INTERNATIONAL STANDARD

ISO/IEC 29157

Second edition
2015-07-15

Information technology — Telecommunications and information exchange between systems — PHY/MAC specifications for short-range wireless low-rate applications in the ISM band

Technologies de l'information — Téléinformatique — Spécifications PHY/MAC pour applications à bas débit sans fil à courte portée dans la bande ISM

Reference number
ISO/IEC 29157:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

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 29157:2010), which has been technically revised.

Introduction

This International Standard is the revision of ISO/IEC 29157. This International Standard was established to provide a unified platform for services of data rates up to 1 Mbps: control data, voice, audio, and video. The purpose of the revision is to accommodate the advancement of technology for higher-quality services in the mobile applications.

The direction of the revision is three-fold: to enhance throughput, to facilitate co-existence with other technologies such as time-division LTE (long term evolution) and WiMAX II, and to increase data rate. For the higher throughput, the length of the preamble of the payload frames is made variable up to the user's need (see 9.2.2). For the co-existence, the duration of the middle frames is reduced to 4 ms from 16 ms to make align with those of the other technologies (see 8.3). With the shorter middleframes, the International Standard does not only harmonise with other technologies, but also attains advantages of shorter communication delay and less paring time. To increase the data rate, the message part of the payload frames may be modulated with QPSK. The modulation format is indicated by the 'PHY version' of the header (see Clause 7 and 9.3). With the addition of the new option, the data rate can be increased to 2 Mbps. In addition, to protect communications against security challenges due to the loss of protection provided by wires, this International Standard provides the optional security mechanism (See 9.3.2 and Annex A).

Information technology — Telecommunications and information exchange between systems — PHY/ MAC specifications for short-range wireless low-rate applications in the ISM band

1 Scope

This International Standard specifies the PHY characteristics and MAC procedures used for short-range, low-data-rate, wireless communications with very low latency and point-to-multipoint connection capability.

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 9798-3, *Information technology — Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques*

3 Terms and definitions

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

3.1
pico-net
small operational range for wireless transmissions within about 10 meters in radius from the user or his/her device

3.2
group
total of devices interoperable within a Pico-net, with their usage of the same group code separating them from other devices in different groups

3.3
master
device transmitting the reference synchronisation signal within a group

3.4
slave
device that is not a master

3.5
scan
process performed by slaves to search for the synchronising signal from the master

3.6
middleframe
basic unit of frame operation, consisting of one control frame and one or more payload frames

Note 1 to entry: Sixteen middleframes constitute a superframe.

3.7
superframe
bigger frame consisting of sixteen middleframes

Note 1 to entry: The superframe is the overall operational unit of pico-net MAC operations.

3.8
scan code
7-bit seed to generate one of the 127 gold codes which has a value between 1 and 127

3.9
open code
code used for broadcasting

3.10
closed code
code used exclusively for a specific communication group or purpose

3.11
group code
code to discriminate among communication groups

Note 1 to entry: Either an open or a closed code may be applied.

3.12
security code
code applied to message data to enhance security or privacy of the communication

Note 1 to entry: Either an open or a closed code may be applied.

4 Abbreviated terms

The following acronyms are used in this International Standard.

AARQF	authentication access request frame
AARSF	authentication access response frame
ACCF	authentication capabilities announcement frame
ADDM	address mode
BF	beacon frame
BPSK	binary phase shift keying
DME	device management entity
FBF	fast beacon frame
FSK	frequency shift keying
GAPF	general authentication process frame
GFSK	Gaussian frequency shift keying
GKAP	group key authentication protocol
GTK	group temporal key
ISM	industrial, scientific, and medical
LSAP	light-weight shared-key authentication protocol

MAC	medium access control
MACF	MCF acknowledge control frame
MCF	master control frame
MLME	MAC sublayer management entity
MLME-SAP	MAC sublayer management entity-service access point
MPDU	MAC protocol data unit
MSDU	MAC service data unit
PD-SAP	PHY data service access point
PDU	protocol data unit
PF	payload frame
PHY	physical layer
PLAS	pico-net light-weight architecture security
PLME	physical layer management entity
PLME-SAP	physical layer management entity - service access point
PPDU	physical layer protocol data unit
PSDU	physical layer service data unit
PTK	pairwise temporal key
QPSK	quadrature phase shift keying
RACF	RCF acknowledge control frame
RCF	request control frame
RF	radio frequency
RSSI	received signal strength indication
SAP	service access point
SDU	service data unit

5 Overview

There may be many applications in the ISM band. Such applications that require a short-range wireless communication channel can be listed as follows in the order of data rates; video, audio, voice, control, sensor, and so on. A different platform for a different application may be an ineffective way in light of cost, time-to-market, compatibility, etc. It would be beneficial to have a single platform which is capable of accommodating all these applications with the least overhead.

This International Standard is intended to provide a unified yet efficient and versatile platform for low-power, low-data-rate, short-range wireless communication applications. It is possible to accommodate diverse services of different nature in a single platform.

For mobile applications, low power consumption is one of the most important factors. To save power, data rate should be traded-off. This International Standard aims for the applications of 2 Mbps or less. To minimise implementation effort, it assumes the use of off-the-shelf RF components for the ISM band.

The International Standard makes use of frequency hopping, time-division multiple access, and time/frequency hybrid diversity. Frequency hopping is adopted to render immunity to the channel variations and to provide independent simultaneous communication channels. Time-division multiple access provides one with the control of interference of strong adjacent signals which otherwise should be avoided using an elaborate manipulation. The diversity technology is the means to maintain quality-of-service in the ISM band where channel fading is of serious concern.

Each device in the pico-net formed by this International Standard is either a master or a slave. In the pico-net, there exists only one single master which transmits a beacon signal to which all the other devices (slaves) are synchronised. The beacon signal contains the time synchronisation information and the frequency hopping pattern table. The frequency hopping pattern table contains the 16 best frequencies which are selected by sounding algorithms (see 7.4.5).

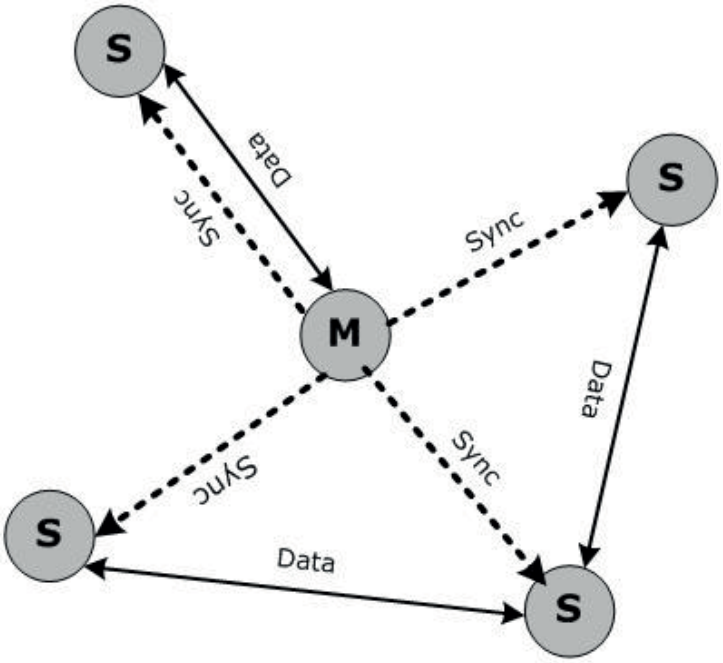

Figure 1 — A group communication example

At start-up, the master checks the frequency channels and selects the best 16 channels out of 80 to form a table of sixteen orthogonal frequency hopping patterns (see 8.4.2). Each frequency hopping pattern corresponds to a channel. The master assigns a communication channel (or channels) using the MCF (Master Control Frame) to be described below (see 7.4). Within each communication channel which is specified by a unique frequency-hopping pattern, the devices communicate with each other using time-division multiple access without any other intervention of the master.

The pico-net may have up to sixteen independent simultaneous communication channels. Within each communication channel, point-to-multipoint communication (broadcasting) is possible not to mention one-to-one communications. Moreover, each device may switch to another communication channel other than the current one if permitted by the master. Figure 1 shows an example of group communication in the pico-net. The master (M) transmits a beacon signal and is communicating with only one slave (S). The other slaves are communicating with another via other channels independently of the master.

Data are encased into the well-tailored standard units of a frame, a middleframe, and a superframe (see 8.3). Figure 2 shows the relationship between these units. These data formats are synchronised to the master beacon signal. To accommodate different applications in a single framework, this Interna tional Standard fixes the length of protocol frames to 4 ms which gives a permissible level of latency in most applications. A middleframe consists of frames. Sixteen middleframes constitute a superframe.

A frame is categorised into one of the seven kinds depending on its use: (1) a beacon frame (BF), (2) a fast beacon frame (FBF), (3) a request control frame (RCF), (4) a master control frame (MCF), (5) an RCF

acknowledge control frame (RACF), (6) an MCF acknowledge control frame (MACF), and (7) a payload frame (PF). All the frames except the payload frame (PF) are control frames. All the control frames have an identical format consisting of Lock Time, Preamble, Header, Message, and EoF Delimiter (see 8.2). The payload frame is used to carry user's data. Unlike the control frames whose preambles are fixed, the length of the preamble of the payload frame is variable to enhance the data throughput (see 9.2.2). Header is used to identify the kind of the frame. The message field is used to convey information and data necessary for communications (see 7.1 to7.7).

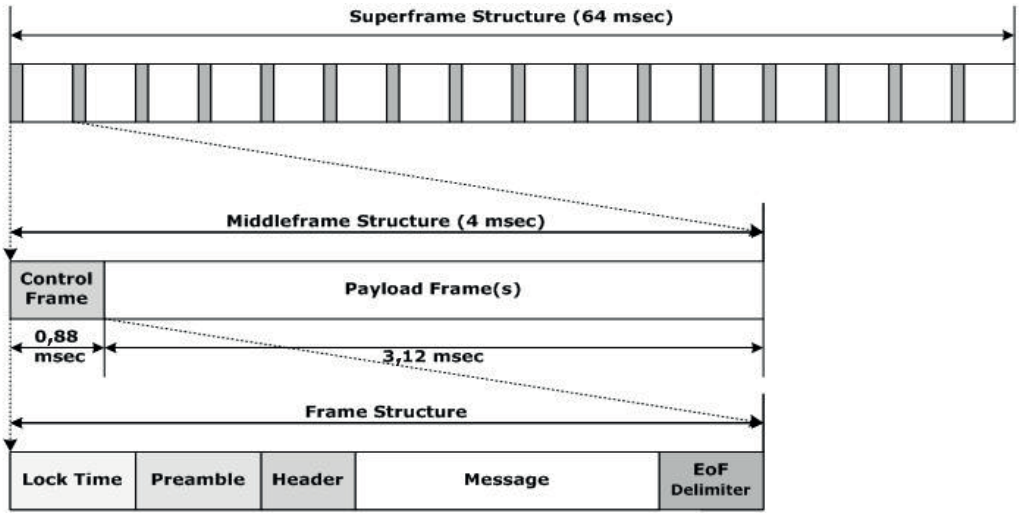

Figure 2 — Data formats: a frame, a middleframe, and a superframe

A middleframe consists of one control frame and one or more payload frames (PF's). The middleframe starts with a control frame whose length is fixed to 0,88 ms. The length of the middleframe is fixed to 4 ms. The length of the payload frames varies depending on applications. Carrier frequencies hop in harmony with the middleframes.

A superframe consists of sixteen middleframes and is of 64 ms. Superframes have two modes: a normal mode and a fast synchronisation mode (see 8.3.2). A fast synchronisation mode is used for robust synchronisation. In a fast synchronisation mode, a frame called 'fast beacon frame (FBF)' is used instead of 'beacon frame (BF)'. Two modes may be interchangeably adopted by the unit of a superframe.

For security reasons, the preamble in the frame uses Gold codes for group identification. The message field data are also encrypted with security codes (see 9.3.2).

The MAC/PHY services and primitives will be defined and described in Clause 6.

This International Standard uses the 2,4 GHz band and offers two classes of power transmission levels. Class one is up to 100 mW and class two is up to 10 mW. As a modulation scheme, the International Standard uses (G)FSK and QPSK (see 9.3).

In addition, this International Standard protects the communications from the security challenges due to the lack of protecting wires. Eavesdroppers may overhear data exchanges not intended for them, whereas imposters may send forged data not using its own identity, may replay previously transmitted data, and may transmit modified data captured from a previous transmission. In order to solve these issues, This International Standard also provides the security mechanism which the user can choose to implement (See 9.3.2 and Annex A).

6 Interlayer service specification

This Clause defines the interface between the MAC and PHY layers, and between the MAC layer and the upper layer.

6.1 Overview

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

The PHY provides data and management services through two SAPs (Service Access Points). The PHY data services are provided through the PD-SAP (PHY Data SAP), and PHY management services are provided through the PLME-SAP. The DME-PLME_SAP is equivalent to MLME-PLME-SAP except that it operates through DME rather than MLME.

The MAC provides data and management services through two SAPs (Service Access Points). The MAC data services are provided through the MAC-SAP, and MAC management services are provided through the MLME-SAP.

In order to provide correct MAC operation, each device must possess a DME (Device Management Entity). The DME is a layer-independent entity and act under the direction of a higher-level management application. Figure below depicts the relationships between the various management entities.

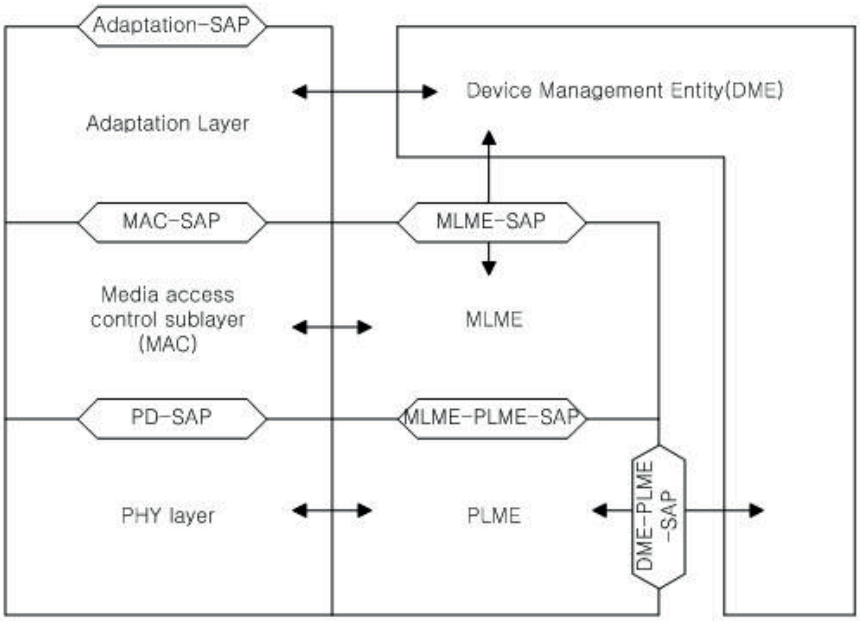

Figure 3 — The protocol model used in this International Standard

6.2 General format of management primitives

Each sublayer's specific management information is organised into the relevant Management Information Base (MIB). Corresponding to the MIB of the PAN, the LAN/MAN contains the 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 MIB is used to define the specifications of each sublayer.

MLME and PLME are assumed to have a MIB for each sublayer, and the management primitives of the MIB are exchanged by means of management SAPs. The manager can "GET" or "SET" the value of the MIB 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 PHY 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	6.2.1	6.2.2
XX-SET	6.2.3	6.2.4

Table 2 — MLME/PLME general management primitive parameters

Name	Type	Valid range	Description
MIBattribute	Octet string	Any MIB attribute	MIB attribute name
MIBvalue	Variable		MIB value
ResultCode	Enumeration	SUCCESS, INVALID_MIB_ATTRIBUTE, READ_ONLY_MIB_ATTRIBUTE, WRITE_ONLY_MIB_ATTRIBUTE	Result of MLME or PLME request

6.2.1 MLME-GET.request and PLME-GET.request

This primitive requests information about the relevant MAC MIB or PHY MIB. The semantics of these primitives are as follows.

XX-GET.request (

MIBattribute

)

The primitive parameters are defined in Table 2.

6.2.1.1 When generated

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

6.2.1.2 Effect of receipt

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

6.2.2 MLME-GET.confirm and PLME-GET.confirm

This primitive returns the result of an information request to the relevant MAC MIB or PHY MIB. The semantics of these primitives are as follows.

XX-GET.confirm (

Status,

MIBattribute,

MIBattributevalue

)

The primitive parameters are defined in Table 2.

6.2.2.1 When generated

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

6.2.2.2 Effect of receipt

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

6.2.3 MLME-SET.request and PLME-SET.request

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

```
XX-SET.request        (
                      MIBattribute,
                      MIBattributevalue
                      )
```

The primitive parameters are defined in Table 2.

6.2.3.1 When generated

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

6.2.3.2 Effect of receipt

The relevant management entity tries to alter the value of the MIB attribute in the database. If the MIB 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.

6.2.4 MLME-SET.confirm and MLME-SET.confirm

This primitive returns the result of the attempt to set the MAC MIB or PHY MIB attribute. The semantics of this primitive are as follows.

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

The primitive parameters are defined in Table 2.

6.2.4.1 When generated

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

6.2.4.2 Effect of receipt

If the Status is SUCCESS, this means that the MIB attribute was set to the requested value. Otherwise, the Status field shows the error description. If the specified MIB 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_MIB_ATTRIBUTE and READ_ONLY_MIB_ATTRIBUTE.

6.3 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. 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-GET	6.3.1			6.3.2
MLME-MASTER-START	6.3.3			6.3.4
MLME-RESET	6.3.5			6.3.6
MLME-SCAN	6.3.7			6.3.8
MLME-SET	6.3.9			6.3.10

Table 4 — MLME-SAP parameters

Name	Type	Valid range	Description
MIBAttribute	Enumeration		Desired physical layer MIB attribute
MIBStatus	Enumeration	SUCCESS, INVALID_ATTRIBUTE, INVALID_VALUE	Result of request for MIB attribute information
MIBAttributeValue	Various	Attribute specific	Desired physical layer MIB attribute value
ResetResult Code	Enumeration	SUCCESS, FAILED	Response to reset request
ScanNumber	integer	0 ~ 65535	Scan duration is ScanNumber * 64 ms
ScanStartFreq	integer	0 ~ 3	Scan start frequency. Scan frequency round as 0 -> 1 -> 2 -> 3-> 0 (0 : freq 0, 1 : freq 26, 2 : freq 52, 3 : freq 78)
ScanResult Code	integer	SUCCESS, FAILED, INVALID_VALUE	Return Scan result code
MasterStart ResultCode	integer	SUCCESS, FAILED, INVALID_VALUE	Return Master-Start result code

6.3.1 MLME-GET.request

This primitive requests Information about a MAC sublayer MIB attribute.

6.3.1.1 Definition of service primitives

The semantics of this primitive are:

MLME-GET.request (
MIBAttribute
)

Table 4 define the parameter of this primitive.

6.3.1.2 When generated

This occurs by DME to obtain information from the MAC sublayer MIB of MLME..

6.3.1.3 Effect of receipt

The receipt of the MLME-GET.request primitive by the MAC sublayer entity extracts the requested MIB attribute from the database and sends the results through a MLME-GET.confirm primitive.

6.3.2 MLME-GET.confirm

This primitive report result of the requested information from the MAC sublayer MIB.

6.3.2.1 Definition of service primitives

The semantics of this primitive are:

MLME-GET.confirm (
MIBstatus,
MIBAttribute,
MIBAttributeValue
)

Table 4 define the parameter of this primitive.

6.3.2.2 When generated

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

6.3.2.3 Effect of receipt

If the state parameter is SUCCESS the requested MAC sublayer MIB value is sent, otherwise an error is indicated.

6.3.3 MLME-MASTER-START.request

This primitive requests the process of creating a new network.

6.3.3.1 Definition of service primitives

The semantics of this primitive are:

MLME-MASTER-START.request (
MIBstatus,
MIBAttribute,
MIBAttributeValue
)

Table 4 define the parameter of this primitive.

6.3.3.2 When generated

This primitive is generated by the next higher layer and issued to its MLME to create a new network.

6.3.3.3 Effect of receipt

This primitive initiate the piconet described in 7.1. The MLME subsequently issues an MLME-MASTER-START.confirm that reflects the results of the creation procedure.

6.3.4 MLME-MASTER-START.confirm

This primitive reports the results of a piconet creation.

6.3.4.1 Definition of service primitives

The semantics of this primitive are:

MLME-MASTER-START.confirm (
MasterStartResultCode
)

Table 4 define the parameter of this primitive.

6.3.4.2 When generated

This primitive is generated by the MLME as a result of an MLME-MASTER-START.request.

6.3.4.3 Effect of receipt

MLME reports the result of the creation process of network. A ResultCode of SUCCESS indicates that the station is now the master.

6.3.5 MLME-RESET.request

This primitive requests a reset of the MAC sublayer.

6.3.5.1 Definition of service primitives

The semantics of this primitive are:

MLME-RESET.request (
)

This primitive has no parameters.

6.3.5.2 When generated

This is generated whenever a MAC sublayer reset is requested.

6.3.5.3 Effect of receipt

The MAC sublayer resets all transceiver state machines to their initial states.

6.3.6 MLME-RESET.confirm

This primitive reports result of reset the MAC sublayer.

6.3.6.1 Definition of service primitives

The semantics of this primitive are:

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

Table 4 define the parameter of this primitive.

6.3.6.2 When generated

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

6.3.6.3 Effect of receipt

DME or Adaptation layer is notified of the result of the reset.

6.3.7 MLME-SCAN.request

This primitive define how a device can determine the presence or absence of PANs in a communications channel.

All devices shall provide an interface for these scan primitives.

6.3.7.1 Definition of service primitives

The semantics of this primitive are:

```
MLME-SCAN.request   (
                    ScanNumber,
                    ScanStartFreq
                    )
```

Table 4 define the parameter of this primitive.

6.3.7.2 When generated

This primitive is generated by the next higher layer and issued to its MLME to initiate a channel scan to search for master device activity within the POS of the device.

6.3.7.3 Effect of receipt

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.

6.3.8 MLME-SCAN.confirm

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

6.3.8.1 Definition of service primitives

The semantics of this primitive are:

MLME-SCAN.confirm (

ScanResultCode

)

Table 4 define the parameter of this primitive.

6.3.8.2 When generated

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

6.3.8.3 Effect of receipt

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

6.3.9 MLME-SET.request

This primitive request to set the MAC sublayer MIB attribute to the specified value.

6.3.9.1 Definition of service primitives

The semantics of this primitive are:

MLME-SET.request (

MIBAttribute,

MIBAttributeValue

)

Table 4 define the parameter of this primitive.

6.3.9.2 When generated

This occurs by DME to set the MAC sublayer MIB attribute to the specified value and sends it to MLME.

6.3.9.3 Effect of receipt

The receipt of the MLME-SET.request primitive by the MAC sublayer entity attempts to store the specified MAC sublayer MIB attribute in the database and reports the result through a MLME-SET. confirm primitive.

6.3.10 MLME-SET.confirm

This primitive reports result of the attempt to set the MAC sublayer MIB attribute to the specified value.

6.3.10.1 Definition of service primitives

The semantics of this primitive are:

MLME-SET.confirm (

MIBstatus,

MIBAttribute

)

Table 4 define the parameter of this primitive.

6.3.10.2 When generated

MLME sends this to Adaptation layer as the response to the MLME-SET.request primitive.

6.3.10.3 Effect of receipt

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

6.4 MAC-SAP

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. Table 5 lists the primitives supported by the MAC-SAP. These primitives are discussed in the subclauses referenced in the table.

Table 5 — MAC-SAP primitive summary

Name	Request	Confirm	Indication	Response
MAC-DATA	6.4.1	6.4.2	6.4.3	

Table 6 — MAC-SAP parameters

Name	Type	Valid range	Description
SourceID	integer		Target DEVID of MLME request
DestinationID	integer		DEVID initiating the MLME request
Length	Unsigned Short Integer	0-65535	MSDU length

Table 6 *(continued)*

Name	Type	Valid range	Description
data	variable length Octet	-	Data portion of MSDU
ResultCode	Enumeration	SUCCESS, INVALID_MIB_ATTRIBUTE	Result of MAC request

6.4.1 MAC-DATA.request

This primitive initiates the data transfer from one MAC entity to another MAC entity or entities.

6.4.1.1 Definition of service primitives

The semantics of this primitive are:

MAC-DATA.request (
SourceID,
DestinationID,
Length,
Data
)

Table 6 define the parameter of this primitive.

6.4.1.2 When generated

This occurs when the adaptation layer entity requests transmission of a MSDU(i.e. APDU) to the MAC sublayer entity.

6.4.1.3 Effect of receipt

When this primitive is received, the MAC formats the MSDU according to the input parameters and sends it to the PHY-SAP; then the MSDU passes through the wireless media and is sent to the peer MAC entity. When the MAC sublayer entity has completed the sending, it will issue the MAC-DATA.confirm primitive with a status of SUCCESS.

6.4.2 MAC-DATA.confirm

This primitive confirm that the Adaptation layer entity has sent a MSDU (APDU) to another Adaptation layer entity.

6.4.2.1 Definition of service primitives

The semantics of this primitive are:

MAC-DATA.confirm (
SourceID,
DestinationID,
ResultCode
)

Table 6 define the parameter of this primitive.

6.4.2.2 When generated

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

6.4.2.3 Effect of receipt

On receipt of the MAC-DATA.confirm primitive, the Adaption layer entity is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter is set to SUCCESS.

6.4.3 MAC-DATA.indication

This primitive indicates to the adaptation layer that a MSDU has been received.

6.4.3.1 Definition of service primitives

The semantics of this primitive are:

MAC-DATA.indication (
SourceID,
DestinationID,
Length,
Data
)

Table 6 define the parameter of this primitive.

6.4.3.2 When generated

This occurs when MSDU received by the MAC has been successfully processed.

6.4.3.3 Effect of receipt

The receipt of the MAC-DATA.indication primitive, the adaptation layer is notified of the arrival of an APDU(i.e. MSDU) across the MAC data service.

6.5 PLME-SAP

The PHY layer management object service access points (MLME-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 MLME-PLME-SAP interface. Table 7 defines the primitives supported by PLME-SAP. Table 8 lays out the individual parameters.

Table 7 — PLME-SAP primitive summary

Name	Request	Confirm	Indication	Response
PLME-GET	6.5.1	6.5.2		
PLME-SET	6.5.3	6.5.4		
PLME-RESET	6.5.5	6.5.6		

Table 8 — PLME-SAP parameters

Name	Type	Valid range	Description
MIBAttribute	Enumeration		Desired physical layer MIB attribute
MIBStatus	Enumeration	SUCCESS, INVALID_ATTRIBUTE, INVALID_VALUE	Result of request for MIB attribute information
MIBAttributeValue	Various	Attribute specific	Desired physical layer MIB attribute value
ResetResult Code	Enumeration	SUCCESS, FAILED	Response to reset request

6.5.1 PLME-GET.request

This primitive requests Information about a PHY layer MIB attribute.

6.5.1.1 Definition of service primitives

The semantics of this primitive are:

PLME-GET.request (
MIBAttribute
)

Table 8 define the parameter of this primitive.

6.5.1.2 When generated

This occurs by DME to obtain information from the PHY layer MIB of PLME..

6.5.1.3 Effect of receipt

The receipt of the PLME-GET.request primitive by the PHY entity extracts the requested MIB attribute from the database and sends the results through a PLME-GET.confirm primitive.

6.5.2 PLME-GET.confirm

This primitive report result of the requested information from the PHY layer MIB.

6.5.2.1 Definition of service primitives

The semantics of this primitive are:

PLME-GET.confirm (
MIBstatus,
MIBAttribute,
MIBAttributeValue
)

Table 8 define the parameter of this primitive.

6.5.2.2 When generated

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

6.5.2.3 Effect of receipt

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

6.5.3 PLME-SET.request

This primitive request to set the PHY layer MIB attribute to the specified value.

6.5.3.1 Definition of service primitives

The semantics of this primitive are:

PLME-SET.request (
MIBAttribute,
MIBAttributeValue
)

Table 8 define the parameter of this primitive.

6.5.3.2 When generated

This occurs by DME to set the PHY layer MIB attribute to the specified value and sends it to PLME.

6.5.3.3 Effect of receipt

The receipt of the PLME-SET.request primitive by the PHY entity attempts to store the specified PHY layer MIB attribute in the database and reports the result through a PLME-SET.confirm primitive.

6.5.4 PLME-SET.confirm

This primitive reports result of the attempt to set the PHY layer MIB attribute to the specified value.

6.5.4.1 Definition of service primitives

The semantics of this primitive are:

PLME-SET.confirm (

MIBstatus,

MIBAttribute

)

Table 8 define the parameter of this primitive.

6.5.4.2 When generated

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

6.5.4.3 Effect of receipt

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

6.5.5 PLME-RESET.request

This primitive requests a reset of the PHY layer.

6.5.5.1 Definition of service primitives

The semantics of this primitive are:

PLME-RESET.request (

)

This primitive has no parameters.

6.5.5.2 When generated

This is generated whenever a PHY layer reset is requested.

6.5.5.3 Effect of receipt

The PHY layer resets all transceiver state machines to their initial states.

6.5.6 PLME-RESET.confirm

This primitive reports result of reset the PHY layer.

6.5.6.1 Definition of service primitives

The semantics of this primitive are:

PLME-RESET.confirm (

ResetResultCode

)

Table 8 define the parameter of this primitive.

6.5.6.2 When generated

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

6.5.6.3 Effect of receipt

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

6.6 PD-SAP

The PD-SAP supports the transmission of MPDUs between peer MAC sublayer entities. Table 9 lists the primitives supported by the PD-SAP. These primitives are discussed in the subclauses referenced in the table.

Table 9 — PD-SAP primitives

Name	Request	Confirm	Indication	Response
PD-DATA	6.6.1	6.6.2	6.6.3	

Table 10 — PD-SAP parameters

Name	Type	Valid range	Description
psduLength	Unsigned Short Integer	≤ aMaxPHYPacketSize	The number of octets contained in the PSDU received by the PHY entity.
psdu	Set of octets	-	The set of octets forming the PSDU received by the PHY entity.
status	Enumeration	SUCCESS (EoF)	The result of the request to transmit a packet.
rssi	Integer		RSSI Value

6.6.1 PD-DATA.request

This primitive requests the transmission of a MPDU from the MAC sublayer to the local PHY entity.

6.6.1.1 Definition of service primitives

The semantics of this primitive are:

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

Table 10 define the parameter of this primitive.

6.6.1.2 When generated

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

6.6.1.3 Effect of receipt

The receipt of the PD-DATA.request primitive by the PHY entity will cause the transmission of the supplied PSDU. When the PHY entity has completed the transmission, it will issue the PD-DATA.confirm primitive with a status of SUCCESS.

6.6.2 PD-DATA.confirm

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

6.6.2.1 Definition of service primitives

The semantics of this primitive are:

PD-DATA.confirm (
status
)

Table 10 define the parameter of this primitive.

6.6.2.2 When generated

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

6.6.2.3 Effect of receipt

On receipt of the PD-DATA.confirm primitive, the MAC sublayer entity is notified of the result of its request to transmit. If the transmission attempt was successful, the status parameter is set to SUCCESS.

6.6.3 PD-DATA.indication

This primitive indicates the received PSDU from PHY layer to the MAC sublayer entity.

6.6.3.1 Definition of service primitives

The semantics of this primitive are:

PD-DATA.indication (
psduLength,
psdu
rssi
)

Table 10 define the parameter of this primitive.

6.6.3.2 When generated

This occurs when the PHY layer sends the received PSDU to the MAC sublayer entity.

6.6.3.3 Effect of receipt

The receipt of the PD-DATA.indication primitive, the MAC sublayer is notified of the arrival of an MPDU across the PHY data service.

7 MAC PDU format

This clause specifies the formats of the MAC PDU (MPDU).

MAC frame consists of one control frame and one or more payload frames. Control frame is categorised into one of the six kinds depending on its use: (1) beacon frame (BF), (2) fast beacon frame (FBP), (3) request control frame (RCF), (4) master control frame (MCF), (5) RCF acknowledge control frame (RACF), and (6) MCF acknowledge control frame (MACF).

In addition, if the link layer authentication and key generation security mechanisms should be implemented in a pico-net, see Annex A Pico-net Light-weight Architecture Security (PLAS) for the related control frames information.

Figure 4 shows the relationship between MPDU and PPDU in a frame structure.

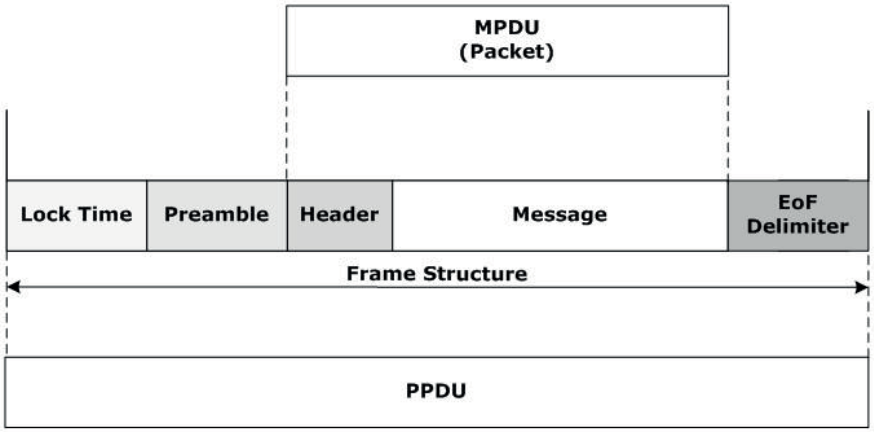

Figure 4 — MPDU in frame structure

The format of each control frame is described in 7.1 through 7.6. The format of payload frame is described in 7.7.

7.1 MPDU of beacon frame (BF)

BFs are used to maintain synchronisation. There are two BFs during a single normal superframe. The master sends information used for synchronisation, and the slaves acquire the synchronisation from BFs.

The MPDU format of BF is shown in Figure 5.

7.1.1 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. This field comprises of GCOF which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

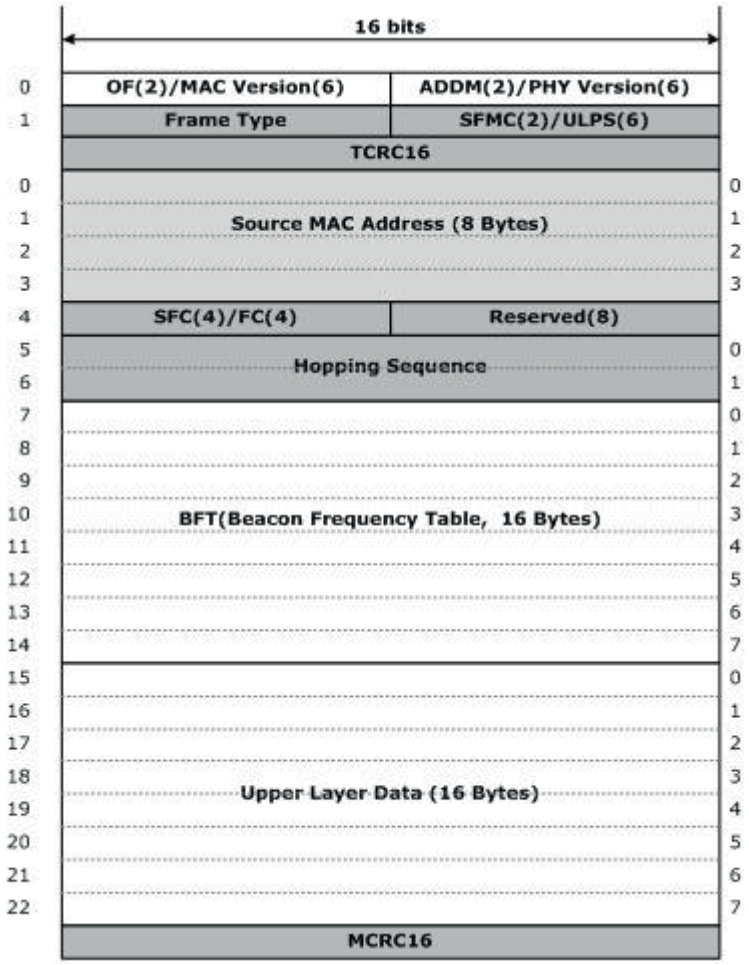

Figure 5 — MPDU format of Beacon Frame

7.1.2 MAC version (6 bits)

This field provides information about the MAC Version, and has a value between 0x00 to 0x3F.

7.1.3 Address mode (ADDM, 2 bits)

This field provides information about whether a MAC address exists. The ADDM comprises of two bits: one bit used for displaying whether a source MAC address exists and the other bit used for displaying the destination MAC address.

7.1.4 PHY version (6 bits)

This field provides information about the PHY version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

7.1.5 Frame type (8 bits)

This field is used to indicate the type of the current frame.

7.1.6 Superframe mode control (SFMC, 2 bits)

This field is used for selecting a mode of the superframe. It comprises of the current superframe mode (CSFM) and the next superframe mode (NSFM). That CSFM is 1 means the current superframe is a normal superframe. When it is 0, the current superframe is a fast synchronisation superframe. That

NSFM is 1 means the next superframe is a normal superframe. When it is 0, the current superframe is a fast synchronisation superframe.

7.1.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

7.1.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending BFs.

7.1.9 Superframe counter (SFC, 4 bits)

This field provides information on the superframe counter. The value circulates from 0 to 15.

7.1.10 Middleframe counter (FC, 4 bits)

This field provides information on the middleframe counter. The value circulates from 0 to 15.

7.1.11 Hopping sequence (32 bits)

This field provides the hopping sequence of the frequencies that were chosen for communication.

7.1.12 Beacon frequency table (BFT, 16 bytes)

This field provides information on the sequence of the frequencies used for the partial band hopping. BFFT contains a table of 16 frequencies being used for communication.

7.1.13 Upper layer data (16 bytes)

This field is used for transmission of the data from the upper layer.

7.1.14 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for header (tag) and MCRC16 for message. Refer to 9.3.2.4.

7.2 MPDU of fast beacon frame (FBF)

This field is used for initial fast synchronisation. There are 16 FBFs in a single fast synchronisation superframe. The master sends information for synchronisation in the FBFs, and the slaves are synchronised using these FBFs.

The MPDU format of FBF is shown in Figure 6.

7.2.1 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. This field comprises of GCOF which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

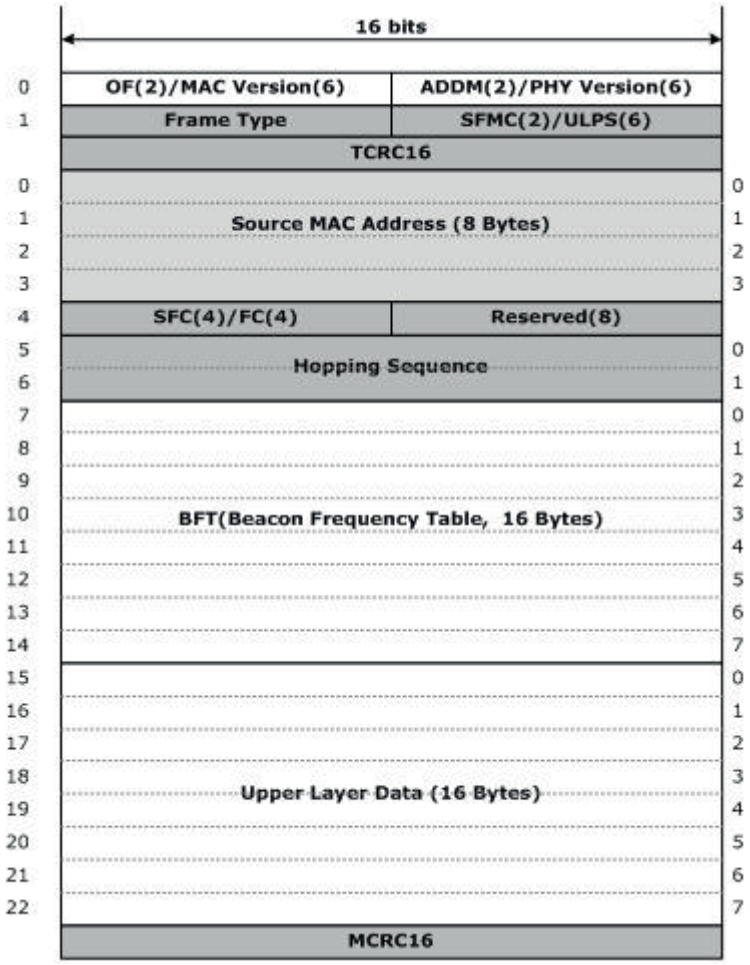

Figure 6 — MPDU format of Fast Beacon Frame (FBF)

7.2.2 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

7.2.3 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists. One means an address exists.

7.2.4 PHY version (6 bits)

This field provides information on the PHY version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

7.2.5 Frame type (8 bits)

This field is used to indicate the type of the current frame.

7.2.6 Superframe mode control (SFMC, 2 bits)

This field is used for selecting the mode of the superframe. It comprises of the current superframe mode (CSFM) and the next superframe mode (NSFM). That CSFM is 1 means the current superframe is a normal superframe. When it is 0, the current superframe is a fast synchronisation superframe. That

NSFM is 1 means the next superframe is a normal superframe. When it is 0, the next superframe is a fast synchronisation superframe.

7.2.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

7.2.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending BFs.

7.2.9 Superframe counter (SFC, 4 bits)

This field provides information on the superframe counter. The value circulates from 0 to 15.

7.2.10 Middleframe counter (SC, 4 bits)

This field provides information on the middleframe counter. The value circulates from 0 to 15.

7.2.11 Hopping sequence (32 bits)

This field provides the hopping sequence of the frequencies that were chosen for communication.

7.2.12 Beacon frequency table (BFT, 16 bytes)

This field provides information on the sequence of the frequencies used for the partial band hopping. BFFT contains a table of 16 frequencies being used for communication.

7.2.13 Upper layer data (16 Bytes)

This field is used for transmission of the data from the upper layer.

7.2.14 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for header (tag) and MCRC16 for message. Refer to 9.3.2.4.

7.3 MPDU of request control frame (RCF)

RCFs are used to exchange control information between devices. In response to RCF, Acknowledge Control Frame of RCF (RACF) is issued. The RCF is used for multiple simultaneous communications among devices in a group. When two or more devices initiate communications at the same time, there may be collision. To ensure reliability of the control information, the random back off must be devised to resolve collisions. All the devices must be either in transmission mode or in reception mode.

The MPDU format of RCF is shown in Figure 7.

7.3.1 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

7.3.2 MAC version (6 bits)

This field provides information on the MAC Version, and has a value between 0x00 to 0x3F.

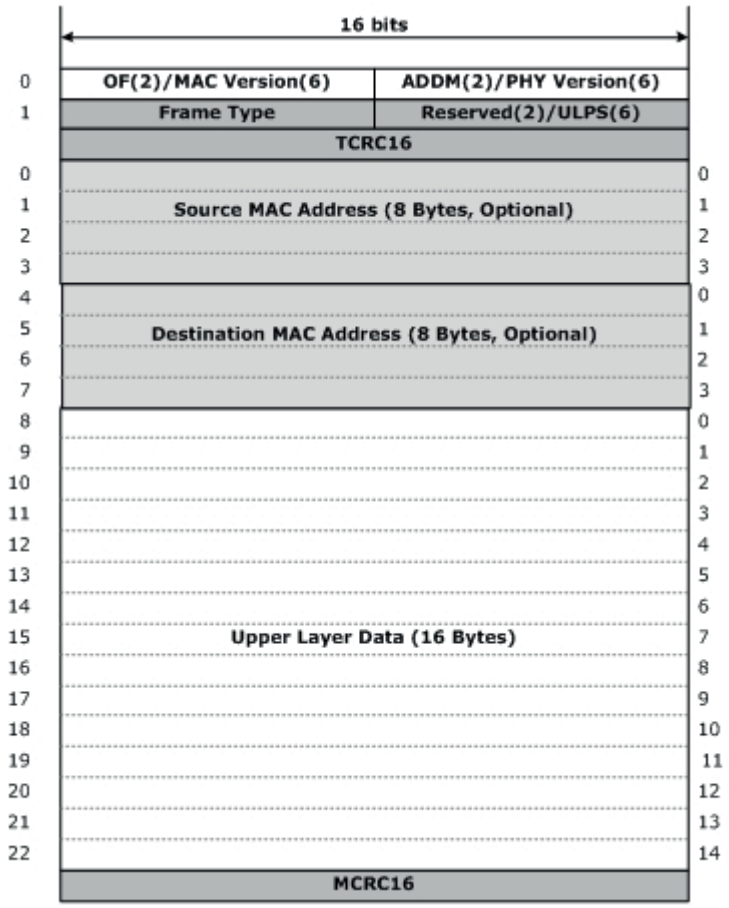

Figure 7 — MPDU format of RCF

7.3.3 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists.

7.3.4 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

7.3.5 Frame type (8 bits)

This field is used to indicate the type of the current frame.

7.3.6 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

7.3.7 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending RCFs.

7.3.8 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive RCFs.

7.3.9 Upper layer data

This field is used for transmission of the data from the upper layer.

7.3.10 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for header (tag) and MCRC16 for message. Refer to 9.3.2.4.

7.4 MPDU of master control frame (MCF)

MCFs are used by the master in a group to send control information to slaves. The master can send control information up to 8 devices by using a single MCF. Slave devices may respond to MCF using the frames called MACF (MCF acknowledgement control frame). MACFs are issued by the responding slaves as controlled by MCFs in a manner to avoid collision. Each responding slave takes up a different MACF slot.

The MPDU format of MCF is shown in Figure 8.

7.4.1 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

7.4.2 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

7.4.3 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists.

7.4.4 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

7.4.5 Frame type (8 bits)

This field is used to indicate the type of the current frame.

7.4.6 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

7.4.7 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending MCFs.

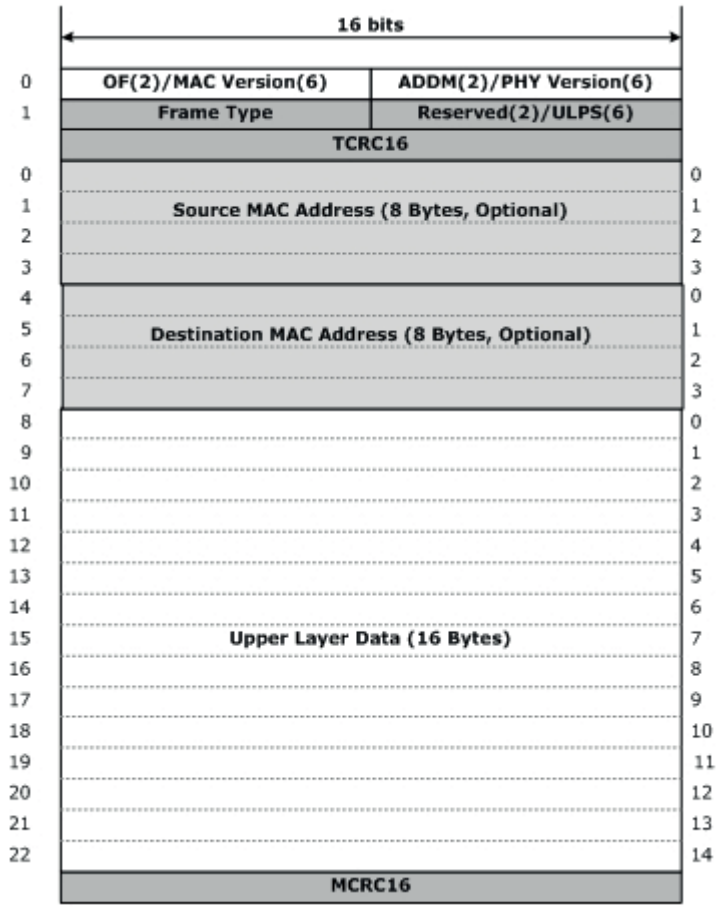

Figure 8 — MPDU format of MCF

7.4.8 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive MCFs.

7.4.9 Upper layer data

This field is used for transmission of the data from the upper layer.

7.4.10 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for header(tag) and MCRC16 for message. Refer to 9.3.2.4.

7.5 MPDU of RCF acknowledge control frame (RACF)

A device receiving RCFs sends RACFs in response. The device that sent RCFs receives RACFs as an acknowledgement. RACFs are control frames that can be sent to both the master and the slave devices. After sending RCF, when a RACF response is not received as is necessary, RCF may be re-sent after backing off in random superframe units.

The MPDU format of RACF is identical to that of RCF except for the Frame Type field.

7.6 MPDU of MCF acknowledge control frame (MACF)

MACFs are issued by the slaves that need to acknowledge the reception of MCFs. The master checks MACFs to verify responses from the slaves with respect to MCFs.

The MPDU format of MACF is identical to that of MCF except for the Frame Type field.

7.7 MPDU of payload frame (PF)

Payload frames are used to transmit data from upper layers. The number of payload frames in a middleframe may vary depending on the requirement of upper layers.

The MPDU format of PF is shown in Figure 9.

7.7.1 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

7.7.2 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

7.7.3 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists.

7.7.4 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

7.7.5 Frame type (8 bits)

This field is used to indicate the type of the current frame.

7.7.6 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

7.7.7 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending PFs.

7.7.8 Destination MAC address (64 bits)

This field provides the MAC address of device which is to receive PFs.

7.7.9 Upper layer data

This field is used for transmission of the data from the upper layer.

7.7.10 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for header (tag) and MCRC16 for message. Refer to 9.3.2.4.

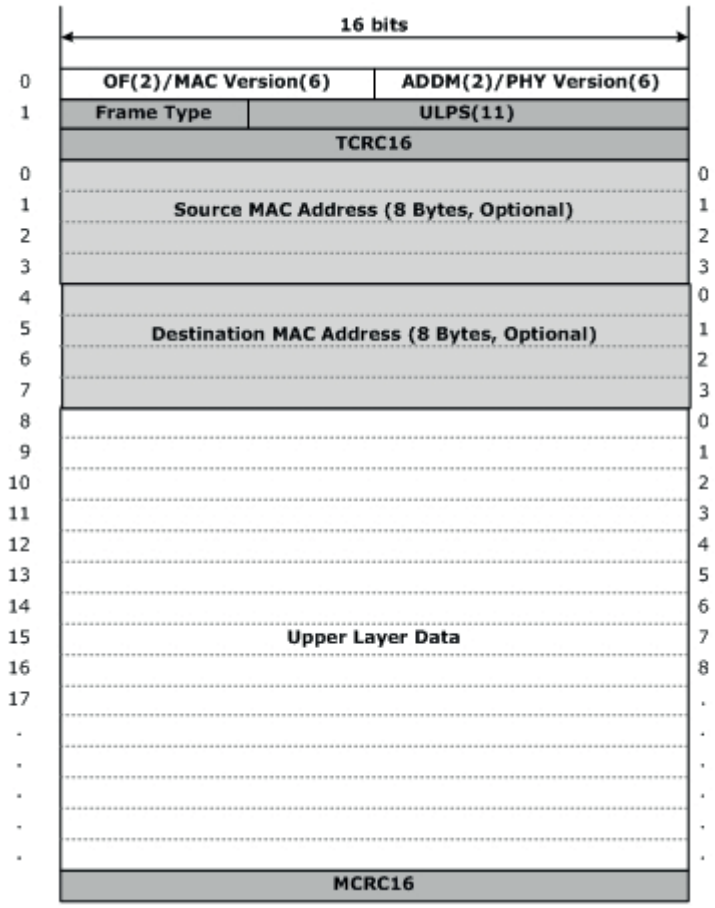

Figure 9 — MPDU format of PF

8 MAC functional description

This clause describes the functions of the MAC layer.

8.1 General description

In communications in a pico-net, it is required to keep the devices within the pico-net synchronised. For this purpose, in a pico-net, one device must operate as a master, which sends synchronising signals periodically, and the other devices operate as slaves in accordance with the synchronizing signals from the master.

Figure 10 shows a pico-net with only two terminals. One terminal must serve as a master while the other as a slave. The master regularly transmits synchronising signals to which the slave is synchronised. Communications between the two terminals are based on this synchronisation.

Figure 11 shows a pico-net with more than two devices. One device must serve as a master while the others as slaves. All the devices must be synchronised to the master's synchronising signals. Communications between any two or more devices are practiced based on this synchronisation with no further master's intervention. The master transmits synchronising signals only to maintain the synchronisation within the network. Communications between slaves are practiced directly without the master's relaying the communications.

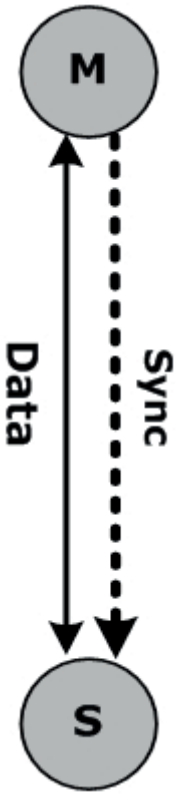

Figure 10 — A pico-net with only two devices.

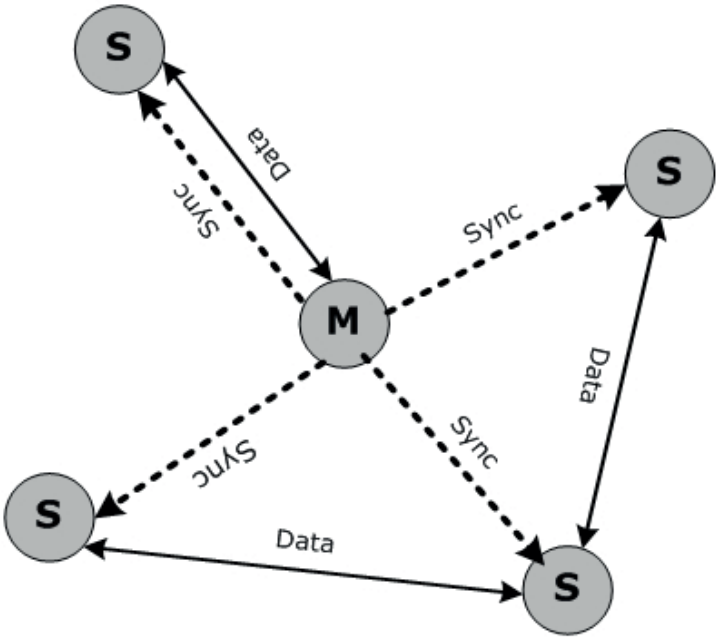

Figure 11 — A pico-net with more than two terminals

8.2 System state diagram

The master/slave operations can be expressed as a finite state machine with 10-phase states. Any device is in one of the ten states. The description of the ten states is summarised in Table 11.

Figure 12 shows the state-transition diagram. The solid line and the dotted line represent the paths of the state changes of the master and the slaves, respectively.

The network operation begins with initialisation. The initialisation starts when more than two devices within the communication range are first turned on.

When the initialisation starts, the master performs the initialisation process, regardless of whatever state it has been in. After initialisation is completed, the master stays in the 'initialised' state. The master in the 'initialised' state can shift into the 'passive sounding' or 'normal/fast master sync' state.

Passive sounding is a state where frequencies are selected by measuring the signal level at each frequency. The master may choose to put itself into the 'passive sounding' state if needed.

After shifting into the 'normal/fast master sync' state, the master sends synchronising signals. In these states, the master can communicate with the slaves synchronised with the synchronising signals. In the 'master sync' states, the master can shift into the 'static sounding' state where frequencies can be selected via special communication between the master and the slaves. In the 'static sounding' state, normal communications between the master and the slaves cannot be practiced. The 'static sounding' state can be selected by the user whenever needed.

After initialisation with the master, the slaves are forced into the 'initialised' state. The slaves are forced into initialisation process once initialisation process is initiated, regardless of whatever state they have been in.

Table 11 — Description of states

State	Description
Not initialised	A state before initialisation.
Initialised	A state after initialisation is completed; the master and slaves are determined.
Normal master sync.	A state where the device is serving as the master of the normal network cycle.
Fast master sync.	A state where the device is serving as the master of the fast synchronisation network cycle.
Scanning	A state where the device is under synchronisation with the master.
Normal slave sync.	A state where the device is serving as a slave in the normal superframe after synchronisation with the master.
Fast slave sync.	A state where the device is serving as a slave in the fast synchronisation superframe after synchronisation with the master.
Passive sounding	A state where the master conducts passive sounding.
Master static sounding	A state where the master conducts static sounding.
Slave static sounding	A state where the slave conducts static sounding

The slaves synchronise with the master in the 'scanning' state after initialisation. With the synchronisation information from the master, the slaves synchronise with the master and shift into the 'normal/fast slave sync' state. When the synchronisation information is not obtained, the slaves are again put into the 'initialized' state. After the slaves are synchronised with the master and put into the 'normal/fast slave sync' state, the slaves can communicate with the master. When a static sounding is requested from the master, the slaves shift into the 'static sounding' state to conduct static sounding.

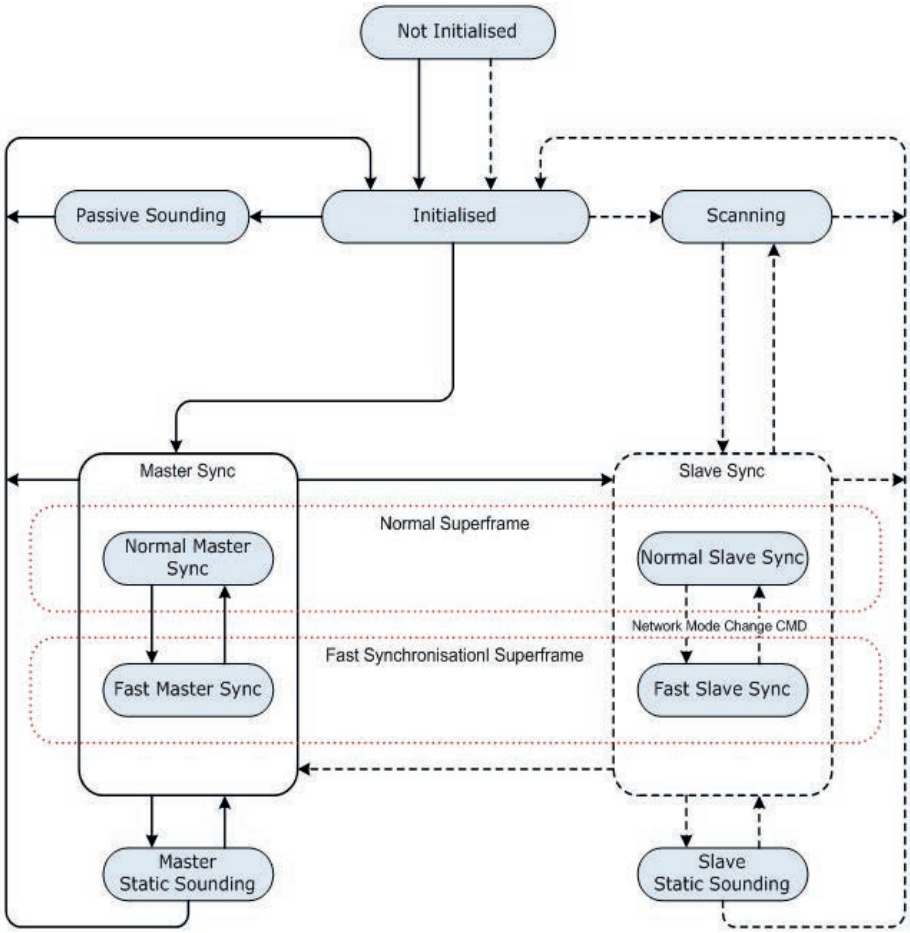

Figure 12 — State transition diagram

8.3 Protocol structure

Figure 13 shows the hierarchical structure of the protocol. The protocol is structured based on the superframes of 64 msec each, which again consists of 16 middleframes of 4 msec each. A middleframe consists of one control frame and one or more payload frames.

The middleframe is the most basic unit structure. The state of each middleframe can be set independently of the others.

The number of payload frames is determined based on the communication type (data, voice, control, etc). However, it is required to maintain the length of superframes equal for synchronised communication. The synchronisation between devices is possible only if the lengths of all superframes are equal. The synchronised communications minimize the frequency interference between devices of different services by controlling the transmission and reception based on time slots.

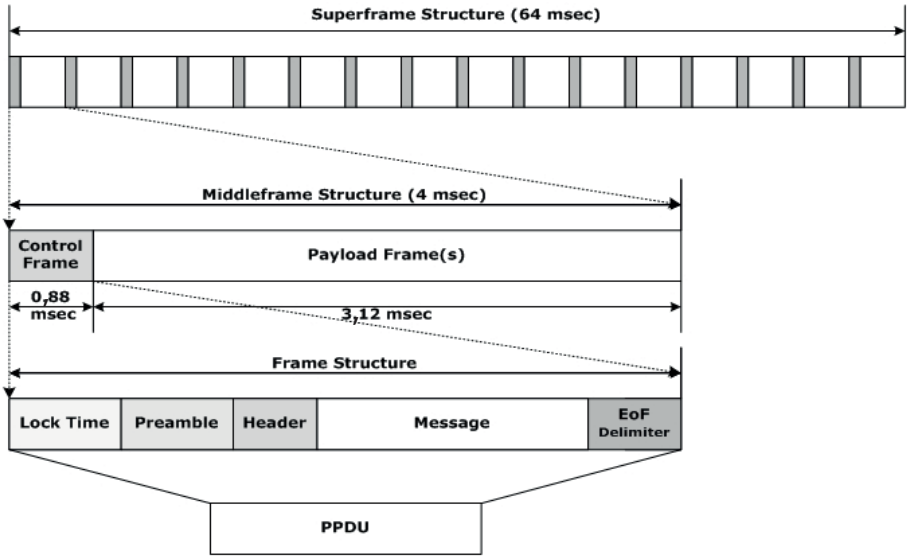

Figure 13 — The protocol structure

8.3.1 Middleframe structure

Figure 14 shows the structure of the middleframe. A middleframe consists of one control frame and one or more payload frames. The length of the control frame is fixed at 880 μsec. The number and length of payload frames within a middleframe can be set as needed. The upper layers may have various data rates depending on the number and length of the payload frames. When high data rate is needed, the number of payload frames can be reduced to lower overhead. However, transmission delay due to buffering increases in proportion to the length of middleframe. The total length of all payload frames should be 3.12 msec.

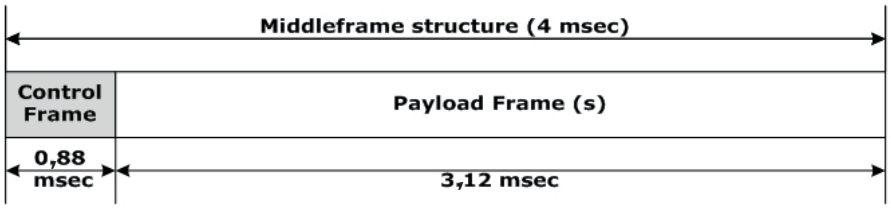

Figure 14 — Middleframe structure

8.3.2 Superframe structure

The length of a single superframe is 64 msec; the superframe consists of 16 frames. Within a superframe are 16 control frames of 0.88 msec each. Overall, 14.08 msec is assigned to control frames and the rest 49.92 msec to payload frame(s). Each control frame has its own unique function as described in Clause 7. There are two types of superframe: a normal superframe and a fast synchronisation superframe.

8.3.2.1 Normal superframe

Figure 15 shows the structure of a normal superframe. There are 16 control frames in a normal superframe: two are beacon frames (BF), one is a request control frame (RCF) used when slaves send request information to the master, one is a master control frame (MCF) used when the master sends control information to the slaves, one is allocated for response to RCF, eight for response to MCF, and the remaining three are reserved.

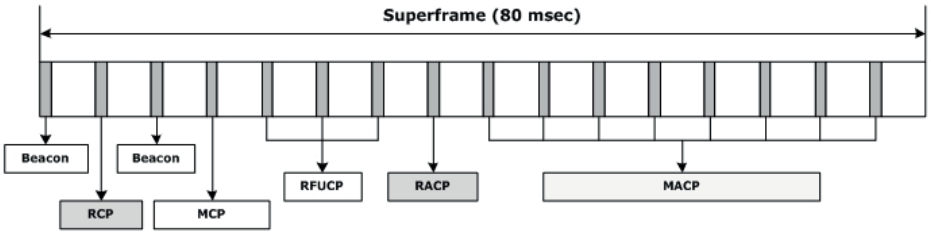

Figure 15 — Structure of a normal superframe

8.3.2.2 Fast synchronisation superframe

Figure 16 shows the structure of a fast synchronisation superframe. For a fast synchronisation, a fast synchronisation superframe should be adopted. All control frames in the fast synchronisation superframe are used for synchronisation. These control frames used for synchronisation are called fast beacon frames (FBFs).

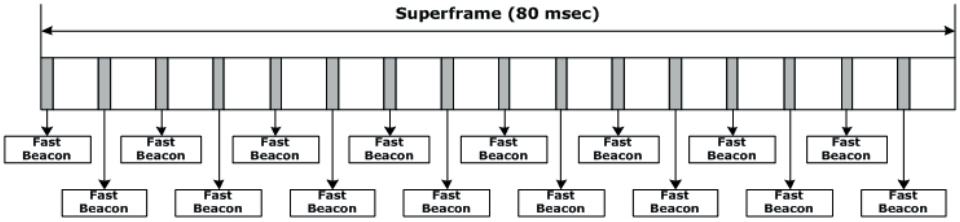

Figure 16 — Structure of a fast synchronisation superframe

8.3.2.3 Alternation between superframes

For a normal operation, the normal superframe repeats itself. For a fast synchronisation, alternation between a normal superframe and a fast synchronisation superframe is practiced. Figure 17 illustrates the operation of the fast synchronisation.

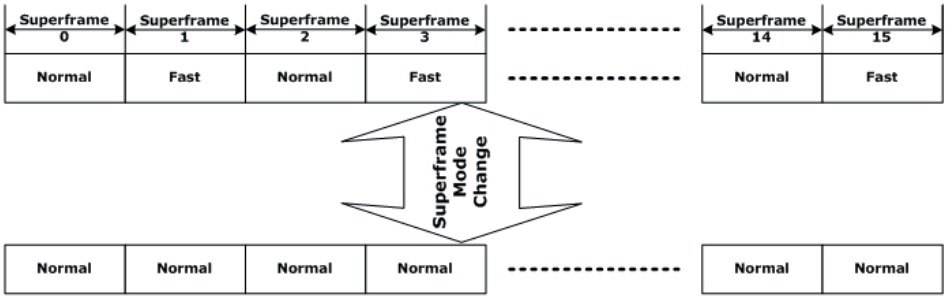

Figure 17 — Superframe mode alternation

8.3.2.4 Middleframe counter and superframe counter

Numbers are assigned to the middleframes and superframes to maintain the structure. The counters are in the message field of each beacon frame or fast beacon frame. A number '0' is assigned to the first middleframe in a superframe. The number increases by one in the frames that follow. A number '15' is assigned to the last middleframe in the superframe. Likewise, a number is assigned to each superframe, which repeats itself with a period of 16. The number starts with 0 and ends with 15. Figures 18 and 19 show the middleframe and superframe counters, respectively.

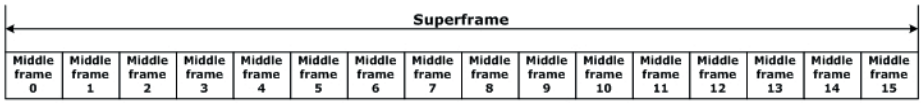

Figure 18 — Middleframe counter

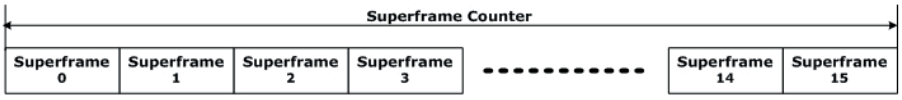

Figure 19 — Superframe counter

8.4 Frequency operation

8.4.1 Frequency hopping control

The protocol carries out communication according to the frequency hopping table. The hopping sequence of the frequencies is in a random order out of the chosen best sixteen frequencies. The random sequence is generated by a hopping sequence generator which works as described below.

Figure 20 shows a hopping sequence generator. The generator uses a 32 stage shift register to generate a maximal length pseudo random sequence. The feedback tabs are [31, 21, 1, 0]. Output values are generated for every single middleframe (in every 4 msec). The sequence repeats itself every $(2^{32} - 1) \times 4$ msec.

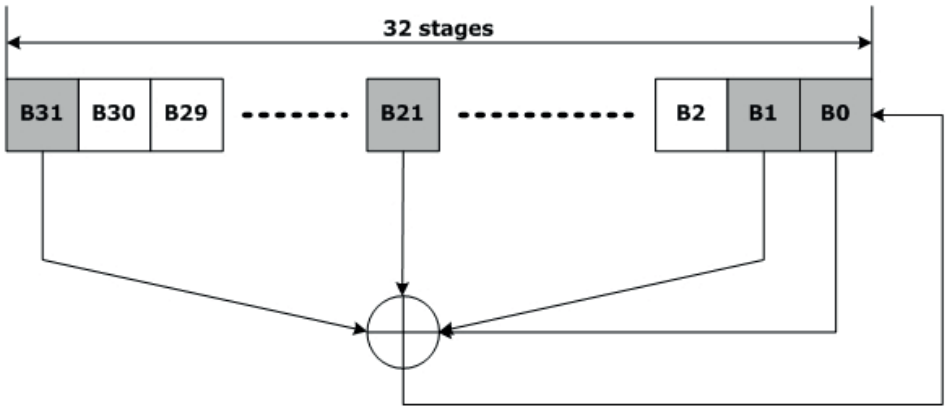

Figure 20 — The hopping sequence generator

8.4.2 Frame frequency mapping

This subclause describes how the outcomes of the hopping sequence generator are mapped to the frequencies.

The 32-bit value of every state of the hopping sequence generator is divided by 16, and the residue is taken. An offset value between 0 and 15 is added to the residue, and the resulting value is again divided by 16 and its residue is taken. The resulting values are used as the indices of the frequency hopping table. Figure 21 illustrates the frame frequency mapping scheme.

The offset values are added to provide more independent frequency channels. A different offset value means a different communication channel. All control frames use '0' as the offset value.

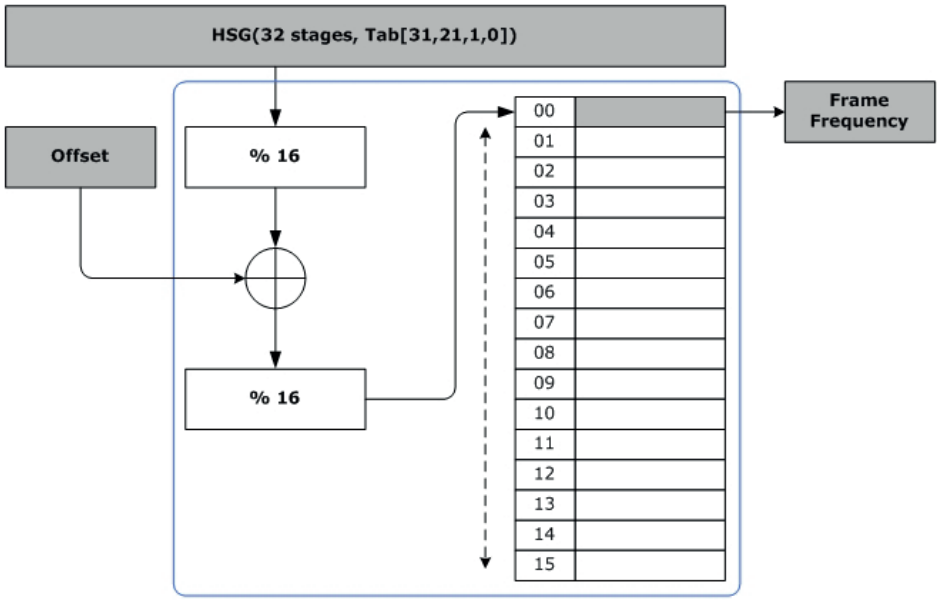

Figure 21 — Frame frequency mapping scheme

8.4.3 Frequency diversity and time diversity

In the frequency band of 2,4 GHz, the communication channels are very vulnerable to fading. To maintain the communication quality, diversity techniques are adopted. Time diversity can be achieved through a multiple transmission of identical frames in the same middleframe. On the other hand, the frequencies hop frame by frame by varying the offset values of the frequency hopping sequence generator. Though the hopping sequence generator creates a new value by the unit of middleframe, a different offset value is applied to each payload frame in order to make the identical payload frames within the middleframe have different frequencies. In this way, frequency and time diversity are achieved simultaneously.

8.4.4 Orthogonal frequency offset

For payload and control frames, their frequencies change according to the hopping sequence determined by the hopping sequence generator. Except for the RCF, control frames can avoid collision as only one device is allowed to transmit at a time. However, when a need of multiple simultaneous independent communication channels in a pico-net arises, collision is inevitable unless means to avoid it is devised. In other words, to allow multiple simultaneous independent communications, means to assign a different frequency to each communication group should be devised. The multiple simultaneous communications are made possible by assigning a different offset value to the hopping sequence generator for each communication group.

8.4.5 Frequency selection

The master uses sounding techniques to initialise the contents of the frequency hopping table with the best 16 frequencies free of interference from the surroundings. Even in the midst of operation, the master updates the table through passive and static sounding techniques. The renewed frequency table is transmitted regularly to the slaves via beacon frames (BFs) so that the identical frequency tables can be maintained within the same group in the pico-net.

8.4.5.1 Passive sounding

Passive sounding is a technique in which the master scans all through the frequency band and selects the best 16 frequencies. It is done in the initialization process to compose the hopping frequency table. Passive sounding requires a RSSI signal from the RF receiver circuit that indicate the level of each frequency.

The procedure of passive sounding is as follows.

a) RSSI is checked for 80 frequencies.

b) The above process is performed N times and the worst value is chosen for each frequency. The RSSI values are sorted in ascending order and the best 16 frequencies are selected.

c) The hopping frequency table is filled out with the selected 16 frequencies.

The time unit for the measurement of the channel is the middleframe. Figure 22 shows the time unit for channel measurement. The structure shown is just to compare with that of the middleframe. The length of each measurement time unit is 640 μsec. 80 sounding units constitute one cycle to measure the RSSI of the 80 frequencies. While passive sounding is being performed, control and payload frames cannot be used.

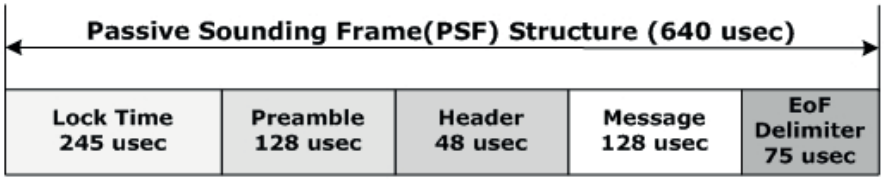

Figure 22 — Middleframe structure of passive sounding

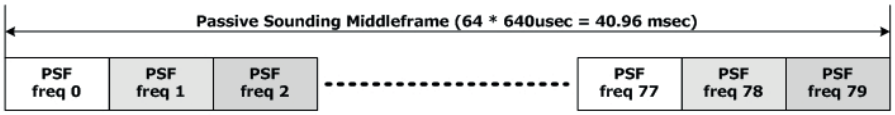

Figure 23 — One cycle of passive sounding frames

8.4.5.2 Static sounding

In static sounding, the master and the slaves cooperate to check the condition of the frequency band and to select the 16 best frequencies. Static sounding is used when passive sounding cannot be adopted due to the lack of function in the RF receiver circuit that measures the RSSI. It is also used when a more accurate channel estimation is needed.

Figure 24 illustrates how static sounding is performed. In response to the control signal from the master, the slaves transmit static sounding signals of known bit pattern in the message field. The master estimates the channel by measuring the bit errors of the known bit pattern. The number of bit errors is called the static sounding value (SSV) and used as the measure of channel quality.

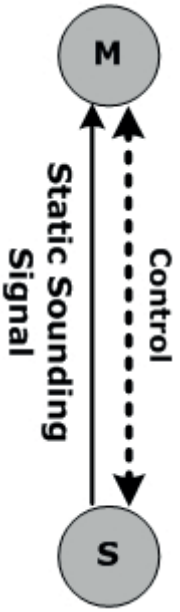

Figure 24 — Illustration of static sounding

The procedure of static sounding is as follows.

a) The master receives the static sounding signals for 80 channel frequencies transmitted by the slaves.

b) The SSV is measured for each of the 80 frequencies.

c) The above process is performed *N* times and the worst value is chosen for each frequency. The values are sorted in ascending order, and the best 16 frequencies are selected.

d) The hopping frequency table is filled out with the selected 16 frequencies.

The structure of the static sounding frame is shown in Figure 25. The structure is similar to that of a normal frame, that is, it consists of lock time, preamble, header, message and EoF. The length of a static sounding middleframe is 945 μsec.

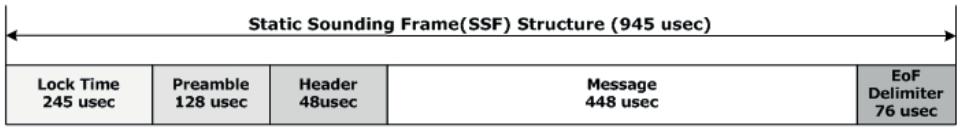

Figure 25 — Middleframe structure of static sounding

Figure 26 shows how the SSVs are measured.

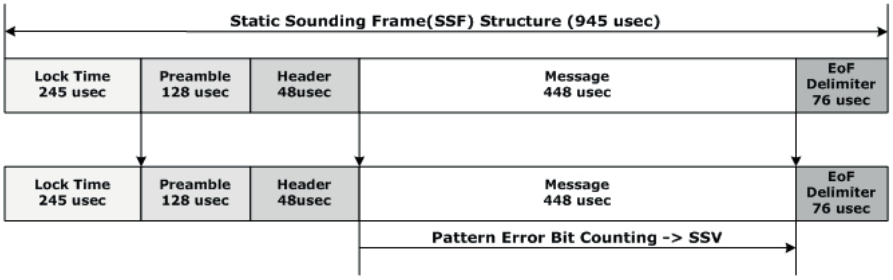

Figure 26 — Static sounding value

As shown in Figure 27, static sounding is performed preserving the superframe-like structure. 16 static sounding frames constitute a static sounding superframe. The control frames may be used as usual but the payload frames are used only for sounding. Therefore, usual communications cannot be practiced during the static sounding period.

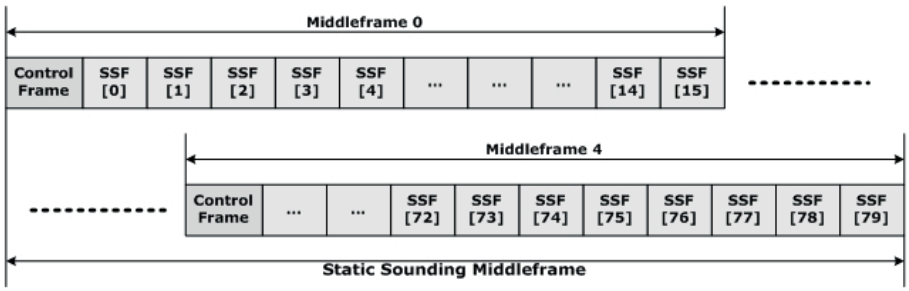

Figure 27 — The structure of the static sounding superframe

9 PHY specification

9.1 General requirements

9.1.1 Operating frequency range

The PHY layer is designed to support the adaptive frequency hopping technology in the 2.4 GHz Industrial Scientific Medical (ISM) band. Using sounding techniques, the best 16 out of all frequencies available are selected to form a frequency table, and the frequency hopping method is used with these frequencies.

The frequency band is between 2 400 MHz and 2 483,5 MHz; a total of 80 frequency channels are available. The channel frequencies are determined as shown below. The RF channel space is 1 MHz, and the number of channel, *k*, is decided in order.

Frequency range	RF channels
2,400 GHz to 2,4835 GHz	f = 2402 + k MHz, k = 0, …, 79

Two guard bands are placed at the edges of the frequency band as shown below.

Lower guard band	Upper guard band
2 MHz	2,5 MHz

9.1.2 Frequency assignment

Frequencies are assigned by the unit of middleframe as described in 8.4. For diversity's sake, payload frames may be repeated and assigned different frequencies by varying the offset value in the hopping sequence generator.

9.1.3 Frequency synthesizer stabilisation time

The frequency of the signal generated by a frequency synthesizer must stabilise within the lock time as defined in 9.2.1. After the lock time, the transmitter should not have any problems in delivering data within the frame.

9.1.4 Frequency synthesizer turn off time

The frequency synthesizer must be turned off completely within EOF (end of frame) as defined in 9.2.5.

9.2 PHY protocol data unit (PPDU) format

Figure 28 shows the structure of a frame which consists of lock time, preamble, header, message and EoF. The frame starts with lock time, which is necessary for the stabilisation of the frequency synthesiser. The lock time is followed by preamble, which is for synchronisation, and then header and message. The header field is used for discrimination of frames and the message field is where data are loaded. At the end of the frame comes the 'end of frame' (EoF) field which indicates the end of the frame and spares time to prepare for next frame. Table 12 summarises the use of each field in the frame. Detailed description is given below.

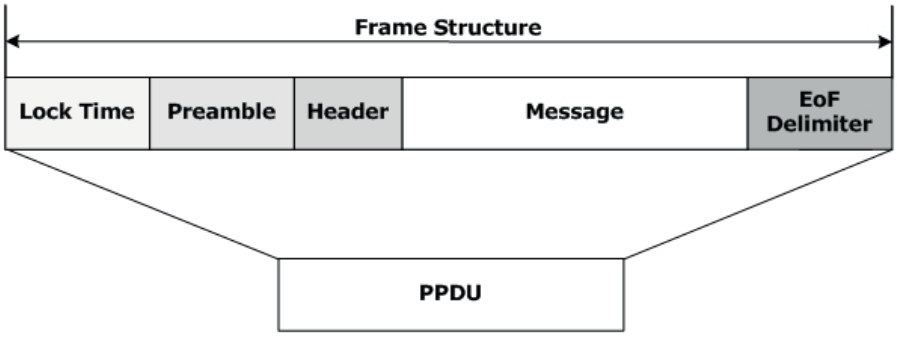

Figure 28 — PHY Protocol Data Unit (PPDU) format

Table 12 — The use of the fields in a frame

Field	Use
Lock time	Time for stabilisation of RF frequency synthesiser.
Preamble	Coded symbols for synchronisation
Header	Designation of frame types (e.g. emergency communications); special control signals
Message	Data
EoF	Indication of the end of a frame; RF circuit stabilisation

9.2.1 Lock time

Since shifting of frequencies and transmission/reception modes are practiced by the unit of frame, 'lock time' is required for the RF circuit to stabilise from frame to frame. Data transmission and reception are forbidden during lock time. When in the transmission mode, modem sends the alternating bits of 0 and 1 to the RF circuit during lock time; when in the reception mode, modem ignores the data.

9.2.2 Preamble

Preamble is a field for transmitting symbols for synchronisation. The length of the preamble is fixed to 128 μsec except for the payload frames. The 128-bit preamble is formed by adding "0" to a 127-bit gold code which is generated by a 7-bit scan code. Figure 29 shows one example of a gold-code generator. The length of the preambles of the payload frames may be varied up to the user's need to enhance the data throughput.

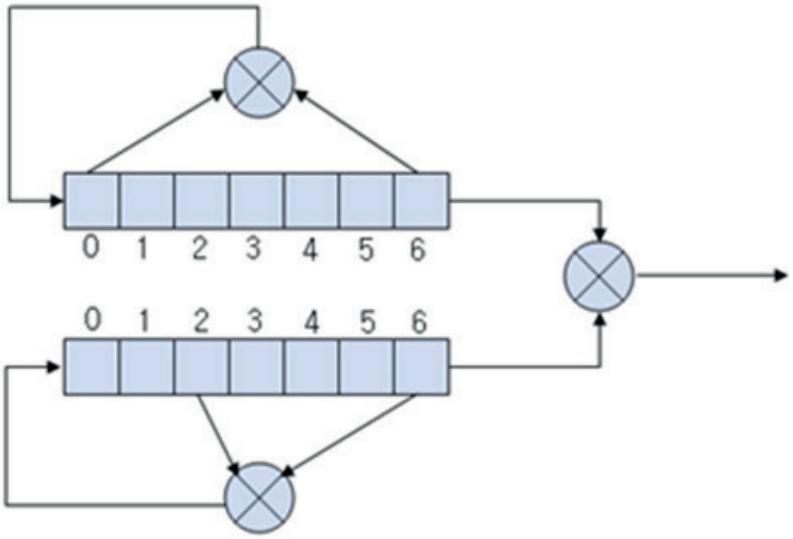

Figure 29 — An example of Gold code generators

9.2.3 Header (48 bits)

Header is a field to indicate the use of the frame. This field clarifies the use of the frame, which cannot be identified only with the preamble. It can also be used for open broadcasting channels or auxiliary signals. The length of a header is fixed to 32 bits; CRC (cyclic redundancy code) of 16 bits named TCRC16 shall be applied to the header.

9.2.4 Message

Message is a field where user data are loaded. The length of the message field is fixed for control frames, but variable for payload frames. For payload frames, the length can be set by the unit of 8 μsec, up to 18 units. CRC of 16 bits named MCRC16 shall be applied to the control frame message. For payload frames, CRC may be applied if needed.

9.2.5 EoF delimiter

EoF is a field to indicate the end of a frame. It is also used for stabilisation of the RF circuit and the modem which alternate transmission and reception modes from frame to frame. The duration of EoF may be set by the unit of 1 μsec.

9.3 Modulation and codes

9.3.1 Modulation

9.3.1.1 Modulation scheme

A binary GFSK or BPSK and QPSK are used for modulation. QPSK may be used for payload message data. When QPSK is used for payload message data, all the other bits shall be modulated with BPSK. The default modulation format is GFSK. If transmitting antenna of directional gain greater than 0 dBi are used, the applicable paragraphs in EN 300 328, EN 301 489-17, and FCC part 15 shall be compensated for.

9.3.1.2 Symbol rate

Symbol rate is 1 Msps.

9.3.2 Codes

9.3.2.1 Scan code

A scan code is 7-bit seed to generate a preamble for synchronisation. The 128-bit gold code (127 gold code with one 0 padded) generated by a scan code is used as a preamble of the middleframe. Communications are possible only when the scan codes of the receiver and the transmitter are identical. A scan code has a value between 1 and 127.

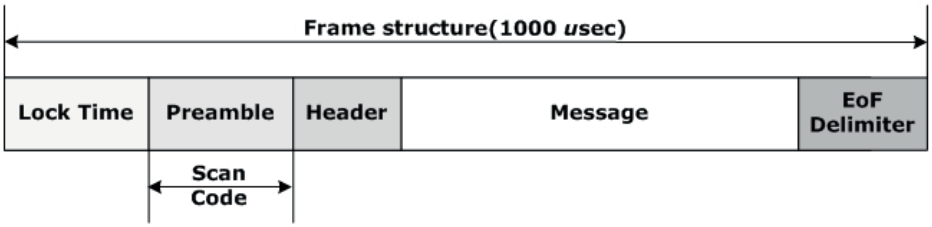

Figure 30 — Scan code

9.3.2.2 Security code

A security code is used for data transmission and reception. The period of the security code is 2^{16} bit long. The security code is multiplied to the message field. Communications are possible only when the phases of the security codes of the receiver and the transmitter are identical.

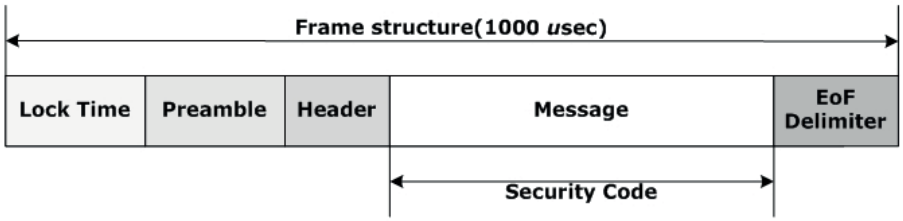

Figure 31 — Security code

Maximal length sequences generated by a 16-stage linear shift register are used for security codes. The generator polynomial for the maximal length sequences shall be

$G(X) = 1 + X^2 + X^{11} + X^{15}$

9.3.2.3 Group code

A group code is used for a group communication. The period of the group code is 2^{64} bit long. The group code is multiplied to the message field. Communications are possible only when only when the phases of the group codes of the receivers and the transmitters in a group are identical.

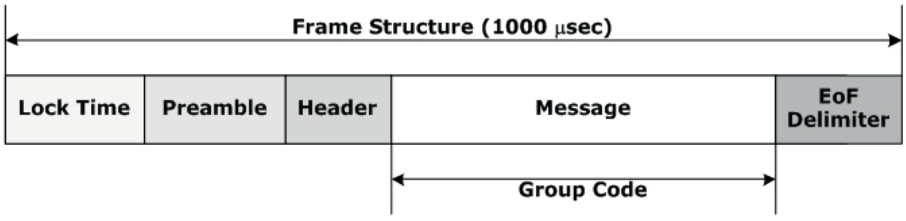

Figure 32 — Group code

Gold codes generated by two 32-stage linear shift registers are used for group codes. The generator polynomials for the group codes shall be

$G_1(X) = (1 + X^1 + X^{21} + X^{31})$

$G_2(X) = (1 + X^1 + X^2 + X^4 + X^6 + X^{31})$

9.3.2.4 Cyclic redundancy code (CRC)

Cyclic redundancy codes (CRC) shall be added to all headers (TCRC16) and the messages (MCRC16) of control frames. The generator polynomial and initial value conform to CRC-CCITT of ITU-T Recommendation V.41. The generator polynomial is

$X^{16} + X^{12} + X^5 + 1$

and the initial value is 0x0000.

9.4 Transmitter specification

The transmitter specification should satisfy international regulations and national laws which regulate the use of radio receivers and transmitters.

9.4.1 Pulse shaping filter

The modulation format of GFSK (Gaussian frequency shift keying) shall have a bandwidth - bit duration product BT = 0,5. The modulation index shall be of 0,32 ± 0,03. The symbol rate is 1 Msps. The symbol-rate deviation shall be less than 20 ppm. A positive frequency deviation shall correspond to a binary data ‘1’ and a negative frequency deviation shall correspond to a binary data ‘0’.

9.4.2 Transmitter power spectrum mask

The in-band spurious emission shall be less than –20 dBc (relative) or –20 dBm (absolute), whichever smaller.

Annex A
(informative)

Pico-net Light-weight Architecture Security (PLAS)

A.1 General

This Annex provides a Pico-net Light-weight Architecture Security (PLAS) based on a pre-shared secret for link layer authentication and key generation in a pico-net. Especially, the Authentication Capability Negotiation (ACN), Light-weight Shared-Key Authentication Protocol (LSAP) and Group Key Authentication Protocol (GKAP) are specified in the Annex. In this mechanism, the Pre-Shared Key (PSK) shall be a uniformly random key with the length of 256 bits and distributed to the pico-net devices by a method outside the scope of this Standard. Guidance on the management of PSKs is provided in ISO/IEC 11770-1 and ISO/IEC 11770-2. Use of the weak secrets such as short password is out of the scope of this standard.

Cryptographic algorithms to be applied to information security mechanism may be subject to national and regional regulations. In this International Standard, cryptographic algorithms are instantiated, which should conform to national laws and regulations, and can be chosen according to specific requirements in different countries and regions.

A.2 Authentication Capability Negotiation (ACN)

A.2.1 MPDU of authentication capabilities announcement frame (ACCF)

A.2.1.1 Format

ACCF is used to announce the authentication capabilities. When the master sends its authentication capabilities used for authentication, the slaves obtain the authentication capabilities of the master from ACCF. When a slave sends its authentication capabilities used for authentication, another slave obtain the authentication capabilities of the slave from ACCF. The MPDU format of ACCF is shown in Figure A.1.

OF(2)/MAC Version(6)	ADDM(2)/PHY Version(6)
Frame Type	Reserved(2)/ULPS(6)
TCRC16	
Source MAC Address (8 Bytes)	
Destination MAC Address (8 Bytes, Optional)	
Upper Layer Data	
MCRC16	

Figure A.1 — MPDU format of ACCF

A.2.1.2 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF, which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

A.2.1.3 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

A.2.1.4 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists. In this frame, the first bit of ADDM is always set to 1, i.e., a source MAC address exists. When the frame is sent from the master to the slaves, the second bit of ADDM is set 0, i.e., a destination MAC address does not exist. When the frame is sent from one slave to the other slave, the second bit of ADDM is set 1, i.e., a destination MAC address exists.

A.2.1.5 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

A.2.1.6 Frame type (8 bits)

This field is used to indicate the type of the current frame.

A.2.1.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

A.2.1.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending ACCF.

A.2.1.9 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive ACCF.

A.2.1.10 Upper layer data

This field is used for transmission of the data from the upper layer. The format of this field is shown in Figure A.2.

Flags(8)	PTK-ASN(4)/GTK-ASN(4)
Authentication Suites for PTK	
Authentication Suites for GTK	

Figure A.2 — The format of the upper layer data field of ACCF

The first bit of the Flags field indicates whether the master/slave supports authentication, the second bit of the Flags field indicates whether a non-authenticated slave can communicate with the master/slave, and the other bits of the Flags field are reserved. If the first bit of the Flags field is set to 0, then the PTK-ASN field, the GTK-ASN field, the Authentication Suites for PTK field and the Authentication Suites for GTK field do not exist. When the frame is sent from one slave to the other slave, the GTK-ASN field is set to 0. The PTK-ASN field indicates the number of authentication suites for Pairwise Temporal Key (PTK), while the GTK-ASN field indicates the number of authentication suites for Group Temporal Key (GTK). The Authentication Suites for PTK field includes all authentication suites for PTK, and each of authentication suites for PTK is identified by a 4 bits integer. The Authentication Suites for GTK field includes all authentication suites for GTK, and each of authentication suites for GTK is identified by a 4 bits integer. The authentication suites for PTK and GTK are shown in Table A.1 and Table A.2, respectively.

Table A.1 — The authentication suites for PTK

Authentication suite identifier	Authentication suite
0001	LSAP(see Annex A.3)
0010-1111	Reserved

Table A.2 — The authentication suites for GTK

Authentication suite identifier	Authentication suite
0001	GKAP(see Annex A.4)
0010-1111	Reserved

A.2.1.11 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for Tag and MCRC16 for message.

A.2.2 MPDU of authentication access request frame (AARQF)

A.2.2.1 Format

AARQF is used to request authentication with the master/slave. A slave sends AARQF, and the master/slave receives AARQF. The MPDU format of AARQF is shown in Figure A.3.

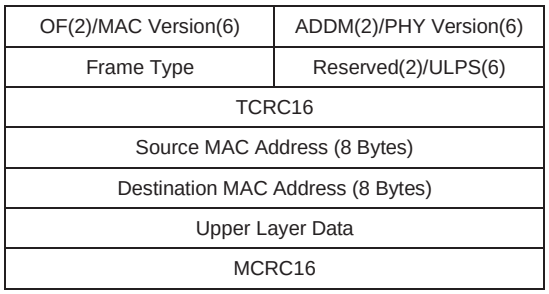

Figure A.3 — MPDU format of AARQF

A.2.2.2 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF, which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

A.2.2.3 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

A.2.2.4 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists. In this frame, the first bit of ADDM is always set to 1, i.e., a source MAC address exists, and the second bit of ADDM is always set 1, i.e., a destination MAC address exists.

A.2.2.5 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

A.2.2.6 Frame type (8 bits)

This field is used to indicate the type of the current frame.

A.2.2.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

A.2.2.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending AARQF.

A.2.2.9 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive AARQF.

A.2.2.10 Upper layer data

This field is used for transmission of the data from the upper layer.

The format of this field is the same as that of the upper layer data field of ACCF. The value of the Flags field of this field is the same as those of the Flags field of upper layer data field of ACCF. When the frame is sent from one slave to the other slave, the GTK-ASN field is set to 0, the PTK-ASN field is set one and the Authentication Suites for PTK field only includes one authentication suites identifier for PTK selected by the slave. When the frame is sent from a slave to the master, the PTK-ASN field is set one and the Authentication Suites for PTK field only includes one authentication suites identifier for PTK selected by the slave, and the GTK-ASN field is set to 1 and the Authentication Suites for GTK field only includes one authentication suites identifier for GTK selected by the slave.

A.2.2.11 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for Tag and MCRC16 for message.

A.2.3 MPDU of authentication access response frame (AARSF)

A.2.3.1 Format

AARSF is used by the master/slave to announce authentication status to a slave. The master/slave sends AARSF, and a slave receives AARSF. The MPDU format of AARSF is shown in Figure A.4.

OF(2)/MAC Version(6)	ADDM(2)/PHY Version(6)
Frame Type	Reserved(2)/ULPS(6)
TCRC16	
Source MAC Address (8 Bytes)	
Destination MAC Address (8 Bytes)	
Upper Layer Data	
MCRC16	

Figure A.4 — MPDU format of AARSF

A.2.3.2 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF, which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

A.2.3.3 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

A.2.3.4 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists. In this frame, the first bit of ADDM is always set to 1, i.e., a source MAC address exists, and the second bit of ADDM is always set 1, i.e., a destination MAC address exists.

A.2.3.5 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

A.2.3.6 Frame type (8 bits)

This field is used to indicate the type of the current frame.

A.2.3.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

A.2.3.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending AARSF.

A.2.3.9 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive AARSF.

A.2.3.10 Upper layer data

This field is used for transmission of the data from the upper layer. The field only includes an authentication status code. The values of the authentication status code are shown in Table A.4.

Table A.4 — The values of the authentication status code

Authentication status code	Authentication status
0x01	Authentication success
0x02	Authentication failure
0x03-0xff	Reserved

A.2.3.11 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for Tag and MCRC16 for message.

A.2.4 MPDU of general authentication process frame (GAPF)

A.2.4.1 Format

GAPF is used to perform authentication process. All devices can send and receive GAPF. The MPDU format of GAPF is shown in Figure A.5.

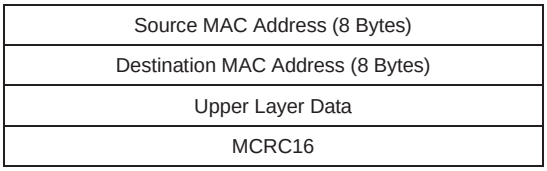

Figure A.5 — The MPDU format of GAPF

A.2.4.2 Open flag (OF, 2 bits)

This field provides information on whether an open code is applied to the current frame. It comprises of GCOF, which is used to indicate whether a group code is applied and SCOF which is used to indicate whether a security code is applied. That GCOF is 1 means an open group code is applied. When it is 0, a closed group code is applied. That SCOF is 1 means an open security code is applied. When it is 0, a closed security code is applied.

A.2.4.3 MAC version (6 bits)

This field provides information on the MAC version, and has a value between 0x00 to 0x3F.

A.2.4.4 Address mode (ADDM, 2 bits)

This field provides information on whether MAC addresses exist. The ADDM comprises of two bits: one bit is used to indicate whether a source MAC address exists and the other bit is used to indicate whether a destination MAC address exists. In this frame, the first bit of ADDM is always set to 1, i.e., a source MAC address exists, and the second bit of ADDM is always set 1, i.e., a destination MAC address exists.

A.2.4.5 PHY version (6 bits)

This field provides information on the PHY Version, and has a value between 0x00 to 0x3F. When (G) FSK is used, this field shall be 0x00. For BPSK, this field shall be 0x01. For QPSK, this field shall be 0x03.

A.2.4.6 Frame type (8 bits)

This field is used to indicate the type of the current frame.

A.2.4.7 Upper layer frame size (ULPS, 6 bits)

This field provides information on the size of the data from the upper layer.

A.2.4.8 Source MAC address (64 bits)

This field provides the MAC address of the master which is sending GAPF.

A.2.4.9 Destination MAC address (64 bits)

This field provides the MAC address of the device which is to receive GAPF.

A.2.4.10 Upper layer data

This field is used for transmission of the authentication protocol data from the upper layer. It shall be supported that the authentication protocol data are encapsulated in a certain message format.

A.2.4.11 TCRC16 (16 bits), MCRC16 (16 bits)

These fields are used for sixteen cyclic-redundancy-check bits. TCRC16 is for Tag and MCRC16 for message.

A.3 Light-weight Shared-Key Authentication Protocol (LSAP)

A.3.1 Message flow

LSAP is used to establish a PTK which may be used to derive a security code (see 9.3.2) for protecting point to point communication in the pico-net. The LSAP message flow is shown in Figure A.6, where the master/slave acts as an initiator and another slave acts as a responder.

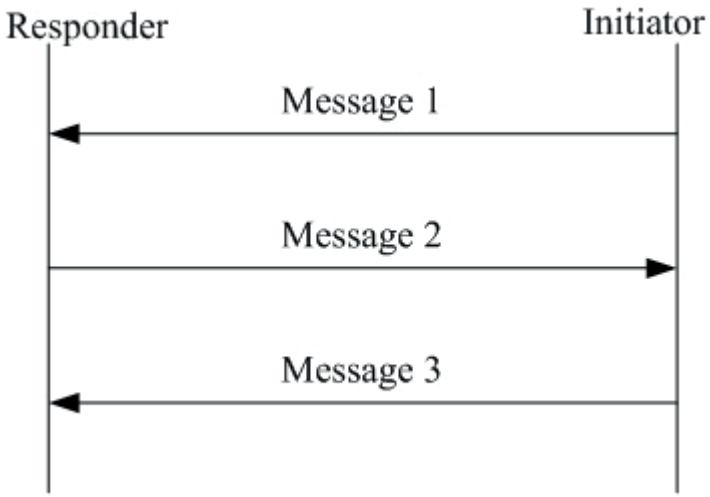

Figure A.6 — LSAP message flow

A.3.2 Message 1

The initiator generates a 128 bits random number N_I, and then sends message 1 to the responder, where message 1 includes N_I.

A.3.3 Message 2

On receipt of message 1, the responder performs the following steps.

a) Concatenate the MAC addresses of the initiator and responder, denoted by ADDID.

b) Generate a 128 bits random number NR.

c) Compute *PTK* = KD-HMAC (*PSK*, *ADDID*||N_I ||N_R||'pairwise key expansion for PTK'), where *PTK* is a pairwise temporal key between the initiator and responder, *PSK* is a pre-shared key between the initiator and responder, and HMAC-SHA-256 is recommended as the KD-HMAC.

d) Generate a message authentication code MAC_R = HMAC (*PTK*, N_I ||N_R).

e) Send message 2 to the initiator, where message 2 includes N_I, N_R and MAC_R.

A.3.4 Message 3

On receipt of message 2, the initiator performs the following steps.

a) Concatenate the MAC addresses of the initiator and responder, denoted by *ADDID*.

b) Determine whether N_I included in message 2 is the same as the N_I included in message 1. If N_I included in message 2 is NOT the same as the value included in message 1, the initiator discards message 2. Otherwise, performs step 3.

c) Compute *PTK* = KD-HMAC (*PSK*, *ADDID*||N_I ||N_R||'pairwise key expansion for PTK'), where *PTK* is a pairwise temporal key between the initiator and responder, *PSK* is a pre-shared key between the initiator and responder, and HMAC-SHA-256 is recommended as the KD-HMAC.

d) Generate a message authentication code MAC_R = HMAC (*PTK*, N_I||N_R), and then determine whether MAC_R generated by the initiator locally is the same as MAC_R included in message 2. If MAC_R generated by the initiator locally is NOT the same as MAC_R included in message 2, then the initiator discards message 2. Otherwise, performs step 5.

e) Generate a message authentication code MAC_I = HMAC (*PTK*, N_R), and then send message 3 to the responder, where message 3 includes N_R and MAC_I.

On receipt of message 3, the responder performs the following steps.

a) Determine whether N_R included in message 3 is the same as N_R included in message 2. If N_R included in message 3 is NOT the same as N_R included in message 2, the responder discards message 3. Otherwise, performs step 2.

b) Generate a message authentication code MAC_I = HMAC (*PTK*, N_R), and then determine whether MAC_I generated by the responder locally is the same as MAC_I included in message 3. If MAC_I generated by the responder locally is NOT the same as MAC_I included in message 3, then the responder aborts. Otherwise, the initiator and responder authenticate successfully each other.

A.4 Group Key Authentication Protocol (GKAP)

A.4.1 Message flow

GKAP is used to establish a GTK which may be used to derive a security code (see 9.3.2) for protecting group communication in the pico-net. Note that the master key announced is also encrypted by security code. The GKAP message flow is shown in Figure A,7, where the master acts as an initiator and the slave acts as a responder.

If the Initiator does not receive a reply to message 1 within the expected time interval, it shall retransmit of message 1. The retransmit timeout value shall be 50 ms for the first timeout, half the listen interval for the second timeout, and the listen interval for subsequent timeouts. If there is no listen interval, then 50 ms shall be used for all timeout values. If it still has not received a response after these retries, then the Initiator should disconnect with the Responder.

If the Responder does not receive Message 1 within the expected time interval, it should disconnect with the Initiator.

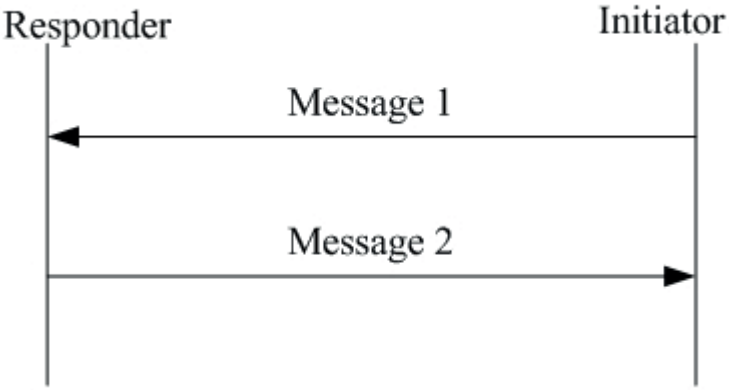

Figure A.7 — GKAP message flow

A.4.2 Message 1

After a successful LSAP, the initiator may initiate a GKAP and perform the following steps.

a) Concatenate the MAC addresses of the initiator and responder, denoted by *ADDID*.

b) Generate a key announcement identifier N_{KN}, which is a 128 bits random number.

c) Generate an announcement master key K_{NMK}, and encrypts K_{NMK} using *PTK*, denote by $\{K_{NMK}\}_{PTK}$. In the meantime, the initiator derives *GTK* = KD-HMAC (K_{NMK}, 'group key expansion for GTK').

d) Compute MAC_I = HMAC (PTK, $ADDID||N_{KN}||\{K_{NMK}\}_{PTK}$).

e) Send message 1 to the responder, where message 1 includes *ADDID*, N_{KN}, $\{K_{NMK}\}_{PTK}$ and MAC_I.

A.4.3 Message 2

On receipt of message 1, the responder performs the following steps.

a) Concatenate the MAC addresses of the initiator and responder, denoted by *ADDID*, and determine whether *ADDID* included in message 1 is the same as *ADDID* generated locally. If *ADDID* included in message 1 is NOT the same as the value generated locally, then the responder discards message 1. Otherwise, performs step 2.

b) Generate a message authentication code MAC_I = HMAC (PTK, $ADDID||N_{KN}||\{K_{NMK}\}_{PTK}$), and then determine whether MAC_I generated by the responder locally is the same as MAC_I included in message 1. If MAC_I generated by the responder locally is NOT the same as MAC_I included in message 1, then the initiator discards message 1. Otherwise, performs step 3.

c) Decrypt $\{K_{NMK}\}_{PTK}$ using *PTK*, and then the responder derives *GTK* = KD-HMAC (K_{NMK}, 'group key expansion for GTK').

d) Generate a message authentication code MAC_R = HMAC (PTK, $ADDID||N_{KN}$).

e) Send message 2 to the initiator, where message 2 includes *ADDID*, N_{KN} and MAC_R.

On receipt of message 2, the initiator performs the following steps.

a) Determine whether *ADDID* included in message 2 is the same as *ADDID* included in message 1. If *ADDID* included in message 2 is NOT the same as *ADDID* included in message 1, then the initiator discards message 2. Otherwise, performs step 2.

b) Determine whether N_{KN} included in message 2 is the same as N_{KN} included in message 1. If N_{KN} included in message 2 is NOT the same as N_{KN} included in message 1, then the initiator discards message 2. Otherwise, performs step 3.

c) Generate a message authentication code MAC_R = HMAC (PTK, $ADDID||N_{KN}$), and then determines whether MAC_R included in message 2 is the same as MAC_R generated locally. If MAC_R included in message 2 is NOT the same as MAC_R generated locally, then the initiator aborts. Otherwise, the initiator and responder complete GKAP successfully.

A.5 Digital Certificate Support

If each device in a pico-net supports digital certificate management, the authentication mechanisms that are defined in ISO/IEC 9798-3 should also be recommended for implementation. Figure A.8 shows a group communication example of pico-net using the digital certificates.

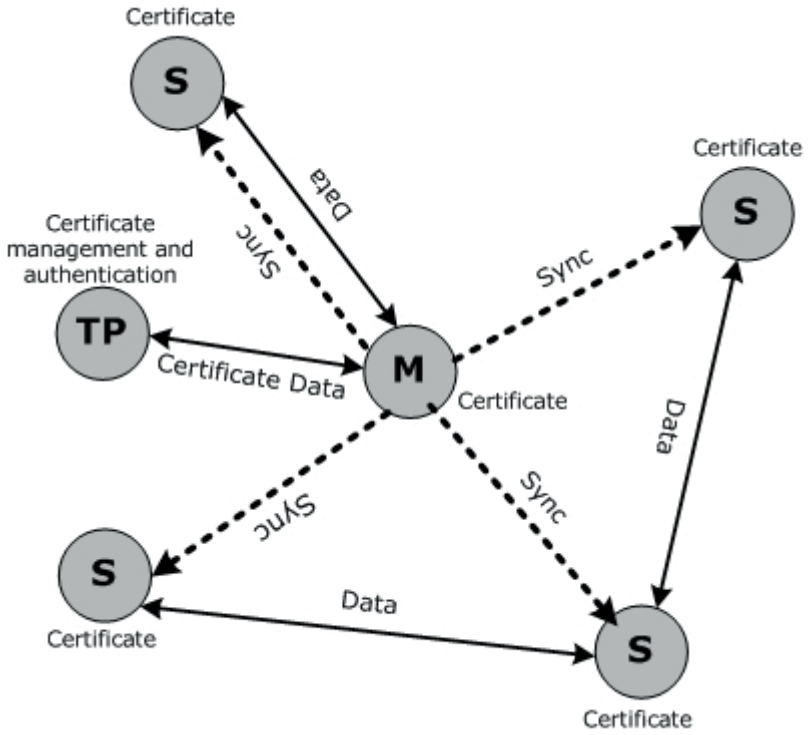

Figure A.8 — A group communication example using the digital certificates

ICS 35.110
Price based on 55 pages