* and *macCo-ordShortAddress*, respectively. The source mesh identifier field shall contain the value of *macMeshId*. The source address field shall contain the value of *aExtendedAddress* if the value of *macShortAddress* is equal to 0xfffe. Otherwise, the source address field shall be set to the value of *macShortAddress*.

6.3.4.6 Mesh ID conflict notification command

6.3.4.6.1 General

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

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

The mesh ID conflict notification command shall be formatted as illustrated in Table 46.

Table 46 – Mesh ID conflict notification command format

Octets: 17	1
MHR fields	Command frame identifier

6.3.4.6.2 MHR field

The fields of the MHR of the general MAC frame format (see Figure 11) shall be specified as indicated in this subclause.

The destination addressing mode and source addressing mode subfields of the frame control field shall both be set to 3 (i.e., 64 bit extended addressing).

The frame shall be processed for security by the sender according to the method defined by the security suite corresponding to *macCo-ordExtendedAddress*. If the security suite identifier is 0x00, the security enabled subfield of the frame control field shall be set to 0. Otherwise, the security enabled subfield shall be set to 1.

The destination mesh identifier field and source mesh identifier field shall each contain the value of *macMeshId*. The destination address field shall contain the value of *macCo-ordExtendedAddress*. The source address field shall contain the value of *aExtendedAddress*.

6.3.4.7 Orphan notification command

6.3.4.7.1 General

The orphan notification command is used by an associated device that has lost synchronisation with its WMC/WRC. All devices shall be capable of transmitting this command. The orphan notification command shall be formatted as illustrated in Table 47.

Table 47 – Orphan notification command format

Octets: 17	**1**
MHR fields	Command frame identifier

6.3.4.7.2 MHR fields

The source addressing mode subfield of the frame control field shall be set to three (e.g., 64-bit extended addressing). The source addressing mode subfield of the frame control field shall be set to two (e.g., 16-bit short addressing). If security is used, the security enabled subfield shall be set to 0. Otherwise, the security enabled subfield shall be set to 0. The destination mesh identifier field and the source mesh identifier shall contain the broadcast short address (e.g., 0xffff). The destination address field shall contain the extended address of the orphaned device if the command is directed to an orphaned device. The source mesh identifier field shall contain the value of *MeshId*, and the source address field shall contain the value of *aExtendedAddress*.

6.3.4.8 Beacon request command

6.3.4.8.1 General

The beacon request command is used by a WED to locate all WMC/WRC within its POS during an active scan. The beacon request command shall be formatted as illustrated in Table 48.

Table 48 – Beacon request command format

Octets: 17	**1**
MHR fields	Command frame identifier

6.3.4.8.2 MHR fields

The destination addressing mode subfield of the frame control field shall be set to two (i.e., 16-bit short addressing), and the source addressing mode subfield shall be set to zero (i.e., source addressing information not present). The destination mesh identifier field shall contain the broadcast mesh identifier (i.e., 0xffff). The destination address field shall contain the broadcast short address (i.e., 0xffff).

6.3.4.9 Co-ordinator realignment command

6.3.4.9.1 General

The co-ordinator realignment command is sent by the WRC either following the reception of an orphan notification command from a device that is recognised to be on its PAN or when any of its PAN configurations attributes change. If this command is sent following the reception of an orphan notification command, it is sent directly to the orphaned device. All devices shall be capable of receiving this command. The co-ordinator realignment command shall be formatted as illustrated in Table 49.

Table 49 – Co-ordinator realignment command format

Octets: 17/23	**1**	**2**	**2**	**1**	**2**
MHR fields	Command frame identifier	Mesh identifier	Co-ordinator short address	Logical channel	Short address

6.3.4.9.2 MHR fields

The destination addressing mode subfield of the frame control field shall be set to three (e.g., 64-bit extended addressing) if the command is directed to an orphaned device or set to two (e.g., 16-bit short addressing) if it is to be broadcast to the PAN. The source addressing mode subfield of the frame control field shall be set to three (e.g., 64-bit extended addressing). If security is used, the security enabled subfield shall be set to 0. Otherwise, the security enabled subfield shall be set to 0.

The destination PAN identifier field shall contain the broadcast mesh identifier (e.g., 0xffff). The destination address field shall contain the extended address of the orphaned device if the command is directed to an orphaned device. Otherwise, the destination address field shall contain the broadcast short address (e.g., 0xffff). The source mesh identifier field shall contain the value of *MeshId*, and the source address field shall contain the value of *aExtendedAddress*.

6.3.4.9.3 Mesh identifier fields

The mesh identifier field shall contain the mesh identifier that the co-ordinator intends to use for all future communications.

6.3.4.9.4 Co-ordinator short address fields

The co-ordinator short address field shall contain the value of *macShortAddress*.

6.3.4.9.5 Logical channel field

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

6.3.4.9.6 Short address fields

The short address field has a length of 16 bit. If the co-ordinator realignment command is broadcast to the PAN, the Short Address field shall be set to 0xffff and ignored on reception. If the co-ordinator realignment command is sent directly to an orphaned device, this field shall contain the short address that the orphaned device shall use to operate on the PAN. If the orphaned device does not have a short address, because it always uses its 64-bit extended address, this field shall contain the value 0xfffe.

6.3.4.10 CFP request command

6.3.4.10.1 General

The CFP request command is used by an associated device that is requesting the allocation of a new CFP or the deallocation of an existing CFP from the WRC. Only devices that have a 16-bit short address less than 0xfffe shall send this command. The CFP request command shall be formatted as illustrated in Table 50.

Table 50 – CFP request command format

Octets: 7	**1**	**2**
MHR fields	Command frame identifier	CFP characteristics

6.3.4.10.2 MHR field

The destination addressing mode subfield of the frame control field shall be set to zero (e.g., destination addressing information not present), and the source addressing mode subfield shall be set to two (e.g., 16-bit short addressing). If security is used, the security enabled subfield shall be set to 0. Otherwise, the security enabled subfield shall be set to 0. The frame pending subfield of the frame control field shall be set to zero and ignored upon reception, and the acknowledgement request subfield shall be set to one. The source mesh identifier field shall contain the value of MeshId, and the source address field shall contain the value of macShortAddress.

6.3.4.10.3 CFP characteristics field

The CFP characteristics field shall be formatted as illustrated Table 51.

Table 51 – CFP characteristics field format

Bits: 0 to 3	**4**	**5**	**6 7**
CFC length	CFP direction	Characteristics type	Reserved

The CFP length subfield shall contain the number of superframe slots being requested for the CFP. The CFP direction subfield shall be set to one if the CFP is to be a receive-only CFP. Conversely, this subfield shall be set to zero if the CFP is to be a transmit-only CFP. CFP direction is defined relative to the direction of data frame transmissions by the device. The characteristics type subfield shall be set to one if the characteristics refer to a CFP allocation or zero if the characteristics refers to a CFP deallocation.

6.3.4.11 Rate reconfiguration request command

6.3.4.11.1 General

The rate reconfiguration request command is used for the request of the data rate reconfiguration. A WED can transmit the command to the WRC/WMC to reconfigure the current data rate. The rate reconfiguration request command has the following format as shown in Table 52.

Table 52 – Rate reconfiguration request format

Octets: 17/23	**1**	**1**
MHR fields	Command frame identifier	Rate change information

6.3.4.11.2 MHR field

The general MHR field for MAC frame consists of source addressing field, destination addressing field, etc. The source addressing mode subfield in the frame control field should be set to 3 (64-bit extended addressing). The destination addressing mode subfield in the frame control field should be set as the address specified in the beacon frame of the beacon-emitting device. If the security mode is enabled, the subfield is set to 1, otherwise 0. The destination *MeshId* field denotes the associated mesh network information.

6.3.4.11.3 Rate change information field

The rate change information field is shown in Table 53.

Table 53 – Data rate

Rate change information	Data rate
0x01	31,25 kbit/s
0x02	62,50 kbit/s
0x03	125 kbit/s
0x04	500 kbit/s
0x05	1 Mbit/s
0x06	2 Mbit/s
0x07 to 0x0F	Reserved

6.3.4.12 Rate reconfiguration response command

6.3.4.12.1 General

The rate reconfiguration response command is used for the confirmation of the data rate reconfiguration. A WRC/WMC transmits the acceptance or reject for the data rate reconfiguration request of WED. The rate reconfiguration response command has the format indicated in Table 54.

Table 54 – Rate reconfiguration request command format

Octets: 17/23	1	1
MHR fields	Command frame identifier	Rate change confirm

6.3.4.12.2 MHR field

The general MHR field for MAC frame consists of source addressing field, destination addressing field, etc. The source addressing mode subfield in the frame control field should be set to 3 (64-bit extended addressing). The destination addressing mode subfield in the frame control field should be set as the address specified in the beacon frame of the beacon-emitting device. If the security mode is enable, the subfield is set to 1, otherwise 0. The destination MeshID field denotes the associated mesh network information.

6.3.4.12.3 Rate change information field

The rate change information field has the 0 or 1 value if the request is accepted by the WRC/WMC.

Bibliography

ISO/IEC 7498-1:1994, *Information technology – Open systems interconnection – Basic reference model: The basic model*

ISO/IEC 8802-2:1998 *(IEEE Std 802.2™, 1998 Edition), Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*

ISO/IEC 9646-1:1994, *Information technology – Open systems interconnection – Conformance testing methodology and framework – Part 1: General concepts*

ISO/IEC 9646-7:1995 (ITU-T Rec. X.296 (1994)), *Information technology – Open systems interconnection – Conformance testing methodology and framework – Part 7: Implementation conformance statements*

ISO/IEC 10039:1991, *Information technology – Open systems interconnection – Local area networks – Medium Access Control (MAC) service definition* (withdrawn)

ISO/IEC 15802-1:1995, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 1: Medium Access Control (MAC) service definition*

ITU-T Recommendation X.210, *Data Networks and Open Systems Communication – Open Systems Interconnection – General*

ITU-T Recommendation Z.100, *CCITT Specification and Description Language – Overview of SDL*

IEEE 802 – 1990, *IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture*

NIST FIPS Pub 197 *Advanced Encryption Standard (AES),* Federal Information Processing Standards Publication 197, U.S. Department of Commerce/N.I.S.T., November, 2001

ZigBee Document 053474r06, Version 1.0 – December 14th, 2004 – ZigBee Alliance

____

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch