INTERNATIONAL STANDARD

ISO/IEC 24771

Second edition
2014-08-01

Information technology — Telecommunications and information exchange between systems — MAC/PHY standard for ad hoc wireless network to support QoS in an industrial work environment

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Norme MAC/PHY pour un réseau ad hoc sans fil qui supporte QoS dans un environnement de travail industriel

Reference number
ISO/IEC 24771:2014(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

List of Figures

List of Tables

Foreword

ISO (the International Organization for Standardization) and IEC (the 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 through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 6, *Telecommunications and information exchange between systems*.

This second edition cancels and replaces the first edition (ISO/IEC 24771:2009), which has been technically revised.

Information technology — Telecommunications and information exchange between systems — MAC/PHY standard for ad hoc wireless network to support QoS in an industrial work environment

1 Scope

This International Standard defines a protocol for the physical layer (PHY) and the data link layer in order to construct a reliable and high-speed data transmission network between devices on industrial sites such as factories and plants. This network specification provides a standardized protocol to provide a framework for various industrial devices to establish a simple, low-cost, energy-efficient, and high-speed network between them. In order to fulfil the service requirements of the factories and large plants, this network specification is designed to enable devices to establish a network by themselves without help of any infrastructure and reliably exchange various kinds of data, including real-time audio and video data, between them. In addition to high transmission rates, Quality of Service (QoS) for multimedia data - such as video - is also provided.

The devices mentioned in this International Standard refer to equipment that is and can be used in industrial sites such as factories and automated assembly lines. Such devices include PLC (Programmable Logic Controller), and CNC (Computerized Numerical Controller) and manufacturing robots. However, beyond such conventional devices, devices mentioned in this International Standard include personal IT devices that workers may carry and use while working, including cellular phones, personal industrial digital assistants (PDA), and laptop PCs.

2 Normative references

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 9797-1, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*

ISO/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

3 Terms and definitions, and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
Access control
control process to prevent unauthorized use of resources or bandwidth

3.1.2
Ad-hoc network
network that is spontaneously formed usually without system installation

NOTE Such networks are mainly characterized by time and space limitations.

3.1.3
Association
service used to connect authorized devices in the network

3.1.4
Authentication
device verification process allowing devices within the network to connect to one another

3.1.5
Camellia
128-bits secure block algorithm defined in the ISO/IEC 18033-3 standard

3.1.6
Coverage area
territory over which two devices can achieve acceptable quality and performance while exchanging data

3.1.7
Dissociation
service used in an established network

3.1.8
Frame
format of bits in a data exchange

3.1.9
K
prefix indicating multiplication by 1024

3.1.10
Kμs
unit of 1024 μs

3.1.11
k
prefix indicating multiplication by 1000

3.1.12
Logical Channel
data link channel sitting distinctly above the physical layer

3.1.13
Master
station that manages the network by periodically transmitting a beacon frame

3.1.14
MAC Management Protocol data Unit
MMPDU
data unit exchanged between two media access control apparatuses in order to implement the media access control management protocol

3.1.15
MAC Protocol data Unit
MPDU
data unit exchanged between two media access control apparatuses by means of utilizing the physical layer services

3.1.16
MAC Service data Unit
MSDU
data unit transmitted between media access control service access points

3.1.17
Mobile Device
device that utilizes communication networks while in motion

3.1.18
Portable Device
station that is normally portable but must be in a fixed location in order to link to the communication network

3.1.19
SEED
128-bits secure block algorithm defined in the ISO/IEC 18033-3 standard

3.1.20
Slave
station in the network other than the master

3.1.21
Station
device that can operate according to this International Standard

3.2 List of Abbreviations

AES	advanced encryption standard
ARQ	automatic repeat request
ARQN	automatic repeat request N
ASN.1	abstract symbol notation 1
BER	bit error rate
CAP	contention access period
CBC	cipher block chaining
CBC-MAC	cipher block chaining-message authentication code
CCA	clear channel assessment

CCM	counter mode encryption and cipher block chaining message authentication code
CDMA	code division multiple access
CODEC	coder/decoder
CRC	cyclic redundancy check
CTS	clear to send
DA	destination address
DBPSK	differential binary phase shift keying
DCE	data communication equipment
DLL	data link layer
DOQPSK	differential offset quadrature phase shift keying
DQPSK	differential quadrature phase shift keying
FCS	frame check sequence
FEC	forward error correction
FER	frame error rate
HCS	header check sequence
IETF	internet engineering task force
IDU	interface data unit
IP	internet protocol
ISM	industrial scientific medicine
IWN	industrial wireless network
LAN	local area network
LFSR	linear feedback shift register
LLC	logical link control
LM	link manager
LME	layer management entity
LMP	link manager protocol
LSB	least significant bit
MAC	medium access control
Master	network coordinator
MC-CDMA	multi-code CDMA

MCDU	MAC command data unit
MCPDU	MAC command protocol data unit
MDF	management-defined field
MIB	management information base
MIC	message integrity code
MLME	MAC layer management entity
MPDU	MAC protocol data unit
MSB	most significant bit
MSC	message sequence chart
MSDU	MAC service data unit
MTU	maximum transmission unit
NID	network ID
PAN	personal area network
PAR	project authorization request
PDU	protocol data unit
PHY	physical layer
PIB	PAN information base
PLME	physical layer management entity
PN	pseudo noise
PPDU	PHY protocol data unit
PPM	parts per million
PRNG	pseudo random number generator
PSDU	PHY service data unit
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quadrature phase shift keying
OQPSK	offset quadrature phase shift keying
RF	radio frequency
RFC	request for comments
RSSI	received signal strength indication

RTS	request to send
RTX	response timeout expired
RX	receive or receiver
SAP	service access point
SDP	service discovery protocol
SDU	service data unit
SEC	security
SECID	security identifier
SEQN	sequential numbering scheme
SFC	security frame counter
SME	station management entity
SQ	signal quality
SRC	short retry count
SRES	signed response
SS	station service
STN	station
TA	transmitter address
TCM	trellis coded modulation
TDD	time division duplex
TDMA	time division multiple access
TX	transmit or transmitter
TXE	transmit enable
WAN	wide area network
WLAN	wireless local area network
WM	wireless medium

4 Overview

This section defines the general attributes of the industrial wireless network and describes the attributes of the physical layer and data link layer. The physical layer is built upon a binary CDMA, and the data link layer is composed of the media access control (MAC) layer.

4.1 Characteristics

This International Standard is designed for the construction and management of an optimal network for industrial use applications.

4.1.1 Ad-hoc network

This International Standard is based upon an ad-hoc network that can be established even without a network infrastructure. A network is made up of two kinds of devices - a master and a slave, which are differentiated according to their functions. All stations can function as a master or a slave and one of them is selected as a master based on the device layout and its capabilities. An independent network structure is feasible without requiring infrastructure.

4.1.2 Quality of Service

The number of devices participating in an industrial wireless network changes vastly over time due to the channel conditions and industrial mobile device operation characteristics of a wireless environment. The bandwidth allocated to each device and the transmission delay time also have a significant effect, making it difficult to support real-time multimedia traffic services that require a certain quality of service.

This document requires one station in the network to be the master, which allocates and controls resources and thereby manages the connection quality of each network traffic.

4.1.3 Binary CDMA technology

This International Standard uses Binary-CDMA technology so that it has strong noise resistance, inherent advantage of CDMA, and has another good capability of changing bandwidth finely,

and thereby has the advantages of noise resistance and finely tuned and flexible resource allocation.

First, Binary-CDMA possesses superior noise resistance that is characteristic of CDMA technology, and this is an outstanding attribute in a wireless network environment which, unlike a wire-line network, has a high noise factor. In addition, the nature of Binary CDMA makes it possible to adjust the bandwidth by changing the number of codes used, thereby allowing flexible and finely-tuned resource allocation.

4.2 Components of network

The components of a network can be roughly depicted as shown in Figure 1. The primary component is the station. The first station trying to connect or establish a network becomes the master of the network and helps other stations to associate with it by periodically transmitting beacons. It also takes responsibilities such as quality of service and power management. The network is made up of two or more stations operating on the same wireless frequency channel in an industrial activity area.

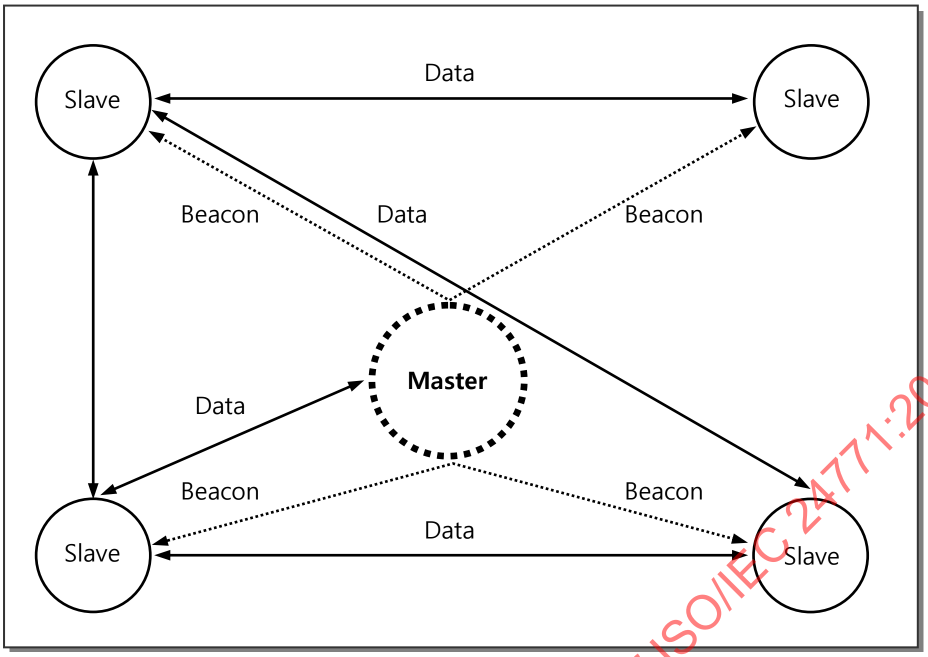

Figure 1- Network

4.2.1 Station

The station is the primary component of the network and is classified as either master or slave depending on its role. The master assumes full management, and no more than one can exist in a particular network. The master controls slaves by broadcasting beacons. Slaves send or receive data as directed by the master. To acquire time slots for data transfer, slaves make resource allocation requests to the master during the contention period.

4.2.2 Resources

Stations in the network should acquire time slots from master to perform their aimed job – exchanging data. After a station acquired rights to use some time slots from master, it can transmit frames exclusively during assigned time slots. In this sense, this International Standard depicts time slots as resources, which can be shared and/or competed among stations in the network. Time slots are supervised by the master and are distributed according to requests from slaves at the discretion of the master.

4.3 Functional overview

The media access control layer provides the following services:

- Network synchronization

- Data transmission

- Power management

- Master handover

- Security

- Data transport with quality of service (QoS)

Data transmission and reception between stations are possible under different standards of quality of service.

4.3.1 Network synchronization

The network is established once the master transmits the beacon frame. The beacon frame contains the status information of the network, and all slaves in the network use this information to sync with the network. The superframe is roughly composed of three parts as shown in Figure 2, and each period has a variable length. (The allocation period must be a multiple of the timeslot length.)

Beacon period	Contention period	Allocation period			
		Time slot 1	Time slot 2	···	Time slot n

Figure 2 - Superframe

a) beacon period: the master transmits the beacon frame containing the network status information to the slaves.

b) contention period: the slaves and master send command frames such as associate/disassociate/grant frames, resource request/grant frames, and connect request/grant frames in a random access fashion.

c) allocation period: this is divided into multiple time slots, each allotted for one station. The station receiving the time slot can send synchronous/asynchronous data or command frames during that interval.

4.3.2 Data transmission

For data transmission, two types of connections – synchronous and asynchronous - are supported. Asynchronous connections, which have minimal overhead when establishing connections but do not have a guaranteed bandwidth, are primarily used when transmitting general data that is relatively insensitive to delays. Synchronous connections, which carry a large overhead when establishing connections but have a guaranteed bandwidth, are used to transmit data for real-time services such as audio and video.

This International Standard manages communication quality in three characteristics – delay, data rate, BER. Each device defines the communication quality of data stream to send in these three parameters and sends the defined quality parameters to master to get channel time allocation. Master collects communication request from slaves and allocate time slots to streams in order that each stream can be transmitted satisfying each one's needed communication quality. Afterward a master monitors quality of communication of stream to check whether it meets initial QoS request. If the communication quality deteriorates below initial request quality, master changes time slot allocation to meet initial QoS request.

This International Standard covers MAC-to-MAC QoS within 1-hop network that MAC layer can control.

4.3.2.1 QoS parameter

Data transfers in this International Standard are differentiated as streams which are largely divided into synchronous stream and asynchronous stream, depending on whether a stream has QoS characteristic or not. Synchronous stream has 3 QoS parameter – Period, Data size to transmit per each period, BER, while asynchronous stream have only data size as a parameter.

Synchronous stream in this International Standard manages QoS with the following three parameters.

- Period: the period with which master allocates time slot. Each device is assigned at least a time slot within this period so that this value is equal to maximum delay.

- Data size to transmit per each period: the size of data that a slave will send during each period. Combined with period, this value decides data rate.

- BER: This value defines quality of communication in Bit Error Rate

Using above QoS parameters, devices can define their communication request with detail characteristics.

4.3.2.2 Maintenance of QoS

A master of a network manages communication resource (time slots) centrally. Master collects network information (usage status of allocated time slots, channel quality of each channel) and communication request by itself or from slaves and allocates time slots to slaves in order that each communication can meet its QoS request. After allocating time slots, a master checks whether each communication meets its initial QoS request. If communication does not meet its QoS requirements, a master makes communication to meet its initial requirements by using the following actions.

4.3.2.2.1 Check and estimation of wireless channel environment

Master continuously monitors BER and SNR value of current wireless channel environment. When it detects the serious change of channel environment, a master takes an action to counterpart the change such as data rate change or frequency channel change

4.3.2.2.2 Dynamic link quality control

A master continuously monitors actual communication status of data streams in a network. When a communication quality of a stream goes worse than its pre-determined level, a master changes resource allocations in order that the communication quality should meet its pre-determined level. To increase communication quality of a stream, a master decreases data rate of the stream and allocates more time slots to compensate reduced data rate through resource allocation and change mechanism.

4.3.2.2.3 Dynamic channel selection and change

When a master estimates that the network-wide communication quality of the current frequency channel goes worse than the acceptable level, it decides to change a frequency channel. Master notifies the new channel where network will be re-established to slaves, slaves leave current frequency channel and rejoin at the notified new channel. This function provides not only the ability to maintain the certain network-wide communication quality, but also interoperability to avoid collision with other kinds of wireless network.

4.3.3 Security

This standard use 128-bits secure block algorithms such as ISO/IEC18033-3 encryption algorithms and regional regulated secure block algorithms. Security for the network is one of three modes as follows:

a) Mode 0—open

Security membership is not required and payload protection (both data integrity and data encryption) is not used by the MAC. The master is allowed to use a list of slaves' addresses to admit or deny entry to the network.

b) Mode 1— inclusive CCM secure membership and payload protection

Stations establish secure membership with the master before they have access to the network's resources. Data sent in the network is allowed to use payload protection (data integrity and/or data encryption). Data integrity is required for most of the frames that are sent in the network through the inclusive CCM security mode based on secure block algorithms. Non-secure frames shall be used to communicate between a station using security mode 0 and a station using security mode 1.

c) Mode 2—exclusive CCM secure membership and payload protection

A device operating in security mode 2 shall not establish a secure relationship with devices operating in either security mode 0 or security mode 1. Stations establish secure membership with the master before they have access to the network's resources. Data sent in the network is allowed to use payload protection (data integrity and/or data encryption). When security mode 2 is enabled (the network adopts security mode 2), the exclusive CCM security mode based on secure block algorithms shall be used to communicate among the stations using security mode 2.

When security is enabled, stations that wish to join the network are required to establish secure membership with the master. The stations are also allowed to establish a secure relationship with other stations for secure communications. A station has established a secure membership or a secure relationship when it gets a management key for the security relationship. The process of establishing secure membership or a secure relationship is outside of the scope of this standard. The master or station that generates and distributes the key is called the key originator.

The payload protection protocol, as described in 8.5, uses a symmetric key that is generated by the key originator and is securely distributed to stations that have established secure membership or a secure relationship with the key originator, as described in 8.4.

4.3.4 Power management

In order to support mobile devices, efficient power management is critical. Each station is notified of the presence and timing of the incoming data during the corresponding superframe by means of the beacon frame, and can reduce power consumption by disabling the physical layer when not transmitting data. During a particular superframe, stations that are not transmitting data can be disabled until the next superframe period starts. Stations receiving data are enabled only during beacon periods and intervals during which they receive frames. Stations sending data or making requests are enabled during intervals when there is data to transmit.

4.3.5 Master handover

The master can decide whether to appoint another slave as the master based on the information it possesses about each slave's capabilities. If a newly connected station is determined to be better for the master, the master role may be handed over. Also, due to the nature of an ad-hoc network, the master may unexpectedly cease operating or depart from the established service area. In such an event, one of the remaining slaves which is most suitable for the master is selected to manage the network.

4.4 Summary of operations

The superframe is divided into the beacon period, contention period, and allocation period, and the stations operate in different modes depending on the period.

4.4.1 Broadcasting during the beacon period

During the beacon period the master broadcasts a beacon frame to all slaves. The beacon frame includes information about the status and resource availability of the network, and the slaves receiving the beacon frame use this information to verify the correctness of the corresponding superframe as well as the allocated resources.

The master places network status information into the beacon frame and sends it out to all slaves.

4.4.2 Random access during the contention period

During the contention period the slave and master communicate via the CSMA/CA scheme. Slave sends command frames such as resource allocation requests to the master during this period, and the master sends back an acknowledgement or response frame.

4.4.3 Exclusive access during the allocation period

During the allocation period, only the stations permitted by the master can communicate using the time slot and code assigned to them in the beacon frame. The master evaluates the currently available resources and decides whether or not to grant the slave's service request and allocate the resources. Data services that use asynchronous connections receive acknowledgements for the dispatched frames by default, and synchronous connections receive acknowledgements selectively.

4.5 Summary of states

A station has the following operational states.

- establishing the network

- associating with the network

- security membership and key establishment

- data transfer

- master handover

- disassociating from the network

- terminating the network

4.5.1 Establishing the network

The master establishes the network by filling a beacon frame with synchronization information such as the length of the superframe, contention period, and allocation period, and sending it out. The beacon frame includes the network ID as well as synchronization information, enabling the stations to identify the network they will connect with.

4.5.2 Associating with the network

Stations attempting to associate with the network periodically scan each frequency channel to listen for the beacon frame from the master. If the received frame is the beacon frame for the desired network, the process of associating with the relevant network's master is started, and if this succeeds, the link to the network is established.

When the station associates with the network, it sends its own capability information to the master. This capability information contains such information as the physical layer data transfer rate that can be supported, power management status, ability to function as the master, and size of memory, and based on this information the master may decide to initiate the process of changing the master.

4.5.3 Security membership and key establishment

Security mechanisms provided by this standard allow security services to be implemented to control the admission of stations into a security relationship between the master and a station or between two ordinary stations and protect the information and integrity of communications between stations in a security relationship. This standard also provides a symmetric cryptography mechanism to assist in providing security services. Additional security services need to be provided by the higher layers to ensure proper management and establishment of the symmetric keys used in this standard.

4.5.4 Data transfer

A slave to send data first requests resources for the transmission from the master during the contention period, and once it is allocated to the resources it uses them to send the data. Data transfer is possible between any stations – not only between master and slave, but also between slaves themselves. The master is involved only during the initial resource allocation process, thus enabling the stations to exchange data freely afterwards.

4.5.5 Master handover

The nature of an ad-hoc network entails frequent changes of the master. The following cases are the two most likely scenarios.

- The current master is no longer able to service requests. (For example, it cannot meet the decided quality of service.)

- The master abruptly vanishes from the network. (It loses power or moves out of the network.)

4.5.6 Disassociating from the network

Stations currently associated with the network can disassociate from the network either spontaneously or at the will of the master.

In the case of spontaneous disassociation, the station sends a disassociate request to the master. If the master wants to disassociate from the network, it first hands over the master role to the most capable slave among the stations before disassociating.

If a certain slave is having trouble in receiving service due to network conditions, the master can forcibly disassociate it by sending a disassociate command.

4.5.7 Terminating the network

Terminating of the network can be categorized in two ways: normal termination by station request and abnormal termination due to unexpected events.

Normal termination refers to the case where the master receives the request from a slave, decides whether or not to terminate, then broadcasts this decision to all slaves, thereby dissolving the network. The master can also initiate this request.

Abnormal termination of the network refers to cases where the network is dissolved due to situations such as a network-wide power outage or exodus out of the industrial coverage area. However, such occurrences are highly unlikely.

5 Inter-layer interfaces

5.1 Summary

Both the MAC layer and the physical layer conceptually have management entities, called the MAC Layer Management Entity (MLME) and the Physical Layer Management Entity (PLME) respectively. These entities provide a service interface for layer management functions.

The physical layer provides an interface between the MAC sublayer and the wireless channel by means of the PHY firmware and the PHY hardware. The physical layer includes a conceptual management entity called a PLME. This entity provides a service interface by means of layer management functions. The PLME maintains a database for managed objects in the physical layer, also called the PAN information base (PIB). The Device Management Entity (DME) is a layer independent from the other management entities and its functionality is beyond the scope of this International Standard. Simply stated, however, it is responsible for the inter-layer interfaces providing device management functions. The physical layer provides services through two Service Access Points (SAP). Data services are provided through the PD-SAP and management services are provided through the PLME-SAP. In addition, the physical layer provides DME-PLME-SAP interface, which is equivalent to the PLME-SAP interface except that it operates through DME rather than MLME.

In order for the MAC to operate correctly, each station must possess a Device Management Entity (DME). DME exists in each layer and is layer-independent. The functions of DME are beyond the scope of this International Standard and vary depending on implementation, but in general they carry out tasks such as receiving data from a layer or initializing another layer. Generally DME acts under the direction of a higher-level management application. Figure 3 shows the relationships between the various management entities.

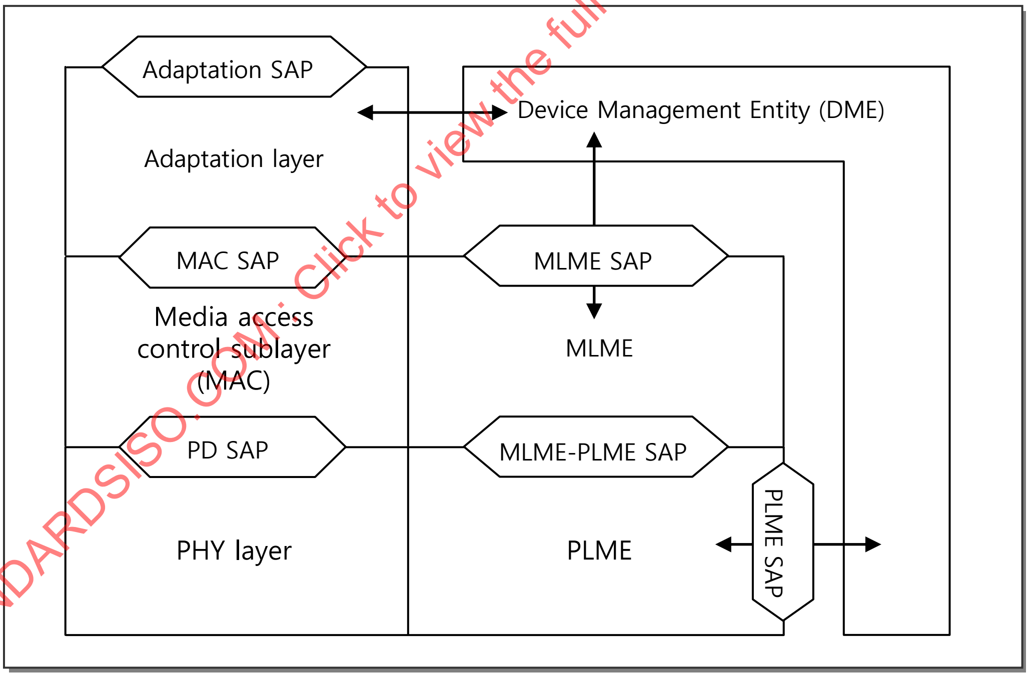

Figure 3 - Protocol stack configuration

5.2 General format of management primitives

Each sublayer's specific management information is organized into the relevant PAN information base (PIB). Corresponding to the PIB of the PAN, the LAN/MAN contains the management Information Base (MIB) that operates according to the Simple Network Management Protocol (SNMP). However, since management within Network is restricted to an individual network (i.e. one network does not interfere in the management of another) the PIB is used to define the specifications of each sublayer.

MLME and PLME are assumed to have a PIB for each sublayer, and the management primitives of the PIB are exchanged by means of management SAPs. The manager can "GET" or "SET" the value of the PIB attribute via the primitives. The "SET" request primitive can also trigger certain actions within the relevant layer.

A "GET" or "SET" primitive may be expressed in the form of a request accompanying a confirm primitive. Such primitives have the prefix MLME or PLME depending on whether the point of exchange is the MAC SAP or the PD SAP. DME utilizes the services provided by MLME through the MLME SAP.

In Table 1, "XX" stands for "MLME" or "PLME", and the parameters of the primitives are defined in Table 2.

Table 1 - General management primitive overview

Name	Request	Confirm
XX-GET	Refer to 5.2.1	Refer to 5.2.2
XX-SET	Refer to 5.2.3	Refer to 5.2.4

Table 2 - MLME/PLME general management primitive parameters

Name	Type	Valid range	Description
PIBattribute	octet string	Any PIB attribute	PIB attribute name
PIBValue	variable		PIB value
ResultCode	enumeration	SUCCESS, INVALID_PIB_ATTRIBUTE, READ_ONLY_PIB_ATTRIBUTE, WRITE_ONLY_PIB_ATTRIBUTE	Result of MLME or PLME request

5.2.1 MLME-GET.request and PLME-GET.request

These primitives request information about the relevant MAC PIB or PHY PIB. The semantics of these primitives are as follows.

```
XX-GET.request  (
                 PIBattribute
                 )
```

The primitive parameters are defined in Table 2.

5.2.1.1 Time of creation

DME and MLME (in the case of a PLME-GET.request) create these primitives to retrieve information from the MAC or PHY PIB.

5.2.1.2 Effect

The relevant management entity fetches the requested PIB attribute from the database and returns the value as the result of XX-GET.confirm.

5.2.2 MLME-GET.confirm and PLME-GET.confirm

These primitives return the result of an information request to the relevant MAC PIB or PHY PIB. The semantics of these primitives are as follows.

```
XX-GET.confirm  (

                Status,

                PIBattribute,

                PIBattributevalue

                )
```

The primitive parameters are defined in Table 2.

5.2.2.1 Time of creation

DME or MLME (in the case of a PLME-GET.confirm) create these primitives in response to an XX-GET.request.

5.2.2.2 Effect

If the status is SUCCESS, these primitives return the value of the relevant PIB attribute, otherwise they return the error code in the status field. Valid error status values include INVALID_PIB_ATTRIBUTE and WRITE_ONLY_PIB_ATTRIBUTE.

5.2.3 MLME-SET.request and PLME-SET.request

These primitives attempt to set the value of the relevant MAC PIB or PHY PIB attribute to the specified parameter. The semantics of these primitives is as follows.

```
XX-SET.request  (

                PIBattribute,

                PIBattributevalue

                )
```

The primitive parameters are defined in Table 2.

5.2.3.1 Time of creation

These primitives are created when DME or MLME (in the case of PLME-SET.request) tries to set the relevant MAC/PHY PIB attribute.

5.2.3.2 Effect

The relevant management entity tries to alter the value of the PIB attribute in the database. If the PIB is a reference to certain actions, this is interpreted as a request to execute the action. The management entity that receives this command responds by returning the result through a call to XX-SET.confirm.

5.2.4 MLME-SET.confirm and PLME-SET.confirm

These primitives return the result of the attempt to set the MAC PIB or PHY PIB attribute. The semantics of this primitive are as follows.

```
XX-SET.confirm   (
                  Status,
                  PIBattribute
                  )
```

The primitive parameters are defined in Table 2.

5.2.4.1 Time of creation

DME or MLME (in the case of PLME-SET.confirm) create these primitives in order to respond to the XX-SET.request.

5.2.4.2 Effect

If the Status is SUCCESS, this means that the PIB attribute was set to the requested value. Otherwise, the Status field shows the error description. If the specified PIB attribute refers to a certain action, the primitive represents the success or failure of the execution of that action. Possible error status values include INVALID_PIB_ATTRIBUTE and READ_ONLY_PIB_ATTRIBUTE.

5.3 MLME SAP

In this subclause, the services that MLME provides to DME are defined. These definitions are conceptual and do not specify a certain implementation or external interface.

The MLME SAP primitive generally follows the format of an ACTION.confirm in response to an ACTION.request. The ACTION.indication is used to inform DME of events from other stations, and the station selectively sends an ACTION.response to the indication. DME uses the services provided by MLME through MLME SAP, and those primitives are outlined in Table 3.

Table 3 - MLME primitive summary

Name	Request	Indication	Response	Confirm
MLME-RESET	5.3.1.1			
MLME-SCAN	5.3.2.1			5.3.2.2
MLME-START	5.3.3.1			5.3.3.2
MLME-SYNCH	5.3.4.1			5.3.4.2
MLME-ATP-EXPIRED		5.3.4.3		
MLME-ASSOCIATE	5.3.5.1	5.3.5.2	5.3.5.3	5.3.5.4
MLME-STN-ASSOCIATION-INFO		5.3.5.5		
MLME-DISASSOCIATE	5.3.6.1	5.3.6.2		5.3.6.3
MLME-MASER-HANDOVER	5.3.10.1	5.3.10.2	5.3.10.3	5.3.10.4
MLME-MASTER-INFO	5.3.11.1	5.3.11.2	5.3.11.3	5.3.11.4
MLME-PROBE	5.3.12.1	5.3.12.2	5.3.12.3	5.3.12.4
MLME-CREATE-STREAM	5.3.13.10			5.3.13.2
MLME-MODIFY-STREAM	5.3.13.3			5.3.13.4
MLME-TERMINATE-STREAM	5.3.13.5	5.3.13.6		5.3.13.7

MLME-CHANNEL-STATUS	5.3.14.1	5.3.14.2	5.3.14.3	5.3.14.4
MLME-REMOTE-SCAN	5.3.15.1	5.3.15.2	5.3.15.3	5.3.15.4
MLME-NETWORK-PARM-CHANGE	5.3.16.1			5.3.16.2
MLME-TX-POWER-CHANGE	5.3.17.1	5.3.17.2		5.3.17.3
MLME-SLEEP	5.3.18.1	5.3.18.2	5.3.18.3	5.3.18.4

5.3.1 Reset

This mechanism describes the process of resetting the MAC. The parameters of this primitive are defined in Table 4.

Table 4 - MLME-RESET primitive parameters

Name	Type	Valid range	Description
SetDefaultPIB	boolean	TRUE, FALSE	If TRUE, all PIB attributes are reset to their default values. If FALSE, the MAC is reset but all PIB values remain what they were prior to the execution of the MLME-RESET.request primitive
ResetTimeout	duration	0 ~ 65535	The time in which the reset process must be completed, in msec

5.3.1.1 MLME-RESET.request

This primitive requests a reset of the MAC entity. The semantics are as follows.

```
MLME-RESET.request      (

                         SetDefaultPIB,

                         ResetTimeout

                         )
```

The primitive parameters are defined in Table 4.

5.3.1.1.1 Time of creation

DME sends this message to MLME to restore the MAC to its initial state.

5.3.1.1.2 Effect

When the primitive is received, if the station is associated with the current network, MLME sends a disassociate request command to the master (see 6.5.1.3). In all cases, the slave MLME sets the MAC to its initial state, and resets all variables but is still consistent with the SetDefaultPIB (see Table 4).

When the master MLME receives this primitive, it behaves the same way as a slave MLME, except for cases when it has been disassociated from the network after transferring the master role or is performing a shutdown of network.

If the ResetTimeout occurs while MLME is still executing network disassociations, master handover, or network shutdown, MLME resets the MAC and cancels the interrupted process.

5.3.2 Scan

This mechanism determines the existence of the network in the communication channel. The parameters of this primitive are defined in Table 5.

PiconetDescriptionSet is the set of PiconetDescriptions, and a PiconetDescription is composed of the elements in Table 6.

Table 5 - MLME-SCAN primitive parameters

Name	Type	Valid range	Description
OpenScan	boolean	TRUE, FALSE	Whether or not the scan is open (an open scan refers to search for any beacon frame)
NID	integer	0-65535	ID of the specific network to scan
ChannelList	Array of integers	0 – maximum channels supported by the physical layer	List of channels to search during a scan
Channel scanDuration	time	0-65535	The time it takes for the station to scan one channel
NumberOfNetworks	integer	0-255	Number of discovered networks (number of PiconetDescriptions in the PiconetDescriptionSet)
PiconetDescriptionSet	Network Description set	As defined in Table 6	The set of all attributes of the discovered networks, returned as the result of a scan request
NumberOfChannels	integer	0 - maximum channels supported by the physical layer	Number of channels to scan
ChannelRating List	Array of integers	0- maximum channels supported by the physical layer	List of channels sorted by quality (best to worst)
ResultCode	enumeration	SUCCESS, INVALID_PARAMETERS	Result of MLME request

Table 6 - PiconetDescription elements

Name	Type	Valid range	Description
MasterSTNAddress	MACaddress	all MACaddress	MACaddress of the master of the discovered network
NID	As defined in Table 5	As defined in Table 5	NID of the discovered network
ScannedFrameType	enumeration	BEACON, NON-BEACON	Type of the discovered frame
ChannelIndex	integer	0-255	Index of frequency channel on which the network was discovered
SuperframeDu	time	0-65536	Length (msec) of the superframe of the discovered

ration			network
CPEndTime	integer	0-65525	The end time of the contention period of the discovered network

5.3.2.1 MLME-SCAN.request

This primitive requests a scan of the MAC entity. The semantics are as follows.

MLME-SCAN.request (

OpenScan,

NID,

ChannelList

Channel scanDuration

)

The primitive parameters are defined in Table 5.

5.3.2.1.1 Time of creation

When a manual SCAN is initiated for a search of a specified NID or an arbitrary NID of a network, DME sends this message to MLME.

5.3.2.1.2 Effect

When MLME receives this primitive from DME, it executes a manual SCAN of the channels in the Channel List. When this SCAN is completed, MLME responds to DME with the result of the SCAN through a call to MLME-SCAN.confirm.

5.3.2.2 MLME-SCAN.confirm

This primitive and its parameters are collected during the SCAN and sent back upon the completion of the SCAN. The semantics are as follows.

MLME-SCAN.confirm (

NumberOfNetworks,

PiconetDescriptionSet,

NumberOfChannels,

ChannelRatingList,

ResultCode

)

The primitive parameters are defined in Table 5.

5.3.2.2.1 Time of creation

This message is sent to DME when MLME completes the requested SCAN or when the parameters of MLME-request are incorrect.

5.3.2.2.2 Effect

DME sends a notification of the result of the SCAN process.

5.3.3 Startup of network

This mechanism supports the process of creating a new network. The parameters used in this primitive are outlined in Table 7.

Table 7 - MLME-START primitive parameters

Name	Type	Valid range	Description
NID	As defined in Table 5	As defined in Table 5	new network NID
ChannelIndex	As defined in Table 6	As defined in Table 6	Index of frequency channel on which to create network
SuperframeDuration	As defined in Table 6	As defined in Table 6	As defined in Table 6
PiconetMaxTxPower	As defined in 6.4.7	As defined in 6.4.7	As defined in 6.4.7
ResultCode	enumeration	SUCCESS, ALREADY_STARTED, NETWORK_DETECTED, INVALID_PARAMETER, CHANNEL_INTERFE-RENCE	Result of MLME request

5.3.3.1 MLME-START.request

This primitive requests the creation of a new network to the MAC entity. The semantics are as follows.

MLME-START.request (

NID,

ChannelIndex,

SuperframeDuration,

PiconetMaxTXPower

)

The primitive parameters are defined in Table 7.

5.3.3.1.1 Time of creation

This is created when DME requests that a station become the master and startup a new network.

5.3.3.1.2 Effect

MLME begins the creation process described in 7.1, and invokes MLME-START.confirm to report the result.

5.3.3.2 MLME-START.confirm

This primitive reports the result of the network creation.

MLME-START.confirm (

ResultCode

)

The primitive parameters are defined in Table 7.

5.3.3.2.1 Time of creation

This is created as a result of an MLME-START.request from DLME.

5.3.3.2.2 Effect

DME reports the result of the creation process of network. A ResultCode of SUCCESS indicates that the station is now the master. If all channels have already been taken by another Network, the ResultCode is set to NETWORK_DETECTED. If network is already started, ResultCode is set to ALREADY_STARTED. If network cannot be started because the noise on the frequency channel indicated for the startup is too severe, the ResultCode is set to CHANNEL_INTERFERENCE. If there is an error in a parameter, the ResultCode is set to INVALID_PARAMETERS.

MLME begins the creation process described in 7.1, then calls MLME-START.confirm to report the result.

5.3.4 Synchronization

This mechanism is a preparatory step for a station to network. This primitive also notifies DME when a station has gone out-of-sync. The primitive parameters are defined in Table 8.

Table 8 - MLME-SYNCH primitive parameters

Name	Type	Valid range	Description
NID	As defined in Table 5	As defined in Table 5	ID of network to sync
ChannelIndex	As defined in Table 6	As defined in Table 6	The frequency channel on which to search network
Channel scan-Duration	As defined in Table 5	As defined in Table 5	Time to scan network
ResultCode	enumeration	SUCCESS, TIMEOUT, INVALID_PARAMETER	Result of MLME-SYNCH.request

5.3.4.1 MLME-SYNCH.request

This primitive initiates the process of synchronizing with a network. The semantics are as follows.

MLME-SYNCH.request (

NID,

ChannelIndex,

ChannelScanDuration

)

The primitive parameters are defined in Table 8.

5.3.4.1.1 Time of creation

DME creates this when initiating the process of synchronizing with a specific beacon.

5.3.4.1.2 Effect

When MLME receives this primitive from DME, MLME scans until it finds the beacon corresponding to the specific channel or the channel scan duration time is exceeded. If the desired beacon is found, an MLME-SYNCH.confirm is sent to DME with the ResultCode set to SUCCESS. If the channel scan duration time limit is exceeded the ResultCode is set to TIMEOUT.

5.3.4.2 MLME-SYNCH.confirm

This primitive notifies DME of the success or failure of the requested network synchronization. The semantics are as follows.

MLME-SYNCH.confirm (

ResultCode

)

The primitive parameters are defined in Table 8.

5.3.4.2.1 Time of creation

MLME creates this when the requested network synchronization task is completed.

5.3.4.2.2 Effect

DME is notified of the result of the synchronization task.

5.3.4.3 MLME-ATP-EXPIRED.indication

This primitive indicates that the station is no longer able to hear the beacon. The semantics are as follows.

MLME-ATP-EXPIRED.indication ()

5.3.4.3.1 Time of creation

An MLME other than the master creates this when it has been unable to receive a beacon for longer than the Association Timeout Period (ATP).

5.3.4.3.2 Effect

DME receives notice that the ATP has timed out.

5.3.5 Association

The following primitives support the association of stations to network. The primitive parameters are defined in Table 9.

Table 9 - MLME-ASSOCIATE.primitive parameters

Name	Type	Valid range	Description
OrigID	integer	Valid addresses as defined in 6.2.1.3	The address to associate or the address assigned by the master
STNAddress	MACaddress	Valid MACaddress	The MACaddress of the station sending the associate request
CapabilityField	As defined in 6.4.5	As defined in 6.4.5	The capability of the station sending the associate request to the master
Association-TimeoutPeriod	duration	As defined in 6.5.1.1	Maximum association time as defined in 6.5.1.1
Max-Associations	integer	0-255	Maximum number of stations that the master can manage
MaxRRBs	integer	0-255	The maximum number of resource request blocks that the master can manage
TxPowerLevel	As defined in 6.4.7	As defined in 6.4.7	As defined in 6.4.7
STNID	integer	Valid address as defined in 6.2.1.3	If the association is successful, the assigned address of the station, otherwise the assigned association address
Association-Status	enumeration	DISASSOCIATED, ASSOCIATED	Indication of whether the station is newly associating or disassociating
AssocTimeout	time	0-65535	The time in which MLME must receive a reply to the associate request
ReasonCode	integer	As defined in 6.5.1.2	The explanation for the reply to the associate request
ResultCode	enumeration	SUCCESS, DENIED, NEIGHBOR_UNSUPPORTED, TIMEOUT	Indication of whether or not the request was granted or denied, or whether the timeout was exceeded

5.3.5.1 MLME-ASSOCIATE.request

This primitive initiates the association process with a network. The semantics are as follows.

MLME-ASSOCIATE.request (

PiconetType,

CapabilityField,

AssociationTimeoutPerid,

MAXRRBs,

TxPowerLevel,

NetworkserviceInquiry,

AssocTimeout

)

The primitive parameters are defined in Table 9.

5.3.5.1.1 Time of creation

This is created when DME attempts to associate with network.

5.3.5.1.2 Effect

When DME receives this primitive through the MLME-SAP, it creates an associate request command frame and sends it to the master MLME.

5.3.5.2 MLME-ASSOCIATE.indication

This primitive reports that an associate request frame has been received.

MLME-ASSOCIATE.indication (

OrigID,

STNAddress,

CapabilityField,

AssociationTimeoutPeriod,

MaxAssociations

)

The primitive parameters are defined in Table 9.

5.3.5.2.1 Time of creation

This is created to notify DME that the master MLME has received an associate request frame from a currently unassociated station.

5.3.5.2.2 Effect

When DME receives this primitive, if the OrigID is the UnassocID, DME determines whether to grant the associate request of the station, and sends back the result via MLME-ASSOCIATE.response through MLME-SAP.

When DME receives this primitive, if OrigID is the STNID that has just been assigned to the station desiring association, the master is being notified that the station has successfully received the assigned STNID, and does not reply with MLME-ASSOCIATE.response.

5.3.5.3 MLME-ASSOCIATE.response

This primitive is created as a response to an MLME-ASSOCIATE.indication. The semantics are as follows.

MLME-ASSOCIATE.response (

OrigID,

STNAddress,

AssociationTimeoutPeriod,

ReasonCode

)

The primitive parameters are defined in Table 9.

5.3.5.3.1 Time of creation

This is created when the master MLME receives an MLME-ASSOCIATE.indication.

5.3.5.3.2 Effect

When the master MLME receives this primitive from DME, it creates an association response frame and sends it back.

5.3.5.4 MLME-ASSOCIATE.confirm

This primitive informs DME that initiated the association whether the associate request was successfully carried out. The semantics are as follows.

MLME-ASSOCIATE.confirm (

STNID,

AssociationTimeoutPeriod,

ReasonCode,

ResultCode

)

The primitive parameters are defined in Table 9.

5.3.5.4.1 Time of creation

MLME sends this to DME when the association is completed or the timeout is exceeded.

5.3.5.4.2 Effect

The DME that initiated the association receives this primitive and determines whether the associate request was successfully carried out. If successful, it receives a unique STNID, otherwise it remains in an unassociated state with no unique STNID.

5.3.5.5 MLME-STN-ASSOCIATION-INFO.indication

This primitive informs all associated stations in network of the station association information in the beacon. The semantics are as follows.

MLME-STN-ASSOCIATION-INFO.indication (

STNAddress,

STNID,

AssociationStatus

)

The primitive parameters are defined in Table 9.

5.3.5.5.1 Time of creation

MLME sends this to DME when it receives the beacon containing the association information of other stations.

5.3.5.5.2 Effect

DME collects information about the station that has just been associated or disassociated from network through this primitive.

5.3.6 Disassociation

The following primitives are used when a station is disassociated from network or when the master is trying to disassociate a station. The primitive parameters are defined in Table 10.

Table 10 - MLME-DISASSOCIATE primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Valid addresses as defined in 6.2.1.3	Address of destination station of MLME request
OrigID	integer	Valid addresses as defined in 6.2.1.3	Address of station initiating the MLME request
DisassocTimeout	time	0-65535	Time in which the task must be completed once the MLME disassociate request is made
ReasonCode	integer	As defined in 6.5.1.2	Reason for disassociate request
ResultCode	enumeration	SUCCESS, ACK_TIMEOUT	Result of MLME request

5.3.6.1 MLME-DISASSOCIATE.request

This primitive initiates the disassociate request. The semantics are as follows.

```
MLME-DISASSOCIATE.request      (
                                TrgtID,
                                ReasonCode,
                                DisassocTimeout
                               )
```

The primitive parameters are defined in Table 10

5.3.6.1.1 Time of creation

DME sends this to MLME when attempting to start the disassociation process.

5.3.6.1.2 Effect

When the slave MLME receives this primitive, the slave MLME sends a disassociate request command to the master DME.

When the master MLME receives this primitive, it sends a disassociate request command to the DME of the station to disassociate.

5.3.6.2 MLME-DISASSOCIATE.indication

This primitive reports that a disassociate request frame has been received. The semantics are as follows.

```
MLME-DISASSOCIATE.indication   (

                                OrigID,

                                ReasonCode

                                )
```

The primitive parameters are defined in Table 10.

5.3.6.2.1 Time of creation

MLME sends this to DME when the master or a station receives a disassociate request frame.

5.3.6.2.2 Effect

The target DME is notified of the reason for the disassociate request.

5.3.6.3 MLME-DISASSOCIATE.confirm

This primitive reports the result of the disassociate request. The master MLME receives notification of the station to disassociate by means of this primitive. The semantics are as follows.

```
MLME-DISASSOCIATE.request      (

                                TrgtID,

                                ResultCode,

                                )
```

The primitive parameters are defined in Table 10.

5.3.6.3.1 Time of creation

After the disassociate request command is sent, once the acknowledgement frame is received or the acknowledgement timeout is exceeded, MLME sends this to DME. If the MLME initiating the disassociate process receives the acknowledgement frame, it considers the disassociate process to have been successfully carried out, and if the acknowledgement time limit has been exceeded, it considers the process to have failed.

5.3.6.3.2 Effect

The result of the disassociation is reported through this primitive which initiated the disassociation.

5.3.7 Key request

The following primitives are used when a station is requesting and receiving a key from the key originator. The parameters used for these primitives are defined in Table 11.

Table 11 - MLME-REQUEST-KEY primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the target of the MLME request.
OrigID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the station that initiated the MLME request.
SECID	2 octets	As defined in 6.2.2.2	As defined in 6.2.2.2
Key	octet string	Any valid key as defined by the symmetric key security operations,	The key to be used as the current payload protection key for this security relationship. The MAC/MLME encrypts the key before it is placed in the Encrypted Key field and decrypts the field before passing the received key to the DME.
KeyRequestTime-out	duration	0-65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	enumeration	SUCCESS, FAILURE, TIMEOUT	Indicates the result of the MLME request.

5.3.7.1 MLME-REQUEST-KEY.request

This primitive requests the transmission of a key from the key originator by a station. The semantics are as follows.

```
MLME-REQUEST-KEY.request      (
                               TrgtID,
                               KeyRequestTimeout
                               )
```

The primitive parameters are defined in Table 11.

5.3.7.1.1 Time of creation

DME generates this for a station to obtain the designated key from the key originator.

5.3.7.1.2 Effect

The MLME creates a Request Key command, as described in 6.5.4.1, and sends it to the indicated station.

5.3.7.2 MLME-REQUEST-KEY.indication

This primitive reports the request of a key from a station. The semantics are as follows.

```
MLME-REQUEST-KEY.indication    (
                                OrigID,
                                TrgtID,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 11.

5.3.7.2.1 Time of creation

The MLME generates this after receiving a Request Key command, as described in 6.5.4.2. If the integrity code is valid, then the ResultCode is set to SUCCESS. Otherwise, the ResultCode is FAILURE.

5.3.7.2.2 Effect

The DME issues an MLME-REQUEST-KEY.response to the MLME if it receives the MLME-REQUEST-KEY.indication with the ResultCode set to SUCCESS.

5.3.7.3 MLME-REQUEST-KEY.response

This primitive responds to a key request from a station with the requested key by a station. The semantics are as follows.

```
MLME-REQUEST-KEY.response    (
                              OrigID,
                              SECID,
                              Key
                              )
```

The primitive parameters are defined in Table 11.

5.3.7.3.1 Time of creation

The DME generates this after receiving an MLME-REQUEST-KEY.indication primitive with ResultCode equal to SUCCESS where the OrigID corresponds to a station that has established secure membership or a secure relationship with the key originator.

5.3.7.3.2 Effect

The MLME generates a Request Key Response command, as described in 6.5.4.2, and sends it to the specified station. The MLME encrypts the key before transmission.

5.3.7.4 MLME-REQUEST-KEY.confirm

This primitive reports the results of a key request and, if the response was received, the requested key to the DME. The semantics are as follows.

MLME-REQUEST-KEY.confirm (

TrgtID,

SECID,

Key,

ResultCode

)

The primitive parameters are defined in Table 11.

5.3.7.4.1 Time of creation

The MLME generates this after receiving a Request Key Response command, as described in 6.5.4.2, from the key originator of this relationship or due to a timeout. If there is no response from the key originator within KeyRequestTimeout, the ResultCode is set to TIMEOUT. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS and the MLME decrypts the key.

5.3.7.4.2 Effect

The DME is informed of the results of a previously issued key request and, if successful, obtains the requested key. The MLME generates MLME-MEMBERSHIP-UPDATE.request.

5.3.8 Key distribution

The following primitives are used when a station is acting as key originator sending a key to another station. The parameters used for these primitives are defined in Table 12.

Table 12 - MLME-DISTRIBUTE-KEY primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the target of the MLME request.
OrigID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the station that initiated the MLME request.
SECID	2 octets	As defined in 6.2.2.2	As defined in 6.2.2.2
Key	octet string	Any valid key as defined by the symmetric key security operations,	The key to be used as the current payload protection key for this security relationship. The MAC/MLME encrypts the key before it is placed in the Encrypted Key field and decrypts the field before passing the received key to the DME.
DistributeKeyTime-out	duration	0-65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ResultCode	enumeration	SUCCESS,	Indicates the result of the MLME request.

		FAILURE, TIMEOUT	

5.3.8.1 MLME-DISTRIBUTE-KEY.request

This primitive distribute a key to another station by a station. The semantics are as follows.

```
MLME-DISTRIBUTE-KEY.request   (
                                TrgtID,
                                SECID,
                                Key,
                                DistributeKeyTimeout
                                )
```

The primitive parameters are defined in Table 12.

5.3.8.1.1 Time of creation

The DME generates this to distribute a key to a station that has established secure membership or a secure relationship with the key originator.

5.3.8.1.2 Effect

The MLME creates a Distribute Key Request command, as described in 6.5.4.3, and sends it to the indicated station. The MLME encrypts the key before transmission.

5.3.8.2 MLME-DISTRIBUTE-KEY.indication

This primitive reports the reception of a key from a key originator. The semantics are as follows.

```
MLME-DISTRIBUTE-KEY.indication      (
                                OrigID,
                                SECID,
                                Key,
                                ResultCode
                                )
```

The primitive parameters are defined in Table 12.

5.3.8.2.1 Time of creation

The MLME generates this after receiving a Distribute Key Request command, as described in 6.5.4.3. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS and the MLME decrypts the key.

5.3.8.2.2 Effect

If the ResultCode is SUCCESS, the DME will save the designated key and generate an MLME-DISTRIBUTE-KEY.response to acknowledge successful receipt of the key.

5.3.8.3 MLME-DISTRIBUTE-KEY.response

This primitive responds to a key distribution from a key originator with an acknowledgement of successful receipt of the key by a station. The semantics are as follows.

```
MLME-DISTRIBUTE-KEY.response          (
                                       OrigID,
                                       SECID
                                       )
```

The primitive parameters are defined in Table 12.

5.3.8.3.1 Time of creation

The DME generates this after receiving an MLME-DISTRIBUTE-KEY.indication primitive from a peer station. It is not generated by the DME as a result of the receipt of an MLME-DISTRIBUTE-KEY.indication primitive from the master.

5.3.8.3.2 Effect

The MLME generates a Distribute Key Response command, as described in 6.5.4.4, and sends it to the specified station.

5.3.8.4 MLME-DISTRIBUTE-KEY.confirm

This primitive reports the results of a distribute key process with a station. The semantics are as follows.

```
MLME-DISTRIBUTE-KEY.confirm  (
                              TrgtID,
                              SECID,
                              ResultCode
                              )
```

The primitive parameters are defined in Table 12.

5.3.8.4.1 Time of creation

The MLME generates this after receiving a Distribute Key Response command, as described in 6.5.4.4, from another station. If there is no response from the station within DistributeKeyTimeout, the ResultCode is set to TIMEOUT. If the integrity code is not valid, then the ResultCode is set to FAILURE. Otherwise, the ResultCode is SUCCESS.

5.3.8.4.2 Effect

The DME is informed of the results of a previously issued key distribution.

5.3.9 Security management

These primitives initialize, update or delete the security information of a membership or key change process or a security event. Primitives are also provided to transfer security messages. These primitives can be used in an authentication process.

The parameters used for the MLME-MEMBERSHIP-UPDATE are defined in Table 13 and MLME-SECURITY-ERROR primitives are defined in Table 14, MLME-SECURITY-MESSAGE primitive are defined in Table 15.

Table 13 - MLME-MEMBERSHIP-UPDATE primitive parameters

Name	Type	Valid range	Description
SecID	2 octets	As defined in 6.2.2.2	As defined in 6.2.2.2
KeyType	enumeration	MANAGEMENT, DATA, GROUP	Specifies the type of key that is being updated.
TrgtID	integer	Any valid STNID as defined in 6.2.1.3 except for the BcstID, the McstID or the UnassocID	The STNID of the target station for this relationship.
MembershipStatus	enumeration	MEMBER, NON-MEMBER	Indicates the membership status of the TrgtID for the provided SECID.
KeyOriginator	boolean	TRUE, FALSE	This station is the key originator for this relationship.
KeyDelete	boolean	TRUE, FALSE	This key is deleted or installed.
Key	octet string	Any valid key as defined by the symmetric key security operations,	The key used for protecting frames between this station and the TrgtID station.

Table 14 - MLME-SECURITY-ERROR primitive parameters

Name	Type	Valid range	Description
ReceivedMACHeader	octet string	Any valid MAC header, Figure 5.	The MAC header of the received frame that induced a failed security check or for which the station is unable to find the designated key.
ReceivedFramePayload	octet string	Any valid Frame Payload, Figure 7	The received Frame Payload that induces a failed security check or for which the station is unable to find the designated key.
ReasonCode	enumeration	INVALID-MODE, UNAVAILABLE-KEY, FAILED-SECURITY-CHECK, BAD-TIME-TOKEN,	The reason for the security error.

		REPLAYED-FRAME	

Table 15 - MLME-SECURITY-MESSAGE primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the target of the MLME request.
OrigID	integer	Any valid STNID as defined in 6.2.1.3	Specifies the STNID of the originator of the MLME request.
VendorOUI	As defined in 6.5.6	As defined in 6.5.6	As defined in 6.5.6
SecurityInformation	octet string	Any valid octet string	Security information that will be passed from one DME to another peer DME in the network.
SecMsgTimeout	integer	0-65535	The time in milliseconds in which the operation initiated by the MLME request needs to be completed before responding with a ResultCode of TIMEOUT.
ReasonCode	enumeration	COMPLETED, TIMEOUT	Indicated if the MAC/MLME was successful in sending the security message.

5.3.9.1 MLME-MEMBERSHIP-UPDATE.request

This primitive requests the membership status, SECID and keying information associated with a security relationship to be included or updated. The semantics are as follows.

MLME-MEMBERSHIP-UPDATE.request (

TrgtID,

MembershipStatus,

SECID,

KeyType,

KeyOriginator,

KeyDelete,

Key

)

The primitive parameters are defined in Table 14.

5.3.9.1.1 Time of creation

The DME sends this request to the MLME after completing a membership change or key update process with the master or a station.

5.3.9.1.2 Effect

This primitive initiates the membership update procedure defined in 8.4.

5.3.9.2 MLME-SECURITY-ERROR.indication

This primitive allows the MLME of any station to indicate a failed security processing operation to the DME. The semantics are as follows.

```
MLME-SECURITY-ERROR.indication          (
                                         ReceivedMACHeader,
                                         Received FramePayload,
                                         ReasonCode
                                         )
```

The primitive parameters are defined in Table 14.

5.3.9.2.1 Time of creation

The MLME issues this when it receives a frame, but security mode is not correct or it is unable to find an appropriate key or fails the security check according to the symmetric key security operations. This primitive is also issued if the time token in the beacon was not within the range of valid time tokens, or replayed frame was received.

5.3.9.2.2 Effect

When a station receives this primitive, the DME is promptly notified of a security error and the reason for the security error and then analyze the reason for the security error.

5.3.9.3 MLME-SECURITY-MESSAGE.request

This primitive sends a security message to a station in the network. The semantics are as follows.

```
MLME-SECURITY-MESSAGE.request           (
                                         TrgtID,
                                         VendorOUI,
                                         SecurityInformation,
                                         SecMsgTimeout
                                         )
```

The primitive parameters are defined in Table 15.

5.3.9.3.1 Time of creation

The DME generates this to send security related information to another station in the network.

5.3.9.3.2 Effect

The MLME creates a Security Message command, as described in 6.5.5.1, and sends it to the appropriate station.

5.3.9.4 MLME-SECURITY-MESSAGE.indication

This primitive reports the reception of a Security Message command, as described in 6.5.5.1 from a station. The semantics are as follows.

```
MLME-SECURITY-MESSAGE.indication      (
                                       TrgtID,
                                       TrgtID,
                                       OrigID,
                                       VendorOUI,
                                       SecurityInformation
                                       )
```

The primitive parameters are defined in Table 15.

5.3.9.4.1 Time of creation

The MLME generates this after receiving a valid Security Message command from a station.

5.3.9.4.2 Effect

The DME receives the security related information. The use of this information is outside of the scope of this standard.

5.3.9.5 MLME-SECURITY-MESSAGE.confirm

This primitive reports the result of an attempt send security information to another station. The semantics are as follows.

```
MLME-SECURITY-MESSAGE.confirm         (
                                       ResultCode
                                       )
```

The primitive parameters are defined in Table 15.

5.3.9.5.1 Time of creation

The MLME generates this after receiving an Imm-ACK for the Security Message command or due to a timeout. If the Imm-ACK is not received within SecMsgtTimeout, the ResultCode is TIMEOUT. Otherwise, the ResultCode is COMPLETED.

5.3.9.5.2 Effect

The DME is informed of whether the message was successfully sent and ACKed or not.

5.3.10 Master handover

The following primitives are used when the current master transfers its authority to the most capable station within network. The primitive parameters are defined in Table 16.

Table 16 - MLME-Master-HANDOVER primitive parameters

Name	Type	Valid range	Description
MasterCapable-STNID	integer	Valid STNID as defined in 6.2.1.3	STNID of station to which the master wants to handover its authority
NumberOfSTNs	integer	0-255	Number of stations in network
NmbrHndOvrBcns	integer	(1+maximum beacon loss)-255	Number of times that a beacon containing the relevant information blocks must be broadcasted prior to the actual handover of master authority
HandoverTimeout	time	0-65535	Time limit in which the MLME request must be actually executed
HandoverStatus	enumeration	STARTED, CANCELLED	Indicates whether the handover of the master role to another station has been started or cancelled
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.10.1 MLME-MASTER-HANDOVER.request

This primitive requests the start of the handover of the master authority to another station. The semantics are as follows.

```
MLME-MASTER-HANDOVER.request      (
                                 MasterCapableSTNID
                                 NmbrHdnOvrBcns,
                                 HandoverStatus,
                                 HandoverTimeout
                                 )
```

The primitive parameters are defined in Table 16.

5.3.10.1.1 Time of creation

This primitive is sent to MLME when the master DME tries to handover its authority to another station or cancels the handover operation.

5.3.10.1.2 Effect

When MLME receives this primitive, it sends a master handover command frame to the destination station.

5.3.10.2 MLME-MASTER-HANDOVER.indication

This primitive reports that a master handover command frame has been received. The semantics are as follows.

```
MLME-MASTER-HANDOVER.indication     (

                                     NumberOfSTNs,

                                     HandoverStatus,

                                     )
```

The primitive parameters are defined in Table 16.

5.3.10.2.1 Time of creation

When MLME receives the master handover command frame from the master, it sends this primitive to DME. It also sends this primitive to DME the first time it broadcasts the beacon as the new master.

5.3.10.2.2 Effect

If HandoverStatus is STARTED, DME receives notice that MLME has begun the process of master handover, and if HandoverStatus is CANCELED, DME learns that the master handover process has been cancelled.

5.3.10.3 MLME-MASTER-HANDOVER.response

This primitive initiates the response to an MLME-MASTER-HANDOVER.indication. The semantics are as follows.

```
MLME-MASTER-HANDOVER.response       ()
```

5.3.10.3.1 Time of creation

Once the master DME receives an MLME-MASTER-HANDOVER.indication with a HandoverStatus of SUCCESS and an MLME-MASTER-INFO.confirm, the master DME sends this primitive to MLME.

5.3.10.3.2 Effect

When the new master MLME receives this primitive from DME, it is notified that DME is ready to take on the role of master.

5.3.10.4 MLME-MASTER-HANDOVER.confirm

This primitive notifies the requesting DME that the job is completed. The semantics are as follows.

```
MLME-MASTER-HANDOVER.confirm        (

                                 ResultCode

                                 )
```

The primitive parameters are defined in Table 16.

5.3.10.4.1 Time of creation

After the master MLME hands over its master function and sends its last beacon, it sets ResultCode to SUCCESS and sends this primitive to DME. If the master was unable to send the desired data to the destination station, it sends a primitive to DME with ResultCode set to TIMEOUT.

5.3.10.4.2 Effect

The master DME is notified of the success or failure of the MLME-MASTER-HANDOVER.request. If the master sends the last beacon before the master handover timeout, ResultCode is set to SUCCESS, otherwise it is set to TIMEOUT.

5.3.11 Data request

This mechanism is used when a station requests information from the master about a specific station or all stations in network. The primitive parameters are defined in Table 17.

Table 17 - MLME-MASTER-INFO primitive parameters

Name	Type	Valid range	Description
QueriedSTNID	integer	Valid addresses as defined in 6.2.1.3	ID of the station about which information is being requested from the master (if set to the broadcasting address, this indicates a request for information about all stations)
OrigID	integer	Valid addresses as defined in 6.2.1.3	ID of station initiating the MLME request
STNInfoSet	As defined in 6.5.7.2	As defined in 6.5.7.2	Set of information about the station being queried
MasterInfo-Timeout	time	0-65535	Time in which the task must be completed once the MLME receives the request
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.11.1 MLME-MASTER-INFO.request

This initiates the process of requesting information concerning a specific station or all stations. The semantics are as follows.

```
MLME-MASTER-INFO.request      (
                               QueriedSTNID,
                               MasterInfoTimeout
                               )
```

The primitive parameters are defined in Table 17.

5.3.11.1.1 Time of creation

This primitive is sent to MLME when DME requests information from the master about a specific station or all stations.

5.3.11.1.2 Effect

When MLME receives this primitive, it sends the station information request frame to the master.

5.3.11.2 MLME-MASTER-INFO.indication

This primitive indicates that the station information request command frame has been received. The semantics are as follows.

```
MLME-MASTER-INFO.indication    (
                                QueriedSTNID,
                                OrigID
                                )
```

The primitive parameters are defined in Table 17.

5.3.11.2.1 Time of creation

MLME sends this to DME upon receiving the station information request frame.

5.3.11.2.2 Effect

When DME receives this primitive, it responds to MLME with MLME-MASTER-INFO.response.

5.3.11.3 MLME-MASTER-INFO.response

This primitive initiates a response to MLME-MASTER-INFO.indication. The semantics are as follows.

```
MLME-MASTER-INFO.response    (
                              OrigID,
                              STNInfoSet
                              )
```

The primitive parameters are defined in Table 17.

5.3.11.3.1 Time of creation

DME sends this primitive to MLME as a response to MLME-MASTER-INFO.indication.

5.3.11.3.2 Effect

When the master MLME receives this primitive, it sends the station information response frame to the requesting station.

5.3.11.4 MLME-MASTER-INFO.confirm

This primitive informs the DME which initiated the station information probe that the process has been completed. The semantics are as follows.

```
MLME-MASTER-INFO.confirm    (
                        TrgtID,
                        InfoElementList,
                        ResultCode
                        )
```

The primitive parameters are defined in Table 17.

5.3.11.4.1 Time of creation

MLME sends this primitive to DME when the station information response command frame is received or the time limit has been exceeded.

5.3.11.4.2 Effect

When the DME requesting the station information receives this primitive, it learns the success or failure of the station information request it sent to the master. If failure, it retransmits another MLME-MASTER-INFO.request for the same information. If success, DME receives the information it requested.

5.3.12 Network node data probe

The following primitives are used when a station requests information about another station in network. The primitive parameters are defined in Table 18.

Table 18 - MLME-PROBE primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Valid addresses as defined in 6.2.1.3	Address of destination station of the MLME request
OrigID	integer	Valid addresses as defined in 6.2.1.3	Address of station initiating the MLME request
InfoElementMap	2 octets	0x0000 – 0xFFFF	Information request as defined in 6.5.7.1
InfoElementList	Variable length of octet	-	Response to information request as defined in 6.5.7.2
ProbeTimeout	time	0-65535	Time limit in which the task must be completed once the MLME has made the request
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.12.1 MLME-PROBE.request

This primitive initiates the process of requesting specific information from a destination station. The semantics are as follows.

```
MLME-PROBE.request     (
                       TrgtID,
                       InfoElementMap,
                       InfoElementList,
                       ProbeTimeout
                       )
```

The primitive parameters are defined in Table 18.

5.3.12.1.1 Time of creation

This primitive is sent to MLME when DME requests information about another station in network.

5.3.12.1.2 Effect

When MLME receives this primitive, it sends an information request command frame to the destination station.

5.3.12.2 MLME-PROBE.indication

This primitive reports that there has been a request made for specific information through an information probe frame. The semantics are as follows.

```
MLME-PROBE.indication  (
                          OrigID,
                          InfoElementMap
                          )
```

The primitive parameters are defined in Table 18.

5.3.12.2.1 Time of creation

MLME sends this to DME when it receives the information probe frame.

5.3.12.2.2 Effect

When DME receives the MLME-PROBE.indication, it responds to MLME through MLME-PROBE.response.

5.3.12.3 MLME-PROBE.response

This primitive initiates a response to MLME-PROBE.indication. The semantics are as follows.

```
MLME-PROBE.response  (
                        OrigID,
                        InfoElementMap,
                        InfoElementList,
                        ProbeTimeout
                        )
```

The primitive parameters are defined in Table 18.

5.3.12.3.1 Time of creation

DME sends this primitive to MLME as a response to MLME-PROBE.indication.

5.3.12.3.2 Effect

When MLME receives this primitive, it sends the information probe command frame to the station that initiated the information probe.

5.3.12.4 MLME-PROBE.confirm

This primitive notifies the DME that initiated the information probe that the process has completed. The semantics are as follows.

```
MLME-PROBE.confirm      (

                        TrgtID,

                        InfoElementList,

                        ResultCode

                        )
```

The primitive parameters are defined in Table 18.

5.3.12.4.1 Time of creation

MLME sends this primitive to DME when it receives the information probe command frame or the time limit has been exceeded.

5.3.12.4.2 Effect

When the DME that requested the information probe receives this primitive, it is notified of the success or failure of the information probe. If failure, another MLME-PROBE.request for the same information is resent. If success, DME is notified of the requested information.

5.3.13 Strean creation, modification, termination

This mechanism supports the creation, modification, and shutdown processes of the resources allocated for streams. The primitive parameters are defined in Table 19.

Table 19 - MLME-CREATE-STREAM, MLME-MODIFY-STREAM, MLME-TERMINATE-STREAM primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Valid addresses as defined in 6.2.1.3	Address of destination station of MLME request
RequestTimeout	time	0-65535	Time limit in which the task must be completed once MLME has made the request
StreamIndex	integer	1-255	Index of stream to create, modify, or terminate
ACKPolicy	enumeration	IMM_ACK, NO_ACK, DLY_ACK	acknowledgement policy to be used by stream
RR-DesiredData	integer	As defined in 6.5.2	As defined in 6.5.2
RR-Period	time	As defined in 6.5.2	As defined in 6.5.2
RR-BER	2 octets	As defined in 6.5.2	As defined in 6.5.2
AllocatedTime	integer	0-65535	Length of periodically allocated time slot, in msec
StreamReqID	integer	0-255	As defined in 6.5.2

ReasonCode	integer	As defined in 6.5.1.2	As defined in 6.5.2
ResultCode	enumeration	SUCCESS, FAILURE, ILLEGAL_DENIED, TIMEOUT	Result of MLME request

5.3.13.1 MLME-CREATE-STREAM.request

This primitive is used to request the allocation of resources. The semantics are as follows.

```
MLME-CREATE-STREAM.request  (
                              TrgtID,
                              StreamReqID,
                              ACKPolicy,
                              Priority,
                              RR-Period,
                              RR-DesiredData,
                              RR-BER
                              )
```

The primitive parameters are defined in Table 19.

5.3.13.1.1 Time of creation

This is created when DME tries to initiate the process of negotiating resource allocation between the slave and master. The purpose of this process is to create a synchronized data stream between stations in network. If streams that are multicast or broadcast try to use an acknowledgment other than No-ACK, it does not send a resource allocation request command frame to the master MLME and instead sets ResultCode to ILLEGAL_ACK_POLICY and returns this in a MLME-CREATE-STREAM.confirm.

5.3.13.1.2 Effect

When MLME receives this primitive through the MLME-SAP, it creates a resource allocation request command and sends it to the master MLME.

5.3.13.2 MLME-CREATE-STREAM.confirm

This primitive is used to grant or deny a resource allocation request. The semantics are as follows.

```
MLME-CREATE-STREAM.confirm          (
                                      StreamReqID,
                                      StreamIndex,
                                      AllocatedTime,
                                      ResultCode
```

)

The primitive parameters are defined in Table 19.

5.3.13.2.1 Time of creation

The MLME sends this primitive to DME when one of the following events occurs.

- Timeout exceeded

- Resource allocation response command frame indicating that the request has been denied

- Resource allocation response command frame indicating that the request has been granted, and a beacon containing the resource allocation information

5.3.13.2.2 Effect

When DME receives this primitive, it learns the success or failure of the stream request.

5.3.13.3 MLME-MODIFY-STREAM.request

This primitive is used to modify a stream that currently has allocated resources. The semantics are as follows.

```
MLME-MODIFY-STREAM.request   (
                              StreamID,
                              RR-Period,
                              RR-DesiredData,
                              RR-BER
                              )
```

The primitive parameters are defined in Table 19.

5.3.13.3.1 Time of creation

This primitive is created when DME wants to modify current resource allocation for a stream.

5.3.13.3.2 Effect

When MLME receives this primitive through the MLME-SAP, it creates a resource allocation request command and sends to the master MLME.

5.3.13.4 MLME-MODIFY-STREAM.confirm

This primitive is used to report the granting or denial of a resource allocation modification request. The semantics are as follows.

```
MLME-MODIFY-STREAM.confirm          (
                                     StreamIndex,
                                     AllocatedTime,
                                     ResultCode
```

)

The primitive parameters are defined in Table 19.

5.3.13.4.1 Time of creation

The MLME sends this primitive to DME when one of the following events occurs.

- Timeout exceeded

- Resource allocation response command frame indicating that the request has been denied

- Resource allocation response command frame indicating that the request has been granted, and a beacon containing the modified resource allocation information

5.3.13.4.2 Effect

When DME receives this primitive, it learns the success or failure of the stream request.

5.3.13.5 MLME-TERMINATE-STREAM.request

This primitive is used to request the termination of a stream. The semantics are as follows.

```
MLME-TERMINATE-STREAM.request          (
                                        StreamIndex,
                                        RequestTimeout
                                        )
```

The primitive parameters are defined in Table 19.

5.3.13.5.1 Time of creation

This is created when MLME requests the termination of an existing stream.

5.3.13.5.2 Effect

When the station MLME receives this primitive, it sets the parameters pertaining to the stream termination and sends a resource allocation request command.

5.3.13.6 MLME-TERMINATE-STREAM.indication

This primitive reports to the station that the stream has been terminated. The semantics are as follows.

```
MLME-TERMINATE-STREAM.indication       (
                                        StreamIndex
                                        )
```

The primitive parameters are defined in Table 19.

5.3.13.6.1 Time of creation

The station MLME sends this to DME after receiving a beacon containing resource allocation information about a stream with not currently allocated resources. It also sends the primitive to the station when it is unable to allocate resources for a period exceeding the specified time limit.

5.3.13.6.2 Effect

DME is notified of the termination of the resource allocation corresponding to the StreamIndex by means of this primitive.

5.3.13.7 MLME-TERMINATE-STREAM.confirm

This primitive is used to report the success or failure of the stream termination request. The semantics are as follows.

```
MLME-TERMINATE-STREAM.confirm         (
                                    StreamIndex,
                                    ResultCode
                                    )
```

The primitive parameters are defined in Table 19.

5.3.13.7.1 Time of creation

This primitive is sent to DME when the station MLME has received an acknowledgement frame for the stream termination resource allocation request frame, or the request time limit has been exceeded.

5.3.13.7.2 Effect

When DME receives this primitive, it learns the success or failure of the stream termination request.

5.3.14 Channel state

This primitive provides the ability to verify a frequency channel state. The primitive parameters are defined in Table 20.

Table 20 - MLME-CHANNEL-STATUS primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	As defined in 6.2.1.3	STNID targeted by the MLME request
OrigID	integer	As defined in 6.2.1.3	STNID initiating the MLME request
MeasurementWindow-Size	integer	0-65535	As defined in 6.5.7.4
TXFrameCount	integer	0-65535	As defined in 6.5.7.4
RXFrameCount	integer	0-65535	As defined in 6.5.7.4
RXFrameErrorCount	integer	0-65535	As defined in 6.5.7.4
RXFrameLostCount	integer	0-65535	As defined in 6.5.7.4
ChannelStatus-Timeout	time	0-65535	Time limit for the completion of the response to the MLME request, in msec
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.14.1 MLME-CHANNEL-STATUS.request

This primitive initiates the process of verifying the channel between two stations in a network. The semantics are as follows.

```
MLME-CHANNEL-STATUS.request (
                                TrgtID,
                                ChannelStatusTimeout
                                )
```

The primitive parameters are defined in Table 20.

5.3.14.1.1 Time of creation

DME sends this to MLME when it wants to know the state of the channel between itself and the TrgtID station.

5.3.14.1.2 Effect

When MLME receives this primitive from DME, it sends a channel state request command frame to the TrgtID station.

5.3.14.2 MLME-CHANNEL-STATUS.indication

This primitive reports that a channel state request command frame has been received. The semantics are as follows.

```
MLME-CHANNEL-STATUS.indication      (
                                    OrigID,
                                    )
```

The primitive parameters are defined in Table 20.

5.3.14.2.1 Time of creation

MLME sends this primitive to DME when it receives a channel state request command frame.

5.3.14.2.2 Effect

When DME receives this primitive, it sends a MLME-CHANNEL-STATUS.response to MLME.

5.3.14.3 MLME-CHANNEL-STATUS.response

This primitive indicates DME’s response to a MLME-CHANNEL-STATUS.indication. The semantics are as follows.

```
MLME-CHANNEL-STATUS.response        (
                                    OrigID,
                                    MeasurementWindowSize,
                                    TxFrameCount,
                                    RxFrameCount,
```

RxFrameErrorCount,

RxFrameLostCount

)

The primitive parameters are defined in Table 20.

5.3.14.3.1 Time of creation

DME responds to MLME with this primitive when it receives a MLME-CHANNEL-STATUS.indication.

5.3.14.3.2 Effect

MLME creates a channel state response command frame and sends it to the requesting station.

5.3.14.4 MLME-CHANNEL-STATUS.confirm

This primitive reports to the DME initiating the channel state request that the request has been completed. The semantics are as follows.

MLME-CHANNEL-STATUS.confirm (

TrgtID,

MeasurementWindowSize,

TxFrameCount,

RxFrameCount,

RxFrameErrorCount,

RxFrameLostCount,

ResultCode

)

The primitive parameters are defined in Table 20.

5.3.14.4.1 Time of creation

MLME sends this primitive to DME when it receives the channel state response command frame or the time limit has been exceeded.

5.3.14.4.2 Effect

The DME that initiated the request learns of the success or failure of the channel state request.

5.3.15 Remote scan

These primitives are used by a master to make a slave to perform channel scan on behalf of it. The primitive parameters are defined in Table 21.

Table 21 - MLME-REMOTE-SCAN primitive parameters

Name	Type	Valid range	Description

TrgtID	integer	As defined in 6.2.1.3	STNID of target STNID
ChannelList	As defined in Table 5	As defined in Table 5	List of channels to be scanned
Remote scan-Timeout	time	0-65535	Time limit for responding to the master, in msec
NumberOfNetworks	integer	0-255	As defined in 6.5.7.7
RemotePiconet-DescriptionSet	RemotePiconetDescription Set	As defined in Table 22	Other network information in the response for the scan result
NumberOfChannels	As defined in Table 5	As defined in Table 5	As defined in Table 5
ChannelRatingList	As defined in Table 5	As defined in Table 5	As defined in 6.5.7.7
ReasonCode	integer	As defined in 6.5.7.7	As defined in 6.5.7.7
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

Table 22 - RemotePiconetDescription elements

Name	**Type**	**Valid range**	**Description**
MasterSTNAddress	MACaddress	valid MACaddress	Master MACaddress of discovered network
NID	integer	As defined in Table 5	NID of discovered network
ScannedFrameType	enumeration	BEACON, NON-BEACON	Discovered frame type
ChannelIndex	integer	0-255	Frequency channel of discovered network

5.3.15.1 MLME-REMOTE-SCAN.request

This primitive is used by the master to request the channel scan of a station. The semantics are as follows.

```
MLME-REMOTE-SCAN.request      (
                               TrgtID,
                               ChannelList,
                               Remote scanTimeout
                               )
```

The primitive parameters are defined in Table 21.

5.3.15.1.1 Time of creation

This primitive is sent to MLME when the master requests the remote channel scan of the TrgtID station.

5.3.15.1.2 Effect

When MLME receives this primitive, it sends a remote channel scan request command frame to the specified station.

5.3.15.2 MLME- REMOTE-SCAN.indication

This primitive reports that a remote channel scan request command frame has been received from the master. The semantics are as follows.

```
MLME-REMOTE-SCAN.indication   (

                                  ChannelList

                                  )
```

The primitive parameters are defined in Table 21.

5.3.15.2.1 Time of creation

MLME sends this primitive to DME when it receives a remote scan request command frame.

5.3.15.2.2 Effect

DME sends MLME either a MLME-SCAN.request to execute the requested channel scan, or a MLME-REMOTE-SCAN.response with a ResultCode indicating that the remote scan could not be executed.

5.3.15.3 MLME- REMOTE-SCAN.response

This primitive is DME's response to a MLME-REMOTE-SCAN.indication. The semantics are as follows.

```
MLME-REMOTE-SCAN.response   (

                                NumberOfNetworks,

                                RemotePiconetDescriptionSet,

                                NumberOfChannels,

                                ChannelRatingList,

                                ReasonCode

                                )
```

The primitive parameters are defined in Table 21.

5.3.15.3.1 Time of creation

This primitive is sent to MLME after either DME denies the MLME-REMOTE-SCAN.request or grants the request, performs the MLME-SCAN.request, and receives a MLME-SCAN.confirm.

5.3.15.3.2 Effect

When MLME receives this primitive, it sends a remote channel scan response command frame to the specified station.

5.3.15.4 MLME- REMOTE-SCAN.confirm

This primitive reports the completion of the request to the master to perform a channel scan of a destination station and obtain the results. The semantics are as follows.

```
MLME-REMOTE-SCAN.confirm      (
                               TrgtID,
                               NumberOfNetworks,
                               RemotePiconetDescriptionSet,
                               NumberOfChannels,
                               ChannelRatingList,
                               ReasonCode,
                               ResultCode
                               )
```

The primitive parameters are defined in Table 21.

5.3.15.4.1 Time of creation

MLME sends this primitive to DME when the remote channel scan response command frame has been received or the time limit has been exceeded.

5.3.15.4.2 Effect

The DME initiating the request is notified of the success of failure of its request. If unsuccessful and the ResultCode is TIMEOUT, the scan request is resent to the same station, and if the ResultCode is REQUEST_DENIED, it is resent to a different station. If successful, DME is notified of the result of the channel scan performed by the destination station.

5.3.16 Network parameter modification

These primitives enable the master to modify network attributes. The primitive parameters are defined in Table 23.

Table 23 - MLME-NETWORK-PARM-CHANGE primitive parameters

Name	Type	Valid range	Description
NewChannelIndex	integer	Dependent on the physical layer	New channel frequency of network
NmbrOfChange-Beacons	integer	0-255	Number of superframes to repeat the information about the network parameter modification
SuperframeLength	time	0-65535	Superframe length, in msec
ChangeType	enumeration	CHANNEL, SIZE, NID,POWER	Indicates what to modify
NID	integer	0-65535	network ID
PiconetMaxTX-Power	As defined in 6.4.7	As defined in 6.4.7	Maximum transmit power allowed within network

ResultCode	enumeration	SUCCESS, TIMEOUT, INVALID_PARAMETERS	Result of MLME request

5.3.16.1 MLME-NETWORK-PARM-CHANGE.request

This primitive is used to initiate the process of changing the frequency channel, superframe length, maximum network transmit power, or NID. The semantics are as follows.

```
MLME-NETWORK-PARM-CHANGE.request (
                                   ChangeType,
                                   NmbrOfChangeBeacons,
                                   NewChannelIndex,
                                   SuperframeLength,
                                   NID,
                                   PiconetMaxTxPower
                                   )
```

The primitive parameters are defined in Table 23.

5.3.16.1.1 Time of creation

This primitive is sent to MLME when the master DME decides to modify a network parameter.

5.3.16.1.2 Effect

If ChangeType is Power, the master MLME sets the maximum transmit power field included in the synchronization information of the beacon to PiconetMaxTXPower. In all other cases the appropriate action is executed depending on the ChangeType parameter.

5.3.16.2 MLME-NETWORK-PARM-CHANGE.confirm

This primitive verifies that the MLME-NETWORK-PARM-CHANGE.request has been executed. The semantics are as follows.

```
MLME-NETWORK-PARM-CHANGE.confirm (
                                   ResultCode
                                   )
```

The primitive parameters are defined in Table 23.

5.3.16.2.1 Time of creation

The master MLME sends out a beacon with the modified parameters, and sends this primitive to DME.

5.3.16.2.2 Effect

The master DME is notified through this primitive that the network parameter modification has been completed.

5.3.17 Adjustment of power

This mechanism supports the ability to increase or decrease the transmission power of the station. The primitive parameters are defined in Table 24.

Table 24 - MLME-TX-POWER-CHANGE primitive parameters

Name	Type	Valid range	Description
TgrtID	integer	As defined in 6.2.1.3	STNID of destination station
TxPowerChange-Value	integer	-127-127	Requested amount of change to transmit power, in dB
TxPowerChange-Timeout	time	0-65535	Time limit for the response to the MLME request to be completed
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.17.1 MLME-TX-POWER-CHANGE.request

This primitive requests the increase or decrease in the transmit power of the destination station. The semantics are as follows.

```
MLME-TX-POWER-CHANGE.request          (

                                      TrgtID,

                                      TxPowerChangeValue,

                                      TxPowerChangeTimeout

                                      )
```

The primitive parameters are defined in Table 24.

5.3.17.1.1 Time of creation

This primitive is sent to MLME when DME wishes to modify the transmit power of the destination station.

5.3.17.1.2 Effect

MLME sends a transmit power change request command frame to the destination station.

5.3.17.2 MLME-TX-POWER-CHANGE.indication

This primitive indicates that the transmit power change request command frame has been received. The semantics are as follows.

```
MLME-TX-POWER-CHANGE.indication       (

                                      TxPowerChangeValue

                                      )
```

The primitive parameters are defined in Table 24.

5.3.17.2.1 Time of creation

When MLME receives the transmit power change request command frame, it sends this primitive to the station.

5.3.17.2.2 Effect

DME receives this primitive and depending on the request, either modifies the power or ignores the request.

5.3.17.3 MLME-TX-POWER-CHANGE.confirm

This primitive indicates that the transmit power change request has been completed. The semantics are as follows.

MLME-TX-POWER-CHANGE.confirm (

ResultCode

)

The primitive parameters are defined in Table 24.

5.3.17.3.1 Time of creation

MLME sends this primitive to DME when it receives an acknowledgement or the acknowledgement time limit has been exceeded.

5.3.17.3.2 Effect

The requesting DME is notified of the success or failure of the power adjustment request. In case the power adjustment fails, such as in the event that the acknowledgement is not received within the time limit, DME may retransmit the MLME-TX-POWER-CHANGE.request. If successful, DME receives notification that the power adjustment request task was successfully carried out by receiving a primitive with a ResultCode of SUCCESS.

5.3.18 Power saving

This mechanism supports a means of saving power by putting a station into sleep state. The primitive parameters are defined in Table 25.

Table 25 - MLME-SLEEP primitive parameters

Name	Type	Valid range	Description
OrigID	integer	As defined in 6.2.1.3	STNID making the MLME request
SleepPeriod	time	0-65535	Period of sleep state
SleepReasonCode	integer	As defined in 6.5.3.2	As defined in 6.5.3.2
SleepTimeout	time	0-65535	Time limit for the completion of the task once MLME makes the request
ResultCode	enumeration	SUCCESS, TIMEOUT	Result of MLME request

5.3.18.1 MLME-SLEEP.request

This primitive requests the master to allow a station to go to sleep. The semantics are as follows.

MLME-SLEEP.request (

SleepPeriod,

SleepTimeout

)

The primitive parameters are defined in Table 25.

5.3.18.1.1 Time of creation

This is created when a station sends a request to the master to go to sleep.

5.3.18.1.2 Effect

When this primitive is received from DME, MLME sends a sleep state request command frame to the master.

5.3.18.2 MLME-SLEEP.indication

This primitive indicates that the master DME has received a sleep state request command frame. The semantics are as follows.

MLME-SLEEP.indication (

OrigID,

SleepPeriod,

)

The primitive parameters are defined in Table 25.

5.3.18.2.1 Time of creation

This is created when the master MLME receives a sleep state request command frame.

5.3.18.2.2 Effect

When MLME receives this primitive, the master DME uses MLME-SLEEP.response to respond. The SleepPeriod is then readjusted to be a multiple of the current superframe length.

5.3.18.3 MLME-SLEEP.response

This primitive is executed by the master as a response to a sleep state request. The semantics are as follows.

MLME-SLEEP.response (

OrigID,

SleepPeriod,

SleepReasonCode

)

The primitive parameters are defined in Table 25.

5.3.18.3.1 Time of creation

This is created when the master DME receives a MLME-SLEEP.indication from MLME.

5.3.18.3.2 Effect

When MLME receives this primitive, it sends a sleep state request frame to the requesting station.

5.3.18.4 MLME-SLEEP.confirm

This primitive reports the completion of the sleep state request to the requesting DME. The semantics are as follows.

```
MLME-SLEEP.confirm       (

                        SleepPeriod,

                        SleepReasonCode,

                        ResultCode

                        )
```

The primitive parameters are defined in Table 25.

5.3.18.4.1 Time of creation

MLME sends this primitive to DME when the sleep state request command frame is received or the time limit has been exceeded.

5.3.18.4.2 Effect

When the DME sending the sleep state request receives this primitive, it learns the success or failure of the sleep state request. If TIMEOUT, it retransmits another MLME-SLEEP.request with a request for the same set of information.

5.4 MAC management

In order to manage the MAC sublayers of a station, the MAC PIB is composed of managed object, attributes, actions, and notifications. The MAC PIB is divided into six groups: master, attributes, association, network security, and media access list. In the following table, ‘static’ indicates that the parameter generally does not change, and ‘dynamic’ indicates that when the station is active the parameter may change.

5.4.1 MAC PIB master group

The MAC PIB master group in Table 26 incorporates the station’s master capability and current network attributes.

Table 26 - MAC PIB master group parameters

Managed object	Octets	Definition	Type
MACPIB_CFPDuration	2	CFP length	dynamic
MACPIB_SuperframeDuration	2	Superframe length	dynamic
MACPIB_MasterCapable	1 bit	1 if the station is capable of being master, otherwise 0	static
MACPIB_MasterDesMode	1 bit	1 if the station desires to be master	dynamic
MACPIB_MaxPSSets	1	The maximum number of PS sets supported by master	static
MACPIB_MaxAssociations	2	Maximum number of slaves that can be managed if the	static

		station becomes master	
MACPIB_MaxCTBs	2	Maximum number of resource blocks that can be managed if the station becomes master	static

5.4.2 MAC PIB attributes group

The MAC PIB attribute group in Table 27 incorporates the station’s capabilities and attributes.

Table 27 - MAC PIB attribute group parameters

Managed object	**Octets**	**Definition**	**Type**
MACPIB_STNAddress	6	MACaddress of station	static
MACPIB_STNID	1	station ID	dynamic
MACPIB_PowerManagementMode	1	power management mode of the station. 0x00: ACTIVE 0x01: PSPS 0x02: SPS 0x03: PSPS and SPS 0x04: HIBERNATE	dynamic
MACPIB_SleepSupported	1	0x00: does not support sleep mode 0x01: supports sleep mode	static
MACPIB_MaxStreams	1	Maximum number of streams that the station can manage	static
MACPIB_PowerSource	2	0x00: wireline power supply 0x01: battery power supply	dynamic

5.4.3 MAC PIB authentication group

The MAC PIB authentication group in Table 28 incorporates the authentication information of the station in the current network.

Table 28 - MAC PIB authentication group parameters

Managed object	Octets	Definition	Type
MACPIB_SecurityOptionImplemented	1	0x00: security mode 0 0x01: security mode 1 0x02: security mode 2	static
MACPIB_AuthenticationResultCode	1	0x00: SUCCESS 0x01: FAILURE 0x02: TIMEOUT	dynamic

5.4.4 MAC PIB association group

The MAC PIB association group in Table 29 incorporates the association and disassociation information of the station in the current network.

Table 29 - MAC PIB association group parameters

Managed object	Octets	Definition	Type
MACPIB_STNServicesBroadcast	1	0x00: station is sending information about its own services 0x01: station is not sending information about its own services	dynamic
MACPIB_MasterServicesBroadcast	1	0x00: master is sending information about its own services 0x01: master is not sending information about its own services	dynamic

5.4.5 MAC PIB network security group

The MAC PIB network security group in Table 30 incorporates the network security information of the station in the current network.

Table 30 - MAC PIB network security group parameters

Managed object	Octets	Definition	Type
MACPIB_encAlgorithm	1	0x00: AES-128 0x01: Camellia-128 0x02:SEED-128 0x03-0xFF:user defined	static
MACPIB_MasterSTNAddress	6	MAC address of the master	dynamic
MACPIB_ManagementSECID	2	current valid security session ID for management key	dynamic
MACPIB_DataSECID	2	current valid security session ID for data key	dynamic
MACPIB_ManagementKeyInfo	variable	Agreed keys to protect command in authentication process	dynamic
MACPIB_DataKeyInfo	variable	Agreed keys to protect data in authentication process	dynamic

5.5 MAC SAP

The MAC provides stream and non-stream services to the higher adaptation layer. This service maps the resource allocation request needed for the service of each traffic link to a particular stream. Streams provide the means by which resource allocation can be managed for uplink (slave to master), downlink (master to slave), and peer-to-peer (slave to slave) data traffic. Resource allocation requests for uplink streams (slave to master), downlink streams (master to slave) and peer-to-peer streams (slave to slave) can be managed.

A stream can be dynamically created, modified, or terminated. An existing stream may need to be changed depending on its service type. Asynchronous traffic can also dynamically reserve resources or terminate. For example, bursty IP services may need to change the resource allocation request.

The MAC SAP is the logical interface between the MAC and the higher adaptation layer. This logical interface incorporates a set of primitives and their definitions. These primitives and definitions are described conceptually here, but through this the process of the parameters exchanged between the MAC and adaptation layer can be understood.

The MAC SAP primitives are defined in Table 31.

Table 31 - MAC SAP primitive summary

Name	Request	Indication	Response	Confirm
MAC-ASYNC-DATA	5.5.1	5.5.3		5.5.2
MAC-ISOCH-DATA	5.5.4	5.5.6		5.5.5

The primitive parameters are defined in Table 32.

Table 32 - MAC-ASYNC-DATA and MAC-ISOCH-DATA primitive parameters

Name	Type	Valid range	Description
TrgtID	integer	Values defined in 6.2.1.3	Target STNID of MAC request
OrigID	integer	Values defined in 6.2.1.3	STNID initiating the MAC request
Priority	integer	0-7	Data priority
ACKPolicy	enumeration	immediate acknowledgement, no acknowledgement, delayed acknowledgement	acknowledgement policy of relevant MSDU
StreamIndex	octet	0-255	Stream to which the data is being sent
Transmission-Timeout	time	0-65535	Time limit for data transfer (msec)
Length	integer	0-2035	MSDU length
Data	variable length octet		Data portion of MSDU
ResultCode	enumeration	SUCCESS, TX_TIMEOUT, DLY_ACK_FAILED, INVALID_ACK_POLICY, INVALID_STREAM	Result of MAC request

5.5.1 MAC-ASYNC-DATA.request

This primitive initiates the asynchronous data transfer from one MAC entity to another MAC entity or entities. The semantics are as follows.

```
MAC-ASYNC-DATA.request        (

                    TrgtID,

                    OrigID,

                    Priority,

                    ACKPolicy,

                    TransmissionTimeout,

                    Length,

                    Data
```

)

The primitive parameters are defined in Table 32.

5.5.1.1 Time of creation

Once the adaptation layer receives the data transfer request from the higher layer, it sends it to the MAC SAP.

5.5.1.2 Effect

When this primitive is received, the MAC formats the MSDU according to the input parameters and sends it to the PD-SAP; then the MSDU passes through the wireless media and is sent to the peer MAC entity. If the acknowledgement policy is set to delayed acknowledgement, the MAC does nothing and returns an error indication with MAC-ASYNC-DATA.confirm.

5.5.2 MAC-ASYNC-DATA.confirm

This primitive reports whether the frame has successfully been sent to the adaptation layer or failed due to a timeout. The semantics are as follows.

```
MAC-ASYNC-DATA.confirm          (
                             TrgtID,
                             OrigID,
                             Priority,
                             ResultCode
                             )
```

The primitive parameters are defined in Table 32.

5.5.2.1 Time of creation

This primitive is created when the MAC successfully completes the frame transfer or fails due to a TX_TIMEOUT event or (in the case of an immediate acknowledgement policy) an acknowledgement has not been received by the time the maximum retry limit is reached. If ACKPolicy is set to delayed acknowledgement, ResultCode is set to INVALID_ACK_POLICY.

5.5.2.2 Effect

When the adaptation layer receives this primitive it notifies the appropriate higher layer of the transfer result.

5.5.3 MAC-ASYNC-DATA.indication

This primitive reports to the adaptation layer that it has received an asynchronous MSDU. The semantics are as follows.

```
MAC-ASYNC-DATA.indication       (
                             TrgtID,
                             OrigID,
                             Length,
                             Data
```

)

The primitive parameters are defined in Table 32.

5.5.3.1 Time of creation

This is created when the asynchronous MSDU received by the MAC has been successfully processed.

5.5.3.2 Effect

When the adaptation layer receives this primitive, it notifies the appropriate higher layer of the received frame.

5.5.4 MAC-ISOCH-DATA.request

This primitive initiates the synchronous data transfer from one MAC entity to another MAC entity or entities. The semantics are as follows.

```
MAC-ISOCH-DATA.request          (

                        StreamIndex,

                        TransmissionTimeout,

                        Length,

                        Data

                        )
```

The primitive parameters are defined in Table 32.

5.5.4.1 Time of creation

After the adaptation layer receives the data transfer request from the higher layer and sets the appropriate StreamIndex, it sends it to the MAC SAP.

5.5.4.2 Effect

When this primitive is received, the MAC formats the MSDU according to the input parameters and sends it to the PD-SAP; then the MSDU passes through the wireless media and is sent to the peer MAC entity. If it is not the station creating the requested stream, MLME does not send the frame, and instead responds through a MAC-ISOCH-DATA.confirm with a ResultCode of INVALID_STREAM.

5.5.5 MAC-ISOCH-DATA.confirm

This primitive indicates the success or failure of the frame transmission to the adaptation layer. The semantics are as follows.

```
MAC-ISOCH-DATA.confirm          (

                        StreamIndex,

                        ResultCode

                        )
```

The primitive parameters are defined in Table 32.

5.5.5.1 Time of creation

This primitive is created when the MAC successfully completes the frame transfer or fails due to a TX_TIMEOUT event or (in the case of an immediate acknowledgement policy) an acknowledgement has not been received by the time the maximum retry limit is reached. A delayed acknowledgement may be used, but if the receiving end does not support this acknowledgement type the ResultCode is set to DLY_ACK_FAILED.

5.5.5.2 Effect

When the adaptation layer receives this primitive it sends the transmission result to the appropriate higher layer.

5.5.6 MAC-ISOCH-DATA.indication

This primitive reports to the adaptation layer that a synchronous MSDU has been received. The semantics are as follows.

```
MAC-ISOCH-DATA.indication        (
                                  TrgtID,
                                  OrigID,
                                  Length,
                                  Data
                                  )
```

The primitive parameters are defined in Table 32.

5.5.6.1 Time of creation

This is created when the synchronous MSDU received by the MAC has been successfully processed.

5.5.6.2 Effect

When the adaptation layer receives this primitive, it notifies the appropriate higher layer of the received frame.

5.6 PHY specification

5.6.1 PD-SAP

The PD-SAP transports the MAC protocol data units (MPDUs) between device MAC sublayer entities. Table 33 describes the primitives supported by the PD-SAP. Each primitive is described in the referenced clause.

Table 33 - PD-SAP primitives

PD-SAP Primitive	Request	Confirm	Indication
PD-DATA	5.6.1.1	5.6.1.2	5.6.1.3
PD-CCA-START	5.6.1.4	5.6.1.5	
PD-CCA-END	5.6.1.6	5.6.1.7	
PD-CCA			5.6.1.8
PD-TX-START	5.6.1.9	5.6.1.10	

PD-TX-END	5.6.1.11	5.6.1.12	
PD-RX-START	5.6.1.13	5.6.1.14	5.6.1.15
PD-RX-END	5.6.1.16	5.6.1.17	5.6.1.18
PD-PS	5.6.1.19	5.6.1.20	

Table 34 - PD-SAP parameters

Name	**Type**	**Valid range**	**Description**
psduLength	Unsigned integer	≤aMaxPHYFrameSize	Number of octets in the PSDU to be transmitted by the physical layer entity
status	enumeration	SUCCESS, RX_ON, TRX_OFF	Result of frame transmit request
LQI	octet	PHY dependent	Link quality index; Varies according to implementation
ChannelStatus	enumeration	BUSY or IDLE	Execution result of CCA request
TXDataRate, RXDataRate	integer	0,1,2,3	See constant amplitude coding method in Table 55 00 = RATE1, 01 = RATE2, 02 = RATE3, 03 = RATE4
TXLength, RXLength	2 octets	*minimum payload length – maximum payload length*	MAC frame length
TXPowerLevel	octet	Varies according to PHY	Transfer power
TXMACHead, RXMACHead	2 octets	valid MAC header	MAC header excluding the sequence check
TXAntSelect	octet	0-255	Antenna used for transmission 0 is always valid but other values may be used depending on implementation
RSSI	octet	PHY dependent	Receive signal strength Varies according to implementation
RXERROR	enumeration	NO_ERROR, FORMAT_VIOLATION, CARRIER_LOST, UNSUPPORTED_RATE	Receive error type
PSLevel	integer	0-PHYPIB_NumPSLevels	Specifies the powersave mode
PSResultCode	enumeration	SUCCESS, FAILED, UNSUPPORTED_MODE	Result of powersave mode request

5.6.1.1 PD-DATA.request

This primitive is sent from the MAC sublayer to the physical layer entity and is used to request transmission of a MPDU (PSDU).

5.6.1.1.1 Definition of service primitives

The following interface is provided.

PD-DATA.request (

psduLength,

)

Table 34 defines the PD-DATA.request primitive parameters.

5.6.1.1.2 Time of generation

This occurs when the MAC sublayer entity requests transmission of a MPDU to the physical layer entity.

5.6.1.1.3 Effect

When it is received by the physical layer entity, it triggers the transmission of the given PSDU. If the transceiver is activated (TX_ON), the physical layer creates a PPDU including the PSDU and sends it. Once the physical layer entity has completed the transmission it sends a PD-DATA.confirm primitive with a SUCCESS indication to the MAC sublayer entity. If the primitive is received when the receiver is activated (RX_ON) or the transceiver is deactivated (TRX_OFF) the physical layer entity sends a PD-DATA.confirm primitive with an RX_ON or TRX_OFF status to the MAC sublayer entity.

5.6.1.2 PD-DATA.confirm

This is used to confirm that the MAC sublayer entity has sent a MPDU (PSDU) to another MAC sublayer entity.

5.6.1.2.1 Definition of service primitives

The following interface is provided.

PD-DATA.confirm (

status

)

Table 34 defines the PD-DATA.confirm primitive parameters.

5.6.1.2.2 Time of generation

The physical layer entity sends this to the MAC sublayer entity as a response to the PD-DATA.request primitive when the requested PSDU is transmitted.

5.6.1.2.3 Effect

The MAC sublayer completes the data transmission.

5.6.1.3 PD-DATA.indication

The physical layer reports the MPDU (PSDU) transmission to the MAC sublayer entity.

5.6.1.3.1 Definition of service primitives

The following interface is provided.

PD-DATA.indication (

psduLength,

LQI

)

Table 34 defines the PD-DATA.indication primitive parameters.

5.6.1.3.2 Time of generation

This takes place when the physical layer sends the received PSDU to the MAC sublayer entity. If the received psduLength is lower than *minimum payload length* or greater than *maximum payload length* nothing happens.

5.6.1.3.3 Effect

The MAC sublayer entity should process received data as a MPDU according to the MAC specification.

5.6.1.4 PD-CCA-START.request

The physical layer requests a clear channel assessment (CCA, 9.6.5).

5.6.1.4.1 Definition of service primitives

The following interface is provided.

PD-CCA.request ()

This primitive has no parameters.

5.6.1.4.2 Time of generation

Whenever the CSMA-CA algorithm needs to adjust the channel, DME generates this and sends it to the physical layer.

5.6.1.4.3 Effect

The receive channel assessment is executed.

5.6.1.5 PD-CCA-START.confirm

The physical layer informs the MAC layer that the receive channel assessment (CCA, 9.6.5) has been started.

5.6.1.5.1 Definition of service primitives

The following interface is provided.

PD-CCA.confirm ()

This primitive has no parameters.

5.6.1.5.2 Time of generation

Whenever a PD-CCA-START.request is received the physical layer generates and sends this to the MAC layer.

5.6.1.5.3 Effect

The MAC may proceed to obtain and update the CCA result

5.6.1.6 PD-CCA-END.request

The MAC layer requests termination of the clear channel assessment (CCA, 9.6.5).

5.6.1.6.1 Definition of service primitives

The following interface is provided.

PD-CCA.request ()

This primitive has no parameters.

5.6.1.6.2 Time of generation

This is generated when the MAC layer wants the physical layer to terminate the receive channel assessment.

5.6.1.6.3 Effect

The receive channel assessment is terminated.

5.6.1.7 PD-CCA-END.confirm

The physical layer informs the MAC layer that the receive channel assessment (CCA, 9.6.5) has been terminated.

5.6.1.7.1 Definition of service primitives

The following interface is provided.

PD-CCA.confirm ()

This primitive has no parameters.

5.6.1.7.2 Time of generation

This is generated by the physical layer as a response to PD-CCA-END.request and sent to the MAC layer once the receive channel assessment is terminated.

5.6.1.7.3 Effect

The MAC stops obtaining the CCA result.

5.6.1.8 PD-CCA.indication

The physical layer reports the current channel state to the MAC layer.

5.6.1.8.1 Definition of service primitives

The following interface is provided.

```
PD-CCA.indication       (
                     Channel Status
                     )
```

Table 34 defines the PD-CCA.indication primitive parameters.

5.6.1.8.2 Time of generation

Whenever the channel state changes from BUSY (IDLE) to IDLE (BUSY) the physical layer reports this to the MAC layer. The physical layer keeps the channel in a BUSY state until the aCCADetectTime period ends.

5.6.1.8.3 Effect

The MAC may use the channel status to perform the CSMA/CA algorithm.

5.6.1.9 PD-TX-START.request

The MAC layer requests the start of a MPDU transmission to the physical layer.

5.6.1.9.1 Definition of service primitives

The following interface is provided.

```
PD-TX-START.request      (
                          TXDataRate,
                          TXLength,
                          TXPowerLevel,
                          TXAntSelect,
                          TXMACHead
                          )
```

Table 34 defines the PD-TX-START.request primitive parameters.

5.6.1.9.2 Time of generation

The MAC layer sends this to the physical layer whenever a MPDU needs to be sent. TXMACHead is sent so that the physical layer can calculate the sequence check.

5.6.1.9.3 Effect

The physical layer starts the frame transmission.

5.6.1.10 PD-TX-START.confirm

The physical layer reports the frame transmission initiation to the MAC layer.

5.6.1.10.1 Definition of service primitives

The following interface is provided.

PD-TX-START.confirm ()

This primitive has no parameters.

5.6.1.10.2 Time of generation

This is triggered by the physical layer when it receives a PD-TX-START.request and is ready to receive data from the MAC layer.

5.6.1.10.3 Effect

The MAC layer sends data to the physical layer.

5.6.1.11 PD-TX-END.request

The MAC layer requests the termination of a MPDU transmission to the physical layer.

5.6.1.11.1 Definition of service primitives

The following interface is provided.

PD-TX-END.request ()

This primitive has no parameters.

5.6.1.11.2 Time of generation

This is generated whenever a PD-DATA.confirm is received from the physical layer.

5.6.1.11.3 Effect

The current transmission is terminated.

5.6.1.12 PD-TX-END.confirm

The physical layer informs the MAC layer of the termination of the frame transmission.

5.6.1.12.1 Definition of service primitives

The following interface is provided.

PD-TX-END.confirm ()

This primitive has no parameters.

5.6.1.12.2 Time of generation

The physical layer generates this after a PD-TX-END.request is received and all data has been transmitted.

5.6.1.12.3 Effect

The MAC can initiate the next transmit, receiver, or power management operation.

5.6.1.13 PD-RX-START.request

MAC layer requests that the physical layer activate the receiver and select the specified antenna.

5.6.1.13.1 Definition of service primitives

The following interface is provided.

```
PD-RX-START.request     (
                             RXAntSelect,
                        )
```

Table 34 defines the PD-RX-START.request parameters.

5.6.1.13.2 Time of generation

The MAC layer sends this to the physical layer whenever it anticipates a transmission of a MPDU to the current device.

5.6.1.13.3 Effect

The PHY begins PHY preamble acquisition. It subsequently issues a PD-RX-START.confirm to the MAC.

5.6.1.14 PD-RX-START.confirm

The physical layer reports to the MAC layer that the receiver has been activated.

5.6.1.14.1 Definition of service primitives

The following interface is provided.

PD-RX-START.confirm ()

This primitive has no parameters.

5.6.1.14.2 Time of generation

This is generated when PD-RX-START.request is received and the physical layer's receiver has been activated.

5.6.1.14.3 Effect

The MAC can receive a PD-RX-START.indication primitive.

5.6.1.15 PD-RX-START.indication

The physical layer generates this when the physical layer header and MAC layer header has been successfully transmitted.

5.6.1.15.1 Definition of service primitives

The following interface is provided.

```
PD-RX-START.indication      (
                             RXDataRate,
                             RXLength,
                             RXMACHead,
                             RSSI
                             )
```

Table 34 defines the PD-RX-START.indication parameters.

5.6.1.15.2 Time of generation

The physical layer sends this to the MAC layer whenever the physical layer successfully completes a sequence check at the start of a new PPDU.

5.6.1.15.3 Effect

The MAC can receive a PD-DATA.indication primitive

5.6.1.16 PD-RX-END.request

The MAC layer requests the deactivation of the receiver to the physical layer.

5.6.1.16.1 Definition of service primitives

The following interface is provided.

PD-RX-END.request ()

This primitive has no parameters.

5.6.1.16.2 Time of generation

This is generated when the MAC layer requests the deactivation of the receiver to the physical layer.

5.6.1.16.3 Effect

The PHY stops the reception and issues a PD-RX-END.confirm to the MAC

5.6.1.17 PD-RX-END.confirm

The MAC layer is informed that the physical layer receiver has been deactivated.

5.6.1.17.1 Definition of service primitives

The following interface is provided.

PD-RX-END.confirm ()

This primitive has no parameters.

5.6.1.17.2 Time of generation

The physical layer generates this after receiving a PD-RX-END.request.

5.6.1.17.3 Effect

The MAC can initiate the next transmit, receiver or power management operation.

5.6.1.18 PD-RX-END.indication

The physical layer informs the MAC layer that the MPDU currently being received has been terminated.

5.6.1.18.1 Definition of service primitives

The following interface is provided.

```
PD-RX-END.indication(
                    LQI,
                    RXERROR
                 )
```

The primitive parameters are defined in Table 34.

5.6.1.18.2 Time of generation

The receive state machine indicates that the reception was completed, regardless of error. At this time, RXERROR has one of the following values.

NO_ERROR : no error occurred in the processing of received data.

FORMAT_VIOLATION : incorrect format of the received PPDU.

CARRIER_LOST : MPDU processing impossible due to failure of carrier restoration.

UNSUPPORTED_RATE : unsupported data rate detected when receiving the PPDU.

This is generated when the MAC layer requests to the physical layer that the receiver be deactivated.

5.6.1.18.3 Effect

The MAC is provided with a reference time for determining the end of the received frame on the local air interface.

5.6.1.19 PD-PS.request

The MAC layer requests to the physical layer that the specified powersave mode be instituted.

5.6.1.19.1 Definition of service primitives

The following interface is provided.

```
PD-PS.request          (
                    PSLevel
                    )
```

The primitive parameters are defined in Table 34.

5.6.1.19.2 Time of generation

This is generated by the MAC layer when a change is needed to the powersave mode in the physical layer.

5.6.1.19.3 Effect

A change to the powersave mode is made if possible, and the result is sent through a PD-PS.confirm primitive.

5.6.1.20 PD-PS.confirm

The physical layer reports the result of the requested powersave mode change to the MAC layer.

5.6.1.20.1 Definition of service primitives

The following interface is provided.

```
PD-PS.confirm       (
                    PSResultCode
```

)

The primitive parameters are defined in Table 34.

5.6.1.20.2 Time of generation

The physical layer receives the PD-PS.request primitive from the MAC layer and sends the result of the specific powersave mode change to the MAC layer.

5.6.1.20.3 Effect

The MAC use PSResultCode to the PD-PS.request primitive.

5.6.2 PLME-SAP

The physical layer management object service access points (PLME-SAP) enable the operational language between MLME and PLME. Additional physical layer management object service access points (DME-PLME-SAP) enable the operational language between DME and PLME, and this interface is equivalent to the PLME-SAP interface. Table 35 defines the primitives supported by PLME-SAP.

Table 36 defines the individual parameters.

Table 35 - PLME-SAP primitives

PLME-SAP Primitive	Request	Confirm	Indication
PLME-ED	5.6.2.1	5.6.2.2	
PLME-GET	5.6.2.3	5.6.2.4	
PLME-SET	5.6.2.5	5.6.2.6	
PLME-RESET	5.6.2.7	5.6.2.8	
PLME-TESTMODE	5.6.2.9	5.6.2.10	

Table 36 - PLME-SAP primitive parameters

Name	Type	Valid range	Comments
Edstatus	enumeration	SUCCESS, TRX_OFF, or TX_ON	Result of energy detection request
EnergyLevel	integer	0x00-0xff	Energy detection level of current channel
PIBAttribute	enumeration	As defined Table 57	Desired physical layer PIB attribute
PIBStatus	enumeration	SUCCESS, INVALID_ATTRIBUTE, INVALID_VALUE	Result of request for PIB attribute information
PIBAttributeValue	Various	Attribute specific	Desired physical layer PIB attribute value
ResetResultCode	enumeration	SUCCESS, FAILED	Response to reset request
TEST_ENABLE	boolean	TRUE, FALSE	if true the physical layer test mode is activated according to the following parameters.
TEST_MODE	integer	1, 2, 3	TEST_MODE is one of the following. 01 = receive mode 02 = continuous transmit mode 03 = 50% efficiency periode mode

DataRate,	integer	0,1,2,3	Refer to constant amplitude coding method in Table 55 00 = RATE1, 01 = RATE2, 02 = RATE3, 03 = RATE4
SCRAMBLE_STATE	boolean	TRUE, FALSE	if true the scrambler operational state is activated
TestResultCode	enumeration	SUCCESS, FAILED, UNSUPPORTED_MODE	Test mode result

5.6.2.1 PLME-ED.request

PLME requests an energy detection measurement

5.6.2.1.1 Definition of service primitives

The following interface is provided.

PLME-ED.request ()

This primitive has no parameters.

5.6.2.1.2 Time of generation

DME generates this and requests an energy detection measurement to PLME.

5.6.2.1.3 Effect

If the receiver is activated (RX_ON) PLME tells the physical layer to perform an energy detection measurement. When PLME completes the energy detection is generates a PLME-ED.confirm primitive.

5.6.2.2 PLME-ED.confirm

The result of the energy detection measurement is reported.

5.6.2.2.1 Definition of service primitives

The following interface is provided.

PLME-ED.confirm (

EDstatus,

EnergyLevel

)

Table 36 defines the primitive parameters.

5.6.2.2.2 Time of generation

PLME generates this as a response to a PLME-ED.request and sends it to DME after the energy detection is measured. The status value of a successful energy detection measurement is SUCCESS and a failed measurement due to a faulty transceiver is reported as the current state of the transceiver (TRX_OFF or TX_ON).

5.6.2.2.3 Effect

The result of the energy detection measurement is reported to DME.

5.6.2.3 PLME-GET.request

Information about a physical layer PIB attribute is requested.

5.6.2.3.1 Definition of service primitives

The following interface is provided.

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

Table 36 defines the primitive parameters.

5.6.2.3.2 Time of generation

DME generates this to obtain information from the physical layer PIB of PLME.

5.6.2.3.3 Effect

PLME extracts the requested PIB attribute from the database and sends the results through a PLME-GET.confirm primitive.

5.6.2.4 PLME-GET.confirm

The result of the requested information from the physical layer PIB is reported.

5.6.2.4.1 Definition of service primitives

The following interface is provided.

```
PLME-GET.confirm        (
                        PIBstatus,
                        PIBAttribute,
                        PIBAttributeValue
                        )
```

Table 36 defines the primitive parameters.

5.6.2.4.2 Time of generation

PLME generates this as a response to a PLME-GET.request primitive and sends it to DME.

5.6.2.4.3 Effect

If the state parameter is SUCCESS the requested physical layer PIB value is sent, otherwise an error is indicated.

5.6.2.5 PLME-SET.request

A request is made to set the physical layer PIB attribute to the specified value.

5.6.2.5.1 Definition of service primitives

The following interface is provided.

```
PLME-SET.request          (
                       PIBAttribute,
                       PIBAttributeValue
                       )
```

Table 36 defines the primitive parameters.

5.6.2.5.2 Time of generation

DME generates this to set the physical layer PIB attribute to the specified value and sends it to PLME.

5.6.2.5.3 Effect

PLME attempts to store the specified physical layer PIB attribute in the database and reports the result through a PLME-SET.confirm primitive.

5.6.2.6 PLME-SET.confirm

This reports the result of the attempt to set the physical layer PIB attribute to the specified value.

5.6.2.6.1 Definition of service primitives

The following interface is provided.

```
PLME-SET.confirm          (
                       PIBstatus,
                       PIBAttribute
                       )
```

Table 36 defines the primitive parameters.

5.6.2.6.2 Time of generation

PLME sends this to DME as the response to the PLME-SET.request primitive.

5.6.2.6.3 Effect

If the state value is SUCCESS it means the PIB attribute was set as requested, otherwise an error is indicated if the PIB attribute was unable to be set for some reason.

5.6.2.7 PLME-RESET.request

This requests a reset of the physical layer. To prevent accidental data transmission while receiving, the state must be changed to receive mode.

5.6.2.7.1 Definition of service primitives

The following interface is provided.

PLME-RESET.request ()

This primitive has no parameters.

5.6.2.7.2 Time of generation

This is generated whenever a physical layer reset is requested.

5.6.2.7.3 Effect

The physical layer resets all transceiver state machines to their initial states and converts to receive mode.

5.6.2.8 PLME-RESET.confirm

This requests reset of the physical layer. To prevent accidental data transmission during receiving data, the state must be changed to receive mode.

5.6.2.8.1 Definition of service primitives

The following interface is provided.

```
PLME-RESET.confirm        (
                           ResetResultCode
                           )
```

Table 36 defines the primitive parameters.

5.6.2.8.2 Time of generation

PLME generates this as the result of a PLME-RESET.request.

5.6.2.8.3 Effect

DME or MLME is notified of the result of the reset.

5.6.2.9 PLME-TESTMODE.request

A conversion of the physical layer to test operational mode is requested. The parameters mentioned here are handled as options when implemented.

5.6.2.9.1 Definition of service primitives

The following interface is provided.

```
PLME-TESTMODE.request          (
                                TEST_ENABLE,
                                TEST_MODE,
                                SCRAMBLE_STATE,
                                DataRate
```

)

Table 36 defines the primitive parameters.

5.6.2.9.2 Time of generation

This is generated whenever a request is made to convert the physical layer to test mode.

5.6.2.9.3 Effect

The physical layer is converted to test mode.

5.6.2.10 PLME-TESTMODE.confirm

This requests the physical layer to change to test operational. The parameters mentioned here are handled as options when implemented.

5.6.2.10.1 Definition of service primitives

The following interface is provided.

```
PLME-TESTMODE.confirm          (
                                TestResultCode
                               )
```

Table 36 defines the primitive parameters.

5.6.2.10.2 Time of generation

PLME generates this as the result of a PLME-TESTMODE.request.

5.6.2.10.3 Effect

DME or MLME is notified of the result of test mode initiation.

5.6.3 Physical layer enumerated description

Table 37 lists the valid physical layer protocol enumerated values

Table 37 - Physical layer enumerated values

Enumerated	value	Comments
BUSY	0x01	busy channel is detected through CCA
IDLE	0x02	idle channel is detected through CCA
INVALID_ATTRIBUTE	0x03	Recognition of an unsupported attribute is requested
INVALID_VALUE	0x04	Invalid value for the setting of the parameter is requested
RX_ON	0x05	Activation of receiver is requested
SUCCESS	0x06	When a SET/GET, energy detection, modification of transceiver, or a synchronous frame transfer succeeds
TRX_OFF	0x07	Transceiver is currently inactive or

		deactivation is requested
TX_ON	0x08	Transceiver is currently active or activation is requested
NO_ERROR	0x09	No error has occurred
FORMAT_VIOLATION	0x0A	Error in format
CARRIER_LOST	0x0B	Carrier not found
UNSUPPORTED_RATE	0x0C	unsupported data rate

6 Mac frame format

This chapter describes the format of the MAC frames.

6.1 is an overview of the frames.

6.2 describes the general format of all frames.

6.3 details the format of each frame.

6.4 describes the format of information blocks used in the network to convey information.

6.5 describes the command blocks carried by command frames.

6.1 Overview

The MAC frames in the MAC sublayer are described as a sequence of fields in a specific order. Each figure in Clause 6 depicts the fields as they appear in the MAC frame and in the order in which they are transmitted in the wireless medium, from left to right where the left-most bit is transmitted first in time.

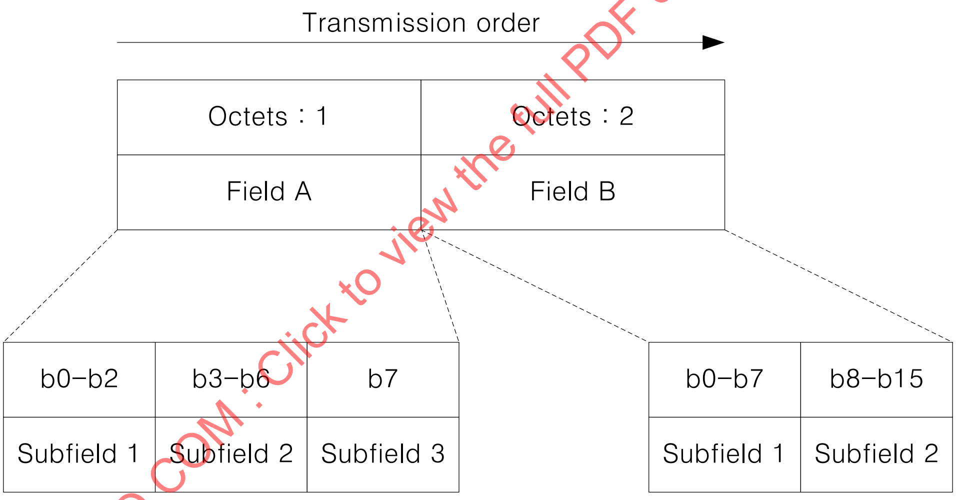

Figure 4 - Transmission order

All MAC frames are composed of the following elements.

a) frame header: frame control information needed to exchange data between stations such as source/destination station ID, frame sequence number, etc. Information located in this portion is used to distinguish frame types and stations exchanging frames. Frame reliability can also be enhanced by error-checking of exchanged frames.

b) frame body: composed of actual payload carrying the data exchanged between stations and FCS to check for payload errors.

6.2 General format of MAC frames

This International Standard uses the common frame formats shown in Figure 5, and each field is defined in detail in the following clauses.

2	2	1	1	1	1	variable	4
network ID	Frame control	Source station ID	Destination station ID	streamID	Sequence number	payload	FCS
Frame header						frame body	

Figure 5 - Format of MAC frame

When the SEC bit is set to zero in the Frame Control field, the non-secure MAC frame body shall be formatted as illustrated in Figure 6.

octet:Ln	4
Payload	FCS
Non-secure MAC frame body	

Figure 6 - Non-secure MAC frame body format

When the SEC bit is set to one in the Frame Control field, the secure MAC frame body shall be formatted as illustrated in Figure 7. The Secure Payload field in the secure MAC frame body is protected as indicated in 8.5.

octet:2	2	2	Ln	8	4
SECID	SFC	Encryption offset	Secure payload	MIC	FCS
Secure frame header				Integrity code	
Payload					
Secure MAC frame body					

Figure 7 - Secure MAC frame body format

6.2.1 Frame header

Frame header contains information for the transmission or reception of the frame, flow control, and error checking.

6.2.1.1 NID

The network ID is the ID that distinguishes a network. This is a unique value that does not overlap with other network IDs and remains as long as the network continues to exist.

6.2.1.2 Frame control

Frame control elements are composed of fields such as frame type, first fragment, last fragment, acknowledgement policy, delayed acknowledgement request, protocol version, security, etc. Their format is shown in Figure 8.

bit 0-3	4-5	6	7	8	9-10	11	12-15
frame type	acknowledgement policy	First fragment	Last fragment	delayed acknowledgement request	Protocol version	SEC	reserved

Figure 8 - Format of frame control fields

6.2.1.2.1 Frame type

The frame type field is 4 bits in length. Table 38 shows the defined frame types and values. Examples of the usage of these types are shown in 6.3.

Table 38 - Frame types

Value	Frame type description
0000	beacon
0001	acknowledgement
0010	command
0011	data (stream or non-stream)
0100	RTS
0101	CTS
0110~1111	reserved

6.2.1.2.2 Acknowledgement policy

Acknowledgement policy field is 2 bits in length. In the case where the received frame is an acknowledgement frame it indicates the type of the acknowledgement frame, otherwise it indicates the acknowledgement policy of the destination station of the frame.

Delayed acknowledgement policy can only be used for data frames, and multicast or broadcast frames do not use acknowledgements. (When multicast or broadcast frames are received, the acknowledgement policy field is ignored and the destination station does not send an acknowledgement.) The following are the possible values of the acknowledgement method field.

0: no acknowledgement: destination station does not acknowledge the transmitted frame, and the source station considers the transmission successful regardless of the actual result. This method can be used for frames that are transmitted 1:1 or 1:N, which do not require acknowledgement.

1: immediate acknowledgement: The station that received the frame sends an acknowledgement frame as a response to the source station after a short frame interval. This acknowledgement policy can only be used for 1:1 frame transmissions.

2: delayed acknowledgement: the station receiving the frame sends a delayed acknowledgement frame per 6.3.2.2. This acknowledgement policy can only be used for 1:1 frame transmissions.

3: implicit acknowledgement: This method is used when the receiving station sends an immediate acknowledgement or other data frames as an acknowledgement. This acknowledgement policy can only be used for 1:1 frame transmissions, and can only be used during the allocation period, not the contention period.

6.2.1.2.3 First fragment

The first fragment field is 1 bit in length. ‘1’ indicates that frame is the start of a data or command (MSDU/MCDU) from a higher layer, while ‘0’ means that it is not the start.

6.2.1.2.4 Last fragment

The last fragment field is 1 bit in length. ‘1’ indicates that frame is the end of a data or command (MSDU/MCDU) from a higher layer, while ‘0’ means that it is not the end.

6.2.1.2.5 Delayed acknowledgement request

The delayed acknowledgement request field is 1 bit in length. This is only valid for data frames of streams using the current delayed acknowledgement policy. This field is set to ‘1’ and sent when the source station of the stream is requesting that a delayed acknowledgement frame be sent to the destination station.

6.2.1.2.6 Protocol version

Protocol version information is 2 bits in length. The size and location is fixed and independent of the protocol version of the system. The current value is 0, and is incremented by 1 each time a new version is released. When a station receives a frame with a version higher than its own, it discards it without notifying the source station.

6.2.1.2.7 SEC

The SEC bit shall be set to one when the frame body is protected using the key specified by the security ID (SECID). The SEC bit shall be set to zero, otherwise. Frames with the SEC bit set to one shall use the secure frame format for that frame type, as described in 6.3.

6.2.1.3 Station ID

In the MAC frame there are two types of station IDs - the destination station ID and source station ID – and each are 8 bits in length. The source station ID is the ID of the station sending the frame, and the destination station ID is the ID of the station receiving the frame. Each station ID is assigned by the master when the station is associated with network, and the possible values are as follows.

- 0x00: used by master of network

- 0xFF: used by broadcast frames

- 0xFE: used by stations in the process of association which have not yet been assigned an ID by the master

- 0xFD: used by multicast frames

6.2.1.4 Stream ID

This is composed of 8 bits and is used to identify data streams. It is only valid for data frames, and in other frame types this is set to 0 and ignored.

bit:7	6:4	3:0
Stream type	priority	stream index

Figure 9 - Format of stream ID field

The field is composed of 3 parts.

The stream index field is 4 bits in length. Each station may freely use any value other than ‘0’ (which indicates non-stream data), but each value must be uniquely assigned by the master.

The priority field is 3 bits in length, and it indicates the stream priority.

The stream type field is 1 bit in length, and ‘1’ means isochronous service, while ‘0’ means asynchronous service.

Frames that do not belong to an existing stream or which do not need a stream connection are non-stream type data, and these are sent with a stream ID of 0. The relevant station determines whether to use an isochronous stream or an asynchronous stream.

6.2.1.5 Sequence number

The sequence number field is 8 bits in length and shows the frame sequence number.

In data frames, a sequence number between 0 and 255 is assigned by means of an incremental counter by each stream, and once it reaches 255 it wraps back to 0.

For frames that are not data type frames, a sequence number between 0 to 255 is assigned by a counter.

6.2.2 Frame body

The frame body portion is variable length and is composed of the payload and FCS. The payload format may vary depending on the frame type of the frame control field, and the FCS is used to check for errors in the frame.

6.2.2.1 Payload

The data that is actually exchanged between stations is carried in a variety of formats. In the case of a secure frame, it also includes the required security information and the secure payload.

6.2.2.2 Secure session ID (SECID)

The frame body of all secure frames shall include the SECID field, which contains a 2-octet identifier for the key that is used to protect the frame. In the lowest order octet of the SECID for all keys except the network group data key, the STNID of the key originator shall be set. The SECID for the network group data key shall have the lowest order octet set to the BcstID, as described in 6.2.1.3. In the higher order octet, a unique value for the key associated with the security relationship shall be designated. The key originator in a security relationship selects the SECID for a given key, as described in 8.3.6.

6.2.2.3 Secure frame counter (SFC)

The frame body of all secure frames includes the SFC field, which contains a 2-octet counter that is used to ensure the uniqueness of the nonce in a secure frame. A station shall not reuse a frame counter with the same beacon sequence number, as described in 6.3.1.1, and key, as described in 8.3.6. The SFC shall be initialized to zero for the first frame sent and increment it for each successive secure frame sent. When the beacon sequence number, as described in 6.3.1.1, is updated, the station may reset the SFC to zero if desired or allow the counter to roll over. In the case where the station receives a new key, the station shall set the SFC to zero. SFC is used for retransmission detect, as described in 8.1.7.

6.2.2.4 Encryption offset (EO)

The Encryption Offset field indicates where encryption starts, in octets, relative to the beginning of the Secure Payload, as shown in Figure 7. A value of zero indicates that the entire Secure Payload is encrypted. A non-zero value in this field indicates that the first EO octets of the Security Payload are not encrypted. Regardless of the value of this field, the entire Secure Payload, along with other appropriate fields, is authenticated by the MIC.

6.2.2.5 Integrity code

The frame body of all secure frames shall include the integrity code field, which contains an 8-octet encrypted integrity code that is used to cryptographically protect the integrity of the MAC header and frame payload. The integrity code is computed as specified in 8.5.3.4.

6.2.2.6 Frame check sequence (FCS)

The Frame Check Sequence (FCS) is 32 bits in length, and is used to verify that the MPDU was received without error. It is generated using the following standard generator polynomial of degree 32:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

The FCS is the 1's complement of the modulo 2 sum of the remainders in (a) and (b) below:

a) The remainder resulting from $(x^k * (x^{31} + x^{30} + \cdots))$ divided (modulo 2) by g(x). The value k is the number of bits in the calculation field.

b) The remainder resulting from the calculation field contents, treated as a polynomial, multiplied by x^{32} and then divided by g(x).

At the transmitter, the initial remainder of the division shall be preset to all ones and is then modified via division of the calculation fields by the generator polynomial g(x). The 1's complement of this remainder is the FCS field.

At the receiver, the initial remainder shall be preset to all ones. The serial incoming bits of the calculation fields and FCS, when divided by g(x) in the absence of transmission errors, results in a unique non-zero remainder value. The unique remainder value is the polynomial:

$$1 + x + x^3 + x^4 + x^5 + x^6 + x^8 + x^{10} + x^{11} + x^{12} + x^{14} + x^{15} + x^{18} + x^{24} + x^{25} + x^{26} + x^{30} + x^{31}$$

6.3 Frame formats

A total of six types of frames are defined, and each frame is coded with different usage codes as shown in Table 39, depending on type.

Table 39 - Usage codes by frame type

Frame type	**Usage code**
beacon	network command code
Acknowledgement	network command code
Command	network command code
data (stream or non-stream)	Variable depending on conditions
RTS	network command code
CTS	network command code

6.3.1 Beacon

6.3.1.1 Non-secure beacon frame

This is used during the beacon period to broadcast information to stations associated with network. The non-secure beacon frame structure is shown in Figure 10.

8	2	2	2	1	1	Variable	4
Frame header	beacon sequence number	superframe length	allocation period start time	network standard code	reserved	beacon payload	FCS
	network synchronization information						

Figure 10 - Non-secure beacon frame format

The beacon sequence number is used to tag each beacon and synchronize slave operations (such as modifying frequency or sleep state). It is incremented by 1 on each transmission.

The superframe length field shows the superframe length in units of usec.

The allocation period start time field shows the start time of the allocation period in the superframe in units of usec.

The network standard code field is used to show the code rates when a slave sends a command code.

The beacon frame payload holds the information blocks described in Table 40, and each is described in detail in 6.4.

Table 40 - Beacon frame body

Information block	Content	When loaded
Station UID	48 bit master IEEE 802 address	as necessary
network synchronization parameter	Synchronization information needed by the network stations to synchronize with network	as necessary
Transmit power	Maximum transmit power within network	as necessary
channel modification	Information needed when changing channels	as necessary
Resource allocation	Present state of current superframe resource distribution	as necessary

The non-secure beacon frame's control field is set to values as shown in Table 41.

Table 41 - Setting the control field of the non-secure beacon frame

Subfield	Value when sent	Action when received
frame type	Value of beacon frame in 6.2.1.2.1	decoded
Acknowledgement policy	0	ignored
First fragment	0	Ignored
Last fragment	0	Ignored
delayed acknowledgement request	0	Ignored
Protocol version	Protocol version	decoded
SEC	0	decoded

6.3.1.2 Secure beacon frame

The secure beacon frame structure is shown in Figure 11. The secure beacon frame format is used when the network is operating in a secure mode.

8	2	2	2	8	Variable	8	4
Frame header	SECID	SFC	Encryption offset	Network synchronization information	Beacon payload	Integrity code	FCS
Secure frame body							

Figure 11 - Secure beacon frame format

The secure beacon frame's control field shall be set and interpreted as described in Table 42.

Table 42 - Setting the control field of the secure beacon frame

Subfield	Value when sent	Action when received
frame type	Value of beacon frame in 6.2.1.2.1	decoded
Acknowledgement policy	0	ignored
First fragment	0	Ignored
Last fragment	0	Ignored
delayed acknowledgement request	0	Ignored
Protocol version	Protocol version	decoded
SEC	1	decoded

6.3.2 Acknowledgement

Acknowledgement frames are of two types - immediate acknowledgement frames and delayed acknowledgement frames – and are encoded with the network standard code rates. While the immediate acknowledgement frame is composed only of a frame header with a frame body, the delayed acknowledgement frame contains a payload and is thus composed of both frame header and frame body. Frame control fields in an acknowledgement frame are set as in Table 43.

Table 43 - Setting the control field of the beacon frame

Subfield	Value when sent	Action when received
frame type	Value of acknowledgement frame as shown in 6.2.1.2.1	decoded
acknowledgement policy	Value of frame type of acknowledgement frame	decoded
First fragment	0	Ignored
Last fragment	0	Ignored
delayed acknowledgement request	0	Ignored
Protocol version	Protocol version	decoded

The acknowledgement frame's source station ID field is set to the Destination station ID of the received frame, and the Destination station ID field is set to the source station ID of the received frame.

The stream ID and stream sequence number are set to 0 and ignored when received.

6.3.2.1 Immediate acknowledgement

The immediate acknowledgement frame is composed of only a frame header without a frame body as in Figure 12, and can be used during any period. It is encoded using the network standard code rates.

2	2	1	1	1	1	4
network ID	Frame control	source station ID	Destination station ID	streamID	Sequence number	FCS
Frame header						

Figure 12 - Immediate acknowledgement frame format

6.3.2.2 Delayed acknowledgement

When the destination station receives a delayed acknowledgement request in a stream using delayed acknowledgement, it uses delayed acknowledgement only during the allocation period. The frame format is as shown in Figure 13, and is encoded using the standard codes.

8	2	7	7	…	7	4
Frame header	length (=7*m)	Record of stream-1	Record of stream-2	….	Record of stream-m	FCS
	payload					
	frame body					

Figure 13 - Delayed acknowledgement frame payload format

The stream record field is as shown in Figure 14.

octet:1	1	1	4
stream ID	Starting sequence number	Ending sequence number	Bitmap of receive status

Figure 14 - Format of record for stream-m

The stream ID field is 2 octets and is used by the data to identify the stream using delayed acknowledgement. When 32 or more frames are being acknowledged in the stream, one or more information blocks can be used.

The starting sequence number is 2 octets and indicates the sequence number of the first frame corresponding to the receive status bitmap. Likewise, the ending sequence number is 2 octets and indicates the sequence number of the last frame corresponding to the receive status bitmap.

The receive status bitmap is 4 octets, and each bit represents the success or failure of each frame received in the stream. The first bit of the receive status bitmap corresponds to the frame containing the starting sequence number and the rest of the bits correspond in order to the frames with the following numbers. A bit value of ‘1’ in the receive status bitmap indicates a successful reception, and ‘0’ indicates that the reception was unsuccessful.

6.3.3 Command

The command frame is used during the contention period for exchanges between master and slave, and during the allocation period for exchanges between slaves. It is sent during the relevant station’s assigned timeslot or the contention period, and one frame can hold one or more command blocks. 0 describes the command types in detail.

6.3.3.1 Non-secure command frame

The non-secure command frame shall be formatted as shown in Figure 15. The command types are described in 6.5.

octet:8	(1+2+L1)	(1+2+L2)	…..	(1+2+Ln)	4
Frame header	Command block-1	Command block-2	….	Command block-n	FCS
	payload				
	frame body				

Figure 15 - Non-secure command frame format

The command block shall be formatted as shown in Figure 16.

octet:1	2	L
Command type	length	Command payload

Figure 16 - Command block format

The frame header stream ID is set to 0 and ignored when received.

While the command payload is of variable length, the command block is laid out in 2-octet increments in the frame body. If the command block length is not in 2-octet units when the command payload is created, it is padded with 0’s to fill the length and the length field indicates only the significant portion.

6.3.3.2 Secure command frame

The secure command frame shall be formatted as shown in Figure 17. The command types are described in 6.5.

octet:8	2	2	2	(1+2+L1)	(1+2+L2)	…..	(1+2+Ln)	4
Frame header	SECID	SFC	Encryption offset	Command block-1	Command block-2	….	Command block-n	FCS
	secure frame header			payload				
	frame body							

Figure 17 - Format of secure command frame

6.3.4 Data (stream or non-stream)

This is used primarily using the allocation period to send data, and it may be used during the contention period to exchange data of short length as well.

6.3.4.1 Non-secure data frame

The format of non-secure data frame is shown in Figure 18, and is encoded using a variety of codes.

octet:8	variable	4
Frame header	Variable length data	FCS

Figure 18 - Non-secure data frame format

6.3.4.2 Secure data frame

The format of secure data frame is shown in Figure 19, and is encoded using a variety of codes.

octet:8	2	2	2	variable	4
Frame header	SECID	SFC	Encryption offset	Variable length data	FCS

Figure 19 - Secure data frame format

6.3.5 RTS (Request To Send)

The RTS frame format is shown in Figure 20.

octet:2	2	1	1	2	4

NID	Frame control	source station ID	Destination station ID	RTS time	FCS

Figure 20 - RTS frame format

6.3.6 CTS (Clear To Send)

The CTS frame format is shown in Figure 21.

octet:2	2	1	2	4
NID	Frame control	source station ID	CTS time	FCS

Figure 21 - CTS frame format

6.4 Information block

The information block is detailed in Table 44, and each individual part is described in the following clauses.

Table 44 - Information blocks

Information block ID	information
0	station UID
1	station name
2	station type
3	network synchronization
4	capability
5	maximum supported time slot
6	channel modification
7	power management parameter
8	maximum transmit power
9	resource allocation
10	new master notification
11	sleep state notification
12-254	reserved
255	vendor specific

The format of the information block is shown in Figure 22. The first octet is the information block ID, and the second octet is the data length (L_n) in octet units. The next L_n octets are the data. More than one information block may be included in a frame and there are no restrictions on the order.

The information block in a frame body is arranged in 2 octets units. When the frame body is made, if the information block is not in units of 2 octets, 0's are appended to the end of the data to fill the 2-octet length. However, the length field of the information block indicates only the significant data length (excluding the zeroes used for padding). When the frame body is opened and the payload read out, only the actual data is read based on the field length, and the padded zeroes are ignored.

octet:1	1	Ln

Information block ID	length (=Ln)	data

Figure 22 - Information block format

6.4.1 Station UID

The station UID information block is shown in Figure 23.

octet:1	1	6
Information block ID (=0)	length (=6)	Station ID

Figure 23 - Station UID information block format

The station UID is the 48 bit IEEE 802 address of the source station.

6.4.2 Station name

The station name information block is shown in Figure 24.

octet:1	1	variable
information block ID (=1)	length (<256)	station name

Figure 24 - Station name information block format

The station name is the name of the source station recognized by the user, and the maximum length is 256 octets.

6.4.3 Station type

The station type information block is shown in Figure 25.

octet:1	1	3
information block ID (=2)	length (=3)	station type

Figure 25 - Station type information block format

The station type field represents the station type. The 3-octet representation of the station type has not yet been defined.

6.4.4 Network synchronization

The network synchronization information block is shown in Figure 26.

octet:1	1	1	1	2
information block ID (=3)	length (=4)	Modification information	Counter	Value to change

Figure 26 - Network synchronization information block format

The modification information field indicates the network synchronization information to be changed in the future. The value to change will be altered according to this field.

If this field is 1, it means the superframe length will change, and if 2 it means that the channel in use will change.

The counter field represents the point at which the modified network synchronization information applies. For example, if the current beacon sequence number is 3 and the counter field is 2, the modified synchronization information will take effect after a beacon sequence number of 5 is received.

The value to change field represents the value of the network synchronization information to be modified, and its meaning varies according to the value of the modification information. If the modification information field is 1, it indicates the new superframe length in usec, and if 2 it indicates the new channel ID.

6.4.5 Capabilities

The capability information block is shown in Figure 27, and its fields are laid out in Figure 28. The values of the capability information block do not change once a station is associated with network until the station disassociates itself. The master's capability field value does not change while it remains master of network.

octet:1	1	2
information block ID (=4)	length (=2)	capability

Figure 27 - Capability information block format

2 octets						
bit:b0	b1	b2	b3	b4~b7	b8~b11	b12~b15
task	master capable	real-time stream support	Sleep state support	Amount of available power	Amount of memory	undefined

Figure 28 - Capability field format

The task bit represents the current station's role, and if the station is master it is set to '1', otherwise '0'.

The master capable bit is set to '1' if the station is capable of becoming master of network, otherwise '0'.

The realtime stream support bit is set to '1' if the station supports real-time streams, otherwise '0'.

The sleep state support bit is set to '1' if the device uses sleep state for power saving, and in the case of the master this bit is always 0.

The available power field indicates the amount of power left in the station. '1111' means that the station is being supplied from a stable source of power.

The memory field indicates the MAC memory size of the station in units of 258KB.

6.4.6 Maximum supported time slot

The maximum supported time slot information block is shown in Figure 29.

octet:1	1	1
Information block ID (=5)	length (=1)	Maximum number of supported time slots

Figure 29 - Maximum support timeslot information block format

The maximum supported time slot field indicates the maximum number of supported time slots that can be allocated to a source station.

6.4.7 Maximum transmit power

The maximum transmit power information block is shown in Figure 30.

octet:1	1	1
information block ID (=8)	length (=1)	Maximum transmit power

Figure 30 - Maximum transmit power information block format

The maximum transmit power field shows the maximum transmit power allowed by network in 2's complement form, in dBm units. For example, +2dBm is encoded as 0x02, and -2dBm is encoded as 0xFE. Also, 0x7F means that the master does not restrict the transmit power.

6.4.8 Resource allocation

The resource allocation information block is shown in Figure 31. Due to restrictions on the length field a information block can be a maximum of 256 octets, therefore if needed the master can divide the resource allocation information and load it into multiple information blocks in the beacon. The destination station puts together all resource allocation information in the received beacon frame. To facilitate the combination of the resource allocation information blocks, the master sequences them by the channel time.

octet:1	1	6	6	….	6
information block ID (=9)	length (=n*6)	resource allocation block-1	resource allocation block-2	….	resource allocation block-n

Figure 31 - Resource allocation information block format

Resource allocation information blocks are made up of many resource allocation blocks, and each block is arranged in the order of the start time of the time slot. The resource allocation block is shown in Figure 32.

octet:1	1	1	1	1	1
source station ID	Destination station ID	streamID	Start of allocated slot	length of allocated slot	Allocated code

Figure 32 - Resource allocation block format

The source station ID indicates the station to which resources have been allocated.

The destination station ID indicates the station to which the source station is sending the frame, and if this ID is the broadcast ID, the source station sends broadcast frames during the time slot.

The start time field is composed of two parts, and the first four bits represent the start time slot index while the last four bits represent the number of allocated time slots.

The stream ID field indicates the stream that is being used for transmission by means of the resources.

The allocated time slot field represents the starting slot for the allocated time slot. '1' is for the first slot of the allocation period, and '0' is for the contention period. If this field is set to '0', it means that the station with destination station ID is waking up during its contention period.

The number of allocated time slots field represents the number of allocated time slots, and this number of time slots is allocated to the source station from the time slots represented in the start time slot index field.

The allocated code field represents the usage mode of the physical layer as shown in 9.3.3

6.4.9 New master notification

The new master notification information block is shown in Figure 33.

octet:1	1	6	1	2
information block ID (=10)	length (=9)	new master station UID	new master ID	Time limit for handover

Figure 33 - New master notification information block format

The new master station UID field represents the unique number of the station to become master.

The new master ID field represents the currently assigned ID of the station to become master.

The handover time limit field represents the time limit in which the station must become the master.

6.4.10 Sleep state notification

The sleep state notification information block, which reports the entry of a station into sleep state and the wakeup time, is shown in Figure 34.

octet:1	1	1	1	…	1	1
information block ID (=11)	length (=2*N)	beacon counter #1	sleeping station ID #1	…	beacon counter #N	sleeping station ID #N

Figure 34 - Sleep state notification information block format

The beacon counter field represents the number of superframes remaining until the sleeping station can wake and hear the beacon. For example, if this value is 2, it means that the station wakes up and hears the beacon after 2 superframes (including the superframe of this beacon), and if 1 it means that it will receive the next superframe in the contention period.

The sleeping station ID field represents the ID of the station in a sleep state.

6.4.11 Vendor specific

The vendor specific IE shall be formatted as illustrated in Figure 35.

octet:1	1	3	Ln
Element ID (=255)	length (=3+Ln)	VendorOUI	Vendor specific information

Figure 35 - Vendor specific information element format

The vendor specific information field is defined by the vendor identified in the VendorOUI field, which is specified in IEEE RAC. Its use by a station is outside of the scope of this standard

6.5 Command block

Each command block is composed of a 2-octet command type field, a 2-octet length field, and a variable-length command payload, as shown in Figure 36.

octet:1		2	L_x
command type		length (=L_x)	Command payload
type(4 bit)	command(4bit)		

Figure 36 - Command block format

Commands are roughly divided into four types as shown in Table 45, and each command is described in detail in the following clauses. The command type field is divided into two subfields – the type subfield and the command subfield.

Table 45 - Command types

Type **(type subfield: 4 bits)**	**Command** **(command value subfield: 4 bits)**
network management (0x1)	associate request (0x1)
	associate response (0x2)
	disassociate request (0x3)
	master handover (0x4)
stream management (0x2)	resource allocation request(0x1)
	resource allocation response(0x2)
	resource allocation modify (0x3)
	resource allocation termination (0x4)
	delayed transmission verification resynchronization (0x5)
power management (0x3)	sleep state request (0x1)
	sleep state response (0x2)
	activation notification (0x3)
	transmit power adjustment (0x4)
other (0x4)	station information request (0x1)
	station information response (0x2)
	data query (0x3)
	channel status request (0x4)
	channel status response (0x5)
	remote channel scan request (0x6)
	remote channel scan response (0x7)
key management (0x5)	request key (0x1)
	request key response (0x2)
	distribute key request (0x3)
	distribute key response (0x4)
security management (0x6)	security message (0x1)
vendor specific (0xF)	vendor specific command

6.5.1 Network management

6.5.1.1 Associate request

This is the command block used to associate a new station with the network built by the master.

The acknowledgement policy is set to immediate acknowledgement, and the first fragment, last fragment, retransmit, and delayed acknowledgement request fields are set to 0 and ignored when received.

The network ID field of an associate request sent by a disassociated station is set to 0, and for an associate request from a station already registered it is set to the current network ID value.

The source station ID is set to ‘0xFE’ to indicate a station with no assigned ID, and the destination station ID is set to ‘0’ to indicate the master.

The command block to be carried as the payload is shown in Figure 37.

octet:1	2	6	2	2
Command type (=0x11)	length (variable)	Station UID	capability	Time limit for association (usec)

Figure 37 - Associate request command block format

The station UID field represents the 48 bit IEEE 802 address of the station requesting association.

The capability field indicates the capability of the station requesting association as shown in Figure 28.

The allocation ID field is set to ‘0xFE’ and ignored when received.

If a response to the associate request is not received within the time specified by the association time limit field, the station is disassociated and an attempt is made to associate again. Also if the master does not receive a frame from the station during this time, it disassociates the station and waits for an association request.

6.5.1.2 Association response

This is the command block that the master sends as a response to the station requesting the association.

The acknowledgement policy is set to immediate acknowledgement, and the first fragment, last fragment, retransmit, and delayed acknowledgement request fields are set to 0 and ignored when received.

The network ID field is set to the current network ID.

The source station ID is set to ‘0’ to indicate the master, and the destination station ID is set to ‘0xFE’.

The command block to be carried in the payload is shown in Figure 38.

octet:1	2	6	1	1
command type (=0x12)	length (variable)	station UID	Allocated ID	reason

Figure 38 - Association response command block format

The station UID field represents the 48 bit unique number of the station requesting association. The master uses the value in the previously received association request frame to fill this field. When two or more stations are awaiting an association response from the master, they use the station UID field of the association response frame to determine whether the frame is intended for themselves.

The allocated ID field is filled with the ID to be assigned to the station requesting association. The station begins communicating using the assigned ID, and it is retained until the station is disassociated. If the master denies the association request, this field is set to ‘0xFE’ and the reason code field is set to the reason for denial.

Valid reason codes are the following.

- 0: the maximum number of stations handled by network which are already associated

- 1: no bandwidth exists for a new station

- 2: poor channel state

- 3: the master is terminating and no alternate master exists in network

- 4: station is attempting to disassociate

- 5: channel is being changed

- 6: master handover is taking place

- 7: station authentication is failed

- 8-255: reserved

6.5.1.3 Disassociation request

The master or slave can make a disassociation request, and the command block format in shown in Figure 39.

octet:1	2	6	1
command type (=0x13)	length (=7)	station UID	Reason code

Figure 39 - Disassociation request payload format

The station UID field represents the 48 bit IEEE 802 address of the station requesting disassociation.

Valid reason codes are the following.

- 0: station status is expired (must associate again)

- 1: poor channel state

- 2: station has exceeded its allotted channel time

- 3: master is terminating and there is no alternate master

- 4: station is attempting to disassociate

- 5-255: reserved

6.5.1.4 Master handover

This command is used when the master hands over its authority to another master-capable station in the network. The master compares the capabilities of the stations in the network and hands over its authority to the most eligible station. The capabilities evaluated when selecting the master are laid out in order of preference in Table 46.

Table 46 - Order of preference when comparing capability

sequence	Information	comments
1	Station status	master state is preferred
2	RTC bit	RTC = 1 is preferred
3	SEC bit	SEC = 1 is preferred
4	PS bit	PS = 0 is preferred
5	Power availability field	Higher value is preferred
6	Memory field	Higher value is preferred
7	Storage type bit	Higher value is preferred
8	Station UID	Higher valued is preferred

The master handover command block format is shown in Figure 40.

octet:1	2	2	6	6.	2
command type (=0x14)	length (=16)	Number of stations	master station UID	alternate master station UID	Handover time limit

Figure 40 - Master handover command block format

The number of stations field is the total number of stations currently connected to the network.

The master station UID represents the unique ID of the master station.

The alternate master station UID represents the unique ID of the alternate master station to take on the role of master.

The handover time limit field represents the interval in which the new master must inherit the relevant information from the current master and generate a beacon frame, starting from the most recent beacon frame sent by the current master. The unit of time is 8μs and the maximum is 524280μs.

Once the current master sends the station information, it no longer transmits beacon frames.

6.5.2 Stream management

6.5.2.1 Resource allocation request

The resource allocation request command block format is shown in Figure 41. The resource allocation request record format is shown in Figure 42.

octet:1	2	6	6		6

command type (=0x21)	length(=n*6)	resource allocation request record for stream-1	resource allocation request record for stream-2	….	resource allocation request record for stream-n

Figure 41 - Resource allocation request command block format

octet:1	1	1	1	2
Destination station ID	Stream request ID	Resource allocation period (Kμs)	Amount of data to transmit each period (octet)	Required BER level

Figure 42 - Resource allocation request record format

The stream request ID field is 1 octet and is generated to prevent duplication of stream connection request identifiers. This identifier is used with the ID of the requesting station, and is maintained while the stream is transmitting or receiving frames later on.

The resource allocation period field represents the allocation period of the resources allocated to the stream.

The minimum necessary data per period field represents the minimum amount of data that must be sent each period for stream transmission, and if the resources to be allocated are smaller than this size, the master terminates the stream services.

The minimum transmitted data per period field indicates the appropriate amount of data to be sent per period for stream transmission.

The required BER level field indicates the BER level required by the stream. This field is divided into a mantissa (first octet) and an exponent (second octet), and the required BER is calculated by mantissa*10^(-1*exponent).

6.5.2.2 Resource allocation response

This frame is sent as a response to a resource allocation request or modification request.

octet:1	2	1	1	1	1	1
command type (=0x22)	length (=5)	stream request ID	streamID	Number of periodically allocated time slots	Number of allocated codes	Reason code

Figure 43 - Resource allocation response command block format

The stream request ID field is defined in 6.5.2.1.

The stream ID field enables the identification of many streams between source/destination stations. The highest 4 bits show the media type, and the lowest 4 bits show the sequence number. The combination of source station ID, destination station ID, and stream ID together identify a stream in network.

The periodic allocation time slot field shows the number of time slots requested by the slave to be allocated in each period.

The number of allocated codes field indicates the number of codes to be assigned to the slave.

The reason code field shows whether the resource allocation request was successfully executed, and the values are as follows.

- 0: success

- 1: request from a station not associated with network

- 2: the requested priority is not supported

- 3: stream is terminated by the master

- 4: stream is terminated by the destination station

- 5: insufficient resources

- 6: destination station is in powersave mode

- 7: request denied

- 8-255: reserved

6.5.2.3 Resource allocation modification request

The resource allocation modification command block format is shown in Figure 44. The resource allocation modification request record format is shown in Figure 45.

octet:1	2	6	6		6
Command type (=0x23)	length (n*6)	Resource allocation modification record for stream-1	Resource allocation modification record for stream-2	….	Resource allocation modification record for stream-n

Figure 44 - Resource allocation modification command block format

octet:1	1	1	1	2
Destination station ID	StreamID	Resource allocation period (Kμs)	Amount of data to transmit per period (octet)	Required BER level

Figure 45 - Resource allocation modification request record format

The stream ID field indicates the ID of the stream requesting the resource allocation modification.

The resource allocation period field represents the allocation period of the resources allocated to the stream.

The minimum necessary data per period field represents the minimum amount of data that must be sent each period for stream transmission, and if the resources to be allocated are smaller than this size, the master terminates the stream services.

The minimum transmitted data per period field indicates the appropriate amount of data to be sent per period for stream transmission.

The required BER level field indicates the BER level required by the stream. This field is divided into a mantissa (first octet) and an exponent (second octet), and the required BER is calculated by mantissa*10^(-1*exponent).

6.5.2.4 Resource allocation termination

This frame is sent as a response to a resource allocation request or modification request.

octet:1	2	1	1	1	1
command type (=0x24)	length (=4)	Source station ID	Destination station ID	stream ID	Reason code

Figure 46 - Resource allocation termination command block format

The source station ID field is the ID of the station transmitting on the stream whose resource allocation is being terminated.

The destination station ID field is the ID of the station receiving on the stream whose resource allocation is being terminated.

The stream ID field is the ID of the stream whose resource allocation is about to be terminated.

The reason code field indicates whether the resource allocation termination request was successful, and the values are as follows.

- 0: stream transmit terminated

- 1: stream receive terminated

- 2: stream transmit abnormally halted

- 3: stream received abnormally halted

- 4: stream terminated by master

- 5: source station terminated

- 6: destination station terminated

- 7: destination station is in powersave mode

- 8-255: reserved

6.5.2.5 Delayed acknowledgement resynchronization

Only stations that are sending streams 1:1 to other stations using a delayed acknowledgement policy may send a delayed acknowledgement resynchronization. The command block format is shown in Figure 47.

octet:1	2	3	3	…	3
command type (=0x25)	length (=3*m)	stream-1 record	stream-2 record	….	stream-m record

Figure 47 - Delayed acknowledgement resynchronization command block format

The record format is shown in Figure 48.

octet:1	2
stream ID	Resynchronization sequence number

Figure 48 - Delayed acknowledgement resynchronization command record format

The stream ID field is 2 octets and represents the stream to be resynchronized between the source station and destination station.

The resynchronization sequence number field is 2 octets and represents the sequence number of the first frame after the command.

6.5.3 Power management

6.5.3.1 Sleep state request

Stations that wish to go into a sleep state while associated with the network must receive permission from the master to do so. This command is used to request permission to go to sleep and the format is shown in Figure 49.

octet:1	2	2
command type (=0x31)	length (=2)	Sleep period

Figure 49 - Sleep state request command block format

The sleep period field represents the time from the entry into sleep state until the station wakes up and receives a beacon.

6.5.3.2 Sleep state response

The master can grant or deny a sleep state request, and the station can only go to sleep if the master grants the request.

octet:1	2	2	2	1
command type (=0x32)	length (=4)	sleep period	beacon sequence number at start of sleep	reason code

Figure 50 - Sleep state response command block format

The sleep period field is the period of time until the sleeping station will wake and hear a beacon. If the master denies the sleep request this field is set to ‘0’ and the reason code is set to the reason for the denial; if this field is not ‘0’ the reason code is ignored.

The beacon sequence number at start of sleep is the sequence number of the beacon that starts the sleep state of the station. The station receives a beacon with this sequence number and begins to sleep from the next superframe.

The reason code values as are follows.

- 0: no resources

- 1: channel change in progress

- 2: master handover in progress

- 3: unknown reason

- 4-255: reserved

6.5.3.3 Activation indication

This command is used by a station in a sleep state or with a granted sleep state request, in order to request reactivation from the master. After acknowledgement of this command, the master halts buffering for the station and sends frames to the station.

octet:1	2
command type (=0x33)	length (=0)

Figure 51 - Activation indication command block format

6.5.3.4 Transmit power adjustment

This command is used to adjust transmit power for a station, and the format is shown in Figure 52.

octet:1	2	1
command type (=0x34)	length (=1)	Transmit power

Figure 52 - Transmit power adjustment command block format

The transmit power field shows the transmit power adjustment amount requested by the station in 2's complement format, in dBm. For example +2dBm is encoded as 0x02, and -2dBm is encoded as 0xFE. Also, 0x7F means that a transmit power adjustment has not been requested.

6.5.4 Key management

This set of commands is used to establish the security and privacy functions between a station and the master and between stations in the network.

6.5.4.1 Request key

This command requests a payload protection key from the key originator, and the SEC field in the frame control field shall be set to one.

Sharing the management key, the requesting station and the key originator protect this command. The format of this command is shown in Figure 53.

octet:1	2
Command type	length (=0)

Figure 53 - Key request command format

6.5.4.2 Request key response

This command sends the requested key in an encrypted format from a key originator to the requesting station, and the SEC field in the frame control shall be set to one. Sharing the management key between the requesting station and the key originator, this command shall be protected and the integrity is generated. The format of this command is shown in Figure 54.

octet:1	2	2	16
Command type	length (=2+16)	SECID	Encrypted key

Figure 54 - Request key response command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID identifies the type of key and the key originator and is defined in 6.2.2.2.The encrypted key field is defined in the symmetric key security operations, as described in 8.3.6. The Encryption Offset field shall be set to five.

6.5.4.3 Distribute key request

This command transmits a key to another station, and the SEC field in the frame control field shall be set to one. If the source ID is the mastered, this command may have the ACK policy field set to no-ACK. Sharing the management key between the requesting station and the key originator, this command shall be protected. The format of this command is shown in Figure 55.

octet:1	2	2	16
command type	length (=2+16)	SECID	Encrypted key

Figure 55 - Request key response command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID identifies the type of key and the key originator and is defined in 6.2.2.2.

The encrypted key field is defined in the symmetric key security operations, as described in 8.3.6.

6.5.4.4 Distribute key response

This command informs the key originator whether or not the key was properly received in a distribute key protocol, and the SEC field in the frame control field shall be set to one. Sharing the management key between the requesting station and the key originator, this command shall be protected. The format of this command is shown in Figure 56.

octet:1	2	2
command type	length (=2)	SECID

Figure 56 - Request key response command format

The SECID is the unique identifier for the security relationship with which the key is associated. The SECID identifies the type of key and the key originator and is defined in 6.2.2.2.

6.5.5 Security management

6.5.5.1 Security message

This command sends security related information to another station in the network, and the SEC field in the frame control field shall be set to zero. The format of this command is shown in Figure 57.

octet:1	1	3	Ln
Element ID	length (=3+Ln)	VendorOUI	Security information

Figure 57 - Vendor specific security information format

The security information field includes security related information defined by the vendor identified in the vendor OUI field. Its use by a station is outside of the scope of this standard.

6.5.6 Vendor specific

6.5.6.1 Vendor specific command

The format of the vendor specific IE is shown in Figure 58.

octet:1	1	3	Ln

Element ID	length (=3+Ln)	VendorOUI	Vendor specific information

Figure 58 - Vendor specific information element format

The vendor specific information field is defined by the vendor identified in the vendor OUI field, which is specified in IEEE RAC. Its use by a station is outside of the scope of this standard

6.5.7 Other

6.5.7.1 Station information request

This is used when a slave requests information about a specific station or stations from the master, and the format is shown in Figure 59.

octet:1	2	1
command type (=0x41)	length (=1)	ID of target of information request

Figure 59 - Station information request command block format

The information request ID field represents the assigned ID of the station about whom the slave is requesting information from the master. If this field is set to the broadcast ID, it means that the request concerns all stations.

6.5.7.2 Station information response

The station information response command block format used by the master to respond to station information requests is shown in Figure 60.

octet:1	2	12	12	…	12
command type (=0x42)	length (=12*N)	station1 station information block	station2 station information block	…	station1 station information block

Figure 60 - Station information response command block format

The requested information about each station currently associated with the network is included in data records, and the format is shown in Figure 61.

octet:6	1	1	2	2
station UID	station ID	state	capability	sleep period

Figure 61 - Station information block format

The station UID field represents the 48 bit IEEE address of the station.

The station ID field represents the ID assigned by the master.

If the state field is 0 it means that the station has been associated but not yet verified, and if 1, it means that the station has been associated.

The capability field shows the station capability as in Figure 28.

The sleep period represents the sleep period if the station is sleeping.

6.5.7.3 Data query

The data query command is used to request information from a station or respond to it. This command can be exchanged between two stations in network. The components of this frame are shown in Figure 62. The stream ID in the Frame header of the data query command is set to 0 and ignored when received.

octet:1	2	2	variable
command type (=0x43)	length	data request	information block

Figure 62 - Data query command block format

The upper 15 bits of the data request field are a bitmap of the requested information. The requesting station sets each bit corresponding to the requested information to 1. The location of bits in the data request field is equivalent to that of the information block ID in the information block. In other words, the n-th bit of the data request field corresponds to the information block with an ID of ‘n’. If all data request fields are set to 0, the station is requesting no information but reporting its own information to the source station.

The data request field’s MSB determines whether the rest of the bits are a bitmap or a representation of the request information block ID.

The information block field is a list of information blocks as explained in 6.4. Each information block can be flexibly arranged, and not all information blocks need to be contained therein.

6.5.7.4 Channel state request

This is sent to a station to find out its current channel state, and the format is shown in Figure 63.

octet:1	2
command type (=0x44)	length (=0)

Figure 63 - Channel state request command block format

6.5.7.5 Channel state response

This command is sent as a response to the station requesting the channel state, and the format is shown in Figure 64.

octet:1	2	2	2	2	2	2
command type (=0x45)	length(=10)	assessment period (kμs)	number of transmitted frames	number of received frames	number of erroneous frames received	number of lost frames

Figure 64 - Channel state response command block format

The assessment period field represents the time for the channel assessment in kμs.

The number of transmitted frames represents the total number of frames sent by the station.

The number of received frames represents the total number of frames received by the station, and this number includes only those frames with the source station ID set to this station.

The number of erroneous frames received represents the total number of erroneous frames received by the station. Here an erroneous frame means a frame that passed the PHY header error-check but did not pass the frame error-check.

The number of lost frames represents the total number of frames whose existence is expected but have not been received. If a received frame's sequence number is not one higher than the previously received frame in the stream, this indicates that a frame has been lost in the interim, and the difference between the consecutive frames' sequence numbers minus 1 is the number of lost frames. These differences are summed up across all streams and placed into the number of lost frames field. Resent frames are excluded from this calculation.

6.5.7.6 Remote channel scan request

This is used by the master to request a remote channel scan to a slave to find out the channel state of a different frequency. Only the master can send this command frame, and the format is shown in Figure 65.

octet:1	2	2
command type (=0x46)	length (=N)	channel bitmap

Figure 65 - Remote channel scan request command block format

The channel bitmap field shows the channels to be scanned in a bitmap format. The Nth bit corresponds to the Nth channel, and if the bit is 1 the channel is scanned, otherwise it is not scanned.

6.5.7.7 Remote channel scan response

This is sent by a station to the master as a response to a remote channel scan request, and the format is shown in Figure 66.

octet:1	2	1	1	5	5	…	5
Command type (=0x47)	length (=1 or 3+5*N)	Reason code	Number of channels (=N)	Channel information record #1	Channel information record #2	…	Channel information record #N

Figure 66 - Remote channel scan response command block format

The reason code field has the following values.

- 0: success

- 1: requested denied

- 2: invalid channel index was requested

- 3-255: reserved

If this field is 0, no fields follow afterwards, and the length of the command is 1.

The number of channels field represents the number of channels that are scanned and reported.

octet:1	1	2	1
Channel ID	Channel quality	Network ID	Scanned frame type

Figure 67 - Channel information block format

The channel ID field represents the ID of the scanned channel.

The channel quality field represents the quality of the scanned channel.

The network ID field represents the network ID discovered during the scan. If this field is 0, it means that no network was discovered.

The scanned frame type field represents the type of the frame discovered during the scan. This field only has relevance when the network ID field is not 0.

6.5.7.8 Application specific

This purpose of this command is to allow vendors to perform the enhanced operation that is out of this standard. For example, by using this command, devices in the network exchange more detailed information such as queue size or number of collision to provide QoS. The frame format is illustrated below.

octet:1	3	1	variable
command type (=0x48)	Vendor Unique ID	length (=3+N)	Vendor defined command

Figure 68 - Application specific command format

7 MAC feature description

This chapter describes the role of the MAC layer.

7.1 describes how each station establishes or associates itself with the network.

7.2 describes how the stations individually access the media by period.

7.3 describes how the master and slave synchronize with the superframe by means of the beacon frame.

7.4 describes the means by which the quality of service of the streams between the stations is ensured. Each stream is allocated to the resources to satisfy their individual quality of service, and the resources are adjusted to support quality of service even if channel conditions change.

7.5 describes the process of fragmentation and defragmentation.

7.6 describes the acknowledgement of a transmitted frame and the retransmission process when transmissions have failed.

7.7 describes the efficient power saving method.

7.8 describes the dynamic channel changing method when data communication becomes difficult due to excessive channel noise or too many stations are occupying one frequency channel.

7.9 lists the values of the parameters used in the MAC layer.

7.1 Network formation and association

For one station to communicate with another, they must first be associated with network. Basically, once the station scans each frequency channel to find a existing network, if one exists it associates itself with it, otherwise it appoints itself master and forms a new network on a specific frequency channel. (Forming a new network means periodically transmitting beacon frames on a certain frequency channel.) However, depending on the station’s capabilities it may simply continue scanning without forming a network, or may form a new network regardless of whether or not one already exists. In the latter case, if a network already exists on every frequency channel, the formation of a new network may be abandoned.

7.1.1 Channel scanning

Each station listens passively for a beacon frame from the master in order to discover an existing network.

The station selects a certain frequency channel and waits for a beacon frame for *channel scan time*, and if it hears a beacon from the master during this interval it determines that there exists a network on this frequency channel, otherwise it determines that no network exists on the frequency channel and moves on to the next frequency channel to scan.

When DME specifies a NID in its scan request, it ignores all beacons other than the beacon from the specified network, and in the case of an open scan where no such designation is made, all beacons are received.

Most stations continuously scan each frequency channel to find an existing network, but stations whose DME has determined not to become the master wake up one per *channel scan period* and perform one channel scan in *channel scan time* in order to reduce power consumption.

7.1.2 Network ID

The master ensures that its newly formed network ID does not overlap with another network. The method of this generation is beyond the scope of this International Standard.

7.1.3 Association

Stations disassociated with a network may request network association during the contention period, and when the master receives the request it responds with an associate request acknowledgement. The master determines whether to grant the station's network associate request and reports the result in an associate response frame. If association has been granted, the associate response frame contains the assigned ID, and when denied the reason for denial is reported. If the station wanting association does not receive an associate response within the associate response time limit, it retries the associate request. The slave sets the associate response time limit.

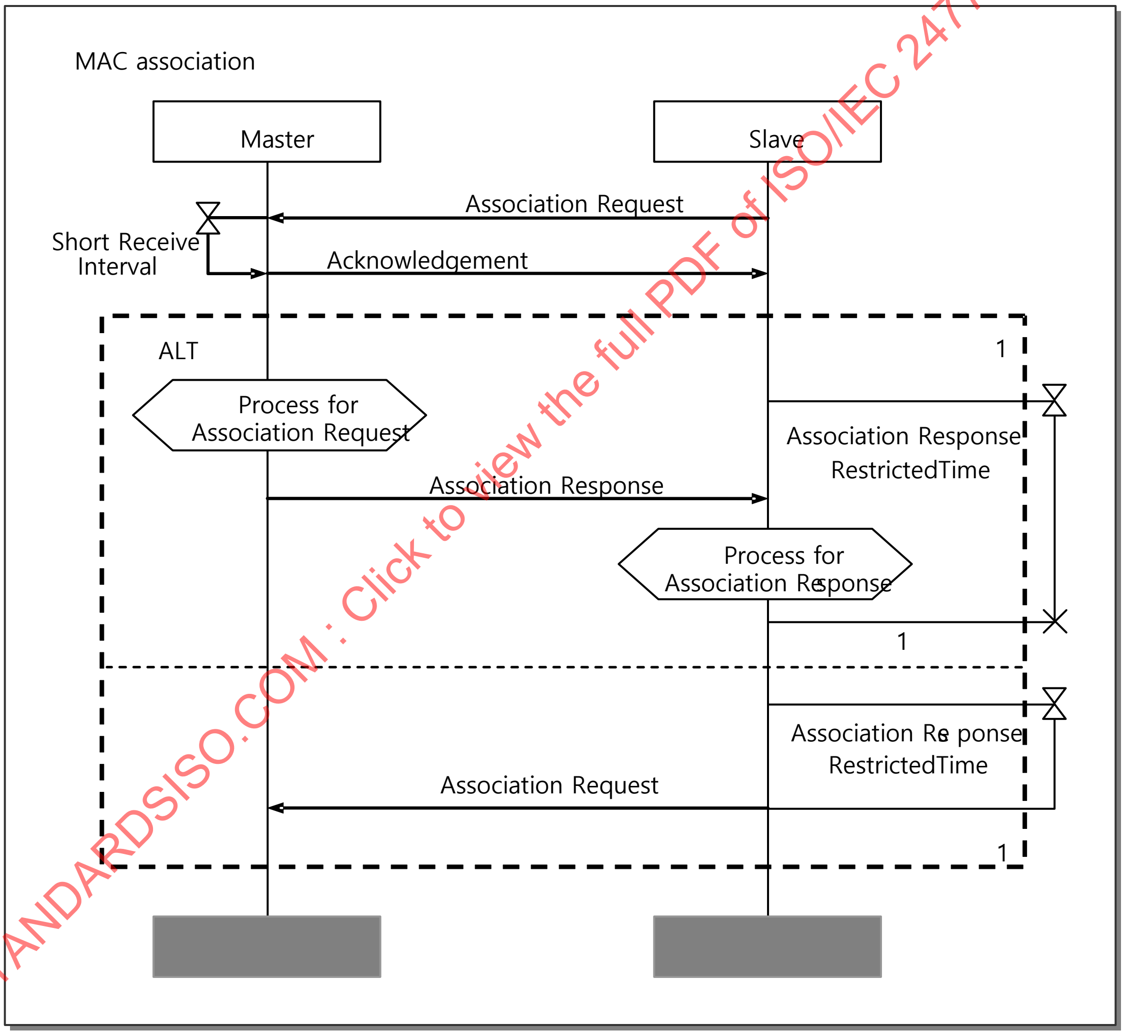
MAC association
Master
Slave
Association Request
Short Receive Interval
Acknowledgement
ALT
1
Process for Association Request
Association Response RestrictedTime
Association Response
Process for Association Response
1
Association Re ponse RestrictedTime
Association Request
1

Figure 69 - Association Process

7.1.4 Disassociation

Stations desiring to disassociate from network send a disassociate request and reason code to the master, and the master acknowledges this disassociate request with acknowledgement frame.

7.1.5 Master handover

Since the master must receive resource allocation requests from associated stations and notify other stations of the resource allocation result through beacon frames, its power consumption is large compared to slaves. Also, since the capabilities of the whole network depend on the processing capabilities of the master station, more hardware capabilities is required. Therefore, the station with the highest power capacity or hardware performance needs to be selected as master of network.

If a station superior to the master exists among the associated stations, the current master may handover its master role to that station.

7.2 Media access

The superframe is divided into three periods – the beacon period, contention period, and allocation period – and each period uses different media access methods. During the beacon period the master uses broadcasting to send beacon frames, and during the contention period and allocation period the stations use CSMA/CA and TDMA/CDMA respectively to access the media.

In the allocation period, media access is achieved by the pair of two resources – time and codes. This means that in order to transmit a frame a station must be allocated a unique time slot and set of codes.

Stations use only the network standard code rates set by the master in the beacon period and contention period of the superframe, and during the allocation period they use the code rates assigned by the master.

7.2.1 Code assignment

The master can specify the codes to be used in network. All frames other than data frame are encoded with the network standard codes set by the master, and data frames are encoded with codes that the master has assigned to streams.

Therefore, frames other than data frames which are exchanged during the beacon period and contention period use the network standard code rates, and in the allocation period several codes allocated by the master may be freely exchanged. For data frames to be transmitted during the contention period, network standard code rates must be used.

There is a trade-off between bandwidth and interference resistance, so when interference is low and channel conditions are good the master maximizes data transfer rate by using as many codes as possible, while if interference is high and channel conditions are poor it improves interference resistance by reducing the codes for data transfer. When the master assigns codes for a data stream it takes into consideration the quality of service requested by the stream.

7.2.2 Inter-frame space

Three types of frame spaces exist - short inter-frame space (SIFS), backoff inter-frame space (BIFS), and retransmission inter-frame space (RIFS). Short inter-frame space is shorter than backoff inter-frame space, and the actual values of these two are determined by physical layer attributes such as Rx/Tx turn around time of RF module, frame processing time, etc. .

Acknowledgement frames for all frames sent during the allocation period or contention period are sent after a period of short inter-frame space. Even if an acknowledgement frame is not needed a short inter-frame space may apply between two consecutive data frames.

BIFS is used during the backoff interval of the contention period. In the contention period, once all slaves have determined that the media is idle, after BIFS they may decrement the backoff counter or send data.

RIFS is used to retransmit frames during the allocation period. (Retransmission of frames during the contention period is also described in 7.2.3.)

7.2.3 Access during the contention period

During the contention period, exclusive access to the media is not ensured through unique allocation of codes and time from the master, therefore each station uses the competitive CSMA/CA method to access the media.

If the station has a frame to send and the media is idle during the backoff period as calculated in 7.2.3.1, the frame is sent.

7.2.3.1 Backoff process

When frames other than the immediate acknowledgement frame are sent during the contention period the following rules apply.

The backoff algorithm uses the following information.

- retry_count: 0, 1, 2, 3

- backoff_window(retry_count): [7, 15, 31, 63]

- pBackoffSlot: as defined in 9.1.2

- bw_random(retry_count): random integer selected from the normal distribution [0. backoff_window(retry_count)]. There is no statistical correlation between one station's random number and another. If a station does not have a random number generator, the station's unique MACaddress or other information can be used with the pseudo-random number generator (PRNG) to generate the random number. The PRGN must be retained to maintain the pseudo-random format of numbers to be generated in the future. It is important to ensure the non-inter-dependency of the random numbers between the stations.

The backoff time in the contention period is measured in the medium and indicates the time for the station to send the data. Once the station determines that the media is idle, before beginning the backoff algorithm it waits for the BIFS time. At the start of the contention period after the station sends the beacon frame it waits during the SIFS time and then starts the backoff algorithm. If the beacon is divided such that two or more beacons occur in sequence, after the final beacon it waits for the SIFS time and then begins.

The station calculates the backoff_count using the formula *backoff_count = bw_random(retry_count)* and maintains this using a counter. The counter is decremented by 1 only when the media is idle during pBackoffSlot. When the frame is first sent, the retry_count is set to 0. When a channel is in use, the backoff counter is suspended. When the backoff counter reaches 0, it starts frame transmission.

The backoff counter must be suspended during periods other than the contention period. It must also be suspended during the contention period when there is not enough time left to send the desired frame. The backoff counter is maintained independent of the superframe period, which means that it is not reset for each beacon. If the time elapsed after the frame is queued for transmission exceeds the transmit time limit, the backoff counter is reset and the frame transmission is aborted.

After a frame needing acknowledgement frame is sent, if no acknowledgement frame is received, the retry_count is incremented. However, the retry_count cannot exceed a maximum of 3, so an attempt to retransmit the failed frame is made up to twice. The backoff_count is recalculated using a new retry_count.

7.2.4 Access during the allocation period

During this period each station has exclusive access to the media during the time slot assigned to itself. The master distributes time slots in the allocation period to each station. In the assigned time slot each station can have exclusive access to the media, and during this assigned slot the station can exchange data in a peer-to-peer with other stations without intervention from the master. The master specifies the start time and length of each time slot, source/destination station, and codes to be used in the beacon, thereby ensuring exclusive access to the media for each station to send frames.

In this period the station uses allocated codes to exchange data with other stations. Since the time slot and code allocation is broadcasted to all stations by means of the beacon frame, each station can know its own time for sending or receiving data.

The inter-frame interval in the allocation period uses SIFS as shown in Figure 70, and the frame transmission must be completed within SIFS + Guard time from the end of the period. The Guard time is calculated as follows.

Guard time = *maximum superframe length* * *clock drift* / 1,000,000 * 2

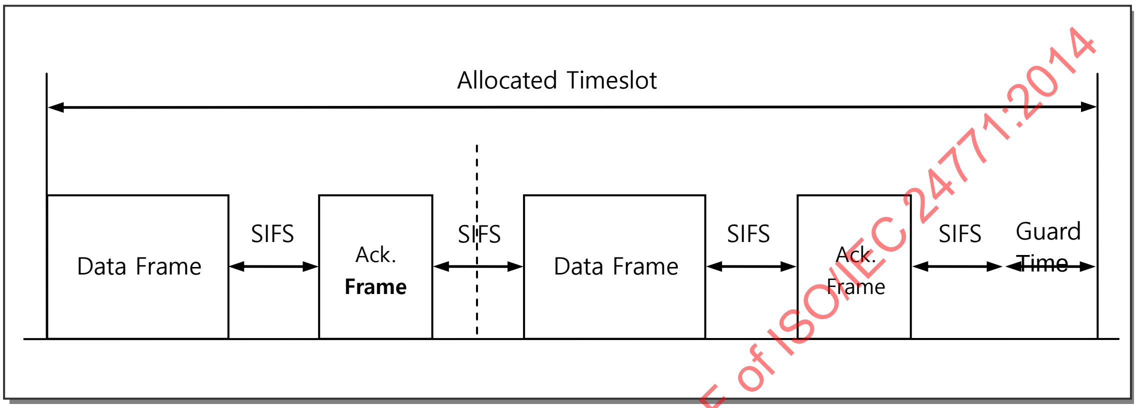

Figure 70 - Inter-frame space in the allocated time slots

7.3 Synchronization

All stations participating in a particular network are synchronized by the beacon frame. In the beacon frame transmitted during the beacon period of each superframe, the synchronization information for the associated station is included.

7.3.1 Superframe synchronization

The superframe is divided into three periods as shown in Figure 71, and slaves synchronize to the master's superframe by using the information in the beacon frame received from the master. The superframe length is set to *superframe length* and the beacon is repeated by that time period, and the length of the allocation period is variable or an multiple of *slot length*. The time between a beacon frame and the allocation period is used for the contention period.

The length of the contention period must be greater than *minimum* contention *period length.*

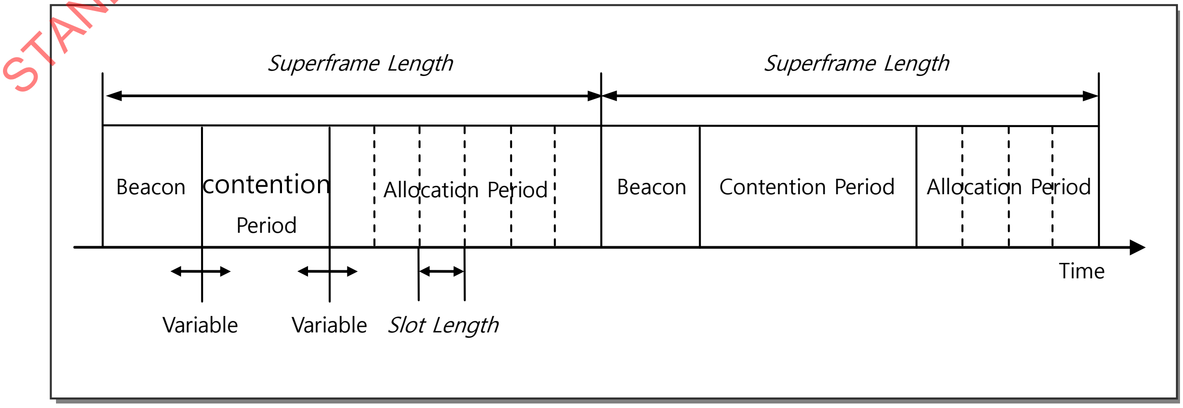

Figure 71 - Superframe synchronization

The beacon frame contains the start time of allocation period, and using this start time and the time the beacon is received, slaves in network can determine start and end time of each superframe.

7.3.2 Beacon generation

The master fills the beacon frame with the allocation period start time and time slot allocation information and sends it.

7.3.3 Beacon reception

Slave can determine the boundaries of the contention period and allocation period from the allocation period's starting offset in the beacon, and determine the start and end time of the time slot allocated to them from the time slot allocation information.

7.3.4 Synchronization

All stations in network synchronize by means of receiving beacon frames from the master. Disassociated stations use manual scanning to hear beacon frames and synchronize. Therefore, stations wanting to associate with network must hear a beacon frame at least once before being able to associate. If an associated station cannot hear a beacon frame, it waits till it hears the next beacon frame in order to synchronize. If a beacon frame is not heard within *connection time limit,* slave determines that the network is terminated and finishes all relevant exchange of network stream.

7.4 Resource allocation

Data transmitted is categorized as synchronous data and asynchronous data. Synchronous data, which has QoS requirements, builds stream for transmission, and the master continuously monitors the transmission status from connection till termination. On the other hand, for asynchronous data which does not have QoS requirements there is no need of a stream connection, and resources are requested and allocated as needed for data transmission. An example algorithm that allocates time slots to meet its QoS requirements is described in “0. Example scheduler and admission control”. However scheduling and admission control implementations are vendor-specific and beyond the scope of this International Standard.

7.4.1 Transmission of synchronous data

The following clauses describe stream management, connection, termination, and prioritization. Each station must support the connection of at least one stream, and the maximum number of supported streams is dependent on the nature of the user application.

7.4.1.1 Stream connection

A stream is built when the station sending the stream requests the setup and resource allocation for the stream, and the master allocates a streamID and resources for that purpose. At the start of the stream creation the streamID is assigned by the master, but afterwards the sending and receiving stations manage the stream through a 1:1 communication method. A station receiving or sending a new stream sends a resource allocation request command to the master, and the connection process is shown in Figure 72. The master decides the time slot and codes for the stream based on the allocation period, size of data, and required BER, and periodically allocates resources for the stream until data transfer for that stream terminates.

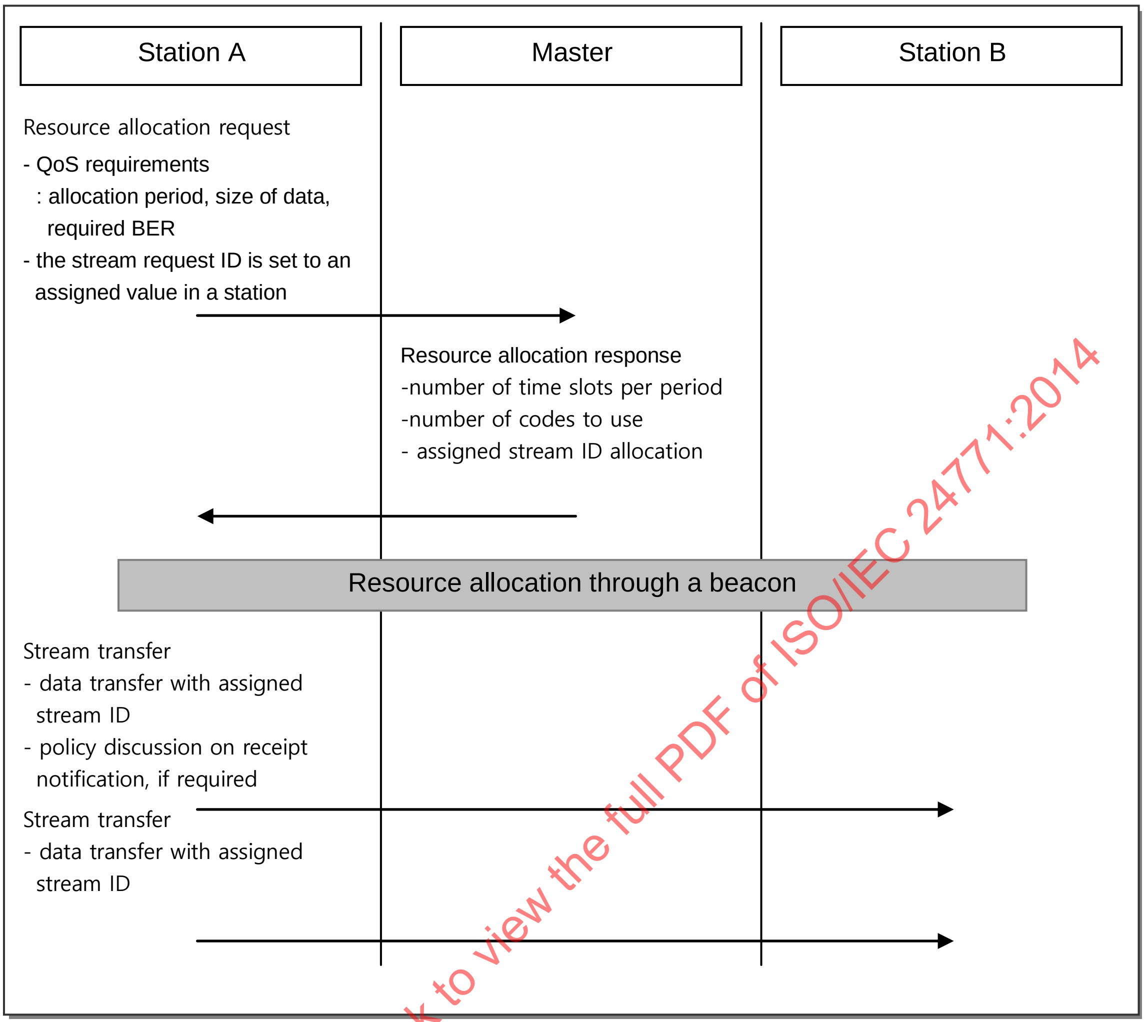
Station A
Master
Station B
Resource allocation request
- QoS requirements
: allocation period, size of data, required BER
- the stream request ID is set to an assigned value in a station
Resource allocation response
-number of time slots per period
-number of codes to use
- assigned stream ID allocation
Resource allocation through a beacon
Stream transfer
- data transfer with assigned stream ID
- policy discussion on receipt notification, if required
Stream transfer
- data transfer with assigned stream ID

Figure 72 - Stream connection process for synchronized data transmission

7.4.1.2 Stream QoS modification

When changing the stream's QoS requirements, the station sending the stream can achieve this by sending a resource allocation request command to the master. In such a case, the stream request ID is set to 0, and the stream ID is set to the formerly assigned stream ID. When the master receives a stream modification request, it sends the response in a resource allocation response command to the slave.

When the master receives a request to change the QoS requirements, if the stream's allocated resources can be modified, the resource allocation changes are put into a resource allocation response command and sent, and the changes apply from the next superframe. If the master wants to deny this request, it simply sets the field of the resource to be allocated (number of time slots or codes) to 0, populates the reason code with the reason for denial, and sends the response.

7.4.1.3 Stream terminating

If the station sending the stream or the station receiving the stream wants to terminate it, it sends the master a resource allocation response command with the reason code set to 'request denied'. When the master receives this stream response request, it stops allocating resources for the stream. When the sending and receiving stream are not allocated resources for more than *allocation period*maximum beacon loss*, they interpret this as a sign that the stream has been terminated and abandon service for that stream.

7.4.1.4 Prioritization

The scheduler in the master determines the actual prioritization, and that is beyond the scope of this International Standard, but here the guidelines for prioritization are presented.

The master gives preference to the connected stream in network with the highest priority when allocating resources. The master remembers the priorities of all currently connected streams, and uses this to allocate channel time and codes to streams.

Each station can decide upon its own transfer method within the allocated channel time, but with the exception of synchronous streams with higher priority must be transmitted before lower priority streams. Asynchronous streams or streams sent to sleeping stations must be transmitted according to a predetermined schedule, and are therefore excluded from the prioritization rules.

7.4.2 Asynchronous data transmission

When a slave transmits asynchronous data with no QoS requirements, it makes resource allocation requests during the contention period, and the master allocates resources enabling asynchronous data transmission. The slave's resource allocation request does not contain the period or BER requirement information, and only holds the size of the data to be sent. The master first allocates resources for synchronous data requiring a stream connection, and allocates any remaining resources for asynchronous data.

A slave sends resource allocation request command to the master during the contention period, and waits for a resource allocation response command during the RequestTimeout time shown in 5.3.13. While waiting for a response command the slave also stays awake during the contention period and waits for a response from the master. If the resource allocation response command is received or resources are allocated for the stream through a beacon even before a response is received, it interprets this as a response to its request and sends the frame using the allocated resources.

7.5 Fragmentation and defragmentation

When the length of data to be sent exceeds the fragmentation limit, the data goes through a process of being divided into many smaller fragments before being sent. The receiving station reassembles the data fragments by means of a defragmentation process and restores the data to its original length.

The source station can freely designate the fragmentation limit depending on the type of transmission stream and channel state, but all fragments of this stream except the last must be of the same length, and once set it must be applied to the fragmentation limit of all fragments.

The first bit of the frame control field of the first fragment to be sent is set to 1, and the last fragment's final bit is set to 1. (If the fragment is both the first and last, both values are set to 1.) All other fragments besides the first and last set their first/final bits to 0/0.

The reassembly takes place in the MAC layer and is transparent to the higher layer. After the fragmented data is completely reassembled in the MAC layer it is sent to the higher layer. The destination station must discard the entire data, even if one fragment is lost. If delayed acknowledgement is used, the receiving station can request a retransmission of the fragment from the most recent fragment's sequence number within the retransmission window.

There is no theoretical limit on the fragmentation limit. However, the fragmentation limit is restricted by implemental considerations to *maximum fragment limit*.

7.6 Acknowledgement and retransmission

To receive acknowledgements for data transmitted, there are four methods: no acknowledgement, immediate acknowledgment, delayed acknowledgement, and implicit acknowledgement. The acknowledgement method to be used is recorded in the Frame header.

7.6.1 No acknowledgement

If the acknowledgement policy is no acknowledgement, the source station that sends the frame does not wait an acknowledgement, and automatically assumes that the data transmission was successful.

7.6.2 Immediate acknowledgement

If immediate acknowledgement is used, the source station expects an acknowledgement frame immediately after sending the data, and the destination station sends an acknowledgement frame after a SIFS once data is received. This acknowledgement method can be used in both the contention period and allocation period.

7.6.3 Delayed acknowledgement

When a delayed acknowledgement policy is adopted, once the source station sends data, the acknowledgement for the data is sent to the source station during the destination station's time slot. If such a method is used the source station can continuously send data frames and receive a one-time acknowledgement afterwards. The source station must store frames up to the pre-decided size of the delay acknowledgement window in order to enable retransmission until an acknowledgement is not received. Destination station should send the delayed acknowledgement before the delayed acknowledgement window of source station is exhausted.

The destination station can send an acknowledgement frame at any time up till the retransmission window is exhausted, and the source station can request the transmission of an acknowledgement frame by setting the delayed acknowledgement request field in the frame header to '1'.

7.6.4 Implicit acknowledgement

The implicit acknowledgement policy is only allowed during the allocation period. It is similar to immediate acknowledgement in that the station sending the data waits immediately for an acknowledgement, but differs in that the destination station acknowledges by sending a data frame that is not a simple acknowledgement frame. Stations sending implicit acknowledgements can request acknowledgement for their transmitted data using methods other than implicit acknowledgement.

7.6.5 Retransmission

During the contention period the acknowledgement method described in 7.2.3 is used.

When immediate acknowledgements or implicit acknowledgements are used during the allocation period, the source station waits for acknowledgement frames after a short inter-frame space after sending the data. When receiving an acknowledgement frame, it transmits other data after waiting for a SIFS after the acknowledgement frame it received, and if it does not receive an acknowledgement frame it retransmits the data after RIFS.

When delayed acknowledgement is used, the source station denies any retransmit request from a destination station for data that is outside the retransmission window. In such cases, by sending a delayed acknowledgement resynchronization command the relayed acknowledgement window between source and destination station is resynchronized.

7.6.5.1 Duplicate frame detection

If the source station does not receive acknowledgement, it may send a duplicate frame to the destination station even if the data has already been received. Such duplicate frames can be weeded out by using the stream ID and sequence number in the frame.

7.7 Power saving

Slaves can know when they transmit or receive frames decoding beacon frames, and can reduce unnecessary power consumption by only activating their physical layer when needed. To reduce power consumption even further, sleep mode can be used.

7.7.1 Saving power in a connected state

When there is no frame to transmit, a slave associated with network activates its physical layer and listens for beacons only during the beacon period. When hearing a beacon frame, if the command or data frame is intended for itself, it activates its physical layer in its allocated time slot during the contention period or allocation period and receives the frame.

Stations to send a command frame activate the physical layer during the contention period to send the frame, and if they need a response to the command frame they activate the physical layer and wait for a response frame during the contention period until the *command response time limit* runs out. Stations to send a data frame activate the physical layer in their allocation period and send the data frame.

When the master has no frame to transmit/receive, it activates the physical layer and transmits or receives frames during the beacon period and contention period. When the master has data to send to the slaves, it notifies them of the time they will receive it (the contention period or allocation period), and communicates with the slaves during the slaves' waking time.

7.7.2 Sleep state

When a slave has no data to send or receive, it can go to sleep by requesting to do so and receiving permission from the master. After master and slave exchange sleep state request and response, the period parameters relating to sleep state are set.

When a slave is granted permission to go into sleep state, it goes to sleep and cannot send or receive frames during that time. Also, it wakes up and receives beacons once per period as arranged with the master, in order to determine whether there is a data frame to receive. If there is data to receive, it receives it during the allocation period, otherwise it goes back to sleep.

Stations that are not sleeping can determine the IDs of the stations that are asleep and their wakeup periods by listening to beacons. If they need to send data to stations that are asleep, they request resource allocation from the master, and if successful, the master allocates resources to the sleeping station via the beacon heard during its wakeup period. Thus the sleeping station remains awake during its allocation period, and the station wishing to transmit will send the data during that period. When data is received, this automatically indicates that the sleeping station has been awakened from its sleep.

7.8 Dynamic channel management

7.8.1 Channel state probe

Stations in network may also use channel state request commands to find out the channel state between stations. Channel state request commands and channel state response commands both use an immediate acknowledgement policy. Stations wanting to access channel quality send a channel state request command to the destination station, and the receiving station reports the channel state information via a channel state response command.

7.8.2 Remote Channel state probe

A remote channel state probe occurs when the master requests another slave in network to scan another frequency channels on behalf of the master and the slave performs the requested channel scan and reports the result. The master may adjust the maximum transmit power or change channels depending on the result of this remote Channel state probe.

The master is unable to scan other frequency channels because of having to periodically generate beacons and allocate resources; therefore it makes use of capable slaves in network to assess the quality of other frequency channels. The master may also use this feature to determine the existence of other networks that are too remotely located to directly communicate with.

The master begins this process by sending a remote channel scan request command to a slave with the frequency channel list. The slave must obey the master's request if possible, but if it cannot perform the remote channel state scan it denies the request by sending a remote channel scan response command with the response code set to "request denied". If the request is denied, the master can send the remote channel scan request to another slave.

If the slave complies with the master's request, it scans the requested channel one by one and puts the result into a remote channel response command and sends it to the master. If the slave unexpectedly receives a beacon from another master, it informs the master that there is an overlap between two networks by sending information about this received beacon in a remote channel response command.

7.8.3 Frequency channel change

If the master checks the current channel state through other slaves and determines that there is excessive interference, it notifies the stations of the new frequency channel and the channel change time limit via a beacon so that they can migrate to the new frequency channel and start a new network. If the beacon is changing the channel, the channel change information must be repeated at least *maximum beacon loss* times. As soon as the slave receives the channel change beacon, it tries to change channels, and when it receives a new beacon on the newly changed channel it means the channel change was completed. If no beacon is received on the new channel by the specified time limit, stations form a new network on that frequency channel.

7.9 MAC parameters

Table 47 - MAC layer parameters

name	value	comments
Minimum contention period length	1 msec	
Command response time limit	100 msec	
Slot usage verification time	250 usec	
Slot length	1 msec	
Connection termination time	400 msec	
resource allocation response time	100msec	
Maximum fragmentation	2047 octets	
Maximum beacon loss	4	
channel scan time	=superframe length	
channel scan period	= 100 * channel scan time	
minimum superframe length	5120 usec	
maximum superframe length	65535 usec.	
Minimum payload length	12 octets	
Minimum allocation period start time	2048 usec	
Maximum payload length	2047 octets	
maximum allocation period start time	5000 usec	
Clock drift	25ppm	

8 Security

Wireless networks need to protect data from cracking and eavesdropping, to check information integrity, to manage key, and to establish a secure network. This standard supports three different modes of security: no security, inclusive CCM, and exclusive CCM. The standard supports the protection of command, beacon, and data frames using CCM mode based on block cipher algorithm, and the distribution of keys for command and data frame protection.

Cryptographic algorithms to be applied to information security mechanism may be subject to the ISO/IEC standardized encryption algorithms or regional regulations. They should conform to national laws and regulations, and can be chosen according to specific requirements in different countries and regions. So, AES, Camellia, SEED is recommended but it is optional.

8.1 Security mechanisms

Security mechanisms specified in this standard control the security operation of devices by setting appropriate security modes. This standard also provides a symmetric cryptography mechanism to assist in providing security services. Additional security services need to be provided by the higher layers to ensure proper management and establishment of the symmetric keys used in this standard. In addition, the security mechanisms provide replay attack prevention measures through the use of secure frame counters (SFCs) and replay counters. The security mechanisms specify the parameters needed in applying the secure block algorithms to protect the data.

8.1.1 Security membership and key establishment

To become a member of a security relationship and to obtain the master key between stations is outside of the scope of this standard. The security message command has been included as a special command to assist in the implementation of vendor specific protocols for establishing security relationships and any related data. It can be achieved with higher layer protocols that are not specified in this standard.

The MAC/MLME is informed of changes to the membership of a security relationship and the key for that relationship with the MLME-MEMBERSHIP-UPDATE primitive, as described in 8.3.2.

8.1.2 Key transport

As described in 8.4.2 for the key request and in 8.4.1 distribute key protocols, all keys that are transmitted from one station to another shall be encrypted. For example, key transport is used to provide a copy of the network group data key to a station.

8.1.3 Data encryption

Data encryption uses a symmetric cipher to protect data from access by parties not possessing the encryption key. Data may be encrypted either by using a key shared by all stations in the network or by using a key shared between only two stations. Secure data frames that are encrypted with unknown SECID are notified to the DME using MLME-SECURITY-ERROR.indicate with the ReasonCode set to INVALID-KEY. This standard adopts counter mode security with secure block algorithms.

8.1.4 Data integrity

Data integrity uses an integrity code to protect data from being modified by parties without the cryptographic key. It provides assurance that data came from a party with the cryptographic key. Integrity may be provided using a key shared by all stations in the network or using a key shared between only two stations. All secure data frames that fail integrity checks are passed to the DME using MLME-SECURITY-ERROR.indicate and no other action is taken on the frame by the MLME. This standard adopts CBC-MAC mode integrity check with secure block algorithms.

8.1.5 Beacon integrity protection

The integrity of beacon may be protected. This provides evidence to all the stations in the network that the master of the secure network transmitted the beacon. Under normal operations, the integrity check on the beacon provides evidence that the network is operating properly and that no security changes have occurred. If the integrity check on the beacon fails, the station is alerted that the station does not have its security state synchronized with the master.

8.1.6 Command integrity protection

The integrity of commands may be protected just like any other data. Using the master-station management key, integrity protected commands sent between the master and a station shall be protected. If integrity checks fail, then all secure commands are passed to the DME using MLME-SECURITY-ERROR.indicate and no other action is taken on the frame by the MLME.

8.1.7 Freshness protection

In the beacon, a strictly-increasing time token is included to prevent replay of old messages. If a received beacon with a time token less than or equal to the current time token, then a station may regard it as invalid. In addition, the time token is included in the CCM nonce, as described in 8.5.2, for each secure frame, as described in 8.5.3, so the integrity check will fail if a frame is replayed in a different superframe.

A station in a secure network maintains time token and secure frame counter for freshness. The CurrentTimeToken is the time token value found in the beacon for the current superframe and is used to protect all messages sent and check all messages received during that superframe.

The secure frame counter, which guarantees uniqueness within a superframe with a given key as long as a station does not send more than 65536 frames to a particular destination within that superframe. If the received secure frame counter is less than or equal to the expected secure frame counter, then the station discard the frame. Secure frame counter is managed for each SECID and it increments for each transmission including retransmission.

8.2 Security modes

The security mode indicates whether a device is permitted or required to establish a secure relationship with another device for data communications. The security mode in use is determined by the MACPIB_SecurityOptionImplemented entry in the MAC PIB.

8.2.1 Security mode 0

A station operating in security mode 0 shall use non-secure frames to communicate with other stations. Such a station shall not establish a secure relationship with any other station.

If a station receives a frame with the SEC field set to one, the station shall discard the frame and the MLME shall return an MLME-SECURITY-ERROR.indication to the higher layer with the ReasonCode set to UNAVAILABLE-KEY.

8.2.2 Security mode 1

A station operating in security mode 1 shall use non-secure frames to communicate with stations operating in security mode 0. The station shall also use non-secure frames to communicate with stations operating in security mode 1 with which it does not have secure relationships. Secured frames shall be used between stations using security mode 1 when the stations establish a secure relationship for secure communications as given in Figure 7.