BS EN 62601:2016

BSI Standards Publication

Industrial networks — Wireless communication network and communication profiles — WIA-PA

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of EN 62601:2016. It is identical to IEC 62601:2015. It supersedes BS IEC 62601:2011 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications: process measurement and control, including fieldbus.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2016

ISBN 978 0 580 85030 1
ICS 25.040.40; 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 April 2016.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62601

April 2016

ICS 25.040.40; 35.100.05

English Version

Industrial networks - Wireless communication network and communication profiles - WIA-PA (IEC 62601:2015)

Réseaux industriels - Réseau de communications sans fil et profils de communication - WIA-PA (IEC 62601:2015)

Industrielle Kommunikationsnetze - Kommunikationsnetze und Kommunikationsprofile - WIA-PA (IEC 62601:2015)

This European Standard was approved by CENELEC on 2016-01-13. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

European foreword

The text of document 65C/821/FDIS, future edition 2 of IEC 62601, prepared by SC 65C “Industrial networks” of IEC/TC 65 “Industrial process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62601:2016.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2016-10-13
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2019-01-13

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 62601:2015 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61804-2	NOTE	Harmonized as EN 61804-2.
IEC 61499-1	NOTE	Harmonized as EN 61499-1.
IEC 61499-2	NOTE	Harmonized as EN 61499-2.
ISO 3166-1	NOTE	Harmonized as EN ISO 3166-1.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu

Publication	Year	Title	EN/HD	Year
ISO/IEC 9899	-	Information technology - Programming languages - C	-	-
ISO 3166-1	-	Codes for the representation of names of countries and their subdivisions - Part 1: Country codes	EN ISO 3166-1	-
IEEE 802.15.4	2011 [1)]	IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)	-	-

[1)] Superseded by IEEE 802.15.4-2015.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

INDUSTRIAL NETWORKS – WIRELESS COMMUNICATION NETWORK AND COMMUNICATION PROFILES – WIA-PA

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62601 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2011. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- changed IEEE STD 802.15.4-2006 to IEEE STD 802.15.4-2011 and added common modification for IEEE STD 802.15.4-2011 MAC profile, PHY profile and IEEE STD 802.15.4-2011 related references;
- added common modifications for regional adoption and added Annex D and Annex E;

- deleted extended MAC management services and added two DLSL management services;
- added specific state machines for DLSL and NL;
- unified representation of frame format and packet format;
- changed format of definition of data types;
- added detailed description of technologies for clearer understanding;
- provided support for CCA modes 1, 2, and 3.

The reader's attention is drawn to the fact that Annex E lists all of the “in-some-country” clauses on differing practices of a less permanent nature relating to the subject of this standard.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/821/FDIS	65C/833/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

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

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INDUSTRIAL NETWORKS – WIRELESS COMMUNICATION NETWORK AND COMMUNICATION PROFILES – WIA-PA

1 Scope

This International Standard specifies the system architecture and the communication protocol of Wireless networks for Industrial Automation – Process Automation (WIA-PA) that is built on IEEE STD 802.15.4-2011.

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 9899, *Information technology – Programming languages – C*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

IEEE STD 802.15.4-2011, *IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

3.1.1
absolute timeslot number
number of timeslots from the start of the network, generally denoting the current timeslot

3.1.2
active leaving
process by which an online field device is allowed to leave the network through applying to its routing device or by which an online routing device is allowed to leave the network through applying to the gateway device

3.1.3
adaptive frequency hopping
change of communication channels according to actual condition of channels in every timeslot during the intra-cluster period of WIA-PA superframe

3.1.4
adaptive frequency switch
change of communication channels according to the actual condition of channels during the beacon frame and active period in a superframe cycle, and using different channels in different superframe cycles

3.1.5
aggregation
merging of data from multiple user application objects or merging of several packets from cluster members into one packet

3.1.6
application sub-layer
protocol sub-layer that provides data and management services for the application layer

3.1.7
beacon
special frame broadcast by the routing devices or the gateway device in the WIA-PA network

3.1.8
broadcast
sending one packet to all WIA-PA devices simultaneously

3.1.9
channel
RF medium used to convey a packet from a sender to a receiver

3.1.10
cluster
logical group of devices that is comprised of a routing device and field devices

3.1.11
clusterhead
manager in a cluster, performed by the routing device

3.1.12
clustermember
data source in a cluster, performed by a field device

3.1.13
coexistence
ability of one network to perform its task in a given shared environment without disturbing or being disturbed by other networks

Note 1 to entry: These networks may or may not have the same set of rules.

3.1.14
communication resource
channels and timeslots used to transport a frame

3.1.15
WIA-PA configuration software
software tool for configuring a WIA-PA network and devices

3.1.16
data link sub-layer
upper layer of the IEEE STD 802.15.4-2011 MAC layer, used to handle the aspects of network topology and communication resources in the WIA-PA network

3.1.17
disaggregation
dividing the merged packet into data of user application objects

3.1.18
field device
device usable in the field, which is connected to or controls the process

3.1.19
frame
format of aggregated bits from the medium access control (MAC) entity that are transmitted together in time

[SOURCE: IEEE STD 802.15.4-2011]

3.1.20
frequency hopping
change of transmitting/receiving frequency to combat interference and fading

3.1.21
gateway device
device connecting the WIA-PA network to other plant networks

3.1.22
handheld device
portable device used for provisioning, firmware updating, and device status monitoring

3.1.23
hop
movement of a packet directly between two adjacent neighbour devices in one network transaction without the participation of any other devices in the WIA-PA network

Note 1 to entry: Multiple hops are used to lengthen the transmission distance, bypass interference sources or avoid obstructions.

3.1.24
host configuration computer
device through which users and maintenance/management personnel perform transactions on the WIA-PA network and the management networks

3.1.25
interconnectable
using the same communication protocols, communication interface and data access

3.1.26
interoperable
able to work together to perform a specific role in one or more distributed application programs

Note 1 to entry: In this case parameters and their application-related functionality fit together both syntactically and semantically. Interoperability is achieved when the devices support complementary sets of parameters and functions belonging to the same profile.

3.1.27
interoperability
ability of two or more network systems to exchange information and to make mutual use of the information that has been exchanged

[SOURCE: ISO/IEC TR 10000-1:1998, 3.2.1 modified by replacing “IT systems” with “network systems”]

3.1.28
joining
process by which a WIA-PA device is authenticated and allowed to participate in the WIA-PA network

3.1.29
link
set of communication parameters necessary to transport a frame between adjacent devices in the network

Note 1 to entry: It includes source/destination address, timeslot, channel, direction, and link type.

3.1.30
mesh
topology formed by routing devices and the gateway device in the WIA-PA network

Note 1 to entry: One routing device may connect to the gateway device and to more than one other routing device.

3.1.31
multicast
sending one packet to a group of WIA-PA devices simultaneously

3.1.32
network address
short address
16-bit unsigned integer uniquely identifying the device in the WIA-PA network

Note 1 to entry: The most significant 8 bits of the network address, assigned by the network manager, identify different clusters.

3.1.33
network manager
logical role for configuring the network, allocating the communication resources, managing the routing tables, and monitoring and reporting the health of the network

3.1.34
packet
formatted, aggregated bits that are transmitted together in time across the physical medium

[SOURCE: IEEE STD 802.15.4-2011, 3.31]

3.1.35
packet lifecycle
maximal packet survival time from being generated to being dropped

3.1.36
passive leaving
process by which an online field device is instructed by its routing device to leave the network or by which an online routing device is instructed by the gateway device to leave the network

3.1.37
physical address
long address
EUI-64 bits uniquely identifying the device in the WIA-PA network

Note 1 to entry: A physical address is assigned by a manufacturer.

3.1.38
routing device
device forwarding packets from one WIA-PA device to another in the WIA-PA network

3.1.39
security manager
logical role for configuring the security strategies of the whole network, managing keys, and authenticating devices

3.1.40
superframe
collection of timeslots repeating over time

Note 1 to entry: It specifies the transmitting or receiving time of periodic communication.

3.1.41
timeslot
basic time unit of data exchange

Note 1 to entry: Its duration is configurable in the WIA-PA network.

3.1.42
timeslot hopping
regular change of transmitting/receiving frequency per timeslot to avoid interference and fading

3.1.43
unicast
sending one packet to a single device in the WIA-PA network

3.1.44
virtual communication relation
communication paths and communication resources between two user application objects

3.1.45
WIA-PA device
device in the WIA-PA network

EXAMPLE: Host configuration computer, gateway device, routing device, field device or handheld device.

3.2 Abbreviations

ACK	Acknowledge
AFH	Adaptive Frequency Hopping
AFS	Adaptive Frequency Switch
AIO	Analog Input Object
AL	Application Layer
AOO	Analog Output Object
ASDU	Application Service Data Unit
ASL	Application Sub-Layer
ASLDE	Application Sub-Layer Data Entity
ASLDE-SAP	ASLDEService Access Point
ASLME	Application Sub-Layer Management Entity
ASLME-SAP	ASLMEService Access Point
ASLPDU	Application Sub-Layer Protocol Data Unit
ASN	Absolute timeSlot Number
CAP	Contention Access Period
CFP	ContentionFree Period

C/S	Client/Server
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DAGO	Data AggreGation Object
DGO	DisaGgregation Object
DIO	Digital Input Object
DLDE	Data Link Sub-Layer Data Entity
DLDE-SAP	DLDE Service Access Point
DLL	Data Link Layer
DLME	Data Link Sub-Layer Management Entity
DLME-SAP	DLME Service Access Point
DLPDU	Data Link Sub-Layer Protocol Data Unit
DLSL	Data Link Sub-Layer
DMAP	Device Management Application Process
DOO	Digital Output Object
EIRP	Equivalent Isotropic Radiated Power
ENC	ENCryption
EUI-64	Extended Unique Identifier-64 bits
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FFD	Full-Function Device
FH	Frequency Hopping
GTS	Guaranteed Time Slot
GW	GateWay device
ID	IDentifier
IDS	Intrusion Detection System
KED	Data Encryption Key
KEK	Key Encryption Key
KJ	Join Key
KS	Share Key
LME-SAP	Layer Management Entity Service Access Point
LSB	Least Significant Bit
LQI	Link Quality Indication
MAC	Medium Access Control sub-layer
MCPS	MAC Common Part Sub-layer
MHR	Medium Access Control Header
MIB	Management Information Base
MIC	Message Integrity Code
MLDE	MAC sub-Layer Data Entity
MLDE-SAP	MLDE Service Access Point
MLME	MAC sub-Layer Management Entity
MLME-SAP	MLME Service Access Point
MPDU	MAC Protocol Data Unit

MSB	Most Significant Bit
NL	Network Layer
NLDE	Network Layer Data Entity
NLDE-SAP	NLDE Service Access Point
NLME	Network Layer Management Entity
NLME-SAP	NLME Service Access Point
NM	Network Manager
NPDU	Network Protocol Data Unit
NSDU	Network Service Data Unit
PAGO	Packet AGgregation Object
PAN	Personal Area Network
PHY	PHYsical layer
PIB	PAN Information Base
P/S	Publisher/Subscriber
RFD	Reduced-Function Device
R/S	Report source/Sink
SAP	Service Access Point
SM	Security Manager
TDMA	Time Division Multiple Access
TH	Timeslot Hopping
UAO	User Application Object
UAP	User Application Process
UAPME-SAP	UAP Management Entity SAP
UTC	Universal Time Coordinated
VCR	Virtual Communication Relationship
VCR_ID	Virtual Communication Relationship IDentifier
WIA-PA	Wireless network for Industrial Automation – Process Automation

4 Definition of data types

Table 1 gives the WIA-PA definition of data types.

Table 1 – Definition of data types

Definition of data types (see ISO/IEC 9899)			
Data type		Range of values	Length
Boolean	Boolean	Non 0: TRUE; 0: FALSE	1 octet
Unsigned integer	Unsigned8	$0 <= i <= 2^8-1$	1 octet
	Unsigned16	$0 <= i <= 2^{16}-1$	2 octets
	Unsigned24	$0 <= i <= 2^{24}-1$	3 octets
	Unsigned32	$0 <= i <= 2^{32}-1$	4 octets
	Unsigned40	$0 <= i <= 2^{40}-1$	5 octets
	Unsigned48	$0 <= i <= 2^{48}-1$	6 octets
	Unsigned64	$0 <= i <= 2^{64}-1$	8 octets
Octet string	Octetstring	8-bit string	1 octet
Float	Float	See IEC 60559	4 octets

5 WIA-PA overview

5.1 Device types

The document specifies five types of WIA-PA devices:

a) host configuration computer;

b) gateway device(GW);

c) routing device;

d) field device; and

e) handheld device.

To improve reliability, there may be redundant gateway devices and redundant routing devices in the WIA-PA network. A primary device connects its redundant device in the wired manner. The wired connection is not defined in this document. The short address (see 6.3.4) of a redundant device is the same as that of its primary device. The start-up time of a redundant device is not later than that of the primary device. When primary devices run, redundant devices do not start up their radio modules. When the information of primary devices is changed, the primary devices should backup in timely manner the changed information to their redundant devices in the wired manner. The wired manner is not specified in this document. If SecEnableFlag is enabled, the redundant devices are authenticated (see 11.2).

5.2 Network topology

WIA-PA network supports two different types of network topologies:

a) a hierarchical network topology that combines star and mesh, and

b) a star-only network topology.

The hierarchical network topology that combines star and mesh is illustrated in Figure 1. The first level of the network is in mesh topology, where routing devices and gateway devices are deployed. The second level of the network is in star topology, where routing devices, field devices, and handheld devices (if they exist) are deployed.

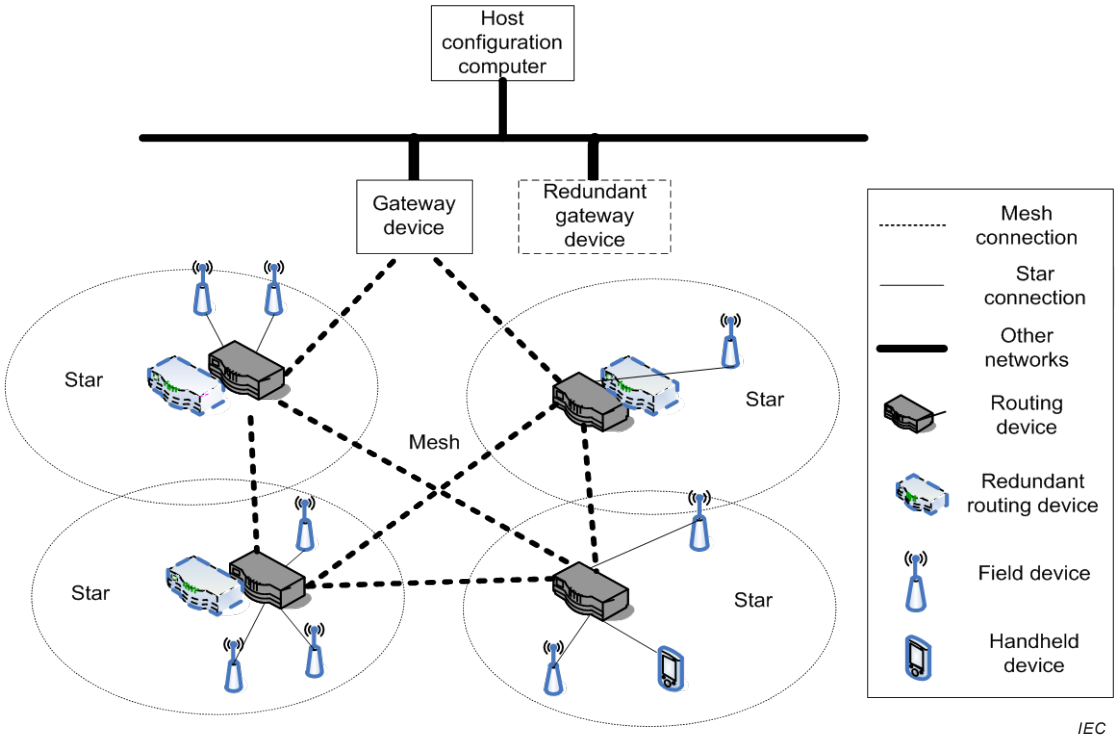

Figure 1 – Example of WIA-PA physical topology (combination of star and mesh)

The star-only network is comprised of a gateway device, field devices, and handheld devices (if they exist). The star network topology is illustrated in Figure 2.

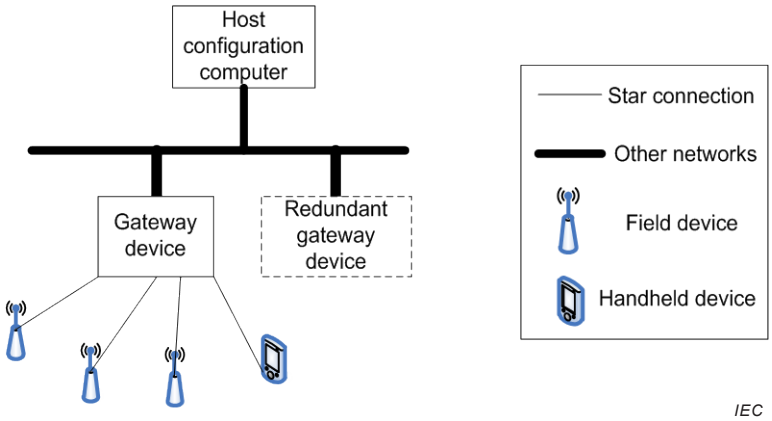

Figure 2 – Example of WIA-PA physical topology (star-only)

The star-only network is a special case of the hierarchical network. In this document, the specification of the hierarchical network covers the star-only network. Therefore, there is no explicit specification provided for the star-only network.

In order to facilitate management, this document specifies five kinds of logical roles.

a) Gateway

Gateway handles protocol-translation and data-mapping between the WIA-PA network and other networks.

b) Network manager

Network Manager (NM) manages and monitors the entire network (see 6.2). There should be one and only one network manager per WIA-PA network.

c) Security manager

Security Manager (SM) deals with security key management and security authentication of gateway devices, routing devices, field devices, and handheld devices (if they exist).

d) Cluster head

Cluster head manages and monitors field devices and handheld devices (if they exist). Cluster head also merges and securely forwards packets from local cluster members and other cluster heads.

e) Cluster member

Cluster member collects field data and sends the data to its cluster head.

The NM and the SM that are used for system management should reside in a gateway device.

One physical device may perform the functions of several logical roles. In the hierarchical network that combines star and mesh, a gateway device may perform the logical roles of gateway, NM, SM, and cluster head. A routing device should act as a cluster head. A field device/handheld device should only act as a cluster member.

The primary device and the redundant device are marked by the parameter RedundantDevFlag in Table 20.

5.3 Protocol architecture

The WIA-PA protocol architecture, which is illustrated in Figure 3, is based on ISO/IEC 7498-1. The WIA-PA protocol architecture defines the Data Link Sub-Layer (DLSL), Network Layer (NL) and Application Layer (AL), while its PHYsical layer (PHY) and Medium Access Control sub-layer (MAC) are based on IEEE Std 802.15.4-2011.

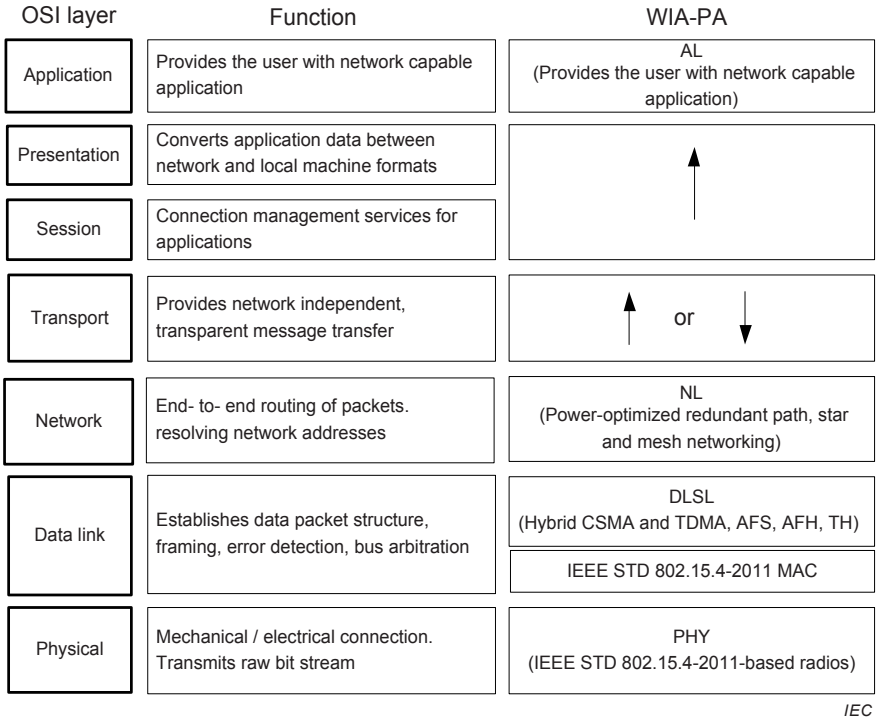

Figure 3 – OSI basic reference model mapped to WIA-PA

5.4 Interconnection

The WIA-PA network interconnects with other networks through the WIA-PA gateway. Besides the communication to the WIA-PA NM and SM, the WIA-PA gateway may communicate with other WIA-PA devices in order to exchange information between devices. Meanwhile, the WIA-PA gateway may connect other networks, such as wired fieldbuses. The architecture of the WIA-PA gateway is shown in Figure 4.

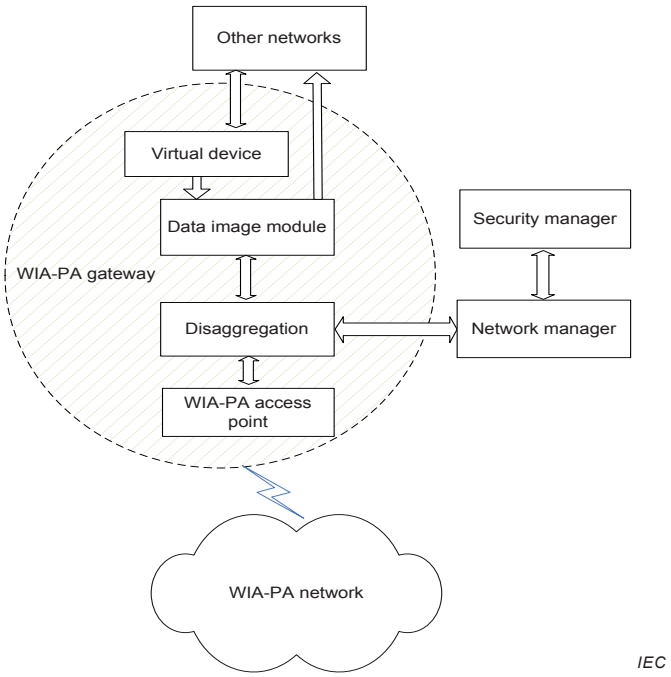

Figure 4 – The architecture of WIA-PA gateway

The WIA-PA gateway includes the following components.

a) WIA-PA access point

The WIA-PA access point realizes the physical connections of the WIA-PA network and transmission of the management information and data.

b) Virtual device

The virtual device defines a communication interface for other networks. The interface is used to map a data source from other networks into a WIA-PA device in order to fulfil communication between the WIA-PA network and other networks.

c) Disaggregation object

This object is used to disaggregate the packets that are aggregated within routing devices and field devices.

d) Data image module

The data image module stores the data of devices in the WIA-PA network and provides an access interface for other networks.

The virtual device and the data image module are optional and implementation-specific, which are not defined in this document.

6 System management

6.1 General

The system management in the WIA-PA network includes both network management and security management. The functions of system management are implemented by the Device Management Application Process (DMAP) in each device. The DMAP, as a system management entity, includes the network management module, the security management module, and the Management Information Base (MIB) module. DMAP is shown as the grey area in Figure 5. The white blocks within the grey area are the function components in the DMAP.

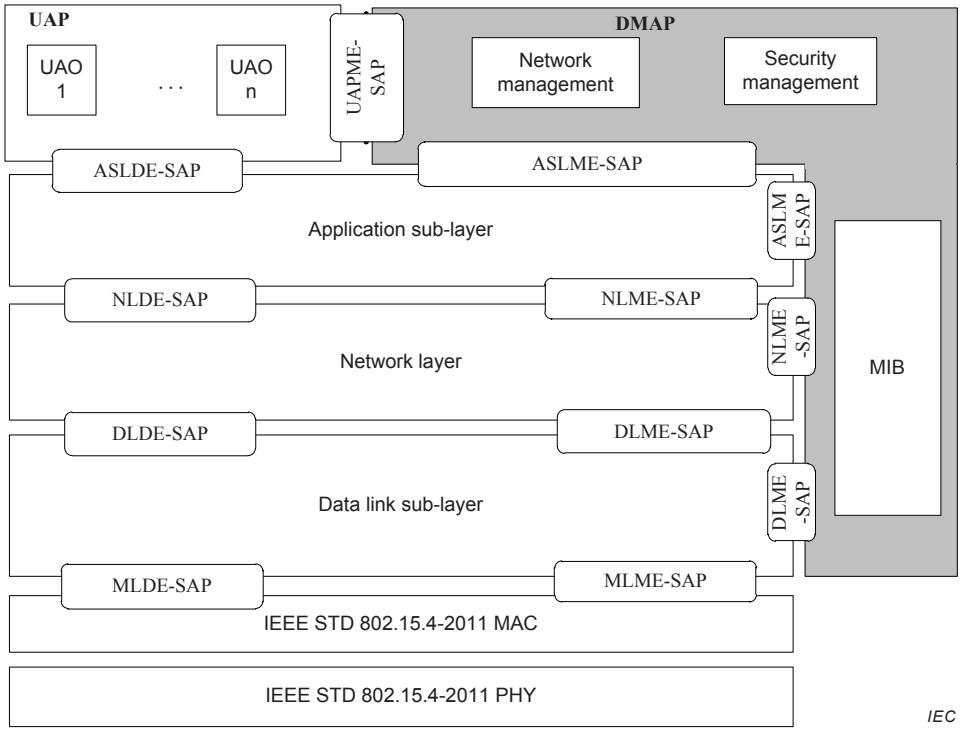

Figure 5 – DMAP in system management

The network management specifies the management of device attributes and attributes that are related to communication and network configuration. The network management functions are accomplished by the DMAPs in the NM, cluster heads, and cluster members. The network management functions include the following actions.

a) Joining and leaving the network

A routing device should join or leave the network through the gateway device or an online routing device. A field device should join or leave the network through an online routing device or the gateway device. The new joining device should be authenticated by the SM.

b) Network address allocation

Each device in the WIA-PA network has a 64-bit Extended Unique Identifier (EUI-64) address and a 16-bit network address. The EUI-64 address is also called long address. The long address of each device is assigned by vendors according to the IEEE EUI-64 address. The 16-bit network address is also called short address. The short address of each network device is assigned by the NM.

c) Routing configuration

Static routing is implemented in the WIA-PA network. The routing table in each routing device is configured by the NM.

d) Communication resource configuration

Communication resources of the WIA-PA network are organized as superframes (see 8.4.4). When a routing device has successfully joined the network, it should be allocated communication resources by the NM. After a field device has successfully joined a cluster, the cluster head should allocate communication resources to it.

e) Time source configuration and services of system time

The WIA-PA network sets only one base time source, which is performed by the gateway device. In order to recognize the order of occurring events, the WIA-PA devices relatively synchronize with the gateway device.

f) Performance monitoring

Performance monitoring is necessary to collect the performance of the WIA-PA network, which includes device status, path failure, and channel condition.

g) MIB maintenance

The MIB module should configure the MIB of the WIA-PA protocol stack.

h) Interfaces for plant operators and maintenance personnel

The NM should provide an interface to allow plant operators and maintenance personnel to monitor and control the performance/activities of the network and devices. The definition of this interface is out of the scope of the current WIA-PA standard.

i) Firmware upgrading

Firmware upgrading should update the protocol stack of WIA-PA devices. Because on-line firmware upgrading is a power-hungry operation, WIA-PA does not recommend this mode and does not specify it in this document. WIA-PA recommends off-line firmware upgrading through handheld devices. Handheld devices connect devices that are to be upgraded by wire or wireless connections. Wired connections may use the serial communication methods. Wireless connections are not specified in this document.

The security management should manage the attributes associated with network security. The security management is performed by the SM and by the security management modules in cluster heads and cluster members. The SM handles the centralized authentication of the whole network, and needs coordination with the NM. See 11.1 for the functions of security management.

The MIB module stores all attributes used in the WIA-PA network, including both structured attributes and unstructured attributes (see 6.9).

6.2 Framework of system management

The WIA-PA network supports the hybrid centralized and distributed management scheme.

The hybrid centralized and distributed management framework is illustrated in Figure 6. The system management is implemented by the NM, the SM, and cluster heads. Cluster heads are directly managed by the NM and the SM. In addition, they are provisioned with the privilege of managing cluster members.

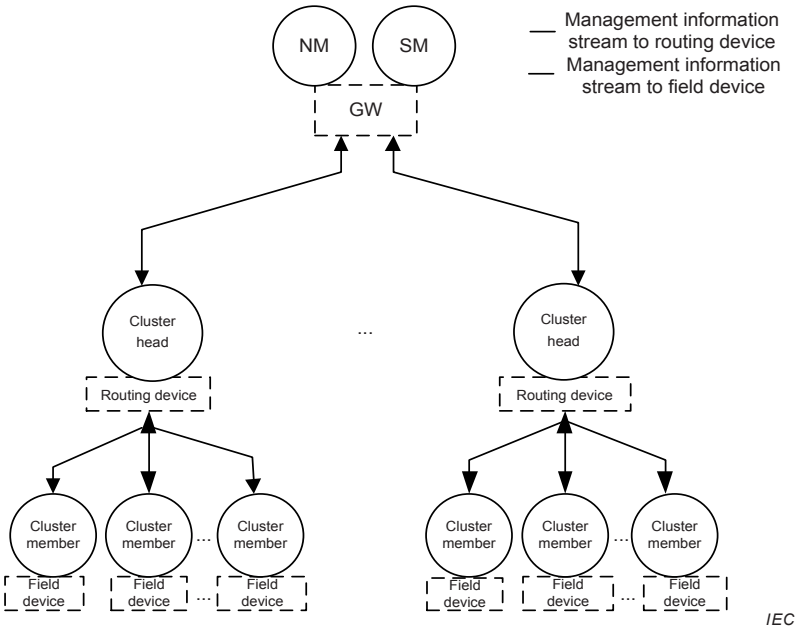

Figure 6 – Hybrid centralized and distributed system management scheme

The NM should perform the following tasks:

a) constructing and maintaining the mesh topology comprised of cluster heads, and star topology comprised of cluster heads and cluster members;

b) allocating communication resources for cluster heads in mesh topology, and allocating communication resources for cluster members to the cluster heads in star topology;

c) monitoring the performance of the WIA-PA network, including device status, path failure, and channel condition.

The SM should perform the following tasks:

a) authorizing the cluster heads and cluster members that are attempting to join the WIA-PA network;

b) managing keys in the entire network, including key generation, key distribution, key recovery, and key revocation;

c) authorizing the relationship of the end-to-end communication.

The cluster head should perform the following tasks:

a) constructing and maintaining the star topology; allocating communication resources that are allocated to the star topology by the NM to cluster members in the cluster; providing the monitoring results of the star topology to the NM;

b) storing and forwarding keys used in the star topology; authorizing the communication relationship among cluster heads; authorizing the communication relationship between a cluster head and cluster members.

6.3 Joining process

6.3.1 Provisioning process

When a WIA-PA device intends to join the network, it should be provisioned by a handheld device via wired manner.

A new device should have 64-bit long address before joining the WIA-PA network. First, the CountryCode value (see 6.9.1.2.1 and 8.4.11) of a new device shall be set, which reflects the regulatory constrains in the region where it might be deployed. The NetworkTopology value

(see 6.9.1.2.1) shall be set, which indicates the type of the network topology. The SecEnableflag value (see 6.9.1.2.1) shall be set, which indicates whether the authentication is enabled over the network. If authentication is enabled, KJ shall also be set (see 11.2).

To join the WIA-PA network, a new routing device or field device should first listen to beacons.

6.3.2 Joining process of routing device

When a routing device intends to join the network, it should be provisioned a Join Key by a handheld device. See Clause 11 for the detailed provisioning process.

The joining process of a routing device includes the following actions.

a) A routing device scans the available channels to get beacons from either the online routing devices or the gateway device.
b) The routing device chooses an online routing device or the gateway device as the temporal parent and synchronizes with the network according to the received beacon.
c) The routing device sends a joining request to its temporal parent, which will then forward the request to the NM.
d) When receiving a joining request, the NM should communicate with the SM to complete the authentication process. Then the NM returns a joining response.
e) The routing device receives the joining response relayed by its temporal parent. If the response is negative, the routing device should restart this joining process; otherwise, the joining process should be finished.

The joining process of a routing device through the gateway device is shown in Figure 7.

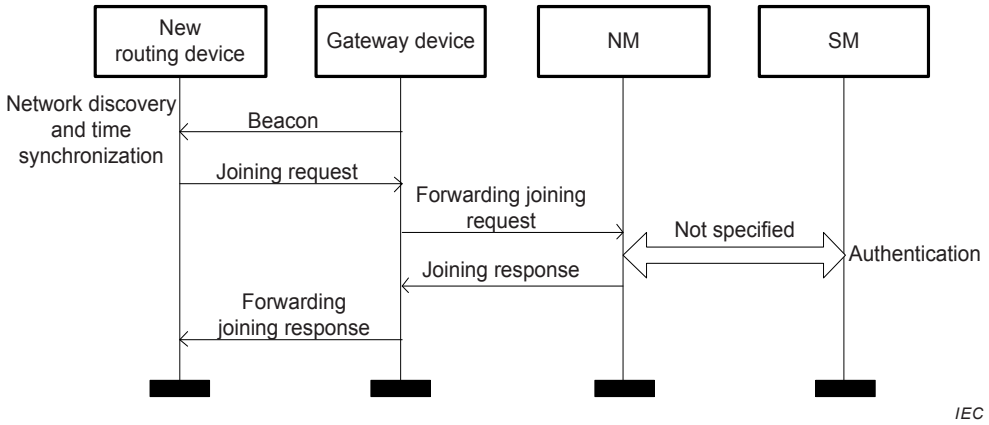

Figure 7 – Joining process of routing device through the gateway device

The joining process of a routing device through an online routing device is shown in Figure 8.

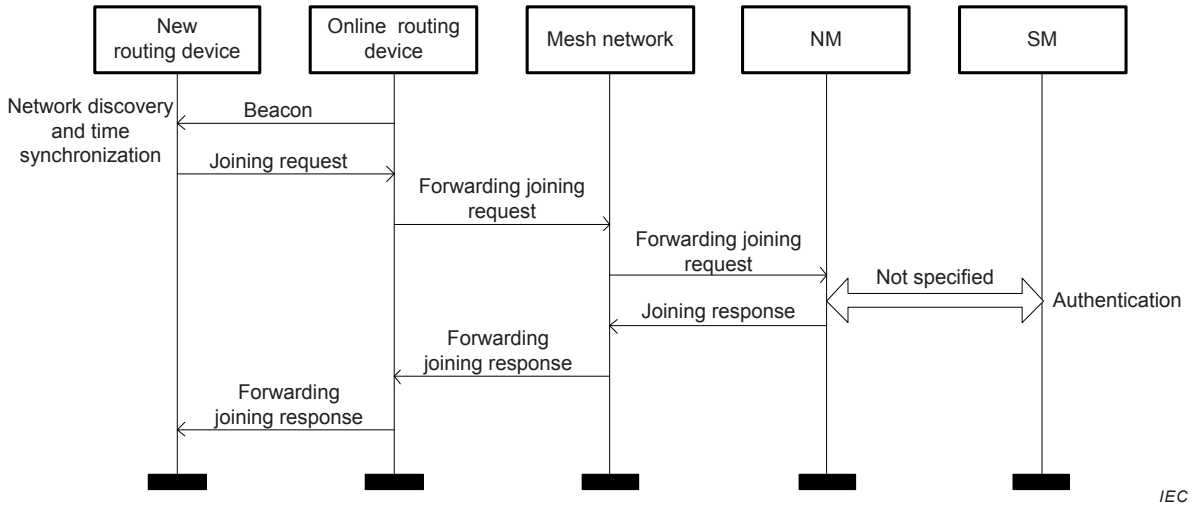

Figure 8 – Joining process of routing device through an online routing device

See 9.5.8 for the allocation of communication resources.

6.3.3 Joining process of field device

The joining process of a field device includes the following actions.

a) A field device scans the available channels to get beacons from the online routing devices or the gateway device.

b) The field device chooses one online routing device or the gateway device as the cluster head and synchronizes with the network according to the received beacon.

c) The field device sends the joining request to the cluster head.

d) When receiving the joining request, the cluster head forwards it to the NM.

e) After receiving the joining response from the NM, the cluster head returns the joining response to the field device according to the type of network topology and the available communication resources of the cluster head. If the network topology is mismatched, the status in the response is set to FAILURE_TOP_DISMATCH; if there are other types of failures, the status in the response is set to FAILURE_ELSE; if the joining is successful, the status in the response is set to SUCCESS. See 8.6.3.4 for definition of Status.

f) The field device receives the joining response from the cluster head. If the status in the response is FAILURE_TOP_DISMATCH, the field device will choose another routing device as its cluster head and restart the joining process as above; if status in the response is FAILURE_ELSE, the field device will restart this joining process; otherwise, if status in the response is SUCCESS, the field device has joined in the network.

The joining processes of a field device through the gateway device and through a routing device are shown in Figure 9 and Figure 10 respectively. See 9.5.8 for the allocation of communication resources.

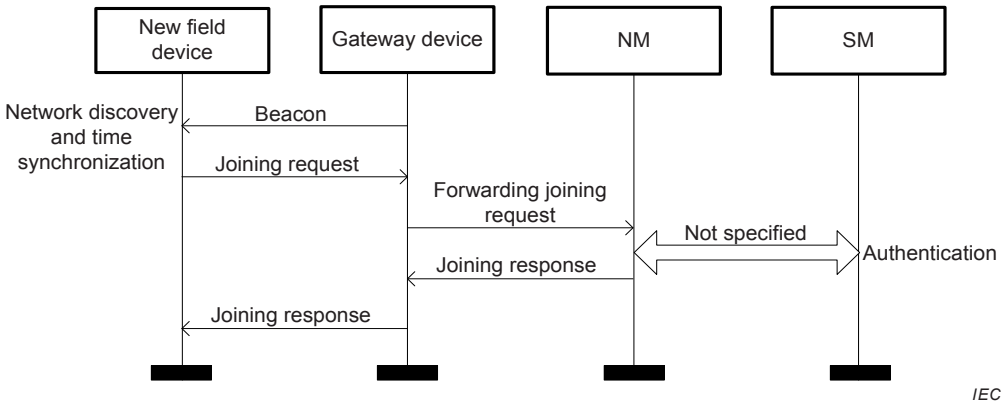

Figure 9 – Joining process of field device through a gateway device

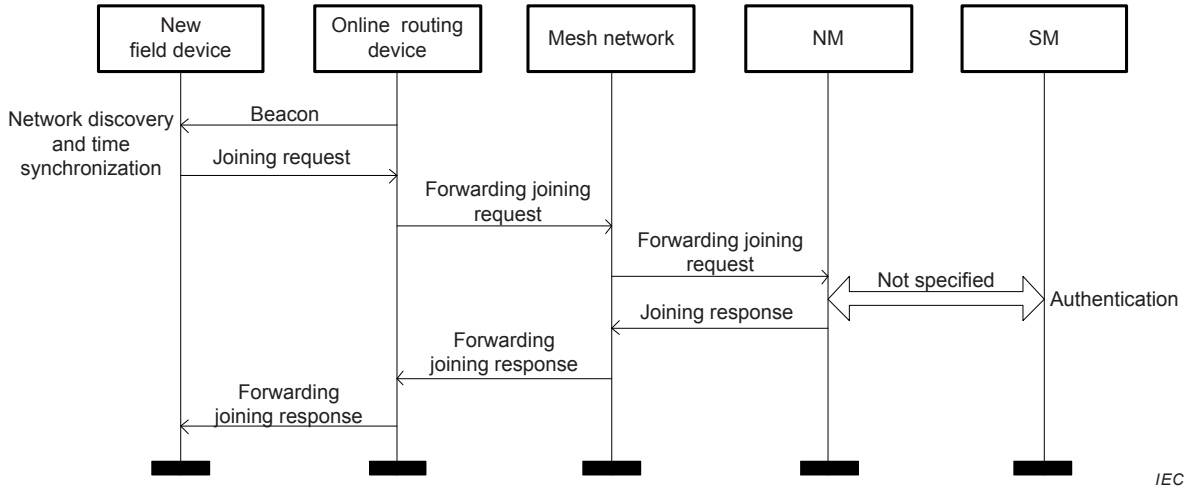

Figure 10 – Joining process of field device through a routing device

A WIA-PA handheld device connects to the WIA-PA network via the gateway device or a routing device. The joining process of the handheld device is the same as other WIA-PA devices. After joining in the network, the WIA-PA handheld device configures the WIA-PA devices and collects the network performance/health information during the Contention Access Period (CAP) period of the WIA-PA superframe.

See Clause 11 for detailed information about security.

6.3.4 Addressing and address assignment

In the WIA-PA network, each WIA-PA device (GW, routing device or field device) has a global unique 64–bit long address and a 16-bit short address. The long address, as shown in Figure 11, is set by manufacturers according to the IEEE EUI-64 address.

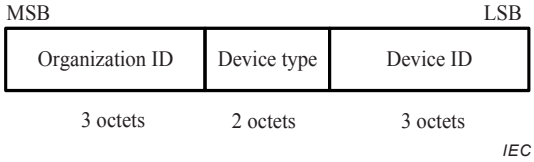

Figure 11 – Long address structure of device

The short address is 16-bit long. The most significant 8 bits of the short address that are assigned by the NM are used to identify different clusters. If the value of the parameter NetworkTopology (see 6.9.1.2.1) is 1, that is the network topology is star-only, the address of the gateway device is also the address of the cluster head.

The short address of GW is allocated by the NM. The NM resides in a gateway device (see 5.2). The address allocation process of the GW is an internal operation, which is not specified in this specification.

The short address of a routing device is shown in Figure 12. For the short address of the routing device, the least significant 8 bits are set to 0.

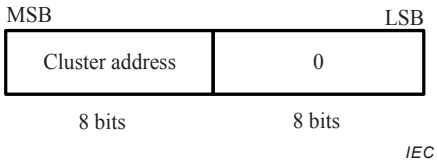

Figure 12 – Short address structure of routing device

The short address of a field device is shown in Figure 13. The least significant 8-bit part of the field device's short address is the intra-cluster address (in the cluster).

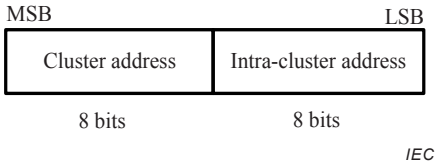

Figure 13 – Short address structure of field device

6.4 Virtual Communication Relationship (VCR)

6.4.1 Definition

Access to the User Application Objects (UAOs) is defined by the VCRs and the endpoints of a VCR are two UAOs. VCRs distinguish the routing and communication resources among different UAOs. Each VCR is identified by a VCR Identifier (VCR_ID). The attributes of a VCR include UAO ID at source side, destination UAO ID, addresses of source device/destination device, VCR type, valid scope, etc (see Table 2).

VCRs are classified into three types according to the application modes:

a) Publisher/Subscriber(P/S) VCR, which is used to publish periodic data;

b) Report source /Sink(R/S) VCR, which is used to support aperiodic events and trend reports;

c) Client/Server(C/S) VCR, which is used to transfer aperiodic and dynamic paired unicast messages.

VCRs are classified into three types according to its aggregation functions:

a) Non-aggregation VCR

   If the aggregation is not supported by a cluster member, the non-aggregation VCR is used by the cluster member to send non-aggregation packets to its cluster head. If the cluster head does not support the aggregation function, it will forward the non-aggregation packets to the gateway through the non-aggregation VCR.

b) Data aggregation VCR

If the aggregation is supported by a cluster member, the data aggregation VCR is used by the cluster member to send the aggregated data from the UAOs to its clusterhead. If the cluster head does not support the aggregation function, it will forward the aggregated data to the gateway through the data aggregation VCR. The effective interval of the data aggregation VCR originates from the Data Aggregation Object (DAGO) of the cluster head to the DisaGgregation Object (DGO) of the gateway.

c) Packet aggregation VCR

If the aggregation is supported by a cluster head, the packet aggregation VCR is used by the cluster head to send aggregated packets to the gateway. The effective interval of the packet aggregation VCR originates from the Packet Aggregation Object (PAGO) of the cluster head to the DisaGgregation Object (DGO) of the gateway.

PAGOs, DGOs, and DAGOs (see 6.6.1) are three special UAOs.

6.4.2 Protocol support for VCR

The protocol supports for VCRs are listed in Table 2.

Table 2 – Protocol support for VCR

Protocol layer support			VCR types		
			P/S VCR	R/SVCR	C/S VCR
AL	User applications		Periodic data transmission of UAO	Aperiodic report e.g. alarm or event	Attribute getting and setting operation in UAO
	Communication modes in Application Sub-Layer (ASL)		P/S	R/S	C/S
	Bidirectional		No	No	Yes
	End-to-end retransmission		No	Optional	Yes
	Packet aggregation supporting		Yes	No	No
NL	Redundant path supporting	Intra-cluster	No	No	No
		Inter-cluster	Yes	Yes	Yes
	Unicast/Broadcast		Broadcast/Unicast	Broadcast/Unicast	Unicast
DLSL	Periodic		Yes	No	No
	Real-time		Yes	Optional	No
	Communication resource	Intra-cluster	Contention Free Period (CFP), exclusive timeslots in intra-cluster communication period	CAP	CAP
		Inter-cluster	Exclusive timeslots in inter-cluster communication period	Shared timeslots	Shared timeslots

6.4.3 VCR establishment

The processes of R/S VCR and C/S VCR establishment are as follows.

a) The NM obtains the UAOs of the field devices when they have joined the WIA-PA network. The UAOs are obtained by the NLME-INFO_GET primitives during the CAP (see 9.5.13 and 9.6.4).

b) Once the NM obtains the UAOs, it allocates reserved R/S VCRs and C/S VCRs for each UAO to field devices.

The process of P/S VCR establishment (if it is needed) is as follows.

a) The NM establishes one P/S VCR for each UAO by using the NLME-INFO_SET primitives (see 9.5.14 and 9.6.4). If a field device does not support the data aggregation, each UAO has one non-aggregation P/S VCR, and different VCRs may use the same routing paths. Otherwise, each field device has one data aggregation VCR. If the routing device supports packet aggregation, one packet aggregation VCR is established for the routing device.

See Table 21 for the VCR information.

6.4.4 VCR release

VCRs are released when routing devices and field devices leave the network (see 6.8).

6.5 Routing configuration and communication resource allocation

6.5.1 Routing configuration

Routing devices use static routing that is configured by the NM. The concrete routing algorithms are not specified in this standard. The details of routing paths are shown in Table 15. The redundant paths are supported and use the same RouteID. The routing configuration is realized by route allocation services (see 9.5.7).

6.5.2 Framework of communication resource allocation

Communication resources consist of timeslots and channels. Allocation of communication resources should be considered in two dimensions: time and channel.

The process of communication resource allocation is as follows.

a) In the mesh network, the communication resources of the cluster heads are allocated by the NM. The communication resources consist of those used by cluster heads in the mesh network and those allocated by cluster heads to cluster members.

b) In the star network, the communication resources are allocated by cluster heads to cluster members. That is, communication resources are bound to cluster members.

An example of the resource allocation in the hierarchical network topology that combines star and mesh is shown in Figure 14. First, the NM residing in the GW allocates communication resources to the routing devices R1, R2 and R3. These communication resources are used for communication among R1, R2 and R3, for communication between routing devices and the GW, and for communication between intra-cluster field devices and routing devices. Second, after receiving communication resources from the NM, routing devices allocate some of the communication resources to their intra-cluster field devices, which are used for intra-cluster communications among field devices and their corresponding routing devices. As shown in Figure 14, after the NM allocates communication resources for R1, R2 and R3, R1 allocates communication resources for F1 and F2, R2 allocates communication resources for F5, and R3 allocates communication resources for F3 and F4.

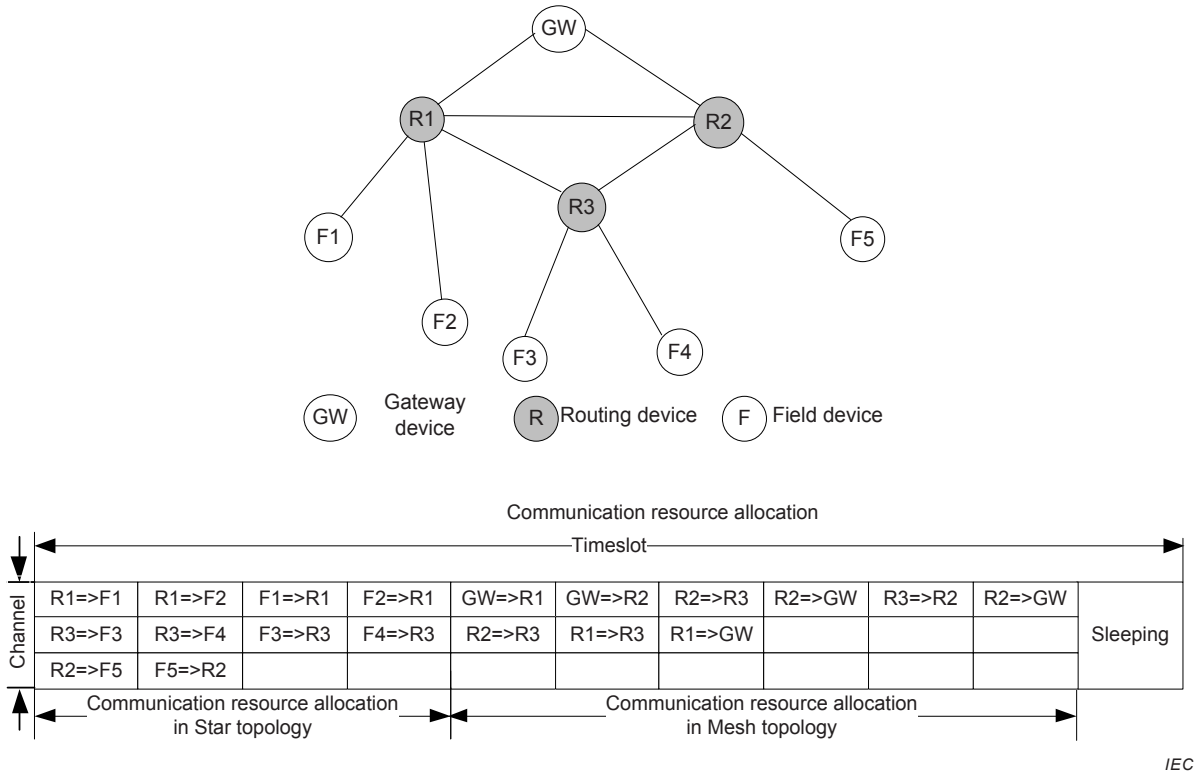

Figure 14 – An example of resource allocation

6.5.3 DLPDU priority and scheduling rules

Four priority levels of DLPDUs are defined.

a) Command (highest priority)

   Any packet containing a payload with network-related diagnostics, configuration, control information, or emergent alarms should be classified with a priority of Command.

b) Process data (secondary priority)

   Any packet containing process data should be classified as secondary priority level, Process Data.

c) Normal (third priority)

   DLPDUs that do not meet the criteria of Command, Process data, or Alarm should be classified as Normal priority.

d) Alarm (lowest priority)

   Packets containing only non-emergent alarm and event payload should assume a priority of Alarm. Devices should buffer no more than one DLPDU having Alarm priority.

The following scheduling rules are recommended in this order for allocating communication resources:

a) allocating channels to the beacon frame and active period;

b) allocating timeslots to the devices with the fastest update rate;

c) allocating resources to the packet with the earliest generating time in multi-hop situations;

d) allocating resources to the highest priority packet prior to other packets.

6.5.4 Communication resource allocation to routing device

The WIA-PA network applies the integrated management strategy of centralized and distributed schemes. After the routing device joins the network, it should scan its neighbour

devices on each channel and report the neighbour information to the NM. After receiving the neighbour information, the NM should allocate paths, a superframe, and a block of links to the routing device by using the communication resource allocation services (see 9.5.8). The information of every routing device superframe is broadcast to its field devices in the beacon frame. The allocation process of routing device communication resources is illustrated in Figure 15.

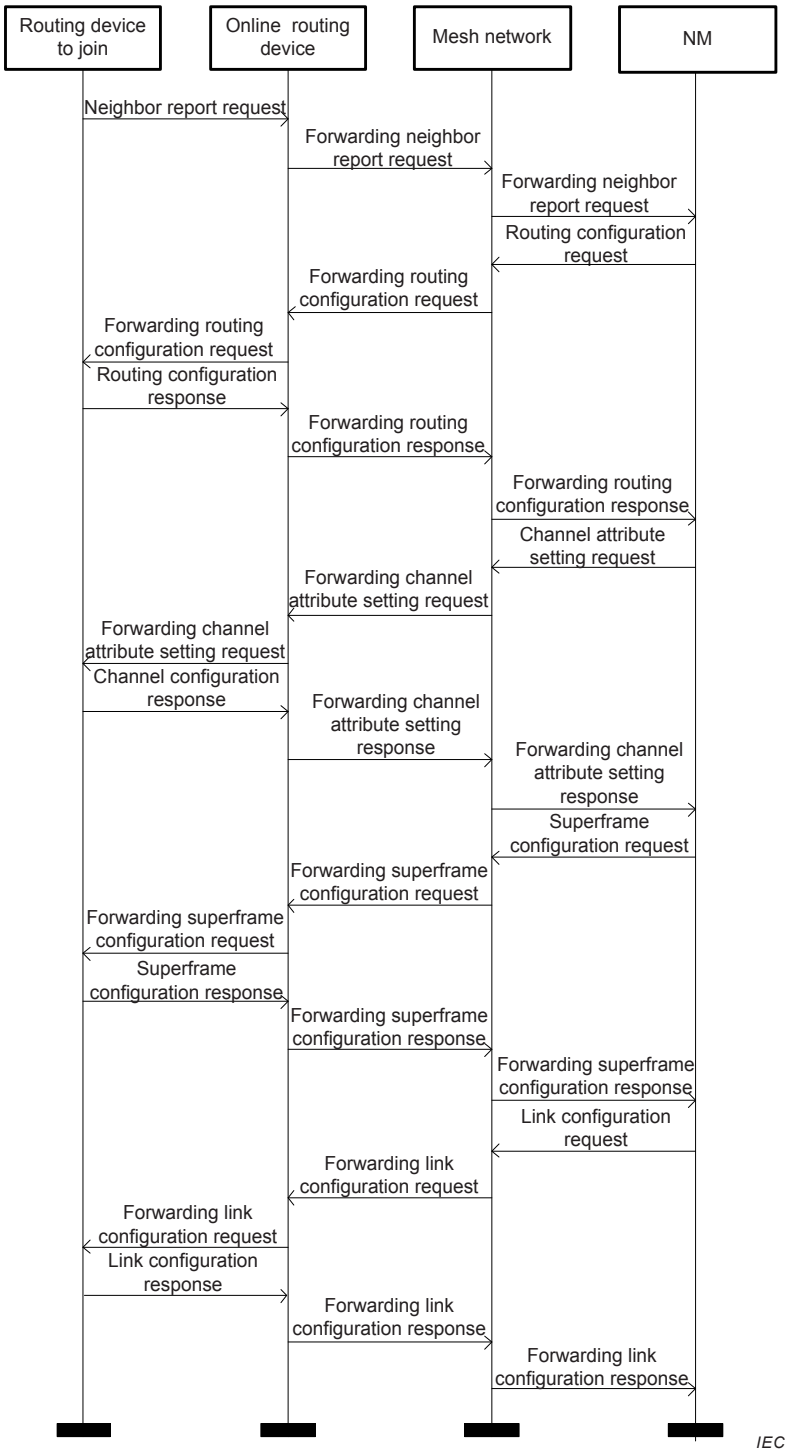

Figure 15 – Allocation process of routing device's communication resources

6.5.5 Communication resource allocation to field device

After VCRs of a field device are established successfully, either the routing device or the gateway device allocates timeslots to its field devices by using the communication resource allocation services (see 9.5.8), which are used in communication between the routing device and the field devices or between the gateway device and the field devices. If the superframes of the routing device and the gateway device are affected by the joining field devices, the routing device and the gateway device should update their RouteTable, Superframe and Link attributes (see 6.9.1.2.2 and 9.5.8).

When a field device joins the network through an online routing device, the communication resource allocation process of the field device is illustrated in Figure 16.

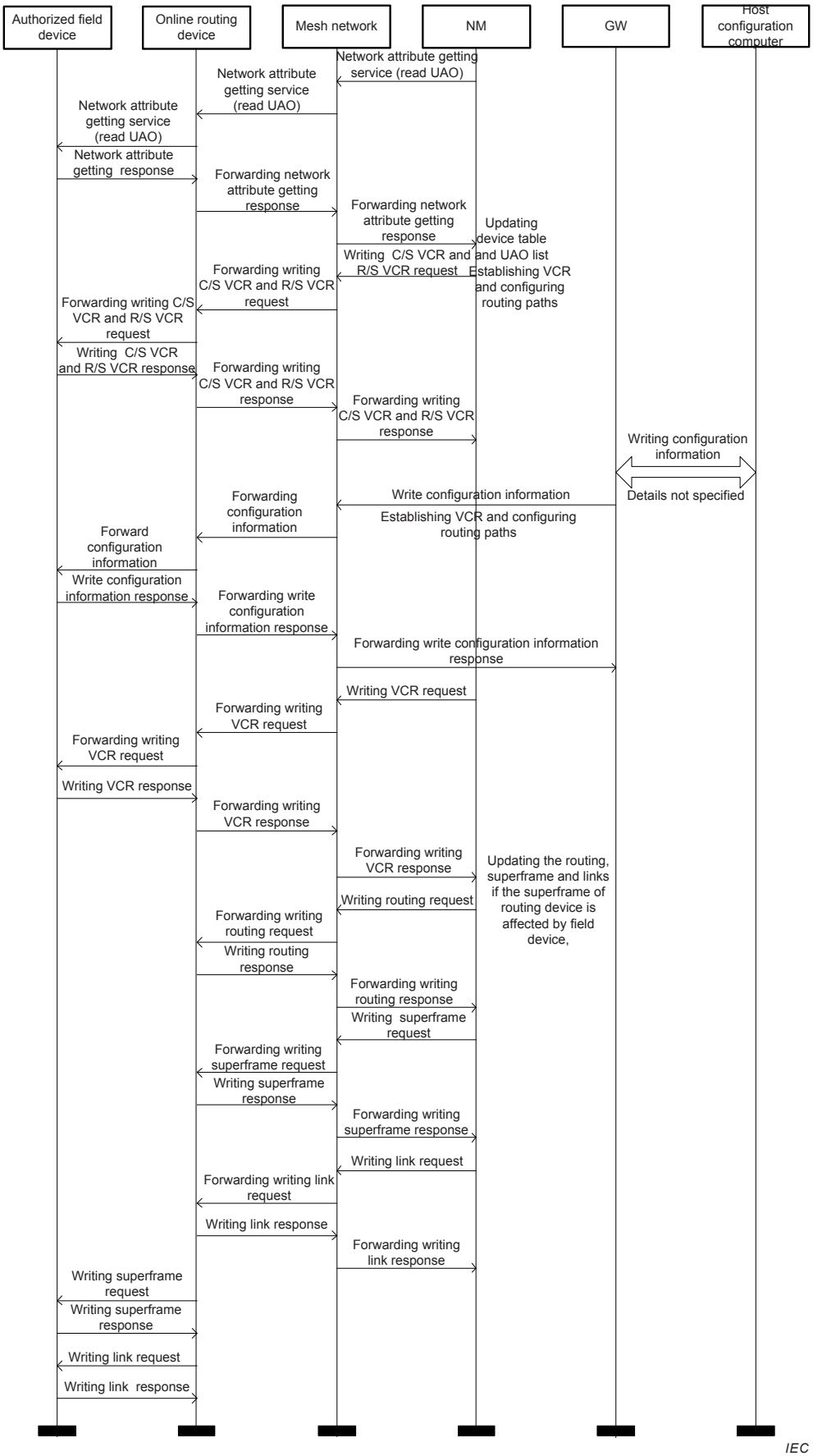

Figure 16 – Allocation process of field device's communication resources

6.6 Aggregation and disaggregation

6.6.1 Aggregation

The aggregation function is optional for field devices and routing devices and is indicated by the aggregation and disaggregation support flag AGGSupportFlag (see Table 20). If the field devices or the routing devices support the aggregation mechanism, the NM sets the aggregation and disaggregation enable flag AGGEnableFlag to 1 (see Table 20).

The WIA-PA network supports a two-level aggregation mechanism in order to reduce the number of packet transmissions.

a) Data aggregation. If a field device has more than one User Application Object (UAO), it should decide whether or not to invoke the data aggregation mechanism according to the AGGEnableFlag. This mechanism should reduce the data communication frequency and enhance network efficiency.
b) Packet aggregation. If a routing device receives packets from more than one field device, it should decide whether invoking packet aggregation mechanism or not according to the AGGEnableFlag. This mechanism should reduce the number of packets from the routing device to the gateway device and enhance network efficiency.

The aggregation function is implemented by the DAGOs in field devices and PAGOs in routing devices. The operation parameters of DAGOs and PAGOs are configured by the NM.

The aggregation function supported by the WIA-PA network includes the following four situations.

1) A field device supports the aggregation function, while its routing device does not.
2) A routing device supports the aggregation function, while its field device does not.
3) Both, a field device and its routing device support the aggregation function.
4) Neither a field device nor its routing device supports the aggregation function.

Relations between the VCRs and the aggregation functions supported by field device, routing device and gateway device are shown in Table 3.

Table 3 – Relations between VCR and aggregation function

	Field device DAGO (Aggregation function)			**Routing device PAGO (Aggregation function)**			**Gateway device DGO (Disaggregation function)**	
	Support	Disable	Enable	Support	Disable	Enable	Disable	Enable
Non-aggregation VCR	NO	X		NO	X		—	—
Data aggregation VCR	YES		X	NO	X[a]			X
	YES		X	NO		X[b]		X
Packet aggregation VCR	NO	—[c]	—[c]	YES		X		X

X Indicates the aggregation/disaggregation function in the corresponding device is enabled or disabled.

— Indicates the VCR is independent of the aggregation/disaggregation function.

[a] The data aggregation VCR starts with DAGO of a field device and ends at DGO of the gateway device.

[b] The data aggregation VCR starts with DAGO of a field device and ends at PAGO of a routing device.

[c] Field devices have no packet aggregation function.

The aggregation configuration process is listed as follows.

i) NM should read the AGGSupportFlag to verify whether devices support the aggregation function.

ii) NM should set the value of the AGGEnableFlag attribute to 1 in the MIB if the routing devices support the aggregation function. The NM should allocate packet aggregation VCRs to the routing devices. The aggregation duration of a routing device should be indicated by attribute AggPeriod (see 6.9.1.2.1) and set to the minimum DataUpdateRate attribute (see Table 21) of its field devices in the cluster. When aggregation duration expires, the routing device aggregates all its packets and sends the aggregated packet out.

iii) NM should set the value of the AGGEnableFlag attribute to 1 in the MIB if the field devices support the aggregation function. The NM should allocate the data aggregation VCRs to the field devices. The aggregation duration of a field device should be indicated by attribute AggPeriod (see 6.9.1.2.1) and set to the minimum DataUpdateRate attribute (see Table 21) of its UAOs. When the aggregation duration expires, the field device aggregates all its packets and sends the aggregated packet to the routing device.

According to whether the aggregation is embedded or not, a field device or a routing device operates as follows.

a) If the AGGEnableFlag attribute of a field device is set to 0, this field device does not support the aggregation function. Field devices that have no aggregation function send their data to the routing device through the non-aggregation VCR and the related RouteID (see Table 20).If the aggregation flag AGGEnableFlag attribute of the routing device is also set to 0, the routing device should forward this aggregated packet to the DGO of the gateway device through the original RouteID.

b) If a field device has more than one UAO and its AGGEnableFlag attribute is set to 1, the DAGO of the field device calculates the length of the P/S data from the UAOs and aggregates these data. The format of the aggregated data is shown in Table 4. The aggregated data is sent to the ASL by the DAGO and is encapsulated with the ASL header. The field of the packet type in the ASL header is set to 0b11, which is used to indicate the aggregated packet (see 10.4.2). The aggregation packet in the ASL is sent to the routing device through the data aggregation VCR and the related RouteID; if the AGGEnableFlag attribute of the routing device is 0, the routing device should forward this aggregated packet to the DGO of the gateway device through the original RouteID.

c) If the aggregation function embedded in a routing device is enabled (AGGEnableFlag = 1) and receives packets from its field devices, it reads the NL packet headers. The routing device should aggregate the packet according to the P/S flags and the Fragment flag in the NL packet headers (see 9.6.3). The aggregation rules are as follows:

   – if the received packet is P/S type and the Fragment flag is 0, the packet should be aggregated;

   – other types should not be aggregated.

If the packet can be aggregated, its source address and NL payload are sent to the PAGO of the DMAP. The PAGO uses the time when the first packet comes (required to be aggregated) as the start time of an aggregation cycle PagoPeriod (see Table 10). Once PagoPeriod expires, the PAGO aggregates the packets according to the format that is shown in Table 5, and searches the packets related to the packet aggregation VCR. The aggregated packet, packet aggregation VCR, and the related packets are then sent to the NL; the NL should send this aggregated packet to the gateway device through the related packets. The format of the aggregated packet in NL is shown in 9.6.3. If the length of the aggregated packet is longer than the maximum length allowed by the NL, the NL should fragment the packet and send the packet to the gateway device by using the packet aggregation VCR and the related packets.

Table 4 – Format of aggregated data followed by field device's DAGO

	Format of aggregated data followed by field device's DAGO					
Length in octet(s)	1	1	Variable length	…	1	Variable length
Field name	Number of aggregated data	Data length	Data	…	Data length	Data

Table 5 – Format of aggregated packet followed by routing device's PAGO

	Format of aggregated data followed by routing device's PAGO							
Length in octet(s)	1	2	1	Variable length	…	2	1	Variable length
Field name	Number of aggregated packet	Source address	Data length	Data	…	Source address	Data length	Data

6.6.2 Disaggregation

The disaggregation function is optional for the gateway device and is indicated by the aggregation and disaggregation support flag AGGSupportFlag (see Table 20). If the gateway device supports the disaggregation mechanism, the NM sets the aggregation and disaggregation enable flag AGGEnableFlag to 1 (see Table 20).

The disaggregation function is implemented by the DGO in the gateway device. The gateway device decides whether to disaggregate the received packet according to the NL header and the ASL header. If the value of packet type in the NL header is 1, the gateway device disaggregates the NL aggregated packet; if the value of the packet type in the ASL header is 0b11, the gateway device continues disaggregating the ASL aggregated packet.

The operation parameters of the DGO are configured by the NM.

The process of disaggregation is as follows.

a) After the gateway device receives the aggregated packets from field devices and routing devices, it will notify the DGO in its DMAP to disaggregate the packets.

b) The DMAP sends the disaggregated packets to UAOs.

6.6.3 An example of the two level aggregation process

The following example illustrates the aggregation and disaggregation processes. The example is based on a network that consists of two field devices, one routing device, the GW, and the host configuration computer. As shown in Figure 17, the routing device R acts as the cluster head of field device A and field device B; there are two UAOs named a1 and a2 residing in field device A and two UAOs named b1 and b2 residing in field device B.

This example supposes that field device A does not support the aggregation function, while field device B and routing device R do. The data update rates of a1 and a2 are configured to 1 s, and the data update rates of b1 and b2 are configured to 4 s. The DAGO of the field device B is responsible for aggregating the packets from b1 and b2, and generates packet p_b. The routing device R has one PAGO and the aggregation duration is the minimum data update rates of a1, a2, b1, and b2, whose value is 1 s. The routing device R has a PAGO, which aggregates the packets from a1, a2 and packet p_b. The gateway device has a DGO, which disaggregates packets from field device A and B.

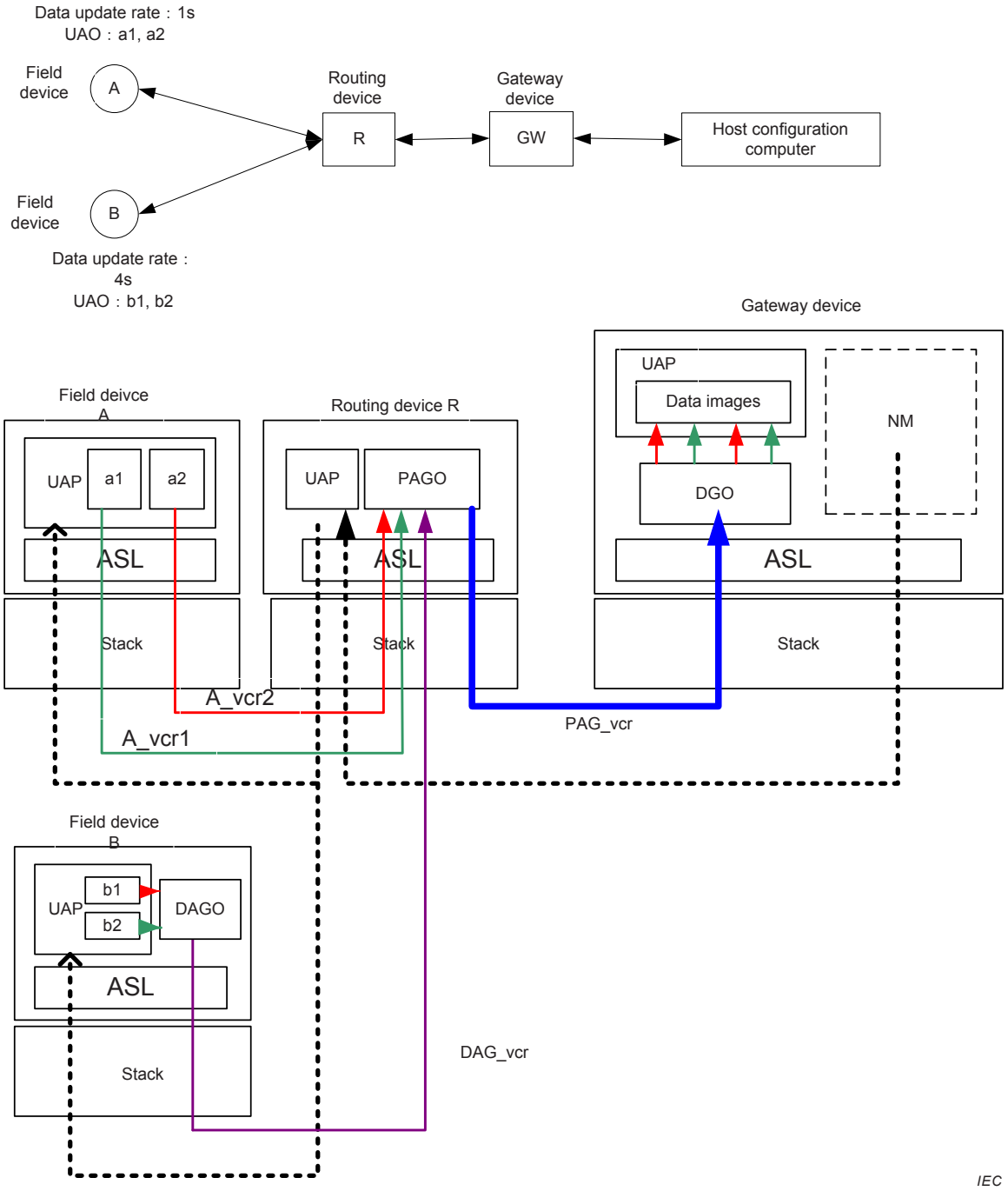

Figure 17 – Example of aggregation and disaggregation

The aggregation and disaggregation process is listed below.

a) After the network is established, the NM allocates the non-aggregation VCRs to a1, a2, b1, and b2 for the transmissions of non-aggregation packets. The non-aggregation VCRs of a1 and a2 in field device A are indicated by A_vcr1 and A_vcr2. The endpoints of A_vcr1 are a1 and the UAP of the GW, and the endpoints of A_vcr2 are a2 and the UAP of the GW. Because the data transmissions of field device B do not use the non-aggregation VCRs, the non-aggregation VCRs of b1 and b2 in field device B are not indicated in this example.

b) After constructing the non-aggregation VCRs, the NM configures the aggregation duration and data aggregation VCR for field device B and configures the aggregation duration and packet aggregation VCR for routing device R. The aggregation durations are the minimum data update rate among all UAOs in a field device or the minimum data update rate among

all field devices for a routing device. Because the data update rates of b1 and b2 are 4 s, the aggregation duration of DAGO in field device B is four seconds; the data aggregation VCR of field device B is indicated by DAG_vcr and its endpoints are DAGO in field device B and the DGO in the GW. The aggregation duration of routing device R is set to the minimum data update rate in a cluster. Because the minimum data update rate among a1, a2, b1, and b2 is one second, the aggregation duration of routing device R is set to 1 s. The packet aggregation VCR of routing device R is indicated by PAG_vcr and its endpoints are PAGO in the routing device R and DGO in the GW.

c) After A_vcr1, A_vcr2, DAG_vcr, and PAG_vcr are established, the DAGO, PAGO and DGO begin to work. The DAGO in field device B uses the time when the first packet comes (required to be aggregated) as the start time, and aggregates the packets and sends them to routing device R through DAG_vcr at the configured aggregation timeslot. The PAGO in routing device R uses the time when the first packet comes (required to be aggregated) as the start time, and aggregates the packets in DMAP and sends them to the DGO through PAG_vcr in the GW at the configured aggregation timeslot.

d) After the DGO of the GW receives the aggregated packets, it disaggregates the packets and finds the corresponding UAOs according to the packet source address and UAO identifiers.

e) The DMAP in GW sends the disaggregated packets to the corresponding UAOs.

6.6.4 Management of aggregation and disaggregation objects

The attributes of the DAGO class are shown in Table 6.

Table 6 – DAGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DagoID	Unsigned8	0	Mandatory	The identifier of the DAGO	READ
1	DagoRevision	Unsigned8	0	Optional	The version defined by the DAGO	READ

The attributes of the DAGO instance are shown in Table 7.

Table 7 – DAGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DagoInsID	Unsigned8	0	Mandatory	The identifier of the DAGO instance	READ
1	DagoPeriod	Unsigned8	1	Mandatory	The aggregation cycle (in seconds)	READ/WRITE
2	DagoMemNum	Unsigned8	1	Mandatory	Count of the aggregated UAOs	READ/WRITE
3	DagoMemLst	MEM_STRUCT structure (see Table 8)		Optional	List of the aggregated UAOs	READ/WRITE
4	DagoMemData Size	Unsigned8	0	Optional	Length of the aggregated data DagoMemData (in octets)	READ/WRITE
5	DagoMemData	Octetstring		Mandatory	Data of the aggregated UAOs	READ/WRITE

The methods supported by the DAGOs are shown in Table 171.

The definition of the MEM_STRUCT structure is shown in Table 8.

Table 8 – MEM_STRUCT structure

Attribute ID	Name	Data type	Description
0	AppObjID	Unsigned8	The identifier of the UAO
1	AppObjInsID	Unsigned8	The identifier of the UAO instance
2	AppObjAttrID	Unsigned8	The attribute identifier of the UAO instance

The attributes of the PAGO class are shown in Table 9.

Table 9 – PAGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	PagoID	Unsigned8	1	Mandatory	The identifier of the PAGO	READ
1	PagoRevision	Unsigned8	0	Optional	The version defined by the PAGO	READ

The attributes of the PAGO instance are shown in Table 10.

Table 10 – PAGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	PagoInsID	Unsigned8	0	Mandatory	The identifier of the aggregation object instance	READ
1	PagoPeriod	Unsigned8	1	Mandatory	The aggregation cycle (in seconds)	READ/WRITE
2	PagoMemNum	Unsigned8	1	Mandatory	Count of the aggregated cluster members	READ/WRITE
3	PagoMemLst	Octetstring		Optional	List of the aggregated cluster members	READ/WRITE
4	PagoDataSize	Unsigned8	0	Optional	Length of the aggregated packet (in octets)	READ/WRITE
5	PagoMemData	Octetstring		Mandatory	Data of the aggregated cluster members	READ/WRITE

The methods supported by the PAGOs are shown in Table 171.

The attributes of the DGO class are shown in Table 11.

Table 11 – DGO class attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DgoID	Unsigned8	2	Mandatory	The identifier of the DGO	READ
1	DgoRevision	Unsigned8	0	Optional	The revision defined by the DGO	READ

The attributes of the DGO instance are shown in Table 12.

Table 12 – DGO instance attributes

Attribute ID	Name	Data type	Default value	Usage	Description	Supporting method
0	DgoInsID	Unsigned8	0	Mandatory	The identifier of the DGO instance	READ

The methods supported by the DGOs are shown in Table 171.

6.7 Performance monitoring

6.7.1 Path failure report

The path failure report is used by routing devices to report path failure events to the NM through the redundant paths. The process of path failure report is shown in Figure 18 (see 9.5.12).

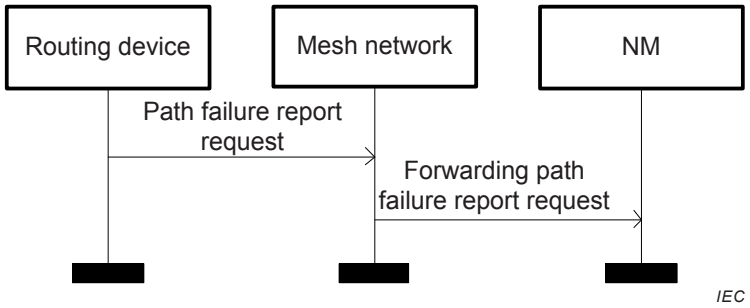

Figure 18 – Process of path failure report

6.7.2 Device status report

After receiving device status reports from intra-cluster field devices, a routing device reports its own health and the health of its field devices to the NM periodically. The NM appraises and diagnoses the network performance according to health information, and replies to the change of network environment timely. The NM should detect abnormal conditions in the WIA-PA device, such as low level of battery power and disconnection from neighbouring devices. This is realized by setting alarm levels to the WIA-PA devices. The device status report processes of the field device and the routing device are shownin Figure 19 and Figure 20 (see 9.5.10).

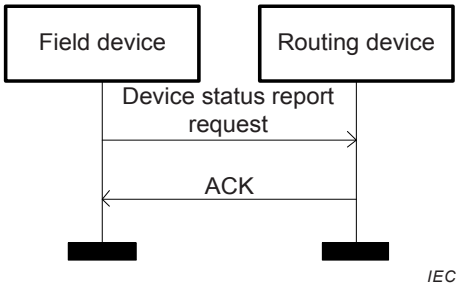

Figure 19 – Device status report process of field device

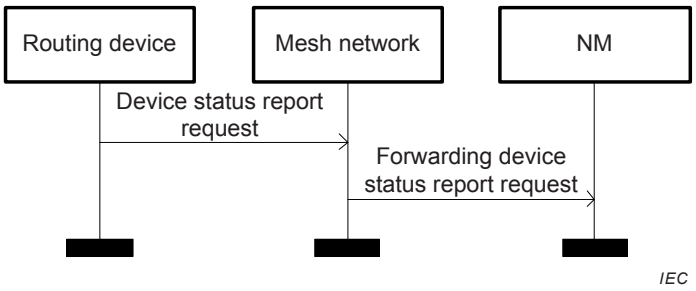

Figure 20 – Device status report process of routing device

6.7.3 Channel condition report

The channel condition report is used for the WIA-PA field devices or routing devices to report the channel condition remotely to the NM. The process of channel condition report is shown in Figure 21 (see 9.5.11).

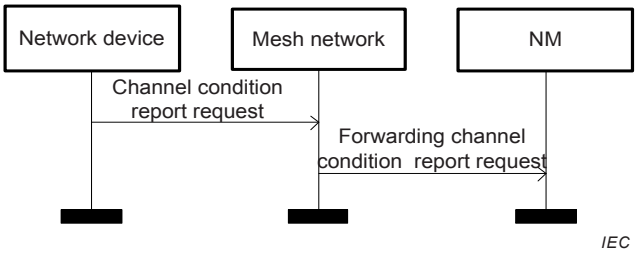

Figure 21 – Process of channel condition report

6.8 Leaving process

6.8.1 General

The leaving process of WIA-PA devices includes the abnormal leaving process, the active leaving process and the passive leaving process. The abnormal leaving process is caused by device failure, device invalidation, and energy depletion, which should be judged by the Keep-alive command frame of DLSL. The active leaving is an optional function and is enabled by setting ActiveLeavingEnable (see 6.9.1.2.1) to 1. The active leaving is requested by field devices from their routing devices or by routing devices from the gateway device. The passive leaving is requested by the gateway device from routing devices or by routing devices from their field devices.

6.8.2 Leaving process of routing device

The WIA-PA network specifies two leaving processes for routing devices: active leaving and passive leaving. The active leaving process of a routing device is as follows (shown in Figure 22).

a) A routing device sends the leaving request to the NM.

b) The NM gives a leaving response to the routing device.

c) After receiving a positive response, the routing device notifies the leaving to its field devices. After passive leaving of field devices, the routing device leaves the network.

d) The NM releases the network address, VCR and the communication resources of the departing routing device, and updates the network topology.

e) The NM notifies the routing devices that have communication relationships with the departed routing device to release the related communication resources.

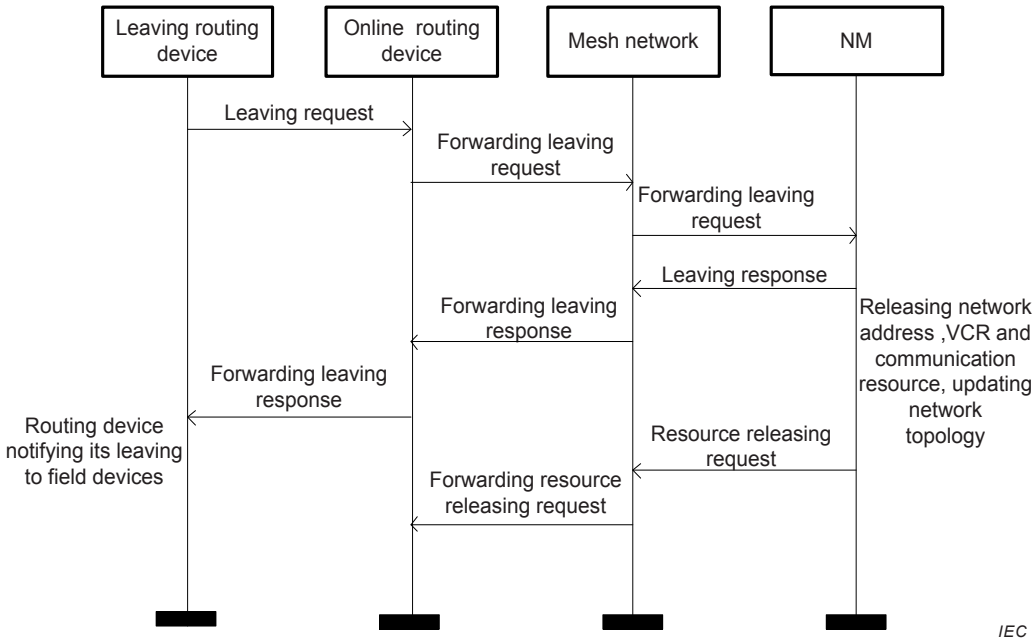

Figure 22 – Active leaving process of routing device

The passive leaving process of a routing device is as follows (shown in Figure 23).

a) The NM sends the leaving request to a routing device.

b) The routing device notifies its leaving to field devices.

c) After passive leaving of field devices, the routing device sends a leaving response to the NM.

d) The NM releases the network address, VCR and communication resources, and updates the network topology after receiving the leaving response from the routing device.

e) The NM notifies the routing devices that have communication relationships with the departed routing device to release the related communication resources.

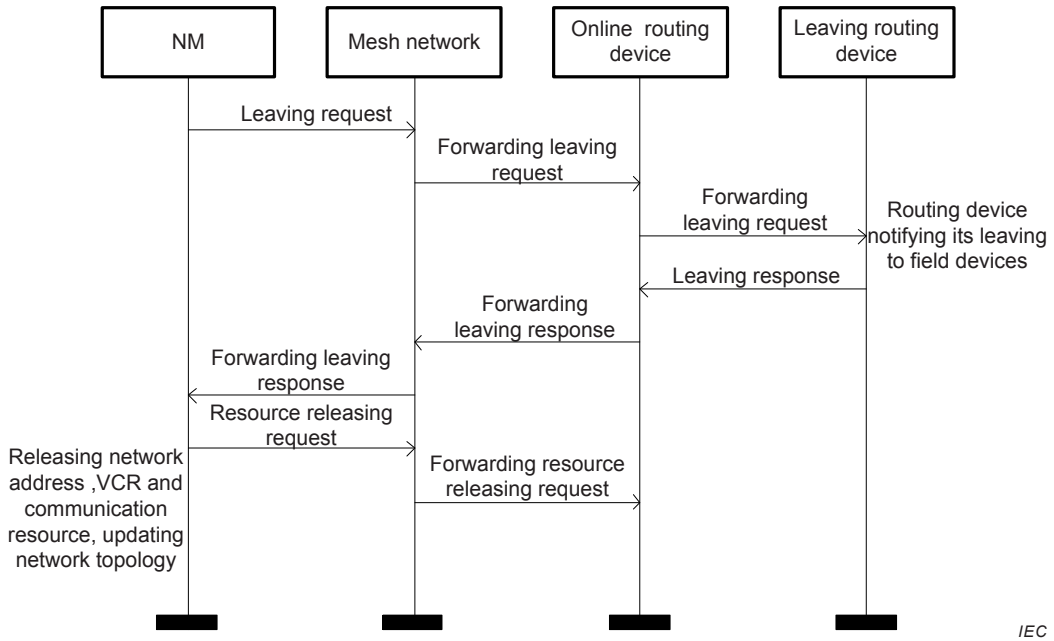

Figure 23 – Passive leaving process of routing device

6.8.3 Leaving process of field device

The WIA-PA network specifies two leaving processes for field devices: active leaving and passive leaving.

The active leaving process of a field device is as follows (shown in Figure 24 and Figure 25).

a) A field device sends the leaving request to the routing device or the gateway device in a cluster.

b) The routing device or the gateway device returns a leaving response to the leaving field device.

c) The routing device or the gateway device releases the intra-cluster network address, VCR, and communication resources and updates the cluster member list and UAO list of the field device that has left the network.

d) After receiving the response, the field device leaves the network.

e) If the field device leaves the routing device, the routing device reports the updated cluster member list and UAO list to the NM.

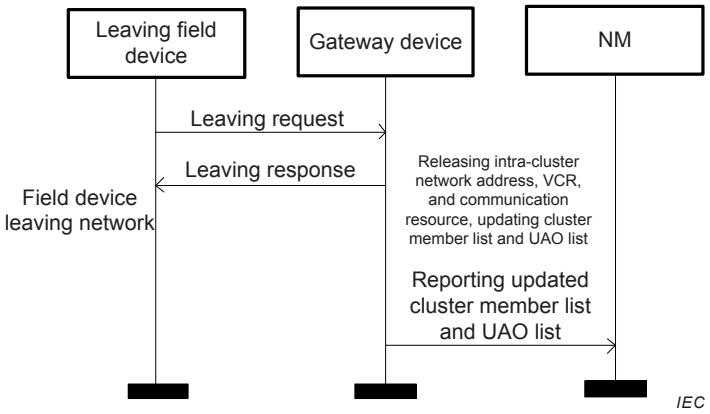

Figure 24 – Active leaving process of field device (leaving from gateway device)

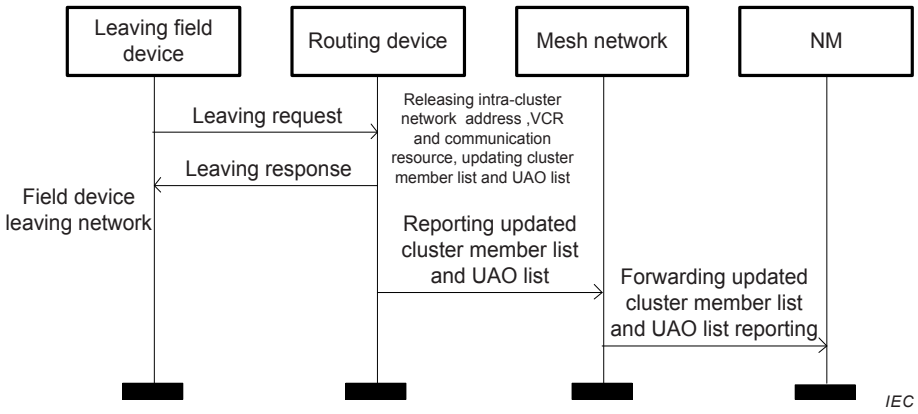

Figure 25 – Active leaving process of field device (leaving from routing device)

The passive leaving process of a field device is as follows (shown in Figure 26 and Figure 27).

a) The routing device or the gateway device sends the leaving request to a field device.

b) The field device returns the leaving response to the routing device or the gateway device.

c) The field device leaves the cluster.

d) The routing device or the gateway device releases the intra-cluster network address, VCR and communication resources of the departing field device and updates the cluster member list and UAO list.

e) If the field device leaves the routing device, the routing device reports the cluster member and its UAO list to the NM.

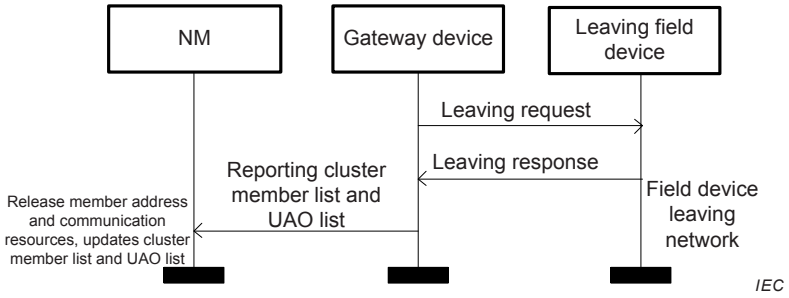

Figure 26 – Passive leaving process of field device (leaving from gateway device)

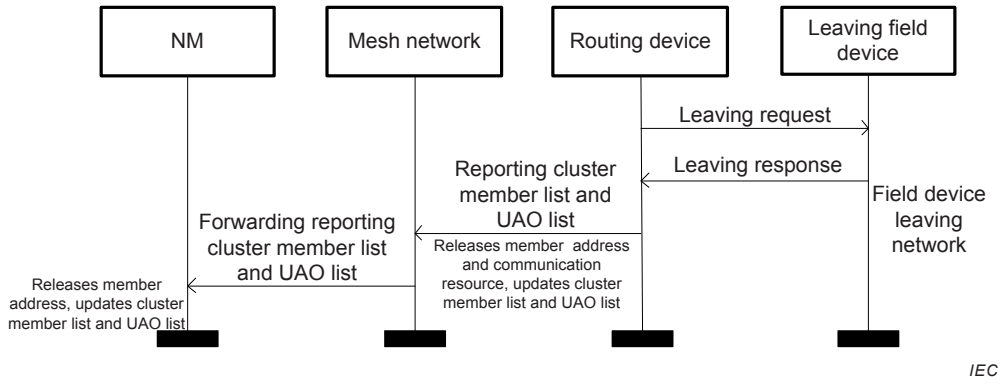

Figure 27 – Passive leaving process of field device (leaving from routing device)

6.9 Management information base and services

6.9.1 Management information base

6.9.1.1 General

Items stored in the MIB are called attributes and are used for monitoring and configuring the WIA-PA network parameters. These attributes can be accessed and updated by the NM.

According to the storage types, the attributes in the MIB are classified into three categories:

a) constant attribute,

b) static attribute, and

c) dynamic attribute.

A constant attribute, such as the serial number of a wireless device, is unchangeable with time. The constant attribute is set when devices leave manufacturers and should not be modified.

A static attribute, such as an alarm limit, changes its value infrequently. The value of the static attribute should be preserved after the warm restart/reset/ power-failure.

A dynamic attribute changes its value frequently without any external command. The value of the dynamic attribute will be lost after the warm restart/reset/power-failure. Warm restart is a sequence of operations that is performed to reset a previously running device after an

unintentional shutdown. When a device is reset, it enters into the idle state. Power-failure indicates that the available power that is used by devices is becoming critically low.

According to the attribute data types, the attributes in the MIB are divided into unstructured attributes and structured attributes.

According to the implementation requirements, the attributes in the MIB are divided into mandatory attributes and optional attributes.

There are two types of the access rights to the attributes in the MIB:

a) R (Read), which means that the values of the attribute may be read by other devices in WIA-PA network; and
b) W (Write), which means that the values of the attribute may be set by other devices in WIA-PA network.

6.9.1.2 MIB attributes

6.9.1.2.1 Unstructured attributes

The unstructured attributes are listed in Table 13. The values of attributes numbered 0 to 28 are identical in the entire network; and the values of attributes numbered 29 to 33 are device specific.

Table 13 – Unstructured attributes *(1 of 5)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
0	ProtocolVersion [de]	Unsigned16	0 to 65 535	R	Static	2013	Protocol version of current WIA-PA protocol stack, which is indicated by years.
1	NetworkID [e#]	Unsigned8	0 to 255	R/W	Static	0	Network identifier, used for multiple networks coexisting
2	StatisticsDuration[e*]	Unsigned16	0 to 65 535	R/ W	Static	0	Configuring the cycle of the statistic collection (in seconds). After this duration, WIA-PA devices update the statistic data in the neighbour tables. The value of 0 indicates that the device does not collect statistic data.
3	MaxNSDUSize[e*]	Unsigned8	0 to 255	R	Static	71	Maximum size of service data unit supported by the NL (in octets). MaxNSDUSize = aMaxPHYPacketSize -maximal length of MAC header – maximal length of DLSL header – maximal length of NL header. See IEEE STD 802.15.4-2011 Table 70 for aMaxPHYPacketSize.

Table 13 *(2 of 5)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
4	BitMap[ae*]	Unsigned32	32 bits	R/W	Dynamic	0x FFFF	Each bit indicates whether a channel can be used: 0 = Not used; 1 = Used. The value of bit i indicates the usage status of channel i. The value of the bit iindicates whether it is used according to IEEE STD 802.15.4-2011 channel number.
5	KeepAliveDuration[e*]	Unsigned24	0 to (2^{24}-1)	R/W	Static	1 000	Duration between two keep-alive frames (in milliseconds) The value of 0 is invalid.
6	TimeSynDuration[e*]	Unsigned24	0 to (2^{24}-1)	R/W	Static	25 000	Duration between two time synchronization frames (in milliseconds). The default value is calculated under the assumptions that the drift of crystal oscillator is 20 parts per million and the synchronization error is 1 ms. The value of 0 is invalid.
7	ChannelThreshold[e*]	Unsigned8	0 to (macMaxFrame-Retries－2)	R/W	Static	1	The channel switch threshold in adaptive frequency hopping, indicated as retry count; see IEEE STD 802.15.4-2011, Table 52 for macMaxFrameRetries
8	SecEnableFlag[e*]	Boolean	0, 1	R/W	Static	0	Security enable flag: 0 = the whole network needs authentication; 1 = the whole network does not need authentication.
9	KeyupdateDur[be*]	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	The updating cycle of key (in milliseconds). The value of 0 indicates that keys are not updated.
10	MaxAttackedCnt [be*]	Unsigned16	0 to 65 535	R/W	Static	5	The maximum count of attacks. The value of 0 indicates that this attribute is invalid.
11	MaxKeyAttacked-Cnt[be*]	Unsigned16	0 to 65 535	R/W	Static	1	The maximum count of key attacked.
12	MICFailerTimeUnit[be*]	Unsigned16	0 to 65 535	R/W	Static	60	The time interval inseconds used to count the MaxAttackedCnt.

Table 13 *(3 of 5)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
13	Alert[be*]	Unsigned16	0 to 65 535	R/W	Static	0	Bit 0 to 6 represent the security alert, the value 1 represents an alert: Bit0 = AttackedCount reaches the MaxAttackedCnt; Bit1 = KS is attacked; Bit2 = KJ is attacked; Bit3 = DLKEK is attacked; Bit4 = ALKEK is attacked; Bit5 = DLKED is attacked; Bit6 = ALKED is attacked.
14	EtoEACKTimeOut[e*]	Unsigned24	0 to (2^{24}-1)	R/W	Static	100*macMaxFrameRetries	The upper limit of waiting time for end to end acknowledge (in milliseconds). See IEEE STD 802.15.4-2011, Table 52 for macMaxFrameRetries.
15	TimeSlotDuration[e*]	Unsigned16	0 to 65 535	R/W	Static	N/A	Indicating the timeslot duration. The timeslot duration is calculated by: (aBaseSlotDuration × $2^{\text{TimeSlotDuratioin}}$). See IEEE STD 802.15.4-2011, Table 51 for aBaseSlotDuration.
16	PLRThreshold[e*]	Unsigned8	0 to 100	R/W	Static	50	The threshold of packet loss rate, which is PLRThreshold divided by 100 and is used for adaptive frequency switch. See 8.4.5 for adaptive frequency switch.
17	CmemRptCycle[ce*]	Unsigned16	0 to 65 535	R/W	Static	Maximal superframe cycle of the whole WIA-PA network	Configuring the reporting cycle of cluster member information (in seconds) The value of 0 is invalid.
18	ChaStaRptCycle[e*]	Unsigned16	0 to 65 535	R/W	Static	Maximal superframe cycle of the whole WIA-PA network	Configuring the reporting cycle of channel condition information (in seconds) The value of 0 is invalid.
19	DevStaRptCycle[e*]	Unsigned16	0 to 65 535	R/W	Static	Maximal superframe cycle of the whole WIA-PA network	Configuring the reporting cycle of device status information (in seconds) The value of 0 is invalid.

Table 13 *(4 of 5)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
20	PathFailRptCycle[ce*]	Unsigned16	0 to 65 535	R/W	Static	Maximal superframe cycle of the whole WIA-PA network	Configuring the reporting cycle of failure path information (in seconds) The value of 0 is invalid.
21	MaxEtoERetry[e*]	Unsigned8	0 to 255	R/W	Static	1	The maximum number of end-to-end retry
22	NetworkTopology[e*]	Boolean	0, 1	R/W	Static	1	Indicating the network topology: 0 = hierarchical network that is the combination of star and mesh; 1 = star-only network topology.
23	UTCTime[f]	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	Universal Time Coordinated (UTC).
24	ActiveLeaving-Enable[e]	Boolean	0, 1	R/W	Static	0	A flag indicates that if the active leaving function is enabled: 0 = Disabled; 1 = Enabled.
25	SecMode[be*]	Unsigned8	0 to 3	R/W	Static	0	Bit 0 represents whether the DLSL uses security service: 0 = Not used; 1 = Used. Bit 1 represents whether the AL uses security service: 0 = Not used; 1 = Used.
26	SecLevel[b*]	Unsigned8	0 to 63	R/W	Static	0	Bit 0, 1 and 2 represent the security levels of the DLSL frames. 000 = None; 001 = MIC-32; 010 = MIC-64; 011 = MIC-128; 100 = Encryption; 101 = Encription-MIC-32; 110 = Encription-MIC-64; 111 = Encription-MIC-128; Bit 3, 4and 5 represent the security levels of the AL packets. 000 = None; 001 = Encryption; 010 = MIC-32; 011 = Encription-MIC-32; Others are reserved.

Table 13 *(5 of 5)*

ID	Name	Data type	Valid range	Access type	Storage type	Default value	Description
27	CountryCode[e*]	Unsigned16	0 to 65 535	R/W	Dynamic	0x3C00	The country code that provides the local and regulatory-constraint guidance. See 8.4.11 for details.
28	EtoEACKEnable[e*]	Bool	0, 1	R/W	Dynamic	0	A flag indicates that if the WIA-PA network supports end to end ACK: 0 = Not support; 1 = Support.
29	AuthenState[be]	Boolean	0, 1	R/W	Dynamic	0	A flag that marks if a WIA-PA device has been authenticated to be a legal device: 0 = Not authenticated; 1 = Authenticated.
30	AuthenTime[be]	Unsigned32	0 to (2^{32}-1)	R/W	Dynamic	0	Time of a device being authenticated (in UTC time)
31	AggPeriod[be*]	Unsigned8	0 to 255	R/W	Static	N/A	The aggregation duration (in milliseconds) This value is valid only if aggsupport=1 and enable=1.
32	ObjectNumber[e*]	Unsigned8	0 to 255	R/W	Static	N/A	The number of UAOs.
33	AbsoluteSlotNumber[e]	Unsigned48	0 to (2^{48}-1)	R/W	Dynamic	0	Number of timeslots from the start of the network, denoting the current timeslot, increasing by one.

[a] Attributes are optional for the field device/handheld device.

[b] Attributes are optional for all devices in the WIA-PA networks.

[c] Attributes are not chosen by the field device/handheld device.

[d] ProtocolVersion is configured by manufacturers.

[e] Attributes numbered 0 to 33 are stored by all types of WIA-PA devices. If the default value of an attribute is N/A, users may set it; otherwise, if the default value of an attribute is specified, users should set it according to actual application scenarios.

[#] Attributes are configured by the handheld device.

[*] Attributes are configured by the GW.

6.9.1.2.2 Structured attributes

The structured attributes are listed in Table 14.

Table 14 – Structured attributes

ID	Name	Data type	Access type	Storage type	Default value	Description
100	RouteTable	NLRoute_Struct structure (see Table 15)	R/ W	Dynamic		Including the route ID, destination address, next-hop address, and retry counter. Each list member is identified by storage index.
101	Superframe	Superframe_Struct structure(see Table 16)	R/ W	Dynamic		Describing the superframe information. Each list member is identified by storage index.
102	Link	Link_Struct structure (see Table 17)	R/W	Dynamic		Describing the link information. Each list member is identified by storage index.
103	Neighbour[a]	Neighbour_Struct structure (see Table 18)	R/W	Dynamic		Describing the information of neighbour devices. Each list member is identified by storage index.
104	ChannelCondition	ChanCon_Struct structure (see Table 19)	R/ W	Dynamic		Recording the statistic information of channel condition. Each list member is identified by storage index.
105	DeviceStruct	Device_Struct structure (see Table 20)	R/W	Dynamic/constant/static		Describing the information of WIA-PA devices. Each list member is identified by storage index.
106	VCRList	VCR_Struct structure (see Table 21)	R/ W	Dynamic		Recording the VCR information. Each list member is identified by storage index.
107	DevConRep	DevConRep_Struct structure (see Table 22)	R/W	Dynamic		Recording the information of device condition. Each list member is identified by storage index. This attribute is recorded by routing devices and the gateway device.
108	KeyTable[b]	Key_Struct structure (see Table 23)	R/W	Dynamic		Including the key type, key length, key update time, key using time, key value, and count of key attack. Each list member is identified by storage index.
109	ObjList	ObjList_Struct structure (see Table 24)	R/W	Static		The list of UAO identifiers. Each list member is identified by storage index.

NOTE The sizes of all the lists specified in Table 14 are implementation dependent.

[a] Attributes are not chosen by the field device/handheld device.

[b] Attributes are optional for all devices in the WIA-PA networks.

The data type of routing attributes RouteTable is shown in Table 15.

Table 15 – NLRoute_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	RouteID	Unsigned16	0 to 65 535	A unique routing identifier
1	SourceAddress	Unsigned16	0 to 65 535	The short address of source device
2	DestinationAddress	Unsigned16	0 to 65 535	The short address of destination device
3	NextHop	Unsigned16	0 to 65 535	The short address of next-hop device
4	RetryCounter	Unsigned8	0 to 255	A counter to record end-to-end retries

The data type of superframe attributes Superframe is shown in Table 16.

Table 16 – Superframe_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	SuperframeID	Unsigned16	0 to 65 535	Unique identifier of the superframe, supplied by the NM
1	SuperframeMultiple	Unsigned8	0 to 255	SuperframeMultiple = maximum data update rates/minimum data update rates. It is used for restricting the WIA-PA superframe length and is also used for processing the long cycle data transmission. See 8.4.6 for details.
2	NumberSlots	Unsigned16	0 to 65 535	Superframe size (counts of timeslots)
3	ActiveFlag	Boolean	0, 1	Superframe active flag: 0 = Inactive; 1 = Active.
4	ActiveSlot	Unsigned48	0 to $(2^{48}-1)$	Absolute timeslot number (ASN) when a superframe begins active

The data type of link attributes Link is shown in Table 17.

Table 17 – Link_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	LinkID	Unsigned16	0 to 65 535	Unique identifier of the link
1	NeighbourAddr	Unsigned16	0 to 65 535	Reference to a neighbour table entry, which is empty when the NM allocates links to a routing device for intra-cluster communication.
2	LinkType	Unsigned8	0 to 31	Bit 0 represents the link type: 0 = Unicast; 1 = Broadcast; Bit 1 and bit 2 represent the character of a link: 00 = Transmitting; 01 = Transmit-shared; 10 = Receiving; Bit 3 represents the type of a timeslot: 0 = Data timeslot; 1 = Management timeslot. Bit 4 represents the aggregation character: 0 = Timeslot for non-aggregated packet; 1 = Timeslot for aggregated packet.
3	RelativeSlotNumber	Unsigned16	0 to 65 535	Relative timeslot number
4	LinkSuperframeNum	Unsigned8	0 to 255	LinkSuperframeNum = data update rate of this device/the minimum data update rate. It is used for processing the long cycle data transmission.
5	ActiveFlag	Boolean	0, 1	Indicating if a link is being used: 0 = Not used; 1 = Being used.
6	ChannelIndex	Unsigned8	0 to 31	The channel sequence numbers for this link, namely, the sequence numbers of the main channels.
7	SuperframeID	Unsigned16	0 to 65 535	Reference to a superframe in the superframe table

The data type of neighbour attributes Neighbour is shown in Table 18.

Table 18 – Neighbour_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	NeighbourAddr	Unsigned16	0 to 65 535	The short address of neighbour device
1	NeighbourStatus	Unsigned8	0 to 3	Bit 0 represents whether this neighbour device is a main time source: 0 = No; 1 = Yes; Bit 1 represents the status of neighbour device: 0 = Normal; 1 = Abnormal.
2	BackoffCounter	Unsigned8	0 to 31	Backoff counter
3	BackoffExponent	Unsigned8	0 to 15	Backoff exponent
4	LastTimeCommunicated	Unsigned48	0 to (2^{48}-1)	Time when last communicated with the neighbourdevice (in ASNs)
5	AveRSL	Unsigned8	0 to 255	The average level of signals received from the neighbour device in StatisticsDuration
6	PacketsTransmitted	Unsigned16	0 to 65 535	The number of un-broadcast frames sent to the neighbour device in StatisticsDuration
7	AckPackets	Unsigned16	0 to 65 535	The number of expected ACK/NACK packets received in StatisticsDuration
8	PacketsReceived	Unsigned16	0 to 65 535	The number of good packets received from the neighbour device in StatisticsDuration
9	BroadcastPackets	Unsigned16	0 to 65 535	The number of good broadcast packets received from the neighbour device in StatisticsDuration

The data type of channel condition attributes ChannelCondition is shown in Table 19.

Table 19 – ChanCon_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	DeviceShortAddress	Unsigned16	0 to 65 535	16-bit address of device
1	ChannelID	Unsigned8	0 to 255	The sequence number of channel
2	NeighbourAddr	Unsigned16	0 to 65 535	16-bit address of neighbour device
3	LinkQuality	Unsigned8	0 to 255	Link Quality Indication (LQI) value of every channel
4	PacketLossRate	Unsigned16	0 to 65 535	Packet loss rate of every channel, which is a percentage and calculated through received ACKs and total sent packets
5	RetryNum	Unsigned8	0 to MaxFrameRetries	The count of retransmission of every channel

The data type of device attributes DeviceStruct is shown in Table 20.

Table 20 – Device_struct structure *(1 of 3)*

Attribute member identifier	Name	Data type	Valid range	Storage type	Access type	Default value	Description
0	LongAddress	Unsigned64	0 to (2^{64}-1)	C	R	N/A	64-bit global unique address Same as for the DeviceType field as defined in 6.3.4: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
1	Redundant-DevFlag	Boolean	0, 1	S	R/W	0	Flag that indicates whether this device is a redundant device: 0 = Irredundant device; 1 = Redundant device.
2	Redundant-DevLongAddr	Unsigned64	0 to (2^{64}-1)	C	R	N/A	64-bit global unique address of a redundant device Same as for the DeviceType field as defined in 6.3.4: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
3	NetAddress-Assign	Boolean	0, 1	S	R/W	N/A	A flag indicating whether the network address (see 6.3.4) has been assigned: 0 = No; 1 = Yes.
4	DeviceShort-Address	Unsigned16	0 to 65 535	S	R/W	N/A	Short address of WIA-PA device The most significant 8 bits are cluster address and the least significant 8 bits are intra-cluster address
5	ManufacturerID	Unsigned24	0 to (2^{24}-1)	C	R	N/A	Manufacturer's identifier
6	DeviceSerial-Num	Unsigned64	0 to (2^{64}-1)	C	R	N/A	Device serial number
7	DeviceVersion	Unsigned8	0 to 255	C	R	N/A	Device version number
8	PowerSupply-Status	Unsigned8	0 to 10	S	R/W	N/A	Power condition: 0 = Fixed power supply; 1 = Highest power; 2 = Second highest power; 10 = lowest power.
9	RouterCapable	Boolean	0, 1	S	R/W	0	A flag to indicate if the device has routing function: 0 = No; 1 = Yes.

Table 20 *(2 of 3)*

Attribute member identifier	Name	Data type	Valid range	Storage type	Access type	Default value	Description
10	Devicestate	Unsigned8	0 to 16	S	R/W		Bits 0and 1 represent device joining state: 00 = Not joined; 01 = Joining; 10 = Security authentication; 11 = Joined. Bit 2-3 represent device running state: 00 = Inactive; 01 = Active; 10 = Failed. Others are reserved.
11	DeviceMemory-total	Unsigned32	0 to (2^{32}-1)	C	R	N/A	Total memory in a device (in octets), including RAM, ROM, Flash…
12	DeviceUsed-Memory	Unsigned32	0 to (2^{32}-1)	D	R/W	N/A	Memory used by device (in octets)
13	ClockMaster-Role	Boolean	0, 1	S	R/W	N/A	The device is a time source or not (see 8.3.3.1): 0 = No; 1 = Yes.
14	ClockUpdate	Unsigned32	0 to (2^{32}-1)	D	R/W	0	The last adjustment of clock (in seconds)
15	PacketsMACTo-DLSL	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from MAC to DLSL
16	PacketsFrom-DLSLRejected	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from the DLSL that are rejected by the NL
17	PacketsFrom-DLSLAccepted	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from the DLSL that are accepted by the NL
18	PacketsFrom-ASL	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from the ASL
19	PacketsFrom-ASLRejected	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from the ASL that are dropped by the NL
20	PacketsOutTo-DLSL	Unsigned32	0 to (2^{32}-1)	D	R/W	0	Count of the packets from the ASL that are forwarded by the NL to the DLSL
21	AGGSupport-Flag	Boolean	0, 1	S	R/W	N/A	Aggregation and disaggregation support flag (whether a routing device supports aggregation mechanism): 0 = Not support; 1 = Support.
22	AGGEnable-Flag	Boolean	0, 1	S	R/W	0	Aggregation and disaggregation enable flag (whether a routing device enables aggregation mechanism): 0 = Disenable; 1 = Enable.
23	IntraChannel-Num	Unsigned8	0 to 31	S	R/W	0	The number of usable channels for intra-cluster communication

Table 20 *(3 of 3)*

Attribute member identifier	Name	Data type	Valid range	Storage type	Access type	Default value	Description
24	IntraChanel	Unsigned8	0 to 31	S	R/W	0	A list to store all communication channels, which is allocated by the GW to routing devices and field devices. The size of the list is IntraChannelNum. See 8.4.5 for details.
25	SuperframeID	Unsigned16	0 to 65 535	S	R/W	0	Unique identifier of the superframe, supplied by the NM
The Storage types in this table are C = constant, D = Dynamic, and S = static.							

The data type of VCR attributes VCR_Struct is shown in Table 21.

Table 21 – VCR_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	VcrID	Unsigned16	0 to 65 535	VCR identifier The VCR is invalid if its identifier is 0xFFFF.
1	VcrType	Unsigned8	0 to 16	Bits 0-1representVCR type: 00 = P/S VCR; 01 = R/S VCR; 10 = C/S VCR; Bits 2 and 3 represent aggregation or non aggregation VCR: 00 = Non-aggregation VCR; 01 = Data aggregation VCR; 10 = Packet aggregation VCR; Others are reserved.
2	SrcObjID	Unsigned8	0 to 255	Source object ID
3	SrcObjInsID	Unsigned8	0 to 255	ID of source object instance
4	DesObjID	Unsigned8	0 to 255	Destination object ID
5	DesObjInsID	Unsigned8	0 to 255	ID of destination object instance
6	DataUpdateRate	Unsigned32	0 to (2^{32}-1)	Data update rate (in milliseconds) The data has no cycle if this value is 0xFFFFFFFF.
7	VcrStatus	Boolean	0, 1	VCR status: 0 = Inactive; 1 = Active.
8	VcrActivationTime	Unsigned32	0 to (2^{32}-1)	The activation time of VCR (in milliseconds)
9	ServiceTime	Unsigned32	0 to (2^{32}-1)	The valid service duration of VCR (in milliseconds)
10	SourceChAddress	Unsigned16	0 to 65 535	The cluster head address of the source device
11	SourceAddress	Unsigned16	0 to 65 535	The network address of the source device
12	DestinationAddress	Unsigned16	0 to 65 535	The network address of the destination device
13	SecurityPolicy	Unsigned8	0 to 255	Security level of packets (see Annex A)
14	RouteID	Unsigned16	0 to 65 535	Route identifier
The time unit used in this table is the timeslot used by the superframe.				

The data type of device condition attributes DevConRep is shown in Table 22.

Table 22 – DevConRep_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	DevShortAddr	Unsigned16	0 to 65 535	16-bit short address of device
1	NumPktSent	Unsigned16	0 to 65 535	The total number of packets sent after the last report
2	NumPktRcvd	Unsigned16	0 to 65 535	The total number of packets terminated in the device after the last report
3	NumMacMicFailure	Unsigned16	0 to 65 535	The total number of Message Integrity Code (MIC) failures after the last report
4	BatLevel	Unsigned8	1 to 10	Residual power level
5	RestartCount	Unsigned8	0 to 255	Restart count of device
6	Uptime	Unsigned32	0 to (2^{32}-1)	Time from the last restart (in seconds)

The data type of key attributes KeyTable is shown in Table 23.

Table 23 – Key_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	KeyID	Unsigned16	0 to 65 535	Key identifier
1	KeyType	Unsigned8	0 to 3	Key type: 0 = Joining key; 1 = Key encryption key; 2 = DLSL encryption key; 3 = AL encryption key.
2	KeyLength	Unsigned8	0 to 255	(KeyLength+1) is the valid length of the key (in bits)
3	KeyActiveTime	Unsigned48	0 to (2^{48}-1)	Absolute timeslot number (ASN) when key begins active.
4	KeyData	Octetstring		Key value
5	KeyAttackCnt	Unsigned8	0 to 255	The total number of key attacks
6	KeyState	Unsigned8	0 to 7	The using state of a key: 0 = Backup; 1 = Using; 2 = Invalid; Others are reserved.

The data type of object list attributes ObjList is shown in Table 24.

Table 24 – ObjList_Struct structure

Attribute member identifier	Name	Data type	Valid range	Description
0	ObjectID	Unsigned8	0 to 255	The unique identifier of the UAO
1	InstanceID	Unsigned8	0 to 255	The unique identifier of an object instance
2	ProfileID	Unsigned16	0 to 65 535	The profile ID for the UAO
3	ParameterNumber	Unsigned8	0 to 255	Number of parameters of the UAO
NOTE ObjectID and InstanceID in ObjList_Struct are identical with Object Identifier and Instance ID (see Annex C). DAGO, PAGO, and DGO are three special UAOs. Therefore, the attributes of DAGO, PAGO, or DGO are included in the ObjList_if such object is implemented in the device. In addition, DagoID and DagoInsID of DAGO, PagoID and PagoInsID of PAGO as well asDgoID and DgoInsID of DGO are respectively corresponding to the ObjectID and InstanceID in ObjList_Struct.				

6.9.2 MIB services

6.9.2.1 Remote MIB services

The attributes in the MIB can be read and written remotely through the attribute-getting and attribute-setting services provided by the NL.

6.9.2.2 Local MIB services

6.9.2.2.1 General

The attributes in the MIB can be read and written locally through the attribute-getting and attribute-setting services provided by the local DMAP.

6.9.2.2.2 DMAP attribute getting services

DMAP-MIB-GET.request is used by protocol layers to request attributes in the MIB.

The semantics of DMAP-MIB-GET.request are as follows:

DMAP-MIB-GET.request (
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count
)

Table 25 specifies the parameters for DMAP-MIB-GET.request.

Table 25 – DMAP-MIB-GET.request parameters

Name	Data type	Valid range	Description
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	The identifier of attribute member, which is used to get the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to get the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to get the structured MIB attributes; Getting all attribute values from FirstValueStorIndex if Count = 0

DMAP-MIB-GET.confirm is used to return the result of DMAP-MIB-GET.request.

The semantics of DMAP-MIB-GET.confirm are as follows:

DMAP-MIB-GET.confirm (
Status,
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count,
AttributeValue
)

Table 26 specifies the parameters for DMAP-MIB-GET.confirm.

Table 26 – DMAP-MIB-GET.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Attribute getting results: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; Others are reserved.
AttibuteID	Unsigned8	0 to 255	The requested attribute ID
AttributeMemID	Unsigned8	0 to 255	The identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple read attribute values
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values
AttributeValue	Octetstring		The value of the attribute

If the operation of getting attributes is successful, the Status should be SUCCESS and the AttributeValue is valid; otherwise, if the MIB does not have the needed attributes, the Status should return UNSUPPORTED_ATTRIBUTE and the AttributeValue is invalid.

6.9.2.2.3 DMAP attribute setting services

DMAP-MIB-SET.request should be used by the protocol layers to write attributes to the MIB.

The semantics of DMAP-MIB-SET.request are as follows:

```
DMAP-MIB-SET.request (
                    AttributeID,
                AttributeMemID,
                FirstValueStorIndex,
                Count,
                AttributeValue
                    )
```

Table 27 specifies the parameters for DMAP-MIB-SET.request.

Table 27 – DMAP-MIB-SET.request parameters

Name	Data type	Valid range	Description
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	Identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to get the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attributes
AttributeValue	Octetstring		Value of the attribute

DMAP-MIB-SET.confirm is used to return the result of DMAP-MIB-SET.request.

The semantics of DMAP-MIB-SET.confirm are as follows:

DMAP-MIB-SET.confirm (

Status,
AttributeID,
AttributeMemID,
FirstValueStorIndex
)

Table 28 specifies the parameters for DMAP-MIB-SET.confirm.

Table 28 – DMAP-MIB-SET.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Attribute setting result: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = INVALID_PARAMETER; Others are reserved.
AttributeID	Unsigned8	0 to 255	Attribute ID in the MIB
AttributeMemID	Unsigned8	0 to 255	Identifier of attribute member The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple written attribute values

If the operation of setting attributes is successful, the Status should be SUCCESS; if the MIB does not have the needed attributes, the Status should be UNSUPPORTED_ATTRIBUTE; otherwise, if the set attributes are not conformable to the specified attributes, the Status should beINVALID_PARAMETER.

7 Physical layer

7.1 General

WIA-PA uses Radio Frequencies (RF) to communicate between devices. The communication distance between two devices depends upon the transmitter power, antenna type, and obstruction between two devices. WIA-PA devices can be powered either by mains power, battery, energy harvesting devices (for example solar, vibration, etc), or a combination there of.

The PHY specification of WIA-PA is based on the IEEE STD 802.15.4-2011 2,4 GHz compliant radio.

WIA-PA PHY defines the signaling method, signal strength, receiver sensitivity, and environment for sending bits across the network media. The PHY is responsible for activation and deactivation of the radio transceiver, energy detection, link quality indicator, channel selection, clear channel assessment, transmitting and receiving packets across the physical medium.

The PHY specification of WIA-PA includes the following:

a) the PHY requirements many of which are adopted directly from IEEE STD 802.15.4-2011 including specific requirements on channels, transmit power, and receiver sensitivity, and

b) the PHY requirements to ensure inter-connectable among WIA-PA devices.

7.2 General requirements based on IEEE STD 802.15.4-2011

The WIA-PA physical layer shall be based on IEEE STD 802.15.4-2011 2,4 GHz DSSS with additional requirements and exceptions as specified herein.

One of the device provisioning parameters, CountryCode (see 8.4.11), provides the locale and regulatory-constraint guidance needed to drive conformance to the relevant regulations. See Annex D.

The device vendors are responsible that these devices are compliant with this standard and any country- or region-specific regulations.

Table 29 specifies the IEEE STD 802.15.4-2011 PHY selection for a WIA-PA device.

Table 29 – PHY protocol selection

Clause of IEEE STD 802.15.4-2011	Header	Presence	Constraints
6	General PHY requirements		
6.1	General requirements and definitions	Partial	Only 2 450 MHz DSSS PHY, employing O-QPSK modulation
6.1.1	Operating frequency range	Partial	Only 2 450 MHz DSSS band is included (see Table 30)
6.1.2	Channel assignments	Partial	Only Channels 11-25 of channel page 0 are included (see Table 32)
6.1.3	Minimum LIFS and SIFS periods	Partial	Only minimum LIFS and SIFS for 2 450 MHz band is included
6.1.4	RF power measurement	YES	
6.1.5	Transmit power	YES	
6.1.6	Out-of-band spurious emission	YES	
6.1.7	Receiver sensitivity definitions	YES	
6.2	PHY service specifications	YES	
6.3	PPDU format	YES	
6.4	PHY constants and PIB attributes		
6.4.1	PHY constants	YES	
6.4.2	PHY PIB attributes	Partial	See 7.3.9
6.5	2 450 MHz PHY specifications	YES	See 7.3.8 for further explanation
6.6	868/915 MHz band binary phase-shift keying (BPSK) PHY specifications	NO	Only 2 450 MHz band is included
6.7	868/915 MHz band (optional) amplitude shift keying (ASK) PHY specifications	NO	Only 2 450 MHz band is included
6.8	868/915 MHz band (optional) O-QPSK PHY specifications	NO	Only 2 450 MHz band is included
6.9	General radio specifications		
6.9.1	TX-to-RX turnaround time	YES	
6.9.2	RX-to-TX turnaround time	YES	
6.9.3	Error-vector magnitude (EVM) definition	YES	
6.9.4	Transmit center frequency tolerance	YES	
6.9.5	Transmit power	NO	See 7.3.6 for replacement of this subclause
6.9.6	Receiver maximum input level of desired signal	YES	
6.9.7	Receiver ED	YES	
6.9.8	Link quality indicator (LQI)	YES	
6.9.9	Clear channel assessment (CCA)	YES	

7.3 Additional requirements

7.3.1 General

Although the IEEE STD 802.15.4-2011 physical layer supports multiple frequency bands and modulation classes, a device compliant with this standard shall operate in the license-exempt 2 400 MHz to 2 483,5 MHz band using DSSS modulation and coding at 250 kbit/s, which is specified in IEEE STD 802.15.4-2011, Table 51 as 2 450 DSSS PHY. This standard does not support any of the other frequency bands or data rates or modulation and coding techniques specified in IEEE STD 802.15.4-2011.

7.3.2 Frequency allocations

The transceiver shall employ DSSS and operate in the license-free ISM band as shown in Table 30.

Table 30 – Frequency band and data rate

Frequency MHz	Communication rate kchip/s	Modulation	Bit rate kb/s	Symbol rate ksymbol/s	Symbols
2 400 to 2 483,5	2 000	O-QPSK	250	62,5	16-ary Orthogonal

7.3.3 Channel numbers and frequency assignments

The 2 450 MHz band supports channel assignments 11 – 26 as specified by IEEE STD 802.15.4-2011. Since channel 26 is not compliant with the regulation of some world regions, it is not included in this document.

The frequency assignment for the channels 11 – 25 is shown in Table 31.

Table 31 – Frequency assignments

Channel number	Frequency MHz		Channel number	Frequency MHz		Channel number	Frequency MHz
11	2 405		16	2 430		21	2 455
12	2 410		17	2 435		22	2 460
13	2 415		18	2 440		23	2 465
14	2 420		19	2 445		24	2 470
15	2 425		20	2 450		25	2 475

7.3.4 Radio transceivers

The radio transceivers of WIA-PA devices are compliant to IEEE STD 802.15.4-2011 and conform to worldwide regulations in Europe, Canada, Japan, China, the United States and other locations. See Annex D.

7.3.5 Unspecified or improved required radio performance

This document includes the following that is not specified in IEEE STD 802.15.4-2011.

a) Recommended time to switch between channels should be 12-symbol periods.

b) Recommended cold start radio turn-on time is 4 ms maximum. It includes frequency lock and settling time.

7.3.6 Transmit power

Transmit power shall be the Equivalent Isotropic Radiated Power (EIRP) of the device. WIA-PA devices shall provide a nominal EIRP of +10 dBm (10 mW) ±3 dB. The maximum radiated power level shall not exceed the regulatory requirements that apply where the device is deployed, as constrained by CountryCode (see 8.4.11).

In both cases the rated transmit power shall be representative of the manufacturer's production device with power source (batteries, solar, or otherwise) at 95 % to 100 % of rated capacity. All devices should be provided with an omni-direction antenna. The rated transmit power data including maximum output power and radiation patterns for all antenna supported by the product shall be available in the product manual or upon request.

7.3.7 Output power control

The device power level shall be programmable at discrete, monotonic levels from −10 dBm to +10 dBm EIRP (within ±4 dBm of error in actual power level). The output power can be controlled by setting the PHY PIB attribute phyTransmitPower.

7.3.8 Receiver sensitivity

Receiver sensitivity of WIA-PA complies with IEEE STD 802.15.4-2011 and shall be capable of achieving a sensitivity of −85 dBm or better.

7.3.9 PHY PIB attributes

This standard includes the attributes specified in the IEEE STD 802.15.4-2011, 6.4.2 and additional attributes that apply only to this document. Some of these attributes are assigned a constant value as shown in Table 32.

Table 32 – PHY PIB attributes *(1 of 2)*

Attribute	Comment
phyCurrentChannel	Set by Ph-Enable request
phyChannelsSupported	Only channels 11 – 25 are supported
phyTXPowerTolerance	1dB. 3 dB, 6 dB
phyTXPower	The values from ‘−10 dBm to +10 dBm’ are supported
phyCCAMode	Only value 2 (carrier sense) is included
phyCurrentPage	Only value‘0’ is included
phyMaxFrameDuration	Only value ‘266’ is included
phySHRDuration	Only value ‘10’ is included
phySymbolsPerOctet	Only value ‘2’ is included
phyPreambleSymbolLength	Only value ‘0’ is included
phyUWBDataRatesSupported	This attribute is invalid in WIA-PA
phyCSSLowDataRateSupported	This attribute is invalid in WIA-PA
phyUWBCoUSupported	This attribute is invalid in WIA-PA
phyUWBCSSupported	This attribute is invalid in WIA-PA
phyUWBCurrentPulseShape	This attribute is invalid in WIA-PA
phyUWBCoUpulse	This attribute is invalid in WIA-PA
phyUWBCSpulse	This attribute is invalid in WIA-PA
phyUWBLCPWeight1	This attribute is invalid in WIA-PA
phyUWBLCPWeight2	This attribute is invalid in WIA-PA
phyUWBLCPWeight3	This attribute is invalid in WIA-PA

Table 32 *(2 of 2)*

Attribute	Comment
phyUWBLCPWeight4	This attribute is invalid in WIA-PA
phyUWBLCPDelay2	This attribute is invalid in WIA-PA
phyUWBLCPDelay3	This attribute is invalid in WIA-PA
phyUWBLCPDelay4	This attribute is invalid in WIA-PA
phyRanging	This attribute is invalid in WIA-PA
phyRangingCrystalOffset	This attribute is invalid in WIA-PA
phyRangingDPS	This attribute is invalid in WIA-PA
phyCurrentCode	Only value '0' is included
phyNativePRF	This attribute is invalid in WIA-PA
phyUWBScanBinsPerChannel	This attribute is invalid in WIA-PA
phyUWBInsertedPreambleInterval	This attribute is invalid in WIA-PA
phyTXRMARKER Offset	This attribute is invalid in WIA-PA
phyRXRMARKER Offset	This attribute is invalid in WIA-PA
phyRFRAMEProcessingTime	This attribute is invalid in WIA-PA
phyCCADuration	This attribute is invalid in WIA-PA

8 Data link layer

8.1 General

The WIA-PA Data Link Layer (DLL) is designed to guarantee communication among WIA-PA devices in a reliable and secure way in real-time. The DLL of WIA-PA extends the IEEE STD 802.15.4-2011 superframe structure. The WIA-PA DLL supports certain key functions, including frequency hopping, retransmission, and Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA) hybrid channel access mechanisms. These mechanisms are used to guarantee reliability and real-time transmission in communication. The WIA-PA DLL is designed to use MIC mechanism and encryption technology to guarantee the integrity and confidentiality of the communication process.

8.2 Protocol stack

The WIA-PA DLL is designed to leverage the IEEE STD 802.15.4-2011 to meet the requirements of process automation. The DLL protocol stack is shown in Figure 28.

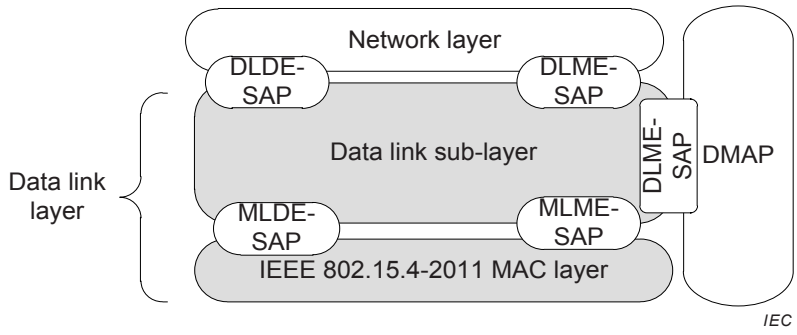

Figure 28 – WIA-PA DLL protocol stack

The WIA-PA DLL includes the following parts:

a) the IEEE STD 802.15.4-2011 MAC, which handles the mechanisms of sending and receiving individual data frames;

b) the DLSL, which handles the aspects of communication resource.

8.3 MAC overview and function extension

8.3.1 MAC overview

The following section specifies the MAC of the WIA-PA standard and the content of IEEE STD 802.15.4-2011 MAC is not stated in this document. This document is based on the IEEE STD 802.15.4-2011 beacon-enabled MAC, which handles all access to the physical radio channel and is responsible for the following tasks:

a) generating network beacons if the device is a gateway device or a routing device,

b) synchronizing to network beacons,

c) supporting network point to point association and disassociation,

d) supporting point to point security,

e) employing the timeslotted CSMA-CA mechanism for device joining,

f) handling and maintaining the GTS mechanism, and

g) providing a reliable link between two peer MAC entities.

Field devices and handheld devices correspond to the Reduced-Function Device (RFD), and routing devices and gateway devices correspond to the Full-Function Device (FFD). None of the optional functions of RFD and FFD in IEEE STD 802.15.4-2011 are required in this standard.

8.3.2 General requirements based on IEEE STD 802.15.4-2011

The WIA-PA MAC shall be based on IEEE STD 802.15.4-2011 MAC with additional requirements and exceptions as specified herein.

The GW acts as the PAN coordinator in IEEE STD 802.15.4-2011 and the routing devices act as coordinators in IEEE STD 802.15.4-2011.

Table 33 specifies the IEEE STD 802.15.4-2011MAC selection for a WIA-PA device.

Table 33 – MAC protocol selection *(1 of 2)*

Clause of IEEE STD 802.15.4-2011	Header	Presence	Constraints
5	MAC protocol		
5.1	MAC functional description		
5.1.1	Channel access	YES	
5.1.2	Starting and maintaining PANs	Partial	Only active scan is used in WIA-PA.
5.1.3	Association and disassociation	YES	
5.1.4	Synchronization	YES	
5.1.5	Transaction handling	NO	Transactions can be instigated from the devices themselves.
5.1.6	Transmission, reception, and acknowledgement	YES	
5.1.7	GTS allocation and management	NO	This function is not used in WIA-PA.
5.1.8	Ranging	NO	This function is not used in WIA-PA.
5.2	MAC frame formats		
5.2.1	General MAC frame format	YES	For WIA-PA MAC frame format, the value of the Security Enabled subfield of MHR shall be set to 0 and the length of the Auxiliary Security Header subfield is 0 octets. The PAN ID Compression is set to 1. The length of Source PAN identifier field is 0 and the Destination PAN Identifier is set to NetworkID (see 6.9.1.2.1) of WIA-PA.
5.2.2	Format of individual frame types	YES	
5.2.3	Frame compatibility	YES	
5.3	MAC command frames	Partial	See following 5.3.1 to 5.3.9.
5.3.1	Association request command	YES	
5.3.2	Association response command	YES	
5.3.3	Disassociation notification command	YES	
5.3.4	Data request command	YES	
5.3.5	PAN ID conflict notification command	NO	PAN ID conflict notification command is not used in WIA-PA.
5.3.6	Orphan notification command	NO	Orphan notification command is not used in WIA-PA.
5.3.7	Beacon request command	NO	Beacon request command is not used in WIA-PA.
5.3.8	Coordinator realignment command	NO	Coordinator realignment command is not used in WIA-PA.
5.3.9	GTS request command	NO	GTS request command is not used in WIA-PA.

Table 33 *(2 of 2)*

Clause of IEEE STD 802.15.4-2011	Header	Presence	Constraints
6	MAC service		
6.1	Overview		
6.2	MAC management service	Partial	See following 6.2.1 to 6.2.17.
6.2.1	Common requirements for MLME primitives	YES	
6.2.2	Association primitives	YES	
6.2.3	Disassociation primitives	YES	
6.2.4	Communications notification primitives	YES	
6.2.5	Primitives for reading PIB attributes	YES	
6.2.6	GTS management primitives	NO	GTS management primitives are not used in WIA-PA.
6.2.7	Primitives for orphan notification	NO	Primitives for orphan notification are not used in WIA-PA.
6.2.8	Primitives for resetting the MAC sublayer	NO	After a node leaving the WIA-PA network, its MAC sublayer resets to the default value.
6.2.9	Primitives for specifying the receiver enable time	YES	
6.2.10	Primitives for channel scanning	YES	
6.2.11	Primitives for writing PIB attributes	YES	
6.2.12	Primitives for updating the superframe configuration	NO	Primitives for updating the superframe configuration are not used in WIA-PA.
6.2.13	Primitives for synchronizing with a coordinator	NO	Primitives for synchronizing with a coordinator are not used in WIA-PA.
6.2.14	Primitives for requesting data from a coordinator	NO	Primitives for requesting data from a coordinator are not used in WIA-PA.
6.2.15	Primitives for specifying dynamic preamble	NO	Primitives for specifying dynamic preamble are not used in WIA-PA.
6.2.16	Primitives for channel sounding	NO	Primitives for channel sounding are not used in WIA-PA.
6.2.17	Primitives for ranging calibration (for UWB PHYs)	NO	Primitives for ranging calibration (for UWB PHYs) are not used in WIA-PA.
6.3	MAC data service	Partial	Following 6.3.1 to 6.3.3 are used in WIA-PA.
6.3.1	MCPS-DATA.request	YES	
6.3.2	MCPS-DATA.confirm	YES	
6.3.3	MCPS-DATA. Indication	YES	
6.3.4	MCPS-PURGE.request	NO	The MCPS-PURGE.request is not used in WIA-PA.
6.3.5	MCPS-PURGE. confirm	NO	The MCPS-PURGE. confirm is not used in WIA-PA.
6.4	MAC constants and PIB attributes		
6.4.1	MAC constants	YES	
6.4.2	MAC PIB attributes	Partial	See Table 34 for details.
6.4.3	Calculating PHY dependent MAC PIB values	YES	

This document includes the attributes specified in the IEEE STD 802.15.4-2011, 6.4.2 and additional attributes that apply only to this document. Some of these attributes are assigned a constant value as shown in Table 34.

Table 34 – MAC PIB attributes

Attribute	Comment
macAckWaitDuration	Only value 54 symbols is included for 2,4 GHz PHY
macAssociatedPANCoord	Only value FALSE is included
macAssociationPermit	Only value TRUE is included
macAutoRequest	Only value FALSE is included
macBattLifeExt	Only value FALSE is included
macBattLifeExtPeriods	This attribute is invalid in WIA-PA
macBeaconPayload	The contents of the beacon payload are extended in WIA-PA
macBeaconPayloadLength	Only value 10 is included
macBeaconTxTime	Only value 0x000000 is included
macCoordExtendedAddress	This attribute is invalid in WIA-PA
macCoordShortAddress	This attribute is invalid in WIA-PA
macGTSPermit	Only value FALSE is included
macMaxFrameTotalWaitTime	Only when phyMaxFrameDuration is 266
macMinLIFSPeriod	Only value 40 is included (for 2,4 GHz PHY)
macMinSIFSPeriod	Only value 12 is included (for 2,4 GHz PHY)
macPromiscuousMode	Only value TRUE is included
macRangingSupported	Only value FALSE is included
macSecurityEnable	Only value FALSE is included
macSyncSymbolOffset	0x000-0x100 for the 2,4 GHz PHY
macTxControlActiveDuration	This attribute is invalid in WIA-PA
macTxControlPauseDuration	This attribute is invalid in WIA-PA

8.3.3 MAC function extension

8.3.3.1 General

This document extends one MAC PIB attribute. The extended PIB attribute is listed in Table 35.

Table 35 – MAC extended PIB attributes

Attribute	Type	Range	Description	Default
macBeaconNum	Integer	1-15	The number of beacon transmissions in one timeslot	1

macBeaconNum is used to indicate the number of beacon transmissions in one timeslot.

8.3.3.2 Beacon payload

The WIA-PA network uses the IEEE STD 802.15.4-2011 MAC beacon payload to distribute superframe information. In IEEE STD 802.15.4-2011, see 5.2.2.1 for the frame format of the MAC beacon.

The beacon payload is shown in Table 36.

Table 36 – Beacon payload

	Beacon payload			
Length in octet(s)	1	6	2	1
Field name	ClusterID	ASN	Timevalue	NextBcnChannel

The subfields in Table 36 are defined as follows:

a) ClusterID indicates the identifier of the cluster;

b) ASN indicates the absolute timeslot number. Its value shall increase by one (does not decrease). Its current value is always the sequence number of the current timeslot. The maximum value shall be (2^{48}-1). After the maximum value, it shall re-count from zero;

c) See Table 60 for Timevalue; and

d) NextBcnChannel indicates the channel used to transmit the next beacon.

8.3.3.3 Association request command frame

The WIA-PA network device uses the IEEE STD 802.15.4-2011 MAC association request command frame to join the WIA-PA network. See 5.3.1 in IEEE STD 802.15.4-2011 for the frame format of the MAC association request command frame.

The types of WIA-PA network devices are indicated by Bit 1 in the Capability Information field (See Figure 50 in IEEE STD 802.15.4-2011) of the MAC association request command frame.

The format of the Capability Information field of the MAC association request command frame is shown in Table 37.

Table 37 – Format of Capability Information field

	Format of Capability Information field						
Length in bit(s)	1	1	1	1	2	1	1
Subfield name	Reserved	Device type	Power source	Receiver on when idle	Reserved	Security capability	Allocated address

The subfields in Table 37 are listed as follows:

a) the Device type subfield is extended to indicate the WIA-PA device type; 0 indicates the routing device and 1 indicates the field device or handheld device;

b) see IEEE STD 802.15.4-2011, 5.3.1.2 for the power source subfield, receiver on when idle subfield, security capability subfield, and allocated address subfield.

8.4 DLSL function description

8.4.1 General

The DLSL provides a service interface between the NL and the MAC. The DLSL conceptually includes a Data Link Sub-Layer Data Entity (DLDE) and a Data Link Sub-Layer Management Entity (DLME). The DLDE provides data service interfaces. The DLME provides layer management services such as the configuration of the parameters of DLSL and the monitoring of the operation status of DLSL.

Figure 29 depicts the components and interfaces of the DLSL.

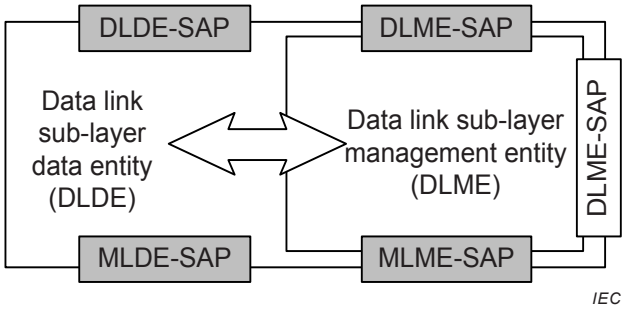

Figure 29 – WIA-PA DLSL reference model

The DLSL provides two services, which are accessed through following two Service Access Points (SAPs):

a) the DLSL data service, which is accessed through the DLDE Service Access Point (DLDE-SAP);

b) the DLSL management service, which is accessed through the DLME-SAP.

In this document, the main function of the DLSL is to allocate communication resources among competitive users in order to avoid collisions, improve throughput, and increase bandwidth utilization. The main concepts of the DLSL include timeslot, superframe and link, which are defined as follows.

a) Timeslot is the basic time unit in packet exchange. The WIA-PA timeslot duration is configurable.

b) Superframe is a collection of timeslots repeating on a cyclic schedule. The number of timeslots in a given superframe determines the communication cycle for the WIA-PA devices that use the timeslots.

c) Link includes time and frequency. A link assignment specifies how the device uses a set of superframe timeslots. The link types include transmitting, receiving and transmit-shared. The sharing link allows more than one device to contend this link for packet exchange at the same time. The transmitting and receiving links should only allow designated devices to exchange packets.

8.4.2 Coexistence

The WIA-PA network should consider the following coexistence strategies:

a) the WIA-PA network extends the IEEE STD 802.15.4-2011 superframe structure;

b) the WIA-PA DLSL together with the NM achieves coexistence with other wireless networks. The WIA-PA DLSL incorporates several strategies to optimize coexistence:
   – timeslot communication;
   – low duty-cycle;
   – multi-channel;
   – frequency hopping (FH); and
   – collision avoidance.

8.4.3 Timeslot communication

The key requirement of timeslot communication is to guarantee that all transactions occur in a timeslot according to specific timing requirements. That is to say, all packets should be exchanged in a prescriptive timeslot and not be delayed. The timeslot length of WIA-PA DLSL is fully compatible with the timeslot length of IEEE STD 802.15.4-2011.

The timeslot duration is configured by the NM after devices join the network.

8.4.4 WIA-PA superframe

In order to guarantee real-time and reliable communication, this document only takes account of the beacon-enabled IEEE STD 802.15.4-2011 superframe structure.

The WIA-PA superframe structure is shown in Figure 30.

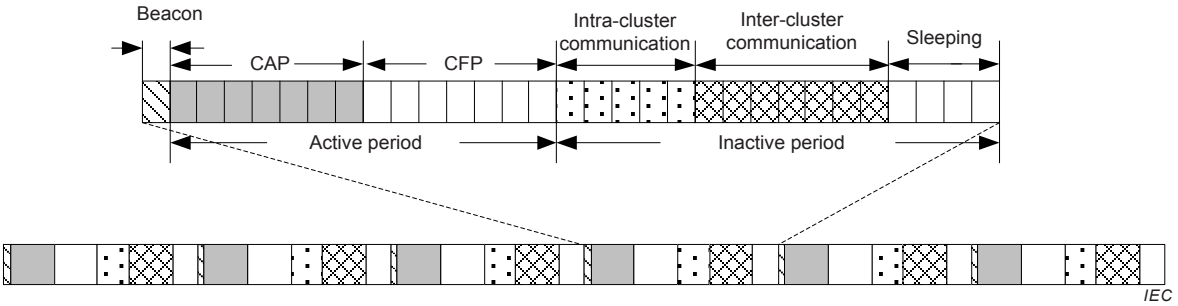

Figure 30 – WIA-PA superframe

The periods of the WIA-PA superframe are defined as follows:

a) the CAP period defined in the IEEE STD 802.15.4-2011 superframe is used for device joining, intra-cluster management and retry in the WIA-PA superframe;

b) the CFP period defined in the IEEE STD 802.15.4-2011 superframe is used for communication between handheld devices and their cluster head in the WIA-PA superframe; and

c) the inactive period defined in the IEEE STD 802.15.4-2011 superframe is used for intra-cluster communication, inter-cluster communication, and sleeping in the WIA-PA superframe.

The NM should generate WIA-PA superframes. The superframe lengths of routing devices are different, and are set as the lowest data update rate of all the members in a cluster. The timeslot types of WIA-PA superframe includes shared timeslots and dedicated timeslots. Shared timeslots are used for transmission of aperiodic data, and dedicated timeslots are used for intra- and inter-cluster transmission of periodic data.

Because the inactive period defined in the IEEE STD 802.15.4-2011 superframe is used for intra-cluster communication, inter-cluster communication, and sleeping in the WIA-PA superframe, the WIA-PA basic superframe duration is defined as thirty-two timeslots. The duration of the WIA-PA superframe is defined as 2^N (N is a natural integer) times the WIA-PA basic superframe duration.

Each routing device has only one superframe and the communication among routing devices happens during the overlapped inter-cluster timeslots. Each field device has only one superframe; the communication among field device and routing device happens during the intra-cluster timeslots.

8.4.5 Frequency hopping

The WIA-PA network supports frequency hopping, and the hopping sequence is designated by the NM. Frequency hopping in the WIA-PA network includes three mechanisms: AFS, AFH, and TH.

Adaptive frequency switch (AFS): in the WIA superframe, the beacon, CAP and CFP use the same channel in the same superframe cycle, and change the channel according to the channel conditions in different superframe cycles. That is to say, bad channel condition, which

means that the packet drop rate is above PLRThreshold, triggers the operation of changing channels. See 6.9.1.2.1 for PLRThreshold.

Adaptive frequency hopping (AFH): irregularly changes communication channels per timeslot of the WIA superframe according to actual channel condition. The channel conditions are measured in retry times. If the channel condition is bad and the retry times of the sender reaches the value of ChannelThreshold, the sender chooses the next channel in sequence from IntraChanel[ ] and notifies the receiver during the next retry timeslot by using the main channel (see Table 17). If the receiver does not receive the notification, it counts its retry times continuously. When the retry times of the receiver reach the value of ChannelThreshold, the receiver chooses the next channel from IntraChanel[] during the (ChannelThreshold+2)th timeslot. If the receiver receives the notification of a channel switch, it changes the communication channel and returns ACK; otherwise, it does not change the communication channel and retriesdata by using the main channel. If the retry times of the sender reach macMaxFrameRetries, the sender discards the current packet and transmits the next packet by using the main channel. If the communication between the sender and the receiver is successful before the retry times of the sender reaches macMaxFrameRetries, the sender transmits the next packet by using the standby channel. The Intra-cluster period adopts the AFH mechanism. See 6.9.1.2.1 for ChannelThreshold and IntraChanel[]; see Table 52 in IEEE STD 802.15.4-2011 for the information of macMaxFrameRetries.

The current channel in IntraChanel[] used by sender and receiver is marked as main channel, while other channels in IntraChanel[] are marked as standby channels.

Timeslot Hopping (TH): regularly changes communication channels per timeslot of the WIA superframe to combat interference and fading. The Inter-cluster period adopts the TH mechanism. The hopping structure is: <timeslot 1, channel 1><timeslot 2, channel 2>… <timeslot i, channel i>.

The specific hopping mechanisms of DLSL in the WIA-PA network are shown in Table 38.

Table 38 – Hopping mechanisms

IEEE STD 802.15.4-2011	WIA-PA	Basic MAC mechanism		DLSL Hopping mechanism
Beacon	Beacon	TDMA	Frequency Division Multiple Access (FDMA) and TDMA	AFS
CAP	CAP	CSMA		
CFP	CFP	TDMA		
Inactive	Intra-cluster period	TDMA		AFH
	Inter-cluster period	TDMA		TH
	Sleeping			

Different routing devices use different channels in the active period. If the number of channels is not enough, the WIA-PA network uses the TDMA mechanism to enhance the system capacity. The start time of a superframe is configured by the NM. For example, there are three routing devices R1, R2, and R3 in the WIA-PA network. The superframe lengths of R1, R2 and R3 are respectively one, two and four WIA-PA basic superframe duration(s), as shown in Figure 31. According to the superframe definition, the active period of R1 cannot be multiplexed with the active periods of R2 and R3. However, the active periods of R2 and R3 may be multiplexed with each other. The active periods of R2 and R3 may use the same channel, while the active period of R1 should use a different channel from that of the active periods of R2 and R3.

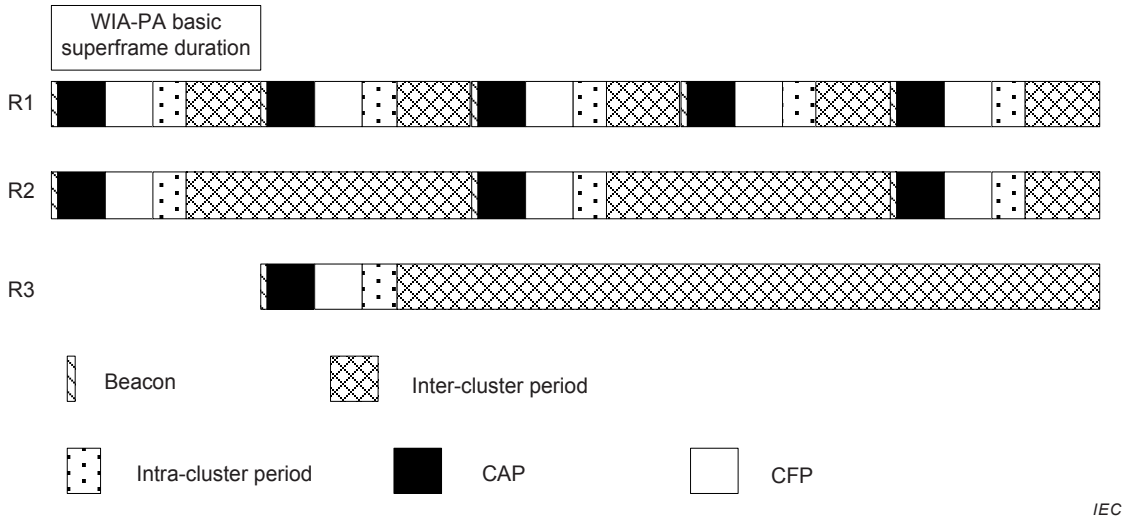

Figure 31 – R1, R2 and R3 superframe structures

8.4.6 Transmission of long cycle data

The long cycle data is defined as the data update rate of a device that is either greater than the maximum superframe length of IEEE STD 802.15.4-2011 or is greater than the data update rate of the routing device in a cluster.

To indicate the transmission of the long cycle data, this document defines a parameter termed TransmitFlag. TransmitFlag is defined as follows:

$$TransmitFlag = \lceil (AbsoluteSlotNumber - ActiveSlot + 1) / NumberSlots \rceil \% SuperframeMultiple$$

where

AbsoluteSlotNumber	see 6.9.1.2;
ActiveSlot	see 6.9.1.2;
NumberSlots	see 6.9.1.2;
SuperframeMultiple	see 6.9.1.2.

In every superframe cycle, field devices receive the beacons and decide whether to send data in this superframe cycle. The process of long cycle data transmission is described as follows:

- If 0 < TransmitFlag < SuperframeMultiple and TransmitFlag = LinkSuperframeNum, then the field device transmits data in this superframe cycle.
- If TransmitFlag = 0 and LinkSuperframeNum = SuperframeMultiple, then the field device transmits data in this superframe cycle.

See Table 17 for LinkSuperframeNum.

Figure 32 is an example of long cycle data transmission.

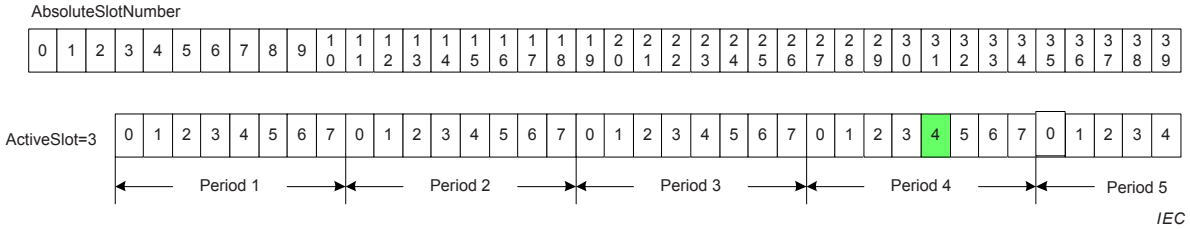

Figure 32 – An example of long cycle data transmission

In Figure 32, ActiveSlot = 3, NumberSlots = 8, SuperframeMultiple = 4, and LinkSuperframeNum = 4.

If the current absolute slot of a beacon is 27 (AbsoluteSlotNumber = 27), *TransmitFlag* is calculated as follows:

$$TransmitFlag = \lceil (27 - 3 + 1) / 8 \rceil \% 4 = 0$$

According to the calculation result, it can be concluded that TransmitFlag = 0 and that SuperframeMultiple = LinkSuperframeNum, which indicates that the packet should be sent during the current superframe cycle.

8.4.7 Retry strategy

The NM in the mesh network and the routing devices in the star network should allocate some timeslots for retries. The retry strategy supports the frequency hopping mechanisms (see 8.4.5).

The number of retry timeslots is bound by the constant macMaxFrameRetries in the IEEE STD 802.15.4-2011 MAC PAN Information Base (PIB).

8.4.8 Management service

The WIA-PA DLSL provides management services to the upper layer and the DMAP with DLME-SAP. The services include network discovery, device joining and leaving, resource allocation and operations of management database attributes (see Clause 6).

8.4.9 Radio link quality and channel condition measurement

Radio link quality and channel condition measurements include the link quality measurement of each pair of neighbours and the channel condition measurement of each channel. The collected information may be accumulated at the DLSL and reported through the DMAP. Link quality and channel condition are used to dynamically allocate communication resources.

Generally, the radio link quality measurement is performed according to the following performance indices:

a) average signal level received from the neighbour devices in the statistics duration;

b) non-broadcast packets sent to the neighbour devices in the statistics duration;

c) count of ACK that is not received in the statistics duration; and

d) count of the non-broadcast packets received from the neighbour devices in the statistics duration.

The channel condition measurement is performed according to the following performance data:

a) LQI per link;

b) packet loss rate per link, which is determined by the count of received ACK and packets that are sent; and

c) average count of retries per link.

8.4.10 Security

The WIA-PA DLSL applies the MIC mechanism and encryption technology to guarantee the integrity and confidentiality of the communication. See Clause 11 for details.

8.4.11 Country code

The regulatory considerations of DLME are configured through its CountryCode attribute (see 6.9.1.2.1), which is a 16-bit value where:

a) bits 0-9 provide a 10-bit country code, using ISO 3166-1 numeric three-digit country codes;

b) bit 10 indicates whether European spectrum regulation apply (0 = no, 1 = yes). A DLME shall operate in compliance with European spectrum regulation when Bit10 = 1;

c) bit 11 indicates whether FCC rules apply (0 = no, 1 = yes). A DLME should operate in compliance with FCC rules when Bit11 = 1;

d) bit 12 indicates whether a 10 dBm EIRP limit applies (0 = no, 1 = yes). A DLME should limit its emissions to ≤ 10 dBm EIRP when Bit12 = 1;

e) bit 13 indicates whether the DLME is considered to be operating under adaptive or non-adaptive rules (0 = non-adaptive, 1 = adaptive);

f) bit 14 indicates whether the DLME is considered to be operating under frequency-hopping spread-spectrum rules (0 = not-FHSS-rules, 1 = FHSS-rules);

g) bit 15 indicates whether the value of this attribute is fixed (0 = no, 1 = yes) while the DLME is operational. Once this bit is set, any subsequent attempt to modify this attribute shall be rejected except when it occurs while the DLME is being provisioned.

NOTE 1 This feature supports device operation in regulatory regimes that prohibit the ability to reconfigure a device in such a way that it would violate regulatory restraints, and while still permitting the repair or refurbishment of devices with subsequent resale into markets where other regulations apply.

When no specific country of intended use has been identified, the default value for CountryCode shall be 0x3C00, indicating that a device in the default state should comply with ETSI rules, FCC rules, the 10 dBm EIRP limit, and be classified as an adaptive non-FHSS device. See Annex D.

NOTE 2 This default value of CountryCode ensures that the equipment, before it has been provisioned, meets the regulatory requirements of most regions in which it might be deployed. The value of CountryCode could be changed during wired provisioning to reflect the intended regulatory regime that applies to the device's locale of deployment.

8.4.12 DLSL state machine

The WIA-PA DLSL state machine is divided into two state machines: one for device joining and the other for in-network running.

The WIA-PA DLSL state machine for device joining is shown in Figure 33.

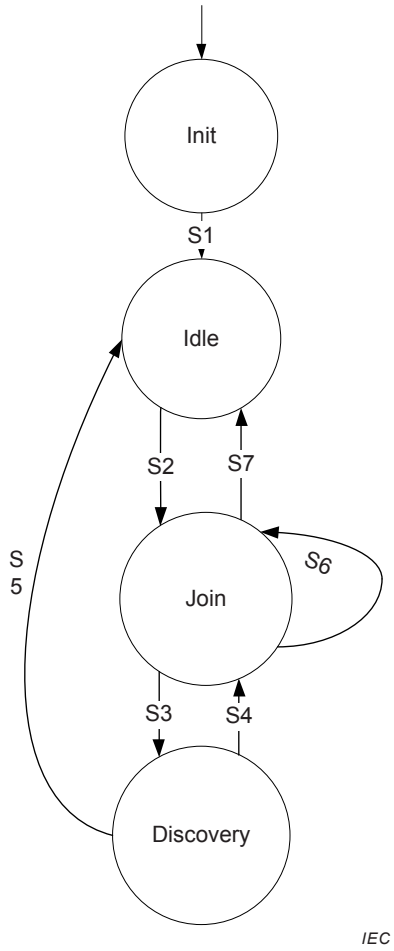

Figure 33 – DLSL state machine for device joining

The DLSL state transitions for device joining are shown in Table 39.

Table 39 – DLSL state transitions for device joining

Sequence number #	Current state	Event or condition	Actions	Next state
S1	Init		Initiation	Idle
S2	Idle	Devicestate = Not joined		Join
S3	Join	DLME-DISCOVERY.request()	Parsing the discovery request message and invoke MLME-SCAN.request()	Discovery
S4	Discovery	MLME-SCAN.confirm (status = SUCCESS) and MLME-BEACON-NOTIFY.indication()	Parsing the scan confirm message and beacon notification message; invoke DLME-DISCOVERY. confirm(SUCCESS)	Join
S5	Discovery	MLME-SCAN.confirm (status != SUCCESS) or no confirm received		Idle
S6	Join	DLME-JOIN.reqeust()	Setting Devicestate = Joining; generating association request message and Invoke MLME-ASSOCIATE. request()	Join
S7	Join	MLME-ASSOCIATE.confirm() or no confirm received	If MLME-ASSOCIATE.confirm() received, parsing the associate confirm message: if status = SUCCESS, invoke DLME-JOIN.confirm(Status = SUCCESS) and setting Devicestate = Joined; if status != SUCCESS, invoke DLME-JOIN.confirm(). If no confirm received, no actions.	Idle

After a device joined WIA-PA network, that is Devicestate = Joined, the DLSL runs its state machine for in-network running.

The WIA-PA DLSL state machine for in-network running is shown in Figure 34.

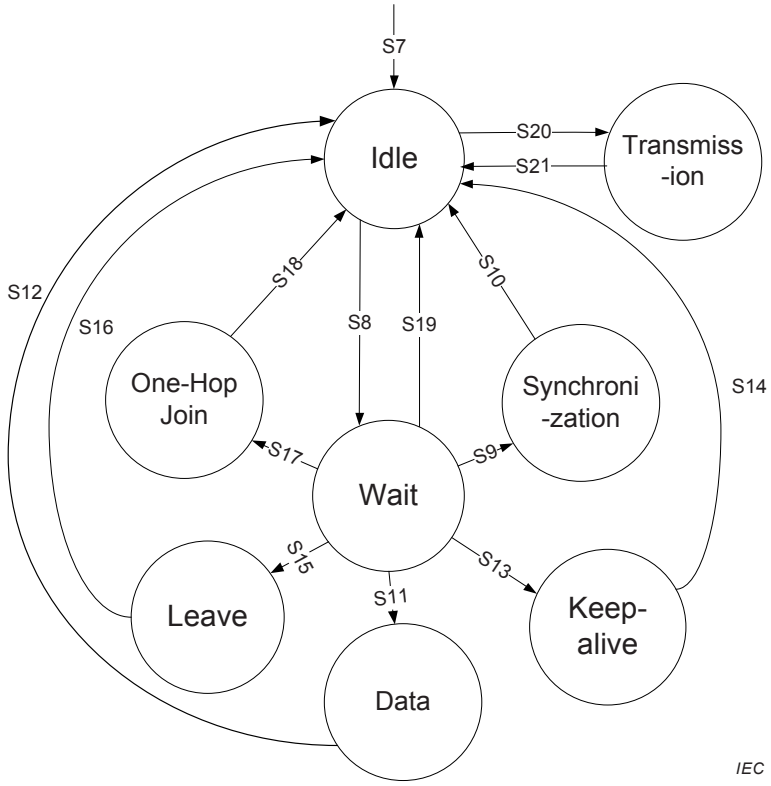

Figure 34 – DLSL state machine for in-network running

The DLSL state transitions for in-network device are shown in Table 40.

Table 40 – DLSL state transitions for in-network running *(1 of 3)*

Sequence number #	Current state	Event or condition	Actions	Next state
S8	Idle	Timeslot timeout LinkType = Receiving or LinkType = Transmit-shared	Invoke MLME-RX-ENABLE. request ()	Wait
S9	Wait	From MAC: MLME-COMM-STATUS.indication() or MCPS-DATA.indication(); From DMAP: DLME-TIME-SYN.request()		Synchronization
S10	Synchronization		From MAC: parsing MLME communication status indication message and invoke DLME-TIME-SYN.confirm() or parsing DLPDU from MAC data frame.If Frametype = 1 and Command frame identifier = 11 in DLPDU, invoke DLME-TIME-SYN.indication(); From DMAP: generating DLSL time request command frame and invoke MCPS-DATA.request().	Idle
S11	Wait	From MAC: MCPS-DATA.confirm()or MCPS-DATA.indication(); From NL: DLDE-DATA.request()		Data

Table 40 *(2 of 3)*

Sequence number #	Current state	Event or condition	Actions	Next state
S12	Data		From MAC: invoke DLSL-DATA.confirm() or parsing DLPDU. If Frame type = 0 in DLPDU, evaluate the LinkQuality and PacketLossRate and invoke DLDE-DATA.indication(); From NL: MCPS-DATA.request()	Idle
S13	Wait	From MAC: MLME-COMM-STATUS.indication() or MCPS-DATA.indication(); From DMAP: DLME-KEEP-LIVE.request()		Keep-alive
S14	Keep-alive		From MAC: parsing MLME communication status indication message and invoke DLME - KEEP-LIVE.confirm() or parsing DLPDU from MAC data frame.If Frametype = 1 and Command frame identifier = 10 in DLPDU, invoke DLME-KEEP-ALIVE.indication(); From DMAP: invoke MLME-KEEP-LIVE.request()	Idle
S15	Wait	From MAC: MLME-DISASSOCIATE.indication() or MLME-DISASSOCIATE.confirm(); From DMAP: DLME-LEAVE.request()		Leave
S16	Leave		From MAC: parsing DLSL leave indication message or confirm message and invoke DLME-LEAVE.indication() or DLME-LEAVE.confirm(); From DMAP: parsing DLSL leave request message and invoke MLME-DISASSOCIATE.request()	Idle
S17	Wait	From MAC:MLME-ASSOCIATE.indication() or MLME-ASSOCIATE.confirm() From DMAP: DLME-JOIN. request () or DLME-JOIN. response ()		One-Hop Join
S18	One-Hop Join		From MAC: parsing MAC association indication message or confirm message and invoke DLME-JOIN.indication() or DLME-JOIN.confirm(); From DMAP: parsing DMAP join request or response message and invoke MLME-ASSOCIATE.request() or MLME-ASSOCIATE.response()	Idle
S19	Wait	From MAC: MLME-COMM-STATUS.indication() and no frame received or received failed From DMAP and NL: no message received	From MAC: invoke DLME-COMM-STATUS.indication(FAILURE_ELSE)	Idle

Table 40 *(3 of 3)*

Sequence number #	Current state	Event or condition	Actions	Next state
S20	Idle	Timeslot timeout LinkType = Transmitting or LinkType = Transmit-shared	Invoke MCPS-DATA.request()	Transmission
S21	Transmission	MCPS-DATA.confirm()	Reschedule or retry frame if transmit failed; Release frame's buffer if MCPS-DATA.confirm (SUCCESS);	Idle

See 8.5 for DLSL related primitives; see 8.7 for DLSL frame formats; see 9.4 and 9.5 for NL related primitives; see IEEE STD 802.15.4-2011, 6.2 for MAC related primitives; see 6.9.1.2.1 for Devicestate.

The states in DLSL state machine are specified as follows.

a) Join state

This state handles the joining procedure of a device. After a device joins the network, the DLSL runs its state machine as in Figure 34.

b) Leave state

This state handles the leaving procedure of a device.

c) Idle state

After a device joins the network, the DLSL enters the Idle state. The following transitions may occur while the DLSL is in the Idle state:

– When a timeslot arrives, the DLSL enters either the Transmission state or the Receive state according to the link options (transmitting, receiving or transmit-shared).

d) Transmission state

When the timeslot arrives and the link option is a transmit link, the DLSL enters the Transmission state. The following events will happen while the DLSL is in the Transmission state.

– Successful propagation of a frame with a broadcast/multicast destination address happens as soon as the frame is transmitted. The frame's buffer is then released.

– Successful propagation of a frame with a unicast destination address occurs when a validated and successful confirmation is received from the local MAC layer. This indicates that message propagation has been completed successfully; the frame's buffer is then released.

– If a failure confirmation is received from the local MAC layer, the frame should be retried.

– If a response with an error or no response is received, the frame should be re-scheduled or retried.

e) Wait state

The frames that a device can receive include frames whose final destination addresses are this device and frames whose final destination addresses are not this device. The functions of the Wait state are to receive, check and process the frame. The following transitions may occur while the DLSL is in the Wait state.

– If there is no frame received, the DLSL will evaluate the link and return to the Idle state.

– If a beacon frame or a time synchronization command frame is received, then the Synchronization state is entered.

– If a keep-alive command frame is received, then the Keep-alive state is entered.

– If a DMAP leave request is received, then the Leave state is entered.

- If a join command is received from DMAP, then the One-Hop Join state is entered.
- Upon successful frame reception, the DLSL will process the frame according to frame priority (receive or discard), and then the device returns directly to the Idle state.

f) Data state

In this state, the DLSL data is generated or parsed. After received the DLSL data frame, the LinkQuality and PacketLossRate are evaluated. After generating the DLSL data frame, the DLSL enters Wait state.

g) Synchronization state

In this state, time synchronization is executed after a device receives a beacon frame or a time synchronization command frame, and then the DLSL enters Idle state.

h) Keep-alive state

In this state, a neighbour node is marked as keep-alive after a device receives a keep-alive command frame, and then the DLSL enters Idle state.

i) One-Hop Join state

In this state, the one-hop join process is supported by GW device.

8.5 Data link sub-layer data services

8.5.1 General

The DLDE-SAP supports the point-to-point transmission of DLSL Protocol Data Units (DLPDUs) between devices. The primitives supported by DLSL data services include DLDE-DATA.request, DLDE-DATA.confirm, and DLDE-DATA.indication.

8.5.2 DLDE-DATA.request

The DLDE receives the payload from the NL through a DLDE-DATA.request and adds it to the message queue of the DLSL.

The semantics of DLDE-DATA.request are as follows:

```
DLDE-DATA.request (
                    NetworkID,
                    SrcAddrMode,
                    SrcAddr,
                    DstAddrMode,
                    DstAddr,
                    Priority,
                    Type,
                    PayloadLength,
                    Payload,
                    PayloadHandle
                  )
```

Table 41specifies the parameters for DLDE-DATA.request.

Table 41 – DLDE-DATA.request parameters

Name	Data type	Valid range	Description
NetworkID	Unsigned8	0 to 255	Network identifier
SrcAddrMode	Unsigned8	0 to 3	Mode of source address: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
SrcAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Source address 64-bit long address is used only in device joining process, and 16-bit short address is used generally.
DstAddrMode	Unsigned8	0 to 3	Mode of destination address: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
DstAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Destination address, 16- or 64-bit
Priority	Unsigned8	0 to 15	Priority of the payload
Data type	Unsigned8	0 to 1	0 = Intra-cluster transmission; 1 = Inter-cluster transmission.
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octetstring		Payload
PayloadHandle	Unsigned8	0 to 255	Handle allocated when call DLDE-DATA.request

8.5.3 DLDE-DATA.confirm

The semantics of DLDE-DATA.confirm are as follows:

```
DLDE-DATA.confirm (
                    PayloadHandle,
                    Status
                  )
```

Table 42 specifies the parameters for DLDE-DATA.confirm.

Table 42 – DLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0 to 255	Handle allocated when call DLDE-DATA.confirm
Status	Unsigned8	0 to 255	Result of the data transmission of DLSL: 0 = SUCCESS; 1 = TRANSACTION_OVERFLOW; 2 = TRANSACTION_EXPIRED; 3 = NO_ACK; 4 = CHANNEL_ACCESS_FAILURE; 5 = UNAVAILABLE_KEY; 6 = FAILED_SUCURITY_CHECK; 7 = INVALID_PARAMETER; Others are reserved. See Table 43 for more detail

Table 43 – Status table

ID	Value	Description
0	SUCCESS	The requested operation is completed successfully. For a transmission request, this value indicates a successful transmission.
1	TRANSACTION_OVERFLOW	Not enough space for storing transactions
2	TRANSACTION_EXPIRED	Transaction has expired and its information is discarded
3	NO_ACK	No acknowledgement message is received after macMaxFrameRetries
4	CHANNEL_ACCESS_FAILURE	Cannot transmit due to channel access failure
5	UNAVAILABLE_KEY	No valid key in access control list
6	SECURITY_ERROR	Cryptographic processing of the received secured frame failed
7	INVALID_PARAMETER	A parameter in the primitives out of value range
8	NO-BEACON	A scan operation failed to find any network beacons
9 to 255	Reserved	

8.5.4 DLDE-DATA.indication

The semantics of DLDE-DATA.indication are as follows:

```
DLDE-DATA.indication (
                        NetworkID,
                        ScrAddrMode,
                        SrcAddr,
                        Type,
                        Priority,
                        PayloadLength,
                        Payload,
                        PayloadLinkQuality,
                      )
```

Table 44 specifies the parameters for DLDE-DATA.indication.

Table 44 – DLDE-DATA.indication parameters

Name	Data type	Valid range	Description
NetworkID	Unsigned8	0 to 255	Network identifier
SrcAddrMode	Unsigned8	0 to 3	The source address mode. Four options are available: 0 = No address; 1 = Reserved; 2 = 16-bit short address; 3 = 64-bit long address.
SrcAddr	Unsigned16/64	0 to 65 535 or (2^{64}-1)	Source address, using 64-bit long address only in device join process, and using 16-bit short address generally
Type	Unsigned8	0 to 1	0 = Intra-cluster transmission; 1 = Inter-cluster transmission.
Priority	Unsigned8	0 to 15	Priority of payload
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octetstring		Payload
PayloadLinkQuality	Unsigned8	0 to 255	LQI value measured during reception of the DLPDU Lower values represent lower link quality.

8.5.5 Time sequence of DLSL data service

Figure 35 shows the data transaction sequence of transporting one data packet from the source device to the destination device.

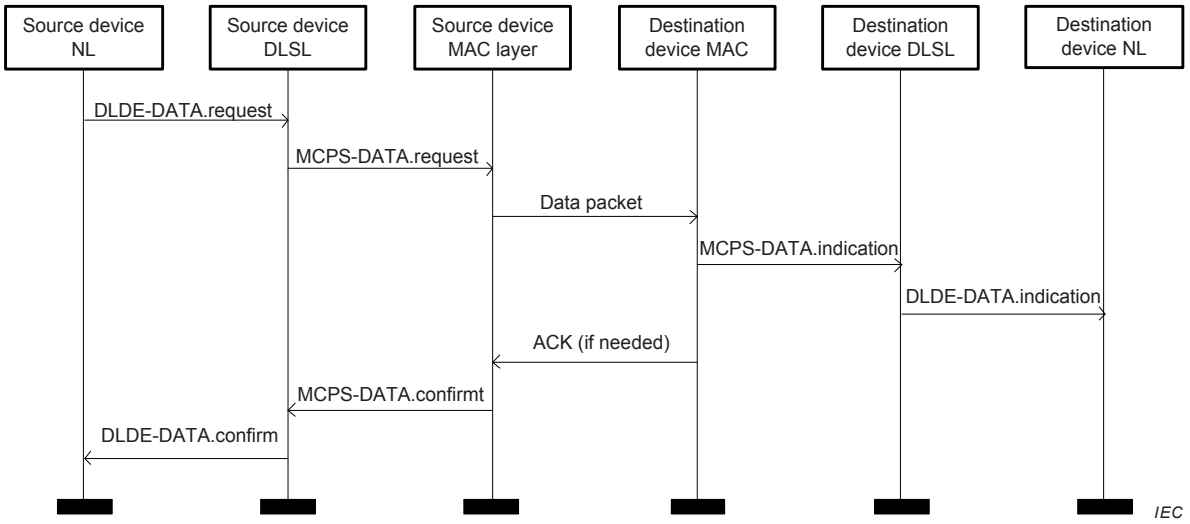

Figure 35 – Time sequence of data service

DLDE-DATA.request is generated by a local Network Layer Data Entity (NLDE) when a Network Protocol Data Unit (NPDU) is to be transferred to a peer NLDE.

On receipt of DLDE-DATA.request, DLDE inserts the message into the sending buffer and begins sending the supplied DLPDU.

DLDE-DATA.indication is generated by the DLDE of a destination device and issued to the NLDE on receipt of a data frame at the local DLDE that passes the appropriate message filtering operations.

DLDE-DATA.confirm is generated by the DLDE of a source device in response to DLDE-DATA.request. DLDE-DATA.confirm returns a status indicating the result of the transmission.

See Figure 27 in IEEE STD 802.15.4-2011 for the detailed message sequence chart between peer MAC entities.

8.6 Data link sub-layer management services

8.6.1 General

The upper layer uses the DLME-SAP to send management commands to the DLSL. The DLSL management services include subnet discovery, device joining and leaving, channel condition collection and report, neighbour information collection and report, and time synchronization.

8.6.2 Network discovery services

8.6.2.1 General

The DLSL network discovery services are used to scan a given list of communication channels. One device may use the network discovery services to search for cluster heads sending beacon frames within its communication scope. The primitives supported by the DLSL network discovery services include DLME-DISCOVERY.request and DLME-DISCOVERY.confirm.

8.6.2.2 DLME-DISCOVERY.request

DLME-DISCOVERY.request is used to request a device to scan channels.

The semantics of DLME-DISCOVERY.request are as follows:

```
DLME-DISCOVERY.request (
                        ScanChannels,
                        ScanDuration
                    )
```

Table 45 specifies the parameters for DLME-DISCOVERY.request.

Table 45 – DLME-DISCOVERY.request parameters

Name	Data type	Valid range	Description
ScanChannels	Unsigned32	32-bit Bits-Map	Bit 11 to bit 26 indicate IEEE STD 802.15.4-2011 channels of 2,4 GHz frequency band; others are set. See IEEE STD 802.15.4-2011, 6.2.10 for ScanChannels.
ScanDuration	Unsigned8	0 to 14	A value used to calculate the length of time to spend scanning each channel of ED, active, and passive scans This parameter is ignored for orphan scans. The time spent scanning each channel is: aBaseSuperframeDuration × (2^n + 1) symbols, where *n* is the value ofScanDuration parameter.
See Table 51 in IEEE STD 802.15.4-2011 standard for the definition and value set of aBaseSuperframeDuration.			

8.6.2.3 DLME-DISCOVERY.confirm

DLME-DISCOVERY.confirm is used to respond to DLME-DISCOVERY.request.

The semantics of DLME-DISCOVERY.confirm are as follows:

```
DLME-DISCOVERY.confirm(
                        Status,
                        NetworkCount,
                        NetworkDescriptor
                    )
```

Table 46 specifies the parameters for DLME- DISCOVERY.confirm. Table 47 specifies the network descriptor list.

Table 46 – DLME- DISCOVERY.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Scan results: 0 = SUCCESS; 7 = INVALID_PARAMETER; 8 = NO_BEACON; Others are reserved. See Table 43 for more detail
NetworkCount	Unsigned8	0 to 255	The count of active network found during scan
NetworkDescriptor	Network descriptor list	0 to NetworkCount	Network Descriptor list of every network found, see Table 47

Table 47 – Network descriptor list

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0 to 31	Logic channel used for joining, chosen from valid channels supported by PHY
BeaconOrder	Unsigned8	0 to 15	The frequency sending beacon frame
SuperframeOrder	Unsigned8	0 to 15	Active period length of the superframe
PermitJoining	Boolean	0, 1	Whether routing device permits field device to join: 0 = No permit; 1 = At least one device is permitted to join

If the scan is successful, DLME-DISCOVERY.confirm returns SUCCESS; however if no beacons are found, DLME-DISCOVERY.confirm returns NO_BEACON; if there are some errors or invalid parameters in DLME-DISCOVERY.request, DLME-DISCOVERY.confirm returns INVALID_PARAMETER.

8.6.2.4 Time sequence of subnet discovery

The time sequence for subnet discovery is shown in Figure 36. See IEEE STD 802.15.4-2011 for the primitives of MLME-SCAN.request and MLME-SCAN.confirm.

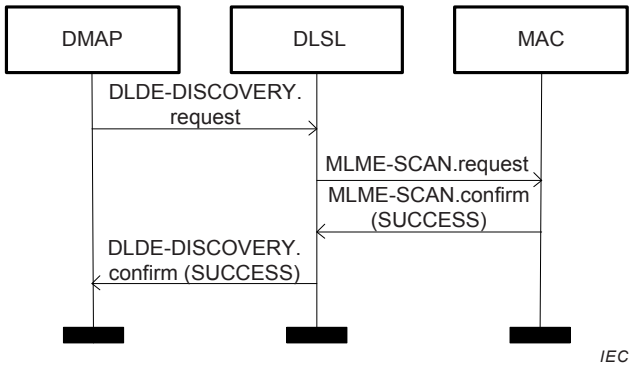

Figure 36 – Time sequence of network discovery

See Figure 12 and Figure 13 in IEEE STD 802.15.4-2011 for the detailed message sequence chart between peer MAC entities.

8.6.3 Device joining services

8.6.3.1 General

There are two instances requiring device joining services:

a) a new field device joining the star network; or

b) a new routing device joining the mesh network. The primitives supported by DLSL device joining services include DLME-JOIN.request, DLME-JOIN.indication, DLME-JOIN.response, and DLME-JOIN.confirm.

8.6.3.2 DLME-JOIN.request

DLME-JOIN.request is used for a device to join the network (star or mesh).

The semantics of DLME-JOIN.request are as follows:

```
DLME-JOIN.request (
                     LogicalChannel,
                     JoinAddr,
                     PhyAddr,
                     DeviceType
                   )
```

Table 48 specifies the parameters for DLME-JOIN.request.

Table 48 – DLME-JOIN.request parameters

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0 to 31	Logic channel used for joining, chosen from valid channels supported by PHY
JoinAddr	Unsigned16	0 to 65 535	The short address of routing device or gateway device that accepts joining request
PhyAddr	Unsgned64	0 to (2^{64}-1)	Physical address of the new device waiting to join
DeviceType	Unsigned8	0 to 255	Type of the new device waiting to join: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

8.6.3.3 DLME-JOIN.indication

DLME-JOIN.indication is used to inform the NLME of a routing device or to inform the gateway device that the joining request from one device has been successfully received.

The semantics ofDLME- JOIN.indication are as follows:

```
DLME-JOIN.indication (
                        PhyAddr,
                        DeviceType
                    )
```

Table 49 specifies the parameters for DLME-JOIN.indication.

Table 49 – DLME-JOIN.indication parameters

Name	Data type	Valid range	Decription
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of new device waiting to join
DeviceType	Unsigned8	0 to 255	Type of device waiting to join: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

8.6.3.4 DLME-JOIN.response

DLME-JOIN.response is the response of DLME-JOIN.indication.

The semantics of DLME-JOIN.response are as follows:

```
DLME-JOIN.response (
                        PhyAddr,
                        ShortAddr,
                        TimeSource,
                        Status
                    )
```

Table 50 specifies the parameters for DLME-JOIN.response.

Table 50 – DLME-JOIN.response parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of device waiting to join
ShortAddr	Unsigned16	0 to 65 535	Short address allocated by the NM to device waiting to join
TimeSource	Unsigned8	0 to 1	Whether this device is set as time source: 0 = Not time source; 1 = Time source.
Status	Unsigned8	0 to 255	Result of joining request: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved.

8.6.3.5 DLME-JOIN.confirm

DLME-JOIN.confirm reports the joining result to the DMAP.

The semantics of DLME-JOIN.confirm are as follows:

```
DLME-JOIN.confirm (
                    ShortAddr,
                    Status
                  )
```

Table 51 specifies the parameters for DLME-JOIN.confirm.

Table 51 – DLME-JOIN.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Short address allocated by NM for device waiting to join
Status	Unsigned8	0 to 255	Result of joining request: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved.

8.6.3.6 Time sequence for device joining in the network

See 9.5.3 for the detailed joining process.

8.6.4 Device leaving services

8.6.4.1 General

There are two instances requiring device leaving services: (1) A field device leaving the star network or (2) a routing device leaving the mesh network. The primitives supported by DLSL network leaving services include DLME-LEAVE.request, DLME-LEAVE.indication, and DLME-LEAVE.confirm.

8.6.4.2 DLME-LEAVE.request

The semantics of DLME-LEAVE.request are as follows:

DLME-LEAVE.request (
ShortAddr
)

Table 52 specifies the parameters for DLME-LEAVE.request.

Table 52 – DLME-LEAVE.request parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	The short address of the device asking to leave

8.6.4.3 DLME-LEAVE.indication

DLME-LEAVE.indication is used to indicate to the upper layer that a device leaving request has been received.

The semantics of DLME-LEAVE.indication are as follows:

DLME-LEAVE.indication (
ShortAddr
)

Table 53 specifies the parameters for DLME-LEAVE.indication.

Table 53 – DLME-LEAVE.indication parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	The short address of the device asking to leave

8.6.4.4 DLME-LEAVE.confirm

DLME-LEAVE.confirm is used to report the result of DLME-LEAVE.request.

The semantics of DLME-LEAVE.confirm are as follows:

DLME-LEAVE.confirm (
Status
)

Table 54 specifies the parameters for DLME-LEAVE.confirm.

Table 54 – DLME-LEAVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of leaving request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.4.5 Time sequence for device leaving

See IEEE STD 802.15.4-2011 for the following primitives: MLME-DISASSOCIATE.request, MLME-DISASSOCIATE.confirm, and MLME-DISASSOCIATE.indication.

The time sequences for devices leaving the network are listed as follows.

a) The time sequence for routing devices leaving the network

Routing devices connect to both the mesh network and the star network. Therefore, the process of routing devices leaving the network includes routing devices leaving the mesh network and routing devices leaving the star network. According to the leaving originator, the process of routing devices leaving the network includes active leaving and passive leaving.

b) The time sequence for field devices leaving the network

According to the leaving source, the process of field devices leaving the network includes active leaving and passive leaving.

See 9.5.4 for detailed process of device leaving.

8.6.5 DLME-CHANNEL-CONDITION.indication

DLME-CHANNEL-CONDITION.indication is used by DLSL to report channel condition information to DMAP. The reported performance data includes LQI, packet loss rate, and count of retries.

The semantics of DLME-CHANNEL-CONDITION.indication are as follows:

DLME-CHANNEL-CONDITION.indication (

ChannelNum,
ChannelStructure
)

Table 55 specifies the parameters for DLME-CHANNEL-CONDITION.indication.

Table 55 – DLME-CHANNEL-CONDITION.indication parameters

Name	Data type	Valid range	Description
ChannelNum	Unsigned8	0 to 255	Count of channels
ChannelStructure	ChanCon_Struct		Information of channels

8.6.6 DLME-NEIGHBOUR-INFO.indication

DLME-NEIGHBOUR-INFO.indication is used to report the collected neighbour information to the DMAP and is used to update the neighbour attributes.

The semantics of DLME-NEIGHBOUR-INFO.indication are as follows:

DLME-NEIGHBOUR-INFO.indication (

NeighbourCount,
NeighbourStructure
)

Table 56 specifies the parameters for DLME-NEIGHBOUR-INFO.indication.

Table 56 – DLME-NEIGHBOUR-INFO.indication parameters

Name	Data type	Valid range	Description
NeighbourCount	Unsigned8	0 to 255	Count of neighbour devices
NeighbourStructure	Neighbour_Struct structure		Information of neighbours

8.6.7 DLME-COMM-STATUS.indication

DLME-COMM-STATUS.indication is used to report the communication status to the upper layer.

The semantics of DLME-COMM-STATUS.indication are as follows:

DLME-COMM-STATUS.indication (

PhyAddr,
Status

)

Table 57 specifies the parameters for DLME-COMM-STATUS.indication.

Table 57 – DLME-COMM-STATUS.indication parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	Address of new device that has just joined the network
Status	Unsigned8	0 to 255	The joining status of a device: 0 = SUCCESS; 1 = FAILURE_TOP_DISMATCH; 2 = FAILURE_ELSE; Others are reserved.

8.6.8 Keep-alive services

8.6.8.1 DLME-KEEP-LIVE.request

DLME-KEEP-LIVE.request is used to send Keep-alive command frames that are requested by the DMAP.

The semantics ofDLME-KEEP-LIVE.request are as follows:

DLME-KEEP-LIVE.request (
)

8.6.8.2 DLME-KEEP-LIVE.confirm

DLME -KEEP-LIVE.confirm is used to respond to DLME -KEEP-LIVE.request.

The semantics of DLME -KEEP-LIVE.confirm are as follows:

DLME -KEEP-LIVE.confirm (

Status
)

Table 58 specifies the parameters for DLME -KEEP-LIVE.confirm.

Table 58 – DLME -KEEP-LIVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of the keep alive: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.8.3 DLME-KEEP-LIVE.indication

DLME-KEEP-LIVE.indication is used to inform the DMAP that the keep-alive command frame has been successfully received.

The semantics of DLME -KEEP-LIVE.indication are as follows:

DLME-KEEP-LIVE.indication (

SrcAddr
)

Table 59 specifies the parameters for DLME -KEEP-LIVE.indication.

Table 59 – DLME -KEEP-LIVE.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address

8.6.9 Time synchronization services

8.6.9.1 DLME-TIME-SYN.request

DLME-TIME-SYN.request is used to send time synchronization command frames that are requested by the DMAP.

The semantics of DLME-TIME-SYN.request are as follows:

DLME-TIME-SYN.request (

TimeValue
)

Table 60 specifies the parameters for DLME-TIME-SYN.request.

Table 60 – DLME-TIME-SYN.request parameters

Name	Data type	Valid range	Description
TimeValue	Unsigned16	0 to (2^{16}-1)	Time difference between the transmission beginning and the timeslot beginning (in microsecond)

8.6.9.2 DLME-TIME-SYN.confirm

DLME-TIME-SYN.confirm is used to respond to DLME-TIME-SYN.request.

The semantics of DLME-TIME-SYN.confirm are as follows:

DLME-TIME-SYN.confirm (

Status
)

Table 61 specifies the parameters for DLME-TIME-SYN.confirm.

Table 61 – DLME -TIME-SYN.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of time synchronization: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

8.6.9.3 DLME-TIME-SYN.indication

DLME-TIME-SYN.indication is used to inform the DMAP that the time synchronization command frame has been successfully received.

The semantics of DLME-TIME-SYN.indication are as follows:

DLME-TIME-SYN.indication (
SrcAddr,
TimeValue
)

Table 62 specifies the parameters for DLME-TIME-SYN.indication.

Table 62 – DLME-TIME-SYN.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address
TimeValue	Unsigned16	0 to (2^{16}-1)	Time difference between the transmission beginning and the timeslot beginning (in microsecond)

8.7 DLSL frame formats

8.7.1 General frame format

The DLSL general frame format is illustrated in Figure 37.

<table>
<tr><th colspan="3">MAC Protocol Data Unit (MPDU)</th></tr>
<tr><th rowspan="2">MHR
(IEEE STD 802.15.4-2011 MAC header)</th><th colspan="2">DLPDU</th><th rowspan="2">FCS</th></tr>
<tr><th>DLSL frame control</th><th>DLSL payload</th></tr>
</table>

IEC

Figure 37 – General frame format

The DLSL frame is composed of:

a) IEEE STD 802.15.4-2011 MAC Header (MHR) (See IEEE STD 802.15.4-2011, 5.2.1);

b) DLSL frame control (see Table 63);

c) DLSL payload; and

d) Frame Check Sequence (FCS).

See 11.3.2 for the content of security.

The format of the WIA-PA DLSL frame control is shown in Table 63.

Table 63 – DLSL frame control field

	DLSL frame control field			
Length in bit(s)	1	1	1	5
Subfield name	Frame type	Time source	Security enable	Index of main channel

The subfields in Table 63 are listed as follows.

a) The Frame type subfield has 1-bit length and is used to specify the frame type. If a frame is a data frame, this subfield is set to 0; otherwise, this subfield is set to 1.

b) The Time source subfield has 1-bit length and is used to indicate whether this device is a time source. If the device is a time source, this subfield is set to 1; otherwise, this subfield is set to 0.

c) The Security enable subfield has 1-bit length and is used to instruct DLSL whether or not to use the security mechanism. A value of 0 means that the security mechanism is disabled and a value of 1 means that the security mechanism is enabled.

d) The Index of the main channel is used to indicate the current communication channel.

All multiple octet fields should be transmitted or received with the least significant octet first and each octet should be transmitted or received with the Least Significant Bit (LSB) first. The same transmission order should apply to data fields transferred between layers.

8.7.2 Date frame format

The format of the DLSL data frame is shown in Table 64.

Table 64 – Date frame format

	Date frame format	
Length in octet(s)	1	Variable length
Field name	DLSL frame control	Data payload

The subfields in Table 64 are listed as follows:

a) the DLSL frame control is defined in Table 63; and

b) the Data payload has variable length and is filled with the DLSL data.

8.7.3 Command frame format

8.7.3.1 General command frame format

The general command frame format is shown in Table 65.

Table 65 – General command frame format

	General command frame format		
Length in octet(s)	1	1	Variable length
Field name	DLSL frame control	Command frame identifier	Command payload

The subfields in Table 65 are listed as follows:

a) the DLSL frame control is defined in Table 63;

b) the Command frame identifier is defined in Table 66; and

c) the Command payload has variable length and is filled with the DLSL management data.

8.7.3.2 DLSL command frame

8.7.3.2.1 General

The DLSL command frames are defined and shown in Table 66.

Table 66 – DLSL command frame

	DLSL command frame		
Command frame identifier	**Command name**	**User**	**Description**
1	Keep-alive command frame	Gateway device/Routing device/Field device	Indicating an existing device
2	Time synchronization command frame	Gateway device/Routing device	Realizing time synchronization of the whole network
3 to 255	Reserved	Reserved	To be extended

The Keep-alive command frame facilitates connection maintenance between neighbour devices. After devices join the network, their DMAPs send keep-alive command frame with the keep-alive request primitive. The GW sends the keep-alive command frame during the inter-cluster communication period in order to indicate that it is alive. Routing devices send the keep-alive command frames during the intra-cluster communication period in order to indicate that they are alive to their cluster members; in addition, routing devices send keep-alive command frames during the inter-cluster communication period in order to indicate that they are alive to other routing devices and to the GW. Field devices send the keep-alive command frames during the intra-cluster communication period in order to indicate that they are alive to their cluster heads.

In order to guarantee the reliability of the TDMA communication mode, devices in a network should synchronize with the time source. Time errors among devices are inevitable in spite of any hardware time sources. In order to overcome time clock drifting, the WIA-PA network performs two kinds of time synchronization: IEEE STD 802.15.4-2011 beacon and specifically designed time synchronization command frame. In the star-only network, the GW is the UTC time source. In the combination of star and mesh network, the GW is the UTC time source of the mesh part and all routing devices synchronize with the gateway device; each routing device is the time source for its field devices in the star part, which synchronize with their routing devices.

This document specifies that the maximum synchronization error is less than 10 % of the basic timeslot length of the maximum superframe duration.

8.7.3.2.2 Keep-alive command frame

The format of the keep-alive command frame is shown in Table 67.

Table 67 – Format of keep-alive command frame

	Format of keep-alive command frame	
Length in octet(s)	1	1
Field name	DLSL frame control	Command frame identifier

The subfields in Table 67 are listed as follows.

a) See Table 63 for DLSL frame control.

b) Command frame identifier is 10.

8.7.3.2.3 Time synchronization command frame

The time synchronization command frame is used to synchronize the entire network. The gateway device and the routing devices send the time synchronization command frames periodically.

If the network topology is star-only, the GW broadcasts the time synchronization command frame periodically. If the network topology is combination of mesh and star, the GW unicasts the time synchronization command frame during the inter-cluster communication period; routing devices broadcast the time synchronization command frames during the intra-cluster communication period to synchronize their star networks and unicast them during the inter-cluster communication period to synchronize the mesh network.

The format of the time synchronization command frame is shown in Table 68.

Table 68 – Format of time synchronization command frame

	Format of time synchronization command frame		
Length in octet(s)	1	1	2
Field name	DLSL frame control	Command frame identifier	Calibrated value of TimeValue

The subfields in Table 68 are defined as follows:

a) see Table 63 for DLSL frame control;

b) command frame identifier is 11; and

c) see Table 60 forCalibrated value of TimeValue.

9 Network layer

9.1 General

The WIA-PA network layer (NL) receives and transports packets over networks, provides interfaces to the ASL, and carries out network layer management, configuration and control.

9.2 Protocol stack

The protocol stack of the WIA-PA network layer is shown in Figure 38.

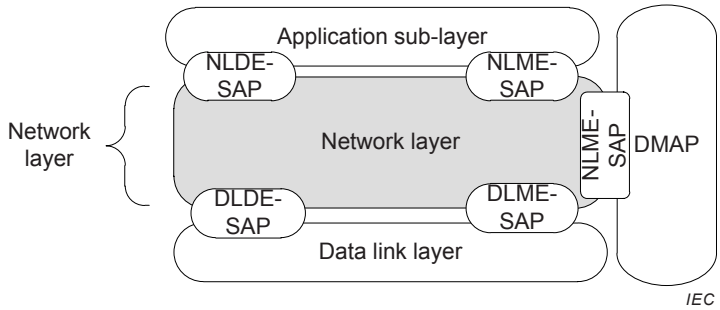

Figure 38 – WIA-PA network layer protocol stack

The layers and entities in Figure 39 are defined as follows:

a) the NL defines the NLDE and the Network Layer Management Entity (NLME);
b) the NLDE provides the service interface through which the ASL transmits and receives data; and
c) the NLME provides the service interfaces through which the layer management functions are invoked by the upper layer and DMAP.

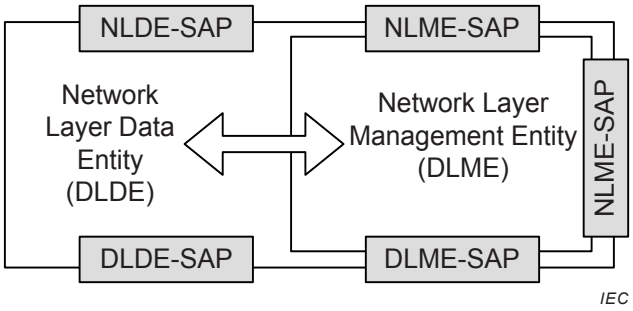

Figure 39 – WIA-PA network layer reference model

The NL provides two kinds of services that are accessed through the following two SAPs:

a) the network layer data service, accessed through the NLDE SAP (NLDE-SAP);
b) the network layer management service, accessed through the NLME SAP (NLME-SAP).

9.3 Function description

9.3.1 General

The NL is designed to perform the following functions:

a) addressing,
b) routing,
c) communication resource allocation,
d) packet lifecycle management,
e) management for device joining and leaving network,
f) end-to-end network performance monitoring, and
g) fragmentation and reassembly.

9.3.2 Addressing

Each device has a global unique 64-bit long address and a 16-bit short address. Each device joins the network by using the long address and communicates with other devices by using the short address after joining the network. The 16-bit short address is indicated as x.y, where x and y are 8-bit integers. The most significant 8-bit in the 16-bit short address is the cluster address, and the least significant 8-bit is the intra-cluster address. The range of the cluster address is 0 to 255, and the range of the intra-cluster address is 0 to 255. The number 255 is used for the broadcast address. A routing device's intra-cluster address is 0.

The classifications of the short address are as follows:

a) Unicast addresses: each device's address is the unicast address, which includes the following types:
   – the gateway device's address is 0.0;
   – the routing device's address is x.0, where x is 1 to 254;

- if the network topology is star-only, that is NetworkTopology=1, the field device's address is 0.y, where y is 1 to 254; otherwise, the field device's address is x.y, where x is 1 to 254 and y is 1 to 254; see 6.9.1.2.1 for NetworkTopology.

b) Broadcast address: according to the different broadcast domains, there are four types of broadcast addresses:
   - the intra-cluster broadcast address is x.255, where x is 0 to 254;
   - the global broadcast address is 255.255;
   - the mesh network's broadcast address is 255.0;
   - the gateway device's broadcast address is 0.255.

9.3.3 Routing

9.3.3.1 General

The WIA-PA network supports static routing algorithms configured by the NM. After getting the neighbour information from each routing device, the NM generates the connection relationship of all routing devices. On the basis of the connection relationship, the NM generates and writes the routing information to each routing device in the form of a routing table. Each route in the table is assigned a route ID. Multiple paths correspond to each VCR, including the main path and the redundant path. The main path and the redundant path use the same route ID.

9.3.3.2 Routing table

Each routing device maintains a routing table, which is generated by the NM. The routing table is used for the path selection in the mesh network. The table has five items. RouteID is the identifier of a route. SourceAddress is the address of the start point of a route, DestinationAddress is the endpoint address of a route, NextHop is the address of the next hop device in a route, and RetryCounter records the number of one-hop retries in a route which reflects the status of the route. The RouteIDs of paths with same source address and destination address are same.

An example of a routing table is shown in Table 69.

Table 69 – Example of a routing table

RouteID	SourceAddress	DestinationAddress	NextHop	RetryCounter
5	F1	N1	N3	0
8	F2	Ng	N2	0
…	…	…	…	…
Ng in the table indicates a gateway address; N1, N2 and N3 indicate routing devices' addresses; F1 and F2 indicate field devices' addresses.				

9.3.4 Packet lifecycle management

Each packet has a lifecycle in the WIA-PA network. The lifecycle is expressed as a maximum surviving time. The NL records the generation time of packets by using timestamps. The surviving time is computed according to the generation time. When the surviving time of a packet exceeds its lifecycle, the packet should be discarded.

9.3.5 Joining and leaving network of device

The WIA-PA NL supports the joining and leaving processes of devices. The joining and leaving processes include the joining and leaving of field devices and of routing devices.

9.3.6 End-to-end network performance monitoring

The WIA-PA network layer monitors each path's health status. The NLME records the number of retries of each path to estimate the path failures. If there is a path failure, the NLME sends an indication to the DMAP (see 9.5.12).

9.3.7 Fragmentation and reassembly

Fragmentation and reassembly are handled at the NL. If the length of an NPDU is longer than the maximum DLSL payload, the NPDU should be fragmented at the NL of the sender. When the fragmented NPDUs reach the receiver, they are reassembled at the NL. See 9.6.2 for detailed packet format.

9.3.8 Network layer state machine

The NL state transitions are shown in Figure 40 and Table 70.

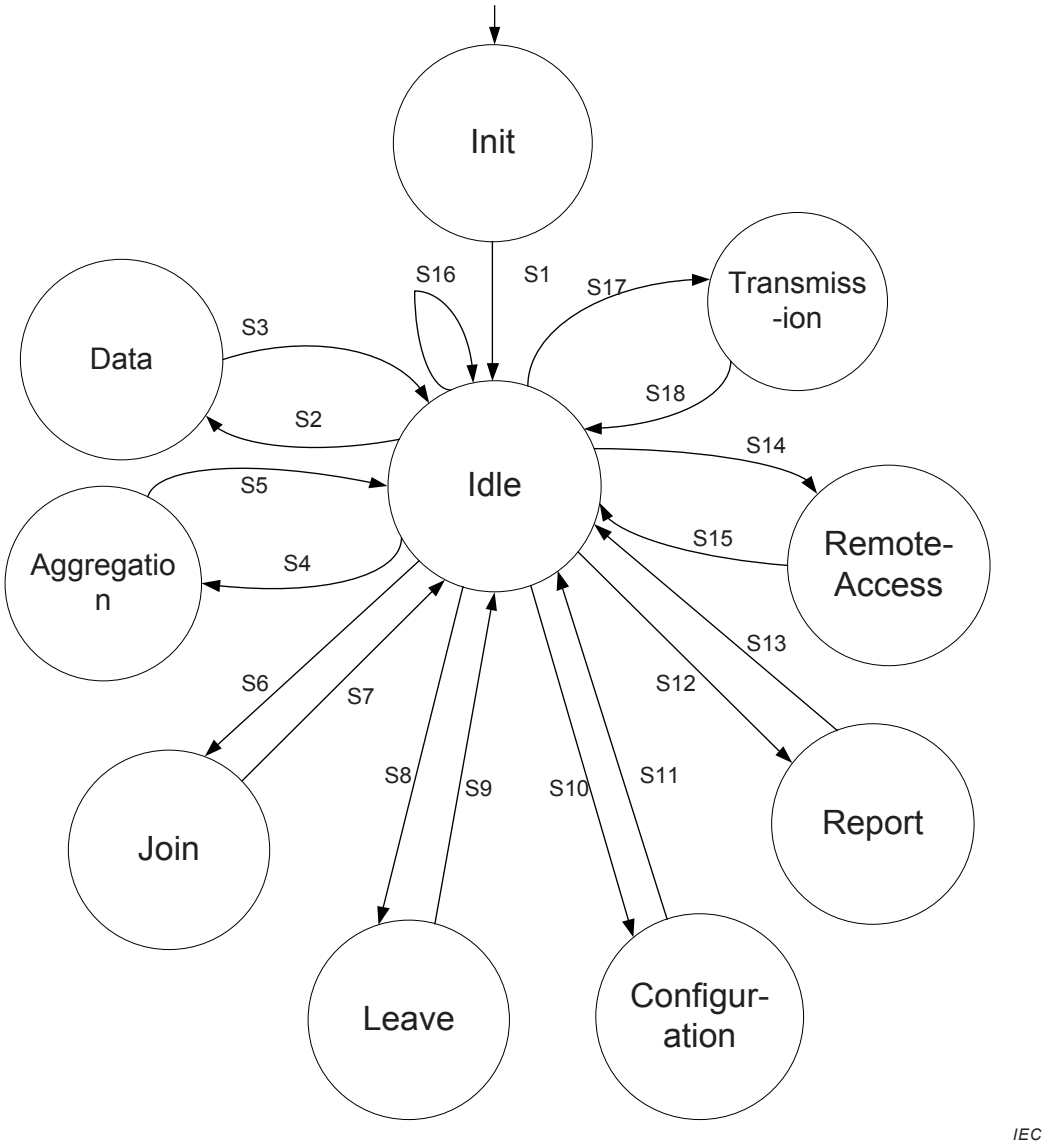

Figure 40 – Network layer state machine

Table 70 – NL state transitions *(1 of 4)*

Sequence number #	Current state	Event or condition	Actions	Next state
S1	Init		Initiation	Idle
S2	Idle	From DLSL: DLSL-DATA.confirm() or DLSL-DATA.indication()&Packet type = 0 in NSDU; From ASL: NLDE-DATA.request()		Data
S3	Data		From DLSL: invoke NLDE-DATA.confirm() to ASL or parsing NSDU and invoke NLDE-DATA.indication() for NSDU; From ASL: packing NPDU and invoke DLDE-DATA.request(); All messages or packets are sent to transmission queue.	Idle
S4	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 1 in NSDU; From DMAP: NLME-AGO-SEND.request()		Aggregation
S5	Aggregation		From DLSL: parsing NSDU according to NL aggregated packet format and invoke NLME-DAG.indication() to DMAP; From DMAP: packing NL aggregated packet; All messages or packets are sent to relate transmission queue.	Idle
S6	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 2 and Command ID = 0, 1 in NSDU; From DMAP: NLME-JOIN. request/response ()		Join
S7	Join		From DLSL: parsing NSDU according to NL joining request packet format if Command ID = 0 or joining response packet format if Command ID = 1 and invoke NLME-JOIN.indication() / NLME-JOIN.confirm()to DMAP; From DMAP: packing NL joining response packet; All messages or packets are sent to related transmission queue.	Idle
S8	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 2 and Command ID = 3, 4 in NSDU; From DMAP: NLME- LEAVE. request/response ()		Leave
S9	Leave		From DLSL: parsing NSDU according to NL leaving request packet format if Command ID = 3 or leaving response packet format if Command ID = 4 and invoke NLME-LEAVE.indication()/NLME-LEAVE.confirm()to DMAP; From DMAP: packing NL leaving response packet; All messages or packets are sent to related transmission queue.	Idle

Table 70 *(2 of 4)*

Sequence number #	Current state	Event or condition	Actions	Next state
S10	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 2 and Command ID = 8-25 in NSDU; From DMAP: NLME-ADD/UPDATE/DELETE_ROUTE. request/response () or NLME-ADD/UPDATE/RELEASE-LINK/SFR. request/response()		Configuration
S11	Configuration		From DLSL: parsing NSDU according to NL route/link/superframe-add/update/delete/release request/response packet format if Command ID = 8-25 and invoke NLME-ADD/UPDATE/DELETE_ROUTE.indication() or NLME-ADD/UPDATE/RELEASE-LINK/SFR.indication() to DMAP; From DMAP: packing NL route/link/superframe - add/update/delete/release response packet; All messages or packets are sent to related transmission queue.	Idle
S12	Idle	From DLSL: DLSL-COMM-STATUS.confirm() or DLSL-DATA.indication()& Packet type = 2 and Command ID =2, 5-7, 26-28; From DMAP: NLME-RPT-CLRMEM.response()		Report

Table 70 *(3 of 4)*

Sequence number #	Current state	Event or condition	Actions	Next state
S13	Report		From DLSL: invoke NLME-COMM-STATUS.cofirm () to DMAP. if Command ID = 2, parsing NSDU according to NL communication status report request packet format and invoke NLME-COMM-STATUS.indication(); if Command ID = 5, 6, parsing NSDU according to NL cluster member report request/response packet format and invoke NLME-RPT-CLRMEM.indication/confirm () to DMAP; if Command ID = 7; parsing NSDU according to NL neighbour information report request packet format and invoke NLME-NEIGHBOUR-INFO.indication() to DMAP; if Command ID = 26; parsing NSDU according to NL device condition report request packet format and invoke NLME-DEVICE-STATUS.indication() to DMAP; if Command ID = 27, parsing NSDU according to NL channel condition report request packet format and invoke NLME-CHANNEL-CONDITION.indication() to DMAP; if Command ID = 28, parsing NSDU according to NL path failure report request packet format and invoke NLME-PATH_FAILURE.indication() to DMAP; From DMAP: packing NL cluster member report response packet; All messages or packets are sent to related transmission queue.	Idle
S14	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 2 and Command ID =29-32; From DMAP: NLME-INFO_GET.response() or NLME-INFO_SET.response()		Remote Access
S15	Remote Access		From DLSL: if Command ID = 29, 30; parsing NSDU according to NL attribute getting request/response packet format and invoke NLME-INFO_GET.indication() / NLME-INFO_GET.confirm to DMAP; if Command ID = 31, 32, parsing NSDU according to NL attribute setting request/response packet format and invoke NLME-INFO_SET.indication() / NLME-INFO_SET.confirm to DMAP; From DMAP: packing NL attribute getting /setting response packet; All messages or packets are sent to relate transmission queue.	Idle

Table 70 *(4 of 4)*

Sequence number #	Current state	Event or condition	Actions	Next state
S16	Idle	From DLSL: DLSL-DATA.indication()& Packet type = 2 and Command ID> 32 or no packet received	Discard related NSDUs	Idle
S17	Idle	Transmission queue is not empty		Transmission
S18	Transmission	Transmission is over		Idle

The states in the NL state machine are specified as follows.

a) Idle state

The following transitions may occur while in the Idle state:

– when the DLSL invokes DLDE-DATA.indication, the NL enters the Receive state;

– when the ASL invokes NLDE-DATA.request, the NL entersthe Transmit state.

b) Transmit state

In the Transmit state, if the NL receives NLDE-DATA.confirm, it enters the Idle state.

c) Receive state

When in the Receive state,the NL enters the Idle state after the NL issues NLDE-DATA.indication to the upper layer.

See 9.4 for NLdata services, e.g. NLDE-DATA; see 9.5 for NL management services, e.g. NLME-; See 9.6.1 for Packet type and 9.6.4 for Command ID.

The states in the NL state machine are specified as follows.

a) Idle state

After a device initiates the NL, the NL enters the Idle state. The following transitions may occur while the DLSL is in the Idle state:

b) When a DLSL, ASL, or DMAP frame arrives, the NL processes the NSDU.

The packets that a device can receive include packets whose final destination addresses are this device. The functions of the Idle state are to receive and process the packet and process the DMAP, ASL and DLSL primitives.

The following transitions may occur while the NL is in the Receive state:

– If a data packet is received from DLSL or NLDE-DATA.request() is received from ASL, the NL enters Data state.

– If an aggregated packer is received from DLSL or NLME-AGO-SEND.request() is received from DMAP, the NL enters Aggregation state.

– If a join request or response command packet is received from DLSL or NLME-JOIN. request/response () is received from DMAP, the NL enters Join state.

– If a leave request or response command packet is received from DLSL or NLME-LEAVE. request/response () is received from DMAP, the NL enters Leave state.

– If a command packet identified from 8 to 25 is received from DLSL or NLME-ADD/UPDATE/DELETE_ROUTE.request/response () or NLME-ADD/UPDATE/RELEASE-LINK/SFR. request/response() are received from DMAP, the NL enters Configuration state.

– If a command packet identified by 2, 5-7, 26-28 is received from DLSL or NLME-RPT-CLRMEM.response() is received from DMAP, the NL enters Report state.

– If an attribute getting or setting command packet identified from 29 to 32 is received from DLSL or NLME-INFO_GET.response() /NLME-INFO_SET.response() is received from DMAP, the NL enters Remote Access state.

– If no packet or primitives is received or command packets identifier bigger than 32 is received from DLSL, the NL enters Idle state.

c) When the transmission queue is not empty, the NL enters the Transmission state.

d) Data state

This state handles the data process procedure of a device. If a data from DLSL is received or an ASL data request is received, the NL enters the Data state.

e) Aggregation state

This state handles the data/packet aggregation and dis-aggregation procedure.

f) Join state

This state handles the joining procedure of a device.

g) Leave state

This state handles the leaving procedure of a device.

h) Configuration state

This state handles the resource allocation procedures, which include the add/update/deleting or release of a route, link, or a superframe.

i) Report state

This state handles the report procedures, which include the cluster member report, neighbour information report, device condition report, channel condition report, and path failure report.

j) Remote access state

This state handles the remote attribute access procedures, which include the attribute getting and setting.

k) Transmission state

This state handles the data or packet transmission procedures. If the transmission queue is not empty, the NL enters the Transmission state.

9.4 Network layer data services

9.4.1 General

The NLDE-SAP is used by the ASL to receive and transmit data. The primitives supported by NL data services include NLDE-DATA.request, NLDE-DATA.confirm, and NLDE-DATA.indication.

9.4.2 NLDE-DATA.request

The NLDE receives the payload from the ASL through NLDE-DATA.request and adds it to the message queue of the NL.

The semantics of NLDE-DATA.request are described as follows:

```
NLDE-DATA.request (
                    VcrID,
                    SrcAddr,
                    Priority,
                    PayloadLength,
                    Payload,
                    PayloadHandle
                )
```

Table 71 specifies the parameters for NLDE-DATA.request.

Table 71 – NLDE-DATA.request parameters

Name	Data type	Valid range	Description
VcrID	Unsigned16	0 to 65 535	The VCR identifier
SrcAddr	Unsigned16	0 to 65 535 (Unicast address)	The 16-bit short address of the NSDU's source
Priority	Unsigned8	0 to 15	Priority of this NSDU
PayloadLength	Unsigned16	0 to 65 535	The length of NSDU to be transmitted
Payload	Octetstring		NSDU
PayloadHandle	Unsigned8	0 to 255	The handle associated with the NSDU to be transmitted

9.4.3 NLDE-DATA.confirm

NLDE-DATA.confirm reports the results of NLDE-DATA.request.

The semantics of NLDE-DATA.confirm are described as follows:

```
NLDE-DATA.confirm (
                    PayloadHandle,
                    Status
                  )
```

Table 72 specifies the parameters for NLDE-DATA.confirm.

Table 72 – NLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0 to 255	Handle of the NSDU, which is used to indicate the NL payload.
Status	Unsigned8	0 to 255	Result of NLDE-DATA.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.4.4 NLDE-DATA.indication

NLDE-DATA.indication informs the ASL when the NL receives a packet.

The semantics of NLDE-DATA.indication are described as follows:

```
NLDE-DATA.indication (
                       SrcAddr,
                       Priority,
                       NSDULength,
                       NSDU
                     )
```

Table 73 specifies the parameters of NLDE-DATA.indication.

Table 73 – NLDE-DATA.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535 (Unicast Address)	The 16-bit short address of the NSDU's source
Priority	Unsigned8	0 to 15	Priority of this NSDU
NSDULength	Unsigned16	0 to 65 535	Length of the NSDU
NSDU	octet		Data of the NSDU

9.4.5 Time sequence of NL data services

Figure 41 shows the basic procedures of packet sending and receiving. NLDE-DATA.request is generated by a local Application Sub-Layer Data Entity (ASLDE) when a data Application Sub-Layer Protocol Data Unit (ASLPDU) is to be transferred to a peer NLDE. On receipt of a NLDE-DATA.request, the NLDE begins transmitting of the NPDU.

NLDE-DATA.confirm is generated by the NLDE of a source device in response to NLDE-DATA.request. NLDE-DATA.confirm returns a status indicating the result of the transmission.

NLDE-DATA.indication is generated by the NLDE of a destination device and is issued to the ASLDE on receipt of a data packet at the local NLDE.

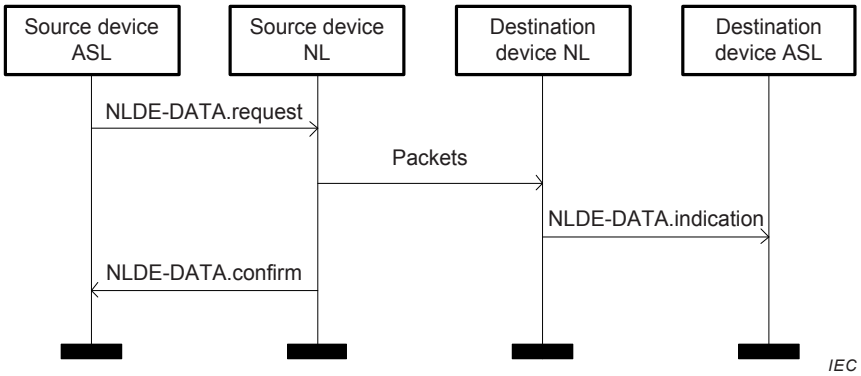

Figure 41 – Time sequence of NL data services

9.5 Network layer management services

9.5.1 General

The DMAP uses the interface supplied by NLME-SAP to configure and control the NL operation.

9.5.2 Network communication status report services

9.5.2.1 NLME-COMM-STATUS.request

The semantics of NLME-COMM-STATUS.request are described as follows:

```
NLME-COMM-STATUS.request (
                          ProxyAddr,
                          PhyAddr,
                          DeviceType,
                          Status
                         )
```

Table 74 specifies the parameters for NLME-COMM-STATUS.request.

Table 74 – NLME-COMM-STATUS.request parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
Status	Unsigned8	0 to 255	Results of the joining process of a new device: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.2.2 NLME-COMM-STATUS.indication

The semantics of NLME-COMM-STATUS.indication are described as follows:

NLME-COMM-STATUS.indication (

ProxyAddr,
PhyAddr,
DeviceType,
Status
)

Table 75 specifies the parameters for NLME-COMM-STATUS.indication.

Table 75 – NLME-COMM-STATUS.indication parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.
Status	Unsigned8	0 to 255	Results of the joining process of a new device: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NL communication status request packet, the NL of the gateway device will invoke NLME-COMM-STATUS.indication and inform the DMAP.

9.5.2.3 NLME-COMM-STATUS.confirm

The semantics of NLME-COMM-STATUS.confirm are described as follows:

```
NLME-COMM-STATUS.confirm (
                          PhyAddr,
                          Status
                          )
```

Table 76 specifies the parameters for NLME-COMM-STATUS.confirm.

Table 76 – NLME-COMM-STATUS.confirm parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-COMM-STATUS.confirm is generated by the NL in response to NLME-COMM-STATUS.request. NLME-COMM-STATUS.confirm returns a status of either SUCCESS, indicating that the requested transmission has been successful, or FAILURE, indicating that the transmission has failed.

9.5.3 Network joining services

9.5.3.1 NLME-JOIN.request

The semantics of NLME-JOIN.request are described as follows:

```
NLME-JOIN.request (
                    ProxyAddr,
                    PhyAddr,
                    SecMaterial,
                    DeviceType
                  )
```

Table 77 specifies the parameters for NLME-JOIN.request.

Table 77 – NLME-JOIN.request parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (Unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
SecMaterial	Unsigned32	0 to (2^{32}-1)	The identity information of a device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

9.5.3.2 NLME-JOIN.indication

The semantics of NLME-JOIN.indication are described as follows:

```
NLME-JOIN.indication (
                       ProxyAddr,
                       PhyAddr,
                       SecMaterial,
                       DeviceType
                     )
```

Table 78 specifies the parameters for NLME-JOIN.indication.

Table 78 – NLME-JOIN.indication parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	Short address of the routing device that generates NL join request
PhyAddr	Unsigned64	0 to (2^{64}-1)	Device's physical address
SecMaterial	Unsigned32	0 to (2^{32}-1)	The identity information of a device
DeviceType	Unsigned8	0 to 255	The type of the new device: 0 = Gateway device; 1 = Routing device; 2 = Field device; 3 = Handheld device; Others are reserved.

On receipt of NL joining request packet, the NL of the gateway device will invoke NLME-JOIN.indication and inform the DMAP.

9.5.3.3 NLME-JOIN.response

The semantics of NLME-JOIN.response are described as follows:

```
NLME-JOIN.response (
                    ProxyAddr,
                    PhyAddr,
                    ShortAddr,
                    Status
                   )
```

Table 79 specifies the parameters for NLME-JOIN.response.

Table 79 – NLME-JOIN.response parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	Short address of the routing device that generates NL join request
PhyAddr	Unsigned64	0 to (2^{64}-1)	Physical address of new device
ShortAddr	Unsigned16	0 to 65 535	Allocated network address (Unicast address) of new device
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved.

The DMAP invokes NLME-JOIN.response to assign a network address to the new device.

9.5.3.4 NLME-JOIN.confirm

The semantics of NLME-JOIN.confirm are described as follows:

```
NLME-JOIN.confirm (
                    ShortAddr,
                    Status
                  )
```

Table 80 specifies the parameters for NLME-JOIN.confirm.

Table 80 – NLME-JOIN.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 to 65 535	Network address (unicast address)
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-JOIN.confirm is generated by the NL in response to NLME-JOIN.request. NLME-JOIN.confirm returns a status of either SUCCESS, indicating that the requested transmission has been successful, or FAILURE, indicating that the transmission has failed.

9.5.3.5 Time sequence for device joining

9.5.3.5.1 Time sequence for field device joining

To join the star network, a field device should initiate a joining request in the MAC layer by DMAP to the routing device or the gateway device. A field device should join the network either through the gateway device or through a routing device.

When the gateway device receives a joining request from a field device, it should indicate the joining to the NM and then return a joining response. If the new field device receives the MAC associate response from the gateway device, it joins the network.

When the routing device receives a joining request from a field device, it should produce a joining request of the NL and send this request to the gateway device. If the new field device receives the joining response, the joining process is finished.

Figure 42 illustrates an example of the joining process of a field device in the hierarchical network that is the combination of star and mesh.

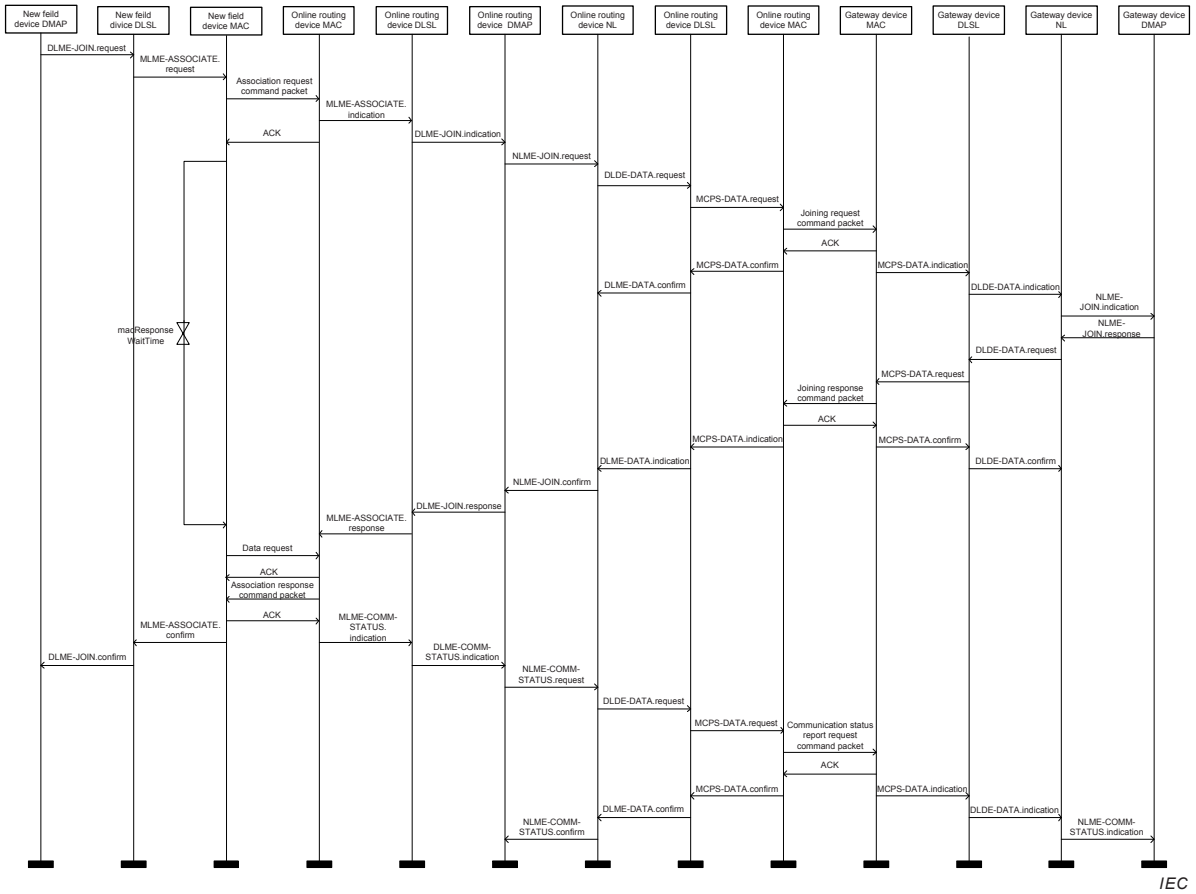

Figure 42 – Time sequence for field device joining through routing device

9.5.3.5.2 Time sequence for routing device joining

If a routing device wants to join the mesh network, it should send a joining request to the gateway device, and the request includes the long address of this routing device. There are two kinds of joining processes for routing devices: one-hop joining and multi-hop joining. If the new routing device sends a joining request to the gateway device, the one-hop joining process is used; if the new routing device joins the mesh network through another existing routing device, the multi-hop joining process is used.

The gateway device receives the joining request and informs the NM. The NM should return a response. If the new routing device receives the response, its joining process is completed.

Figure 43 and Figure 44 illustrate examples of the joining process of routing devices.

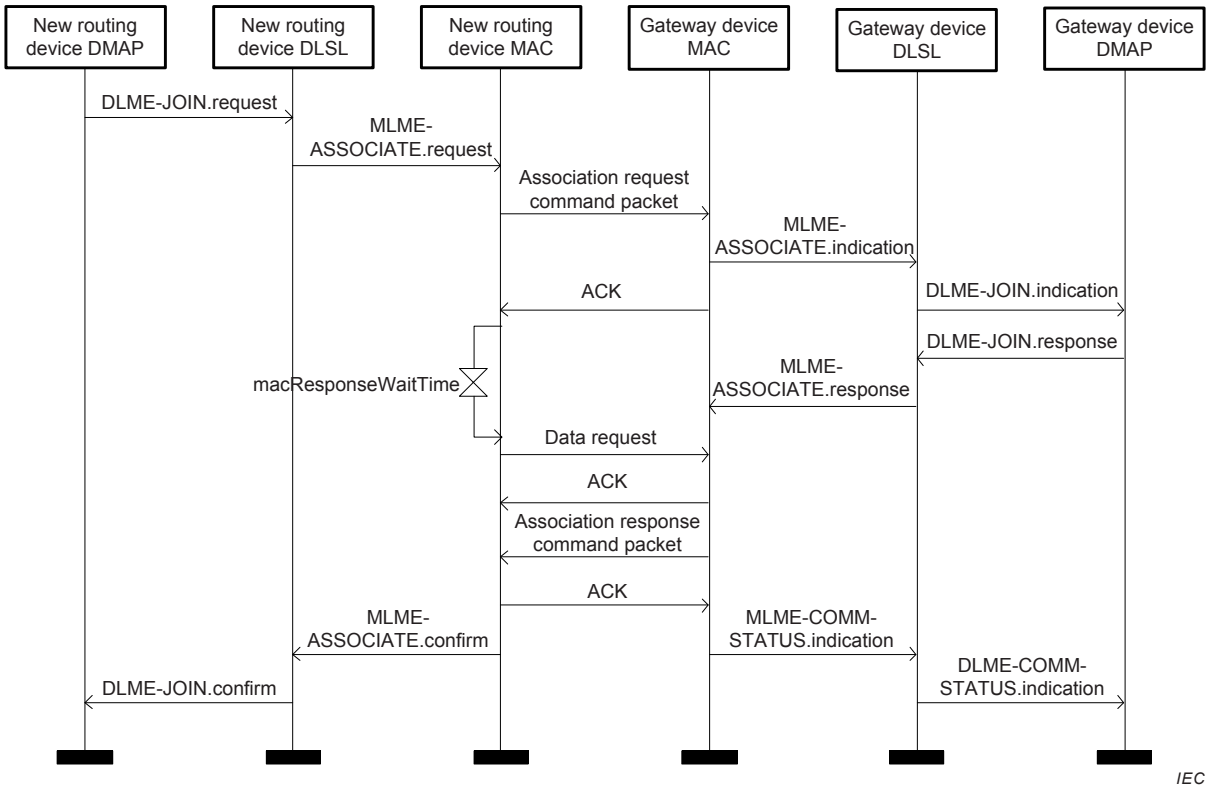

Figure 43 – One-hop joining process for routing device

See IEEE STD 802.15.4-2011, Figure 17 and Figure 18 for the detailed message sequence chart between peer MAC entities.

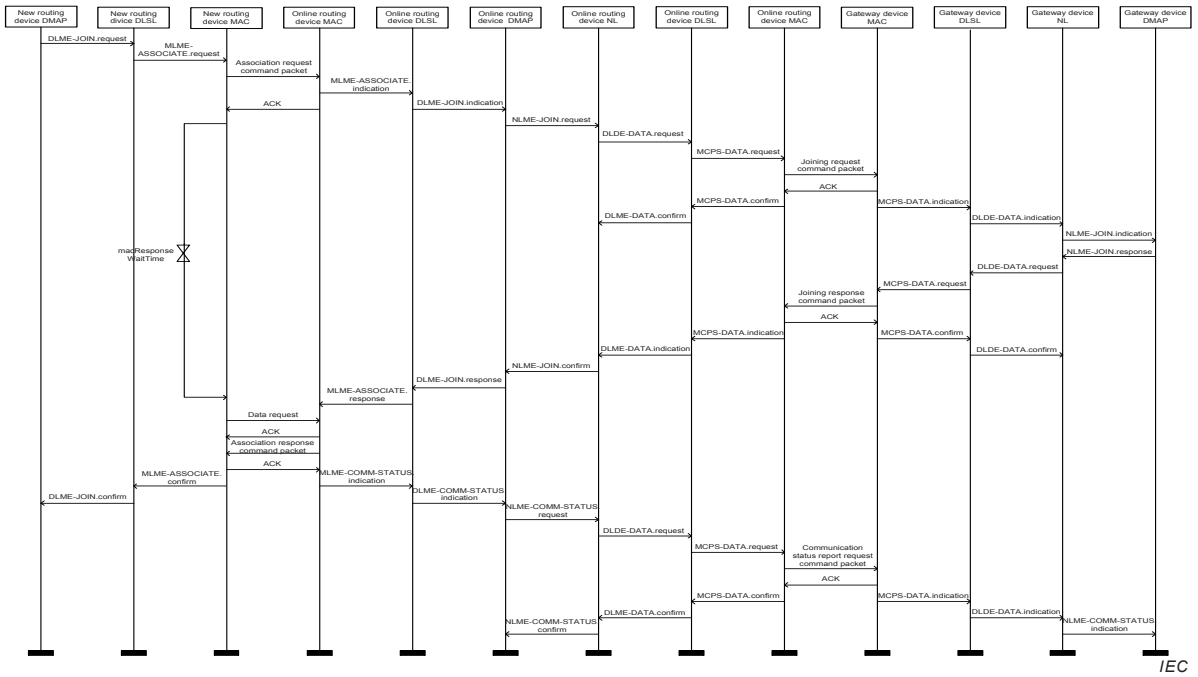

Figure 44 – Multi-hop join process of routing device

Before joining the network, the field devices and the routing devices set the DeviceState (see Table 20) to 0 to indicate that the devices have not joined the network. If the new devices have sent out the joining request and have not received the joining response, they set the

DeviceState to 1 to indicate that the devices are joining the network. If the new devices have sent out the joining request and have received the joining response successfully, they set the DeviceState to 3 to indicate that the devices have joined the network. If the joining process needs security authentication, the devices set the DeviceState to 3 during the authentication process. See Table 20 for DeviceState.

9.5.4 Networkleaving services

9.5.4.1 NLME-LEAVE.request

The semantics of NLME-LEAVE.request are described as follows:

```
NLME-LEAVE.request (
                    SrcAddr,
                    DstAddr,
                    LeavingReason
                )
```

Table 81 specifies the parameters for NLME-LEAVE.request.

Table 81 – NLME-LEAVE.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The address of the device to which to send the leaving response
LeavingReason	Unsigned8	0 to 255	1 = Passive leaving of a routing device or a field device 2 = Active leaving of a routing device Others are reserved.

When a routing device wants to leave the network, or the gateway device wants a device to leave the network, the DMAP invokes NLME-LEAVE.request and requests its NL to start the leaving process.

9.5.4.2 NLME-LEAVE.indication

The semantics of NLME-LEAVE.indication are described as follows:

```
NLME-LEAVE.indication (
                    DeviceAddr,
                    LeavingReason
                    )
```

Table 82 specifies the parameters for NLME-LEAVE.indication.

Table 82 – NLME-LEAVE.indication parameters

Name	Data type	Valid range	Description
DeviceAddr	Unsigned16	0 to 65 535	The address of the device invoking Leaving request
LeavingReason	Unsigned8	0 to 255	1 = Passive leaving of a routing device or a field device 2 = Active leaving of a routing device Others are reserved.

On receipt of a leaving request command packet, the NL will invoke NLME-LEAVE.indication and inform the DMAP.

9.5.4.3 NLME-LEAVE.response

The semantics of NLME- LEAVE.response are described as follows:

```
NLME-LEAVE.response (
                        DevAddr,
                        Status
                    )
```

Table 83 specifies the parameters for NLME-LEAVE.response.

Table 83 – NLME-LEAVE.response parameters

Name	Data type	Valid range	Description
DevAddr	Unsigned16	0 to 65 535	The address of the device that has requested leaving
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-LEAVE.responseis used by DMAP or NM to inform the leaving device whether or not the leaving request has been accepted.

9.5.4.4 NLME-LEAVE.confirm

The semantics of NLME-LEAVE.confirm are described as follows:

```
NLME-LEAVE.confirm (
                        DevAddr,
                        Status
                    )
```

Table 84 specifies the parameters for NLME-LEAVE.confirm.

Table 84 – NLME-LEAVE.confirm parameters

Name	Data type	Valid range	Description
DevAddr	Unsigned16	0 to 65 535	The address of the device that has either requested leaving or been instructed to leave by Gateway device
Status	Unsigned8	0 to 255	Execution result of the request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

NLME-LEAVE.confirm is generated by the NL in response to NLME-LEAVE.request.NLME-LEAVE.confirm returns a status of either SUCCESS, indicating that the request to transmit is successful, or FAILURE, indicating that the request to transmit is failed.

9.5.4.5 Time sequence for device leaving

9.5.4.5.1 Time sequence for field device leaving

The leaving processes of field devices include the following two types.

a) Active leaving. For a field device, the DMAP invokes DLME-LEAVE.request to send a leaving request to the routing device or the gateway device that is its cluster head. After

receiving the leaving request command packet from the field device, the NL of the routing device or the gateway device informs the DMAP with DLME-LEAVE.indication. If the field device leaves the routing device, the routing device sends NLME-RPT-CLRMEM.request to the gateway device. Figure 45 illustrates the time sequence of a field device's active leaving.

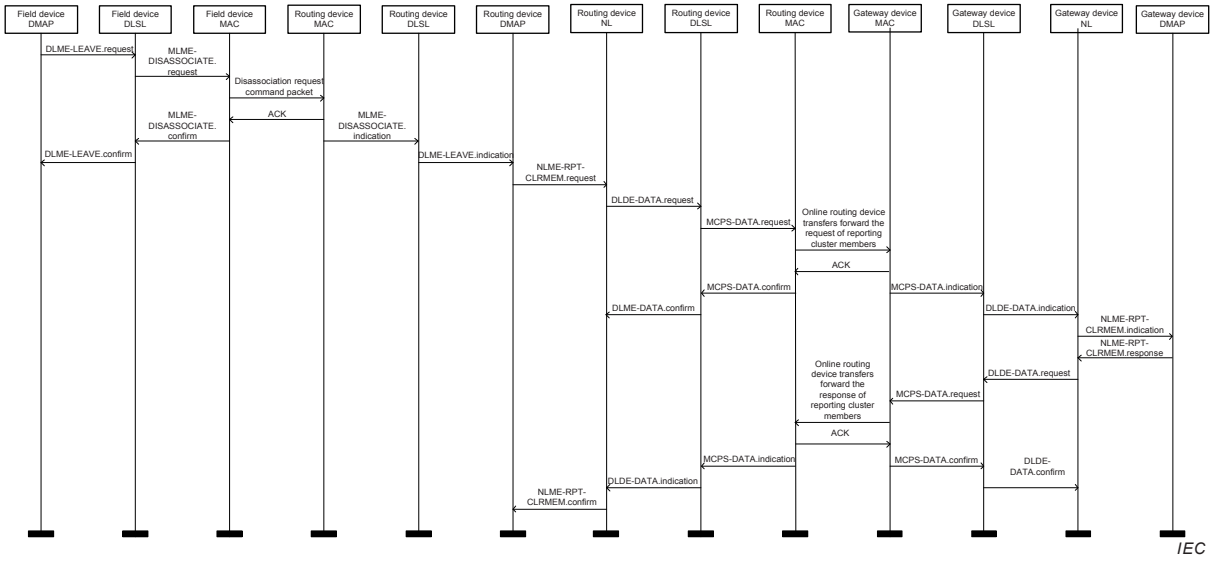

Figure 45 – Active leaving process of field device (leaving routing device)

See IEEE STD 802.15.4-2011, Figure 27 for the detailed message sequence chart between peer MAC entities.

b) Passive leaving. The DMAP of a routing device requests for its field device to leave the network by DLME-LEAVE.request. After receiving this request, the field device returns Acknowledge (ACK). If the routing device receives DLME-LEAVE.confirm, it sends a cluster member report request to the gateway device. Figure 46 illustrates the time sequence of a field device passive leaving.

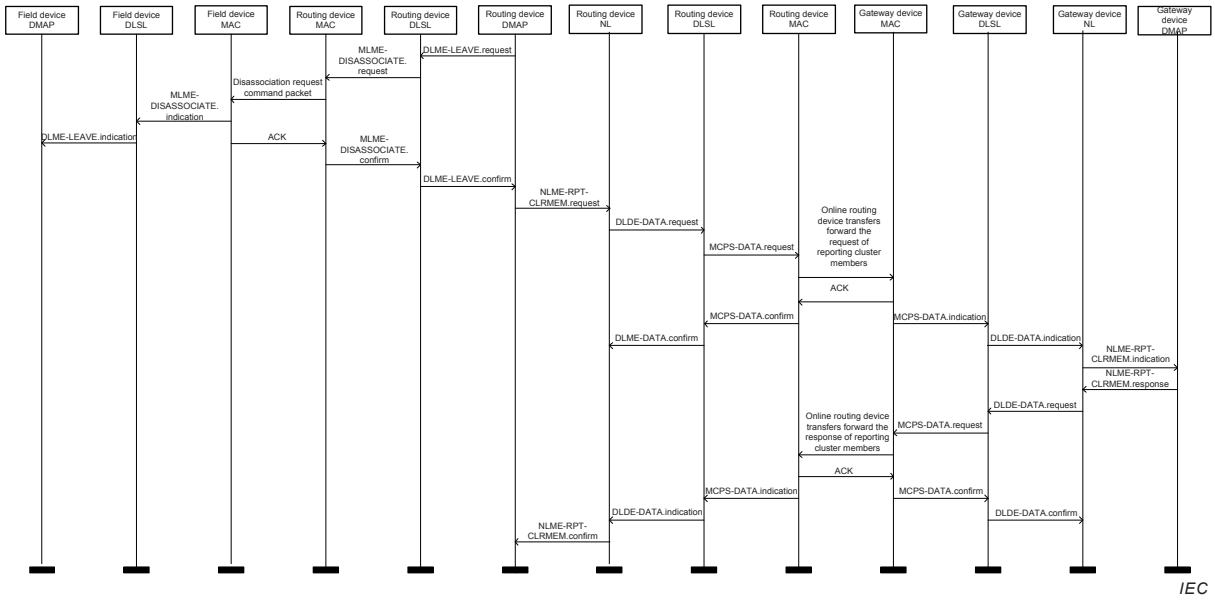

Figure 46 – Passive leaving of field device

Routing devices connect to both the mesh network and the star network. Therefore, the routing device should inform both the gateway device and its field devices of its leaving.

See IEEE STD 802.15.4-2011, Figure 27 for the detailed message sequence chart between peer MAC entities.

9.5.4.5.2 Time sequence for routing device leaving

The leaving processes of routing devices include the following two types.

a) Active leaving. The DMAP of a routing device sends its leaving request to the gateway device by NLME-LEAVE.request. After receiving the leaving request, the gateway device returns a leaving response. When the DMAP of the leaving routing device receives NLME-LEAVE.confirm, it informs its cluster members. Figure 47 illustrates the time sequence of a routing device's active leaving.

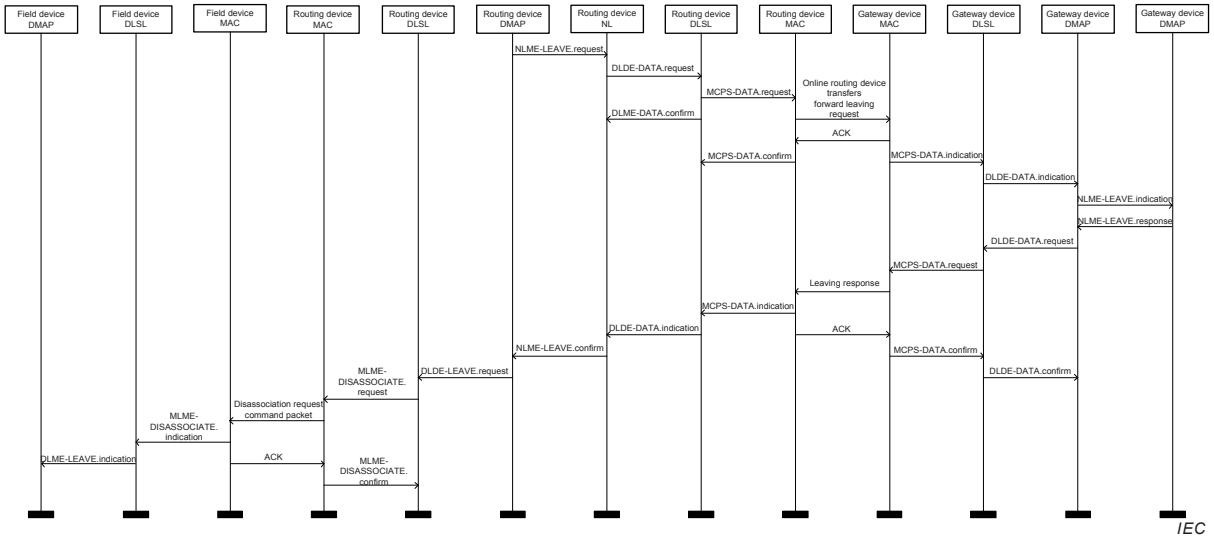

Figure 47 – Active leaving process of routing device

See IEEE STD 802.15.4-2011, Figure 27 for the detailed message sequence chart between peer MAC entities.

b) Passive leaving. If the gateway device wants a routing device to leave the network, it produces an NL leaving request packet, and sends it to the designated routing device. After receiving the leaving request packet from the gateway device, the routing device informs all its cluster members of its leaving. Figure 48 illustrates the time sequence of a routing device's passive leaving.

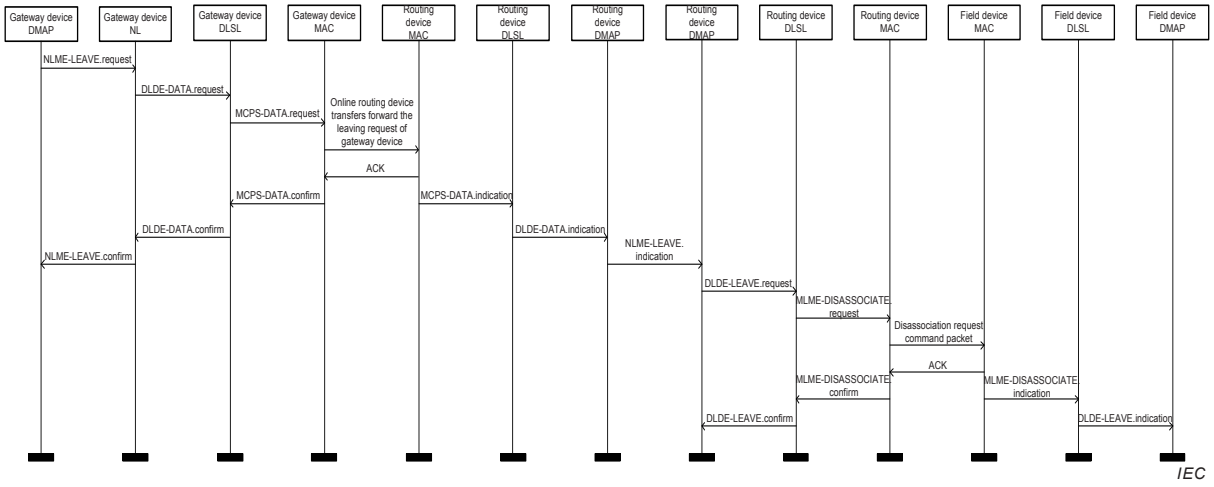

Figure 48 – Passive leaving process of routing device

See IEEE STD 802.15.4-2011, Figure 20 for the detailed message sequence chart between peer MAC entities.

9.5.5 Cluster member report services

9.5.5.1 NLME-RPT-CLRMEM.request

NLME-RPT-CLRMEM.request is used by routing devices to report the information of the cluster members to the gateway device.

The semantics of NLME-RPT-CLRMEM.request are described as follows:

```
NLME-RPT-CLRMEM.request (
                          SrcAddr,
                          DstAddr,
                          ClrMemFlag,
                          ClrMemAddr
                        )
```

Table 85 specifies the parameters for NLME-RPT-CLRMEM.request.

Table 85 – NLME-RPT-CLRMEM.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	The source address
DstAddr	Unsigned16	0 to 65 535	The destination address
ClrMemFlag	Unsigned8	0 to 255	The flag of the cluster member modification: 0 = Add; 1 = Delete; Others are reserved.
ClrMemAddr	Unsigned16	0 to 65 535	The network address of the modified cluster member

9.5.5.2 NLME-RPT-CLRMEM.confirm

NLME-RPT-CLRMEM.confirm is used by the NL to return the result of NLME-RPT-CLRMEM.request to the DMAP.

The semantics of NLME-RPT-CLRMEM.confirm are described as follows:

```
NLME-RPT-CLRMEM.confirm (
                          Status
                        )
```

Table 86 specifies the parameters for NLME-RPT-CLRMEM.confirm.

Table 86 – NLME-RPT-CLRMEM.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of the cluster member report: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.5.3 NLME-RPT-CLRMEM.indication

NLME-RPT-CLRMEM.indication is used by the NL to report the successful receipt of cluster member report request packets.

The semantics of NLME-RPT-CLRMEM.indication are described as follows:

```
NLME-RPT-CLRMEM.indication (
                              SrcAddr,
                              ClrMemFlag,
                              ClrMemAddr
                            )
```

Table 85 specifies the parameters for NLME-RPT-CLRMEM.indication.

9.5.5.4 NLME-RPT-CLRMEM.response

NLME-RPT-CLRMEM.response is used to respond to NLME-RPT-CLRMEM.indication.

The semantics of NLME-RPT-CLRMEM.response are described as follows:

```
NLME-RPT-CLRMEM.response (
                              DstAddr,
                              Status
                            )
```

Table 87 specifies the parameters for NLME-RPT-CLRMEM.response.

Table 87 – NLME-RPT-CLRMEM.response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The destination address
Status	Unsigned8	0 to 255	The result returned from cluster member report: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.5.5 Time sequence for cluster member reporting

The time sequence diagram for reporting cluster member is shown in Figure 49.

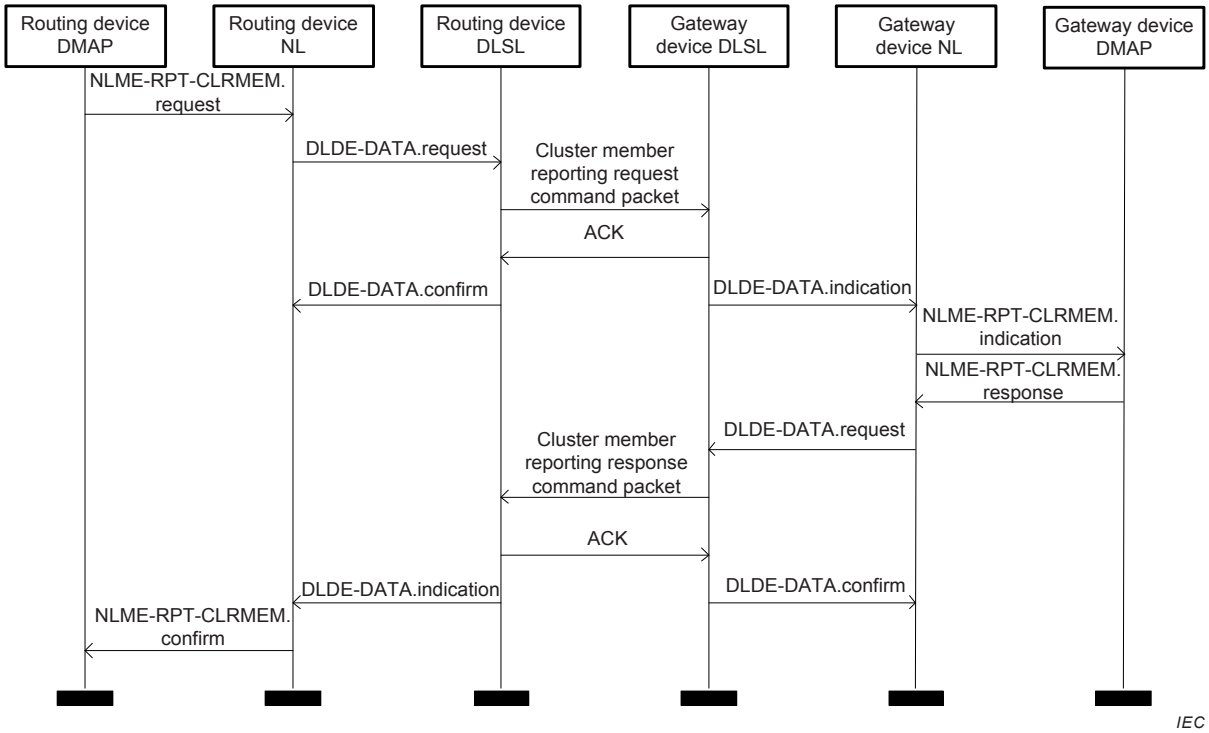

Figure 49 – Cluster member reporting process

9.5.6 Neighbour information report services

9.5.6.1 NLME-NEIGHBOUR-INFO.request

NLME-NEIGHBOUR-INFO.request is used for a routing device to report its one-hop neighbours' information to the gateway device. This usage of NLME-NEIGHBOUR-INFO.request is triggered only when a neighbour joins or leaves the WIA-PA network. The neighbour itself and its next hop neighbour to GW reports neighbour information to GW.

The semantics of NLME-NEIGHBOUR-INFO.request are described as follows:

```
NLME-NEIGHBOUR-INFO.request (
                                    SrcAddr,
                                       DstAddr,
                                        NeighbourCount,
                                        NeighbourStructure
                                   )
```

Table 88 specifies the parameters for NLME-NEIGHBOUR-INFO.request.

Table 88 – NLME-NEIGHBOUR-INFO.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
NeighbourCount	Unsigned8	0 to 255	Number of the neighbours
NeighbourStructure	Neighbour_Struct structure (see Table 18)		Information of the neighbours

9.5.6.2 NLME-NEIGHBOUR-INFO.indication

NLME-NEIGHBOUR-INFO.indication is used for the DLSL to report the received NLME-NEIGHBOUR-INFO.request packet to the DMAP.

The semantics of NLME-NEIGHBOUR-INFO.indication are described as follows:

NLME-NEIGHBOUR-INFO.indication (

SrcAddr,
NeighbourCount,
NeighbourStructure
)

Table 88 specifies the parameters for NLME-NEIGHBOUR-INFO.indication.

9.5.6.3 NLME-NEIGHBOUR-INFO.confirm

NLME-NEIGHBOUR-INFO.confirm is used to report the successful sending of neighbour information report request packets.

The semantics of NLME-NEIGHBOUR-INFO.confirm are described as follows:

NLME-NEIGHBOUR-INFO.confirm (

Status
)

Table 89 specifies the parameters for NLME-NEIGHBOUR-INFO.confirm.

Table 89 – NLME-NEIGHBOUR-INFO.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Reporting the result of the neighbour information: 0 = SUCCESS; 1 = .FAILURE; Others are reserved.

9.5.6.4 Time sequence for neighbour information reporting

The time sequence diagram for reporting neighbour information is shown in Figure 50.

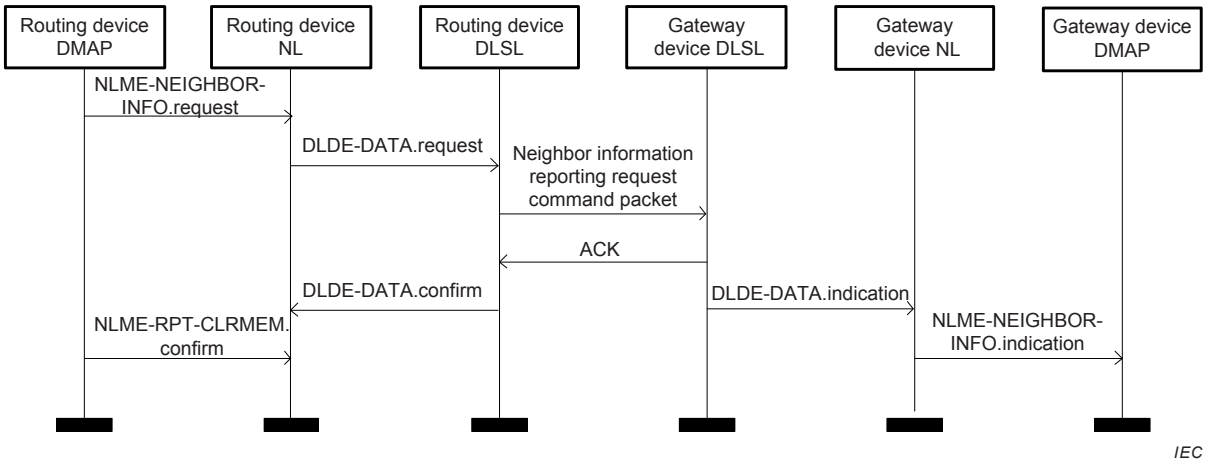

Figure 50 – Neighbour information reporting process

9.5.7 Route allocation services

9.5.7.1 Route adding services

9.5.7.1.1 NLME-ADD_ROUTE.request

NLME-ADD_ROUTE.request is used to add a record to the routing table of a routing device.

The semantics of NLME-ADD_ROUTE.request are described as follows:

```
NLME-ADD_ROUTE.request (
                          DstAddr,
                          RoutingTableRecord
                        )
```

Table 90 specifies the parameters for NLME-ADD_ROUTE.request.

Table 90 – NLME-ADD_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The 16-bit short address of the routing device
RoutingTableRecord	NLRoute_Struct structure (see Table 15)		A routing table item

9.5.7.1.2 NLME-ADD_ROUTE.confirm

NLME-ADD_ROUTE.confirm reports the result of a NLME-ADD_ROUTE.request.

The semantics of NLME_ADD-ROUTE.confirm are described as follows:

```
NLME-ADD_ROUTE.confirm(
                          Status
                        )
```

Table 91 specifies the parameters for NLME-ADD_ROUTE.confirm.

Table 91 – NLME-ADD_ROUTE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of an NLME-ADD_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-ADD_ROUTE.request, the NLshouldtransmit an adding-route-request packet to the routing device and should return an NLME-ADD_ROUTE.confirm to report the result.

9.5.7.1.3 NLME-ADD_ROUTE.indication

NLME-ADD_ROUTE.indication is used to report to the DMAP that a device has successfully received a route adding request packet.

The semantics of NLME_ADD-ROUTE.confirm are described as follows:

```
NLME-ADD_ROUTE.indication (
                              RoutingTableRecord
                          )
```

Table 90 specifies the parameters for NLME-ADD_ROUTE.indication.

9.5.7.1.4 NLME-ADD_ROUTE.response

NLME-ADD_ROUTE.response is the response of NLME-ADD_ROUTE.indication.

The semantics of NLME-ADD_ROUTE.response are described as follows:

```
NLME-ADD_ROUTE.response (
                              Status
                          )
```

Table 91 specifies the parameters for NLME-ADD_ROUTE.response.

9.5.7.1.5 Time sequence for route adding

The time sequence diagram for adding a record to the routing table of a routing device is shown in Figure 51.

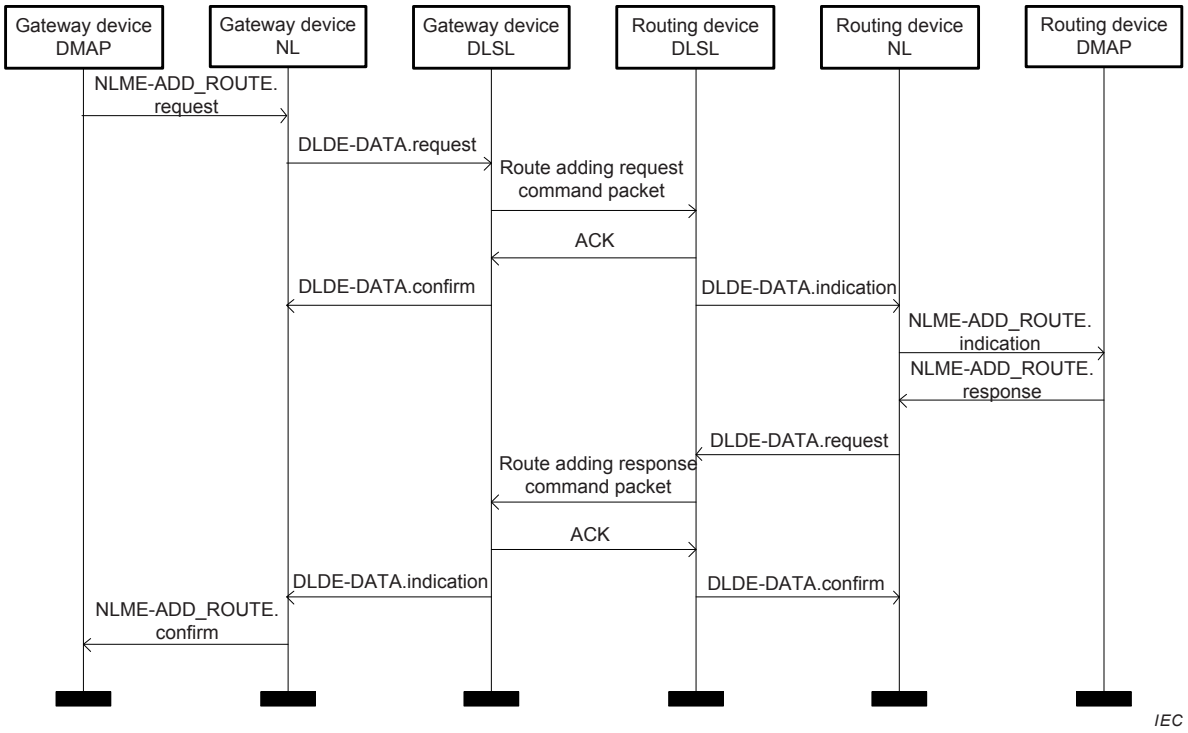

Figure 51 – Time sequence for route adding

9.5.7.2 Route update services

9.5.7.2.1 NLME-UPDATE_ROUTE.request

NLME-UPDATE_ROUTE.request updates a record in the routing table of a routing device.

The semantics of NLME-UPDATE_ROUTE.request are described as follows:

```
NLME-UPDATE_ROUTE.request (
                              DstAddr,
                              RoutingTableRecord
                            )
```

Table 92 specifies the parameters for NLME-UPDATE_ROUTE.request.

Table 92 – NLME-UPDATE_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of routing device
RoutingTableRecord	NLRoute_Struct structure (see Table 15)		A routing table item

The NM invokes NLME-UPDATE_ROUTE.request to update a record in the routing table of routing devices.

9.5.7.2.2 NLME-UPDATE_ROUTE.confirm

The NLME_UPDATE-ROUTE.confirm reports the execution result of an NLME-UPDATE_ROUTE.request.

The semantics ofNLME_UPDATE-ROUTE.confirm are described as follows:

NLME-UPDATE_ROUTE.confirm (
Status
)

Table 93 specifies the parameters for NLME-UPDATE-ROUTE.confirm.

Table 93 – NLME-UPDATE_ROUTE.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of NLME-UPDATE_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-UPDATE_ROUTE.request, the NL will transmit a route update request packet to the routing device and return NLME-UPDATE_ROUTE.confirm to report the result.

9.5.7.2.3 NLME-UPDATE_ROUTE.indication

NLME-UPDATE_ROUTE.indication is used to report to the DMAP that a device has successfully received a route update request packet.

The semantics of NLME-UPDATE_ROUTE.confirm are described as follows:

NLME- UPDATE_ROUTE.indication (
RoutingTableRecord
)

Table 92 specifies the parameters for NLME- UPDATE _ROUTE.indication.

9.5.7.2.4 NLME-UPDATE _ROUTE.response

NLME-UPDATE_ROUTE.response is the response of NLME- UPDATE_ROUTE.indication.

The semantics of NLME-UPDATE_ROUTE.response are described as follows:

NLME-UPDATE_ROUTE.response (
Status
)

Table 93 specifies the parameters for NLME-UPDATE_ROUTE.response.

9.5.7.2.5 Time sequence for route updating

The time sequence diagram for updating a record in the routing table of a routing device is shown in Figure 52.

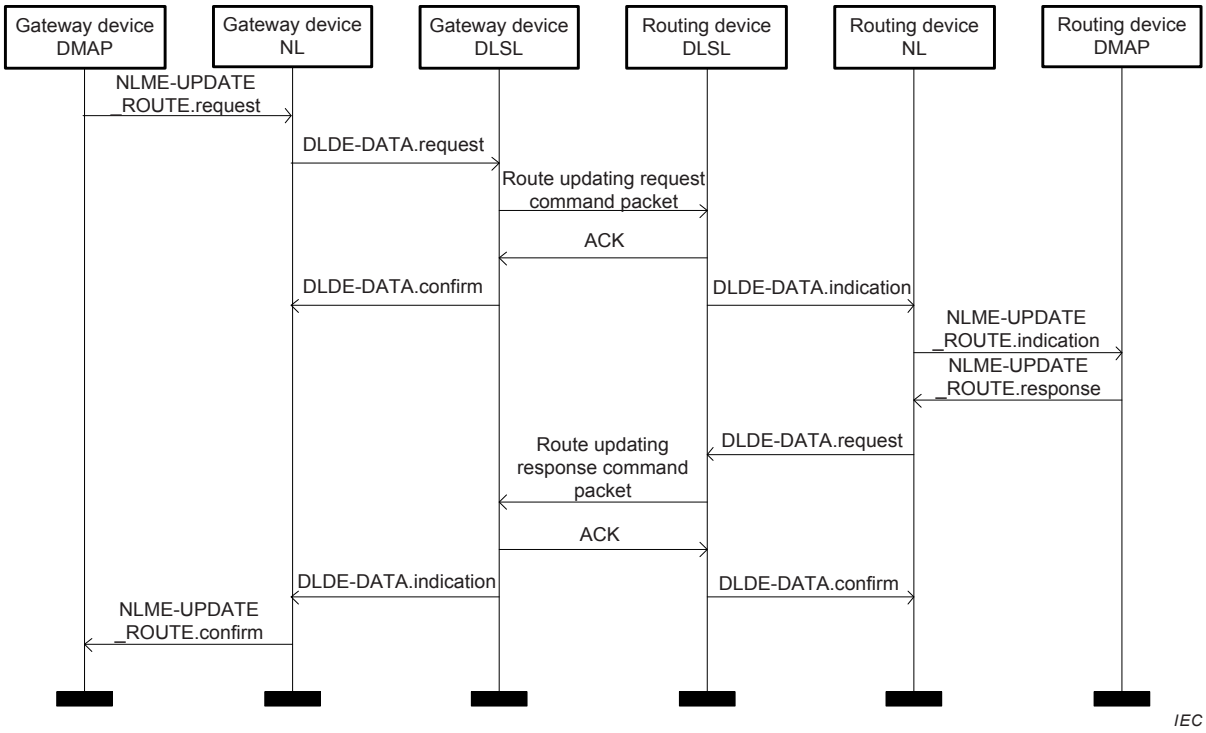

Figure 52 – Time sequence for route updating

9.5.7.3 Route deleting services

9.5.7.3.1 NLME-DELETE_ROUTE.request

The NM invokes NLME-DELETE_ROUTE.request to delete a record in the routing table of a routing device.

The semantics of NLME_DELETE-ROUTE.confirm are described as follows:

```
NLME-DELETE_ROUTE.request(
                                DstAddr,
                                RouteID
                            )
```

Table 94 specifies the parameters for NLME-DELETE-ROUTE.request.

Table 94 – NLME-UPDATE_ROUTE.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	The 16-bit short address of the destination routing device
RouteID	Unsigned16	0 to 65 535	Route ID

9.5.7.3.2 NLME-DELETE_ROUTE.confirm

The NLME_DELETE-ROUTE.confirm reports the result of executing an NLME-DELETE_ROUTE.request.

The semantics of NLME_DELETE-ROUTE.confirm are described as follows:

NLME-DELETE_ROUTE.confirm(

Status
)

Table 95 specifies the parameters for NLME-DELETE_ROUTE.confirm.

Table 95 – NLME-DELETE_ROUTE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of an NLME-DELETE_ROUTE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

On receipt of NLME-DELETE_ROUTE.request, the NL should transmit a route deleting request packetto the routing device and return an NLME-DELETE_ROUTE.confirm to report the result.

9.5.7.3.3 NLME-DELETE_ROUTE.indication

NLME-DELETE_ROUTE.indication is used to report to the DMAP that the device has successfully received a route deleting request packet.

The semantics of NLME- DELETE_ROUTE.confirm are described as follows:

NLME-DELETE_ROUTE.indication (
RouteID
)

Table 94 specifies the parameters for NLME-DELETE_ROUTE.indication.

9.5.7.3.4 NLME-DELETE_ROUTE.response

The NLME-NLME-DELETE_ROUTE.response is the response of NLME-DELETE _ROUTE.indication.

The semantics of NLME-DELETE_ROUTE.response are described as follows:

NLME-DELETE_ROUTE.response (
Status
)

Table 95 specifies the parameters for NLME-DELETE_ROUTE.response.

9.5.7.3.5 Time sequence for route deleting

The time sequence diagram for deleting records in the routing tables of routing devices is shown in Figure 53.

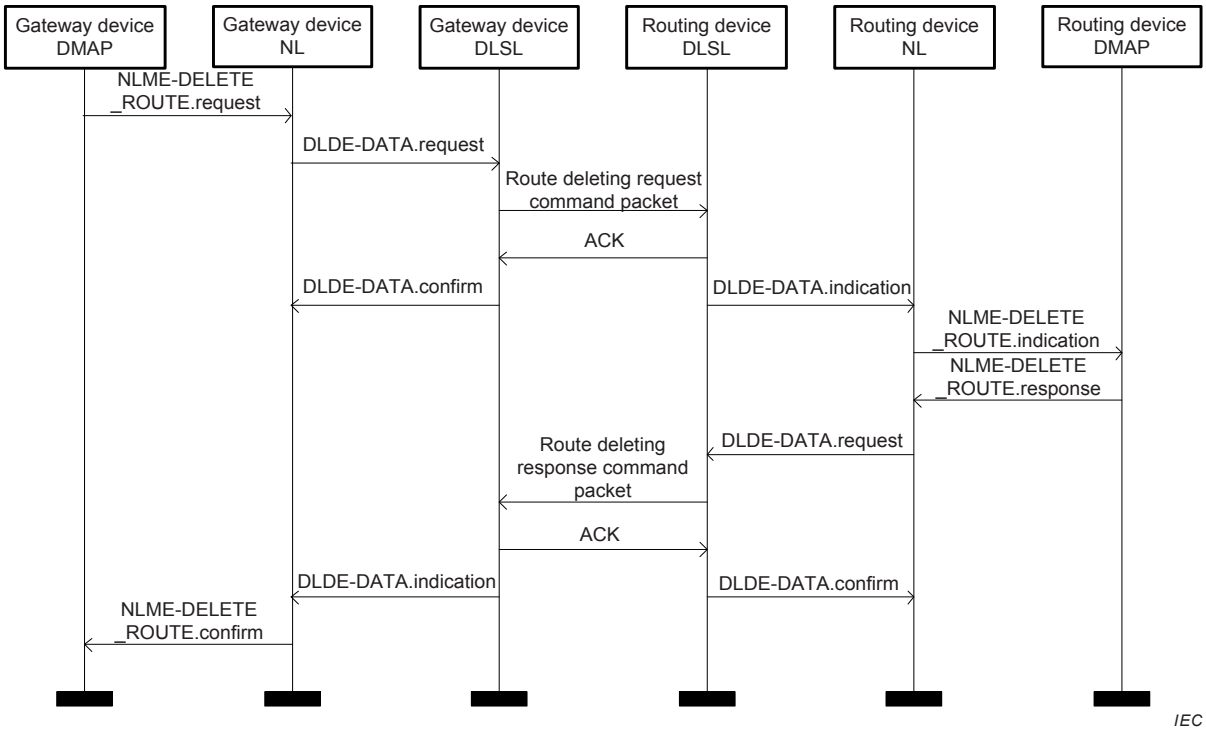

Figure 53 – Time sequence for route deleting

9.5.8 Communication resource allocation services

9.5.8.1 General

Communication resource allocation includes the allocation of link and superframe. These services provide the following primitives:

- Link adding services:
  - NLME-ADD-LINK.request,
  - NLME-ADD-LINK.confirm,
  - NLME-ADD-LINK.response, and
  - NLME-ADD-LINK.indication;
- Link update services:
  - NLME-UPDATE-LINK.request,
  - NLME-UPDATE-LINK.confirm,
  - NLME-UPDATE-LINK.response, and
  - NLME-UPDATE-LINK.indication;
- Link release services:
  - NLME-RELEASE-LINK.request,
  - NLME-RELEASE-LINK.confirm,
  - NLME-RELEASE-LINK.response, and
  - NLME-RELEASE-LINK.indication;
- Superframe adding services:
  - NLME-ADD-SFR.request,
  - NLME-ADD-SFR.confirm,
  - NLME-ADD-SFR.response, and

  - NLME-ADD-SFR.indication;
- Superframe update services:
  - NLME-UPDATE-SFR.request,
  - NLME-UPDATE-SFR.confirm,
  - NLME-UPDATE-SFR.response, and
  - NLME-UPDATE-SFR.indication;
- Superframe release services:
  - NLME-RELEASE-SFR.request,
  - NLME-RELEASE-SFR.confirm,
  - NLME-RELEASE-SFR.response, and
  - NLME-RELEASE-SFR.indication.

9.5.8.2 Link adding services

9.5.8.2.1 NLME-ADD-LINK.request

NLME-ADD-LINK.request is used to add one or more record(s) of new link(s), which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-ADD-LINK.request are described as follows:

```
NLME-ADD-LINK.request (
                        DstAddr,
                        LinkCount,
                        LinkStructure
                      )
```

Table 96 specifies the parameters for NLME-ADD-LINK.request.

Table 96 – NLME-ADD-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support adding multiple links each time
LinkStructure	Link_Struct structure (See Table 17)		Information of the links

9.5.8.2.2 NLME-ADD-LINK.confirm

NLME-ADD-LINK.confirm reports the results of NLME-ADD-LINK.request.

The semantics of NLME-ADD-LINK.confirm are described as follows:

```
NLME-ADD-LINK.confirm (

                        Status
                        )
```

Table 97 specifies the parameters for NLME-ADD-LINK.confirm.

Table 97 – NLME-ADD-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of link adding request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.2.3 NLME-ADD-LINK.indication

NLME-ADD-LINK.indication is used to report to the DMAP that the device has successfully received a link adding request packet.

The semantics of NLME-ADD-LINK.indication are described as follows:

```
NLME-ADD-LINK.indication (
                          LinkCount,
                          LinkStructure
                         )
```

Table 96 specifies the parameters for NLME-ADD-LINK.indication.

9.5.8.2.4 NLME-ADD-LINK.response

NLME-ADD-LINK.response is the response of NLME-ADD-LINK.indication.

The semantics of NLME-ADD-LINK.response are described as follows:

```
NLME-ADD-LINK.response (
                                Status
                                )
```

Table 97 specifies the parameters for NLME-ADD-LINK.response.

9.5.8.2.5 Time sequence for link adding

The time sequence for adding a link originating from the gateway device to a routing device is shown in Figure 54.

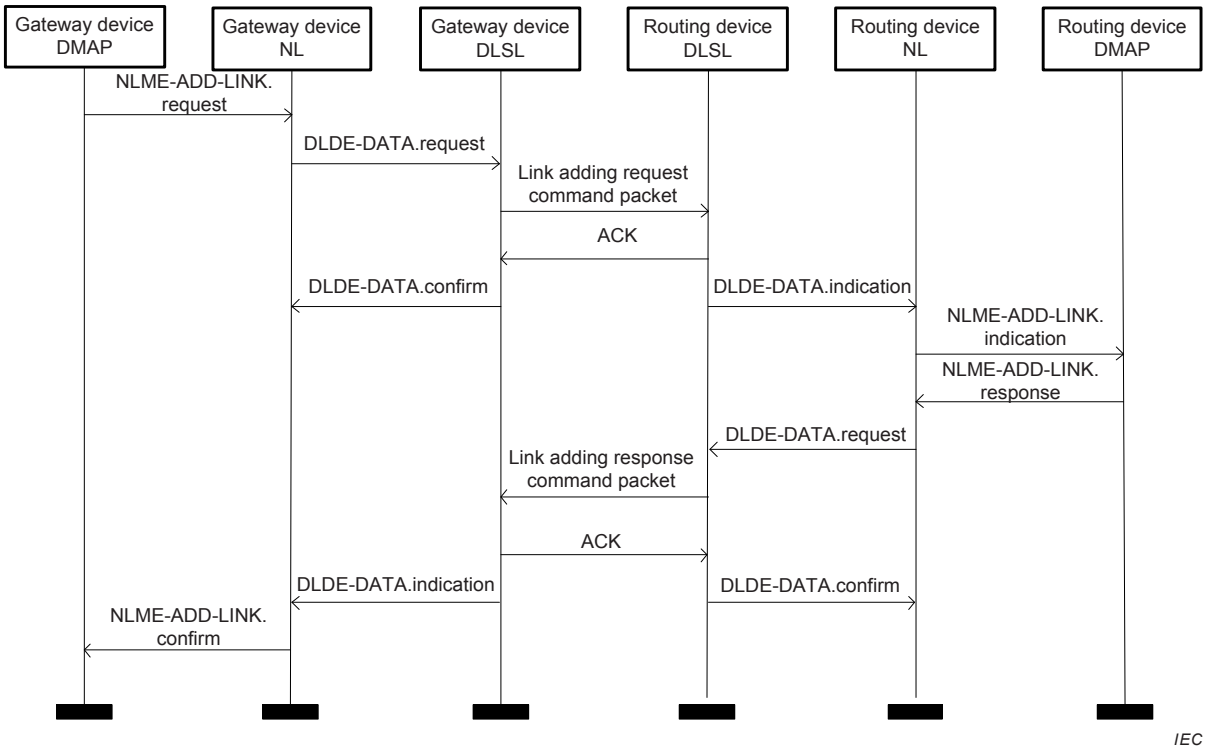

Figure 54 – Adding a link originating from gateway device to routing device

The time sequence for adding a link originating from a routing device to a field device is shown in Figure 55.

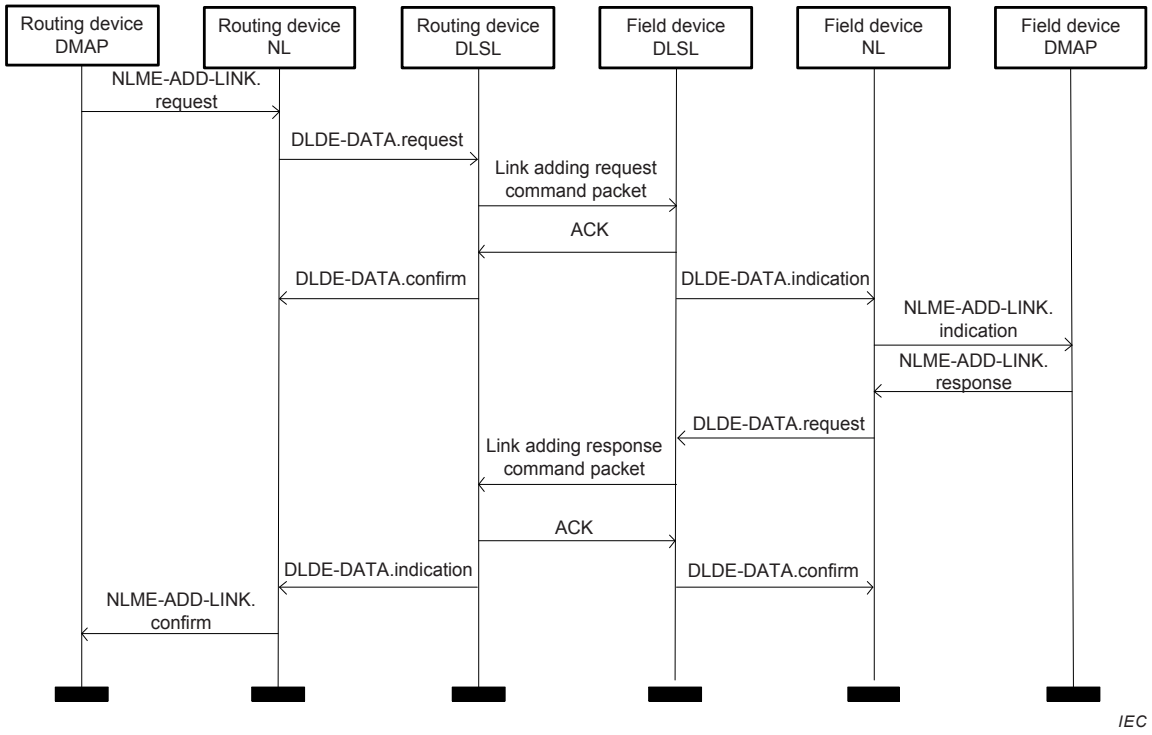

Figure 55 – Adding a link originating from routing device to field device

9.5.8.3 Link update services

9.5.8.3.1 NLME-UPDATE-LINK.request

NLME-UPDATE-LINK.request is used to update one or more record(s) of existing link(s), which is originating either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-UPDATE-LINK.request are described as follows:

```
NLME-UPDATE-LINK.request (
                                DstAddr,
                                LinkCount,
                                LinkStructure
                            )
```

Table 98 specifies the parameters for NLME-UPDATE-LINK.request.

Table 98 – NLME-UPDATE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support updating multiple links each time
LinkStructure	Link_Struct structure (See Table 17)		Information of the links

9.5.8.3.2 NLME-UPDATE-LINK.confirm

NLME-UPDATE-LINK.confirm reports the results of NLME-UPDATE-LINK.request.

The semantics of NLME-UPDATE-LINK.confirm are described as follows:

```
DLME-UPDATA-LINK.confirm (
                                Status
                                )
```

Table 99 specifies the parameters for NLME-UPDATE-LINK.confirm.

Table 99 – NLME-UPDATE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of the link update request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.3.3 NLME-UPDATE-LINK.indication

NLME-UPDATE-LINK.indication is used to report to the DMAP that the device has successfully received a link update request packet.

The semantics of NLME-UPDATE-LINK.indication are described as follows:

```
NLME-UPDATE-LINK.indication     (
                                LinkCount,
                                LinkStructure
                                )
```

Table 98 specifies the parameters for NLME-UPDATE-LINK.indication.

9.5.8.3.4 NLME-UPDATE-LINK.response

NLME-UPDATE-LINK.response is the response to NLME-UPDATE-LINK.indication.

The semantics of NLME-UPDATE-LINK.response are described as follows:

```
NLME-UPDATE-LINK.response (
                                    Status
                                    )
```

Table 99 specifies the parameters for NLME-UPDATE-LINK.response.

9.5.8.3.5 Time sequence for link update

The time sequence for updating a link originating from the gateway device to a routing device is shown in Figure 56.

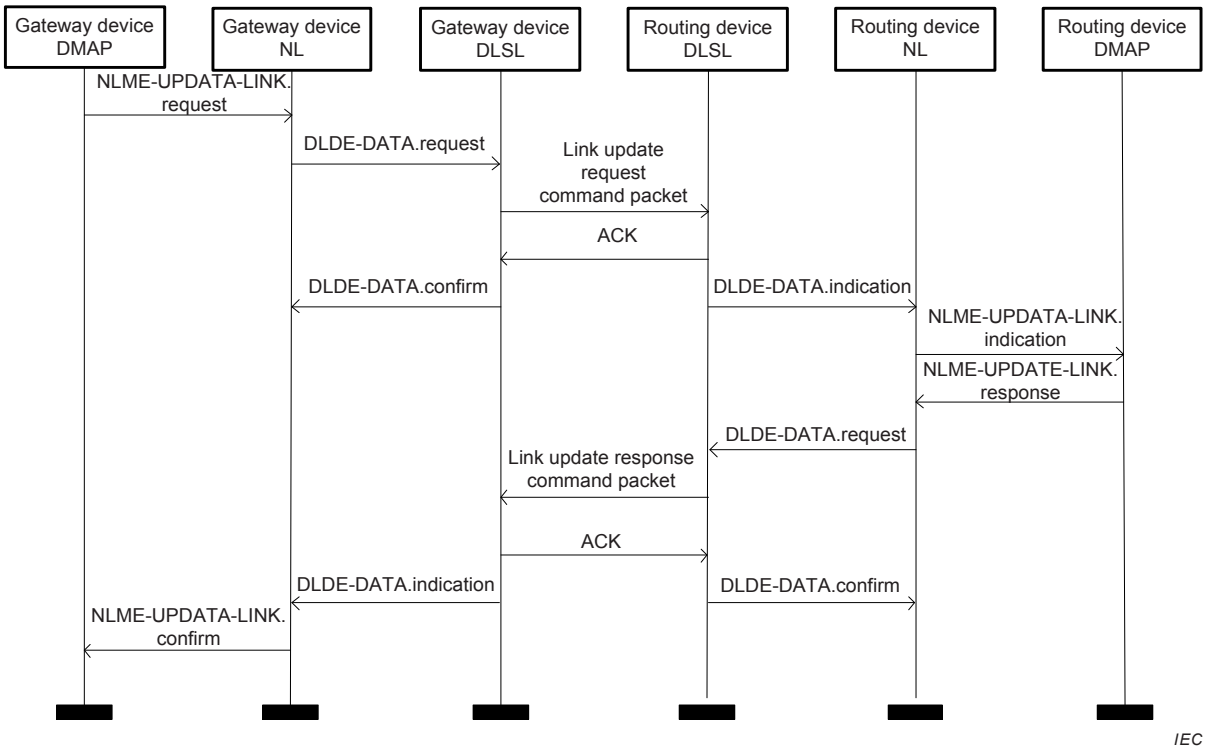

Figure 56 – Updating a link originating by gateway device to routing device

The time sequence for updating a link originating from a routing device to a field device is shown in Figure 57.

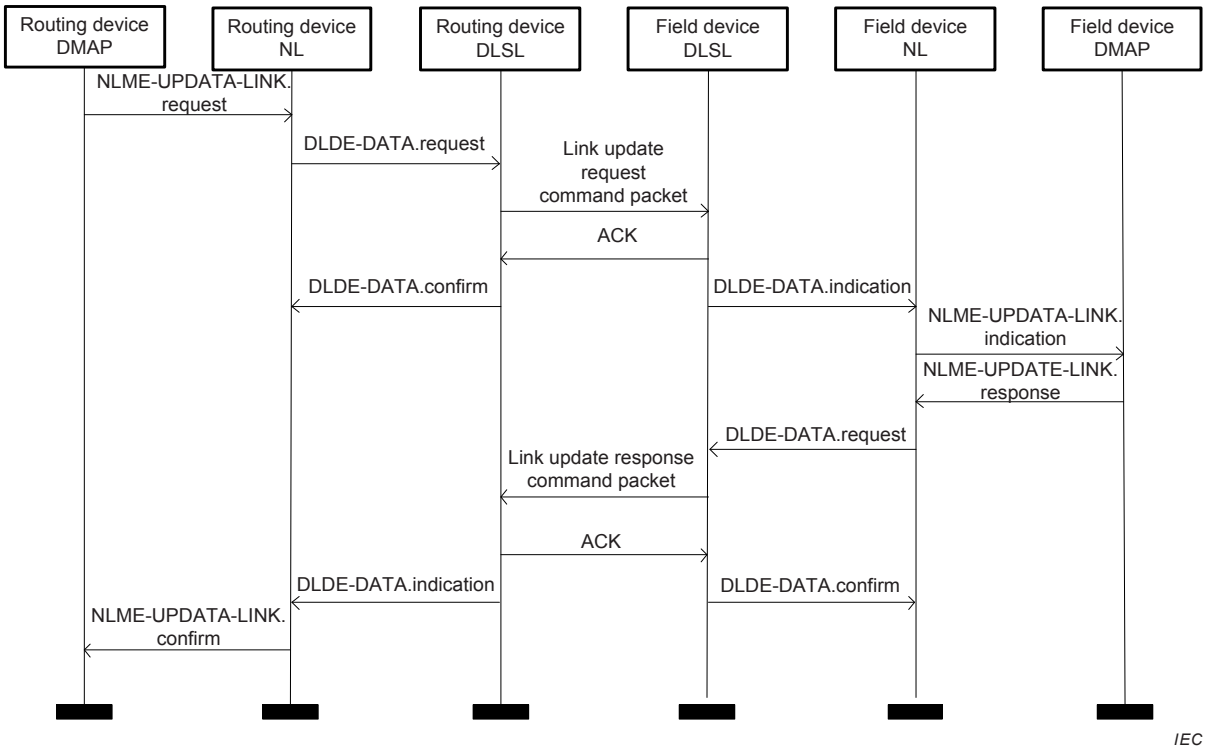

Figure 57 – Updating a link originating from routing device to field device

9.5.8.4 Link release services

9.5.8.4.1 NLME-RELEASE-LINK.request

NLME-RELEASE-LINK.request is used to delete one or more existing link(s), which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-RELEASE-LINK.request are described as follows:

```
NLME-RELEASE-LINK.request (
                              DstAddr,
                              LinkCount,
                              LinkID[LinkCount]
                            )
```

Table 100 specifies the parameters for NLME-RELEASE-LINK.request.

Table 100 – NLME-RELEASE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
LinkCount	Unsigned16	0 to 65 535	Count of added links to support releasing multiple links each time
LinkID[LinkCount]	Unsigned16	0 to 65 535	IDs of the deleted links

9.5.8.4.2 NLME-RELEASE-LINK.confirm

NLME-RELEASE-LINK.confirm reports the results of NLME-RELEASE-LINK.request.

The semantics of NLME-RELEASE-LINK.confirm are described as follows:

```
NLME-RELEASE-LINK.confirm (
                                Status
                            )
```

Table 101 specifies the parameters for NLME-RELEASE-LINK.confirm.

Table 101 – NLME-RELEASE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of the link release request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.4.3 NLME-RELEASE-LINK.indication

NLME-RELEASE-LINK.indication is used to report to the DMAP that the device has successfully received a link release request packet.

The semantics of NLME-RELEASE-LINK.indication are described as follows:

```
NLME-RELEASE-LINK.indication    (
                                  LinkCount,
                                  LinkID[LinkCount]
                                   )
```

Table 100 specifies the parameters for NLME-RELEASE-LINK.indication.

9.5.8.4.4 NLME-RELEASE-LINK.response

NLME-RELEASE-LINK.response is the response to NLME-RELEASE-LINK.indication.

The semantics of NLME-RELEASE-LINK.response are described as follows:

```
NLME-RELEASE-LINK.response (
                                    Status
                                )
```

Table 101 specifies the parameters for NLME-RELEASE-LINK.response.

9.5.8.4.5 Time sequence for link release

The time sequence for releasing a link originating from the gateway device to a routing device is shown in Figure 58.

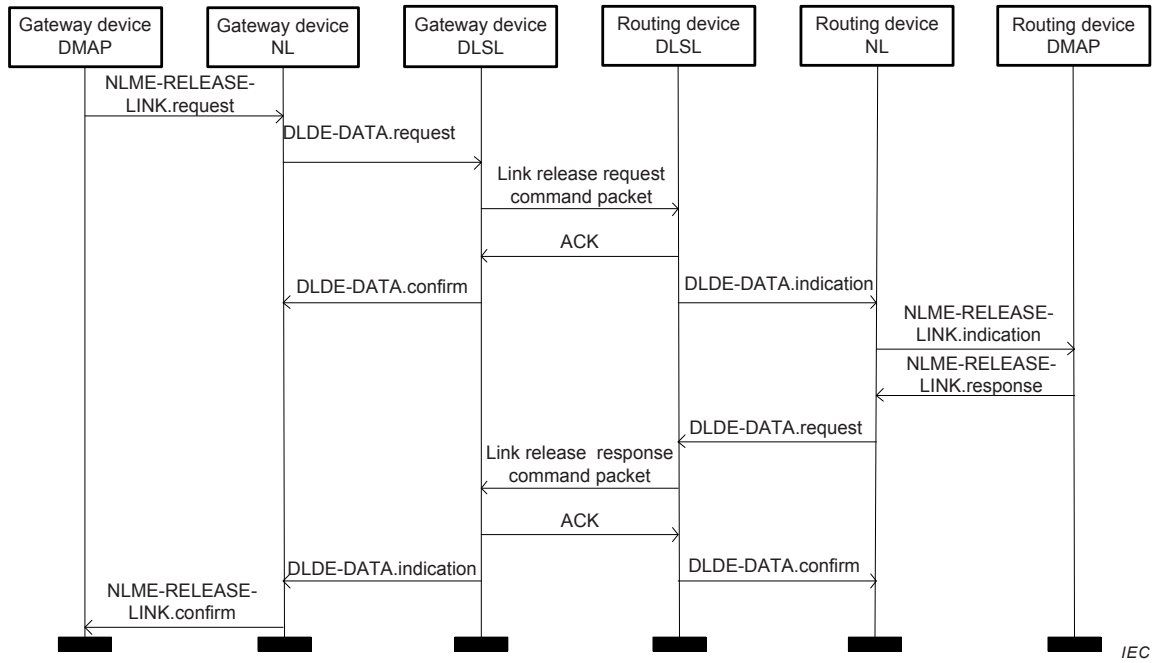

Figure 58 – Releasing a link originating from gateway device to routing device

The time sequence for releasing a link originating from a routing device to a field device is shown in Figure 59.

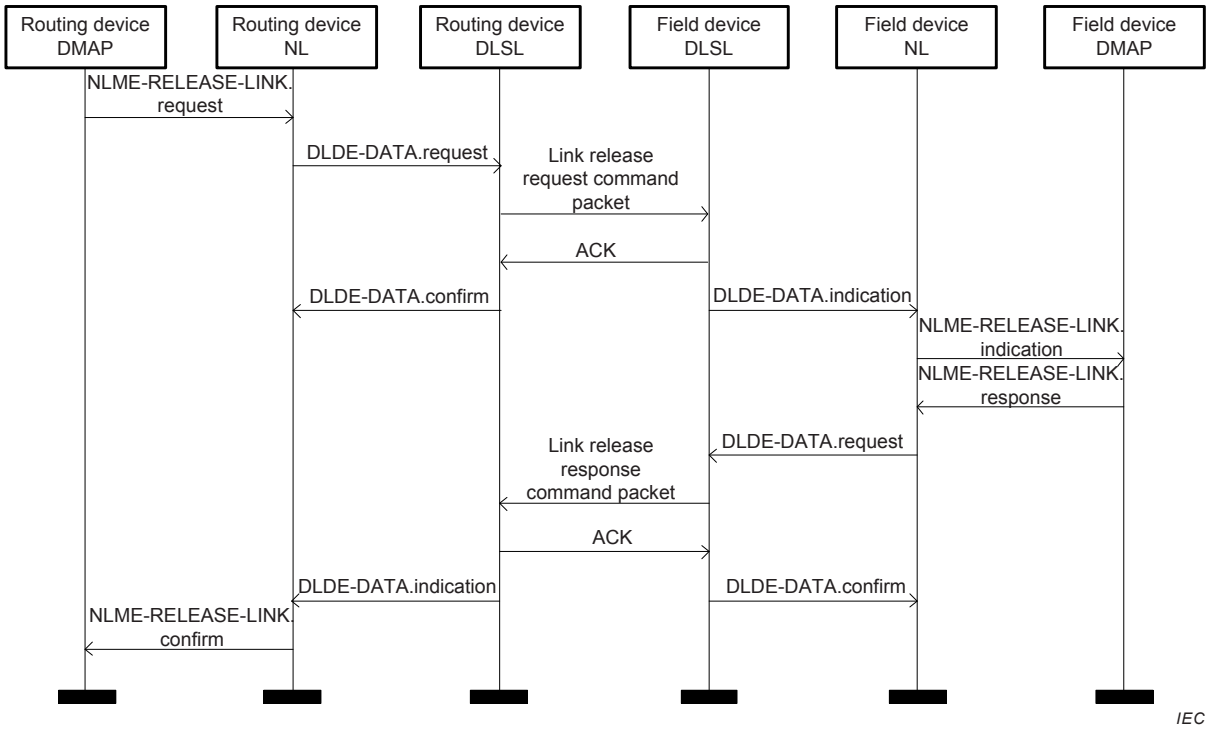

Figure 59 – Releasing a link originating from routing device to field device

9.5.8.5 Superframe adding services

9.5.8.5.1 NLME-ADD-SFR.request

NLME-ADD-SFR.request is used to add a record of a new superframe, which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-ADD-SFR.request are described as follows:

```
NLME-ADD-SFR.request (
                        DstAddr,
                        SuperframeStructure
                    )
```

Table 102 specifies the parameters for NLME-ADD-SFR.request.

Table 102 – NLME-ADD-SFR.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
SuperframeStructure	Superframe_Structstructure (See Table 16)		Information of superframe attribute

9.5.8.5.2 NLME-ADD-SFR.confirm

NLME-ADD-SFR.confirm reports the result of NLME-ADD-SFR.request.

The semantics of NLME-ADD-SFR.confirm are described as follows:

```
NLME-ADD-SFR.confirm (
                        Status
                    )
```

Table 103 specifies the parameters for NLME-ADD-SFR.confirm.

Table 103 – NLME-ADD-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe release request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.5.3 NLME-ADD-SFR.indication

NLME-ADD-SFR.indication is used to report to the DMAP that the device has successfully received a superframe adding request packet.

The semantics of NLME-ADD-SFR.indication are described as follows:

```
NLME-ADD-SFR.indication    (
                            SuperframeStructure
                            )
```

Table 102 specifies the parameters for NLME-ADD-SFR.indication.

9.5.8.5.4 NLME-ADD-SFR.response

NLME-ADD-SFR.response is the response to NLME-ADD-SFR.indication.

The semantics of NLME-ADD-SFR.response are described as follows:

```
NLME-ADD-SFR.response        (
                             Status
                             )
```

Table 103 specifies the parameters for NLME-ADD-SFR.response.

9.5.8.5.5 Time sequence for superframe adding

The time sequence for adding a superframe originating from the gateway device to a routing device is shown in Figure 60.

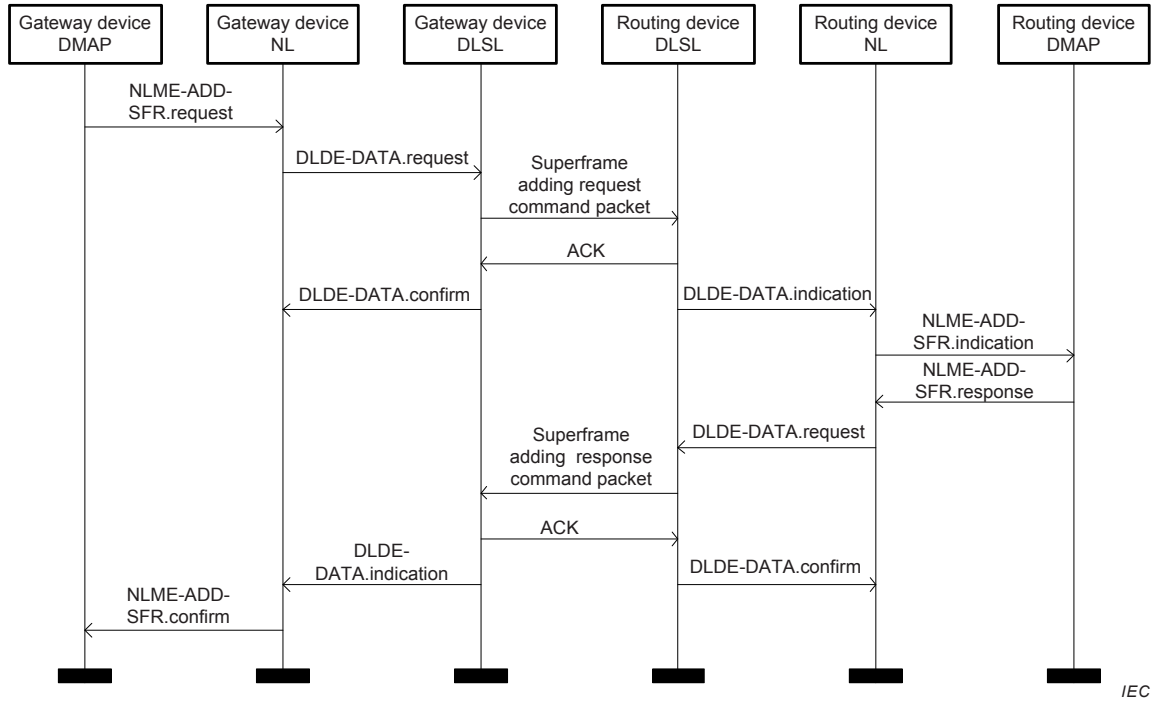

Figure 60 – Adding a superframe originating from gateway device to routing device

The time sequence for adding a superframe originating from a routing device to a field device is shown in Figure 61.

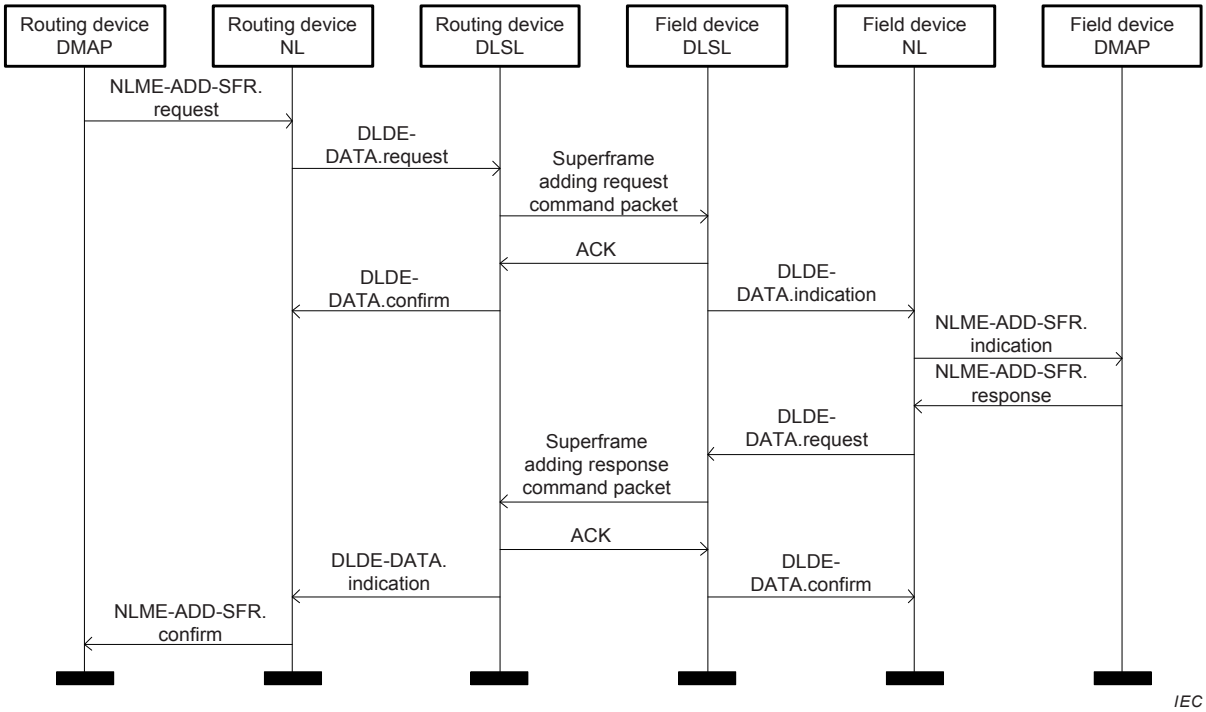

Figure 61 – Adding a superframe originating from routing device to field device

9.5.8.6 Superframe update services

9.5.8.6.1 NLME-UPDATA-SFR.request

NLME-UPDATA-SFR.request is used to modify a record of an existing superframe, which is originated from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-UPDATA-SFR.request are described as follows:

```
NLME-UPDATA-SFR.request (
                          DstAddr,
                          SuperframeStructure
                        )
```

Table 104 specifies the parameters for NLME-UPDATA-SFR.request.

Table 104 – NLME-UPDATA-SFR.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit address of destination device
SuperframeStructure	Superframe_Struct structure (See Table 16)		Information of superframe attribute

9.5.8.6.2 NLME-UPDATE-SFR.confirm

NLME-UPDATE-SFR.confirm reports the results of NLME-UPDATA-SFR.request.

The semantics of NLME-UPDATE-SFR.confirm are described as follows:

```
NLME-UPDATE-SFR.confirm (
                          Status
                          )
```

Table 105 specifies the parameters for NLME-UPDATE-SFR.confirm.

Table 105 – NLME-UPDATE-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe update request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.6.3 NLME-UPDATE-SFR.indication

NLME-UPDATE-SFR.indication is used to report to the DMAP that the device has successfully received a superframe update request packet.

The semantics of NLME-UPDATE-SFR.indication are described as follows:

```
NLME-UPDATE-SFR.indication    (
                              SuperframeStructure
                               )
```

Table 104 specifies the parameters for NLME-UPDATE-SFR.indication.

9.5.8.6.4 NLME-UPDATE-SFR.response

NLME-UPDATE-SFR.response is the response to NLME-UPDATE-SFR.indication.

The semantics of NLME-UPDATE-SFR.response are described as follows:

```
NLME-UPDATE-SFR.response (
                                    Status
                                    )
```

Table 105 specifies the parameters for NLME-UPDATE-SFR.response.

9.5.8.6.5 Time sequence for superframe update

The time sequence for updating a superframe originating from the gateway device to a routing device is shown in Figure 62.

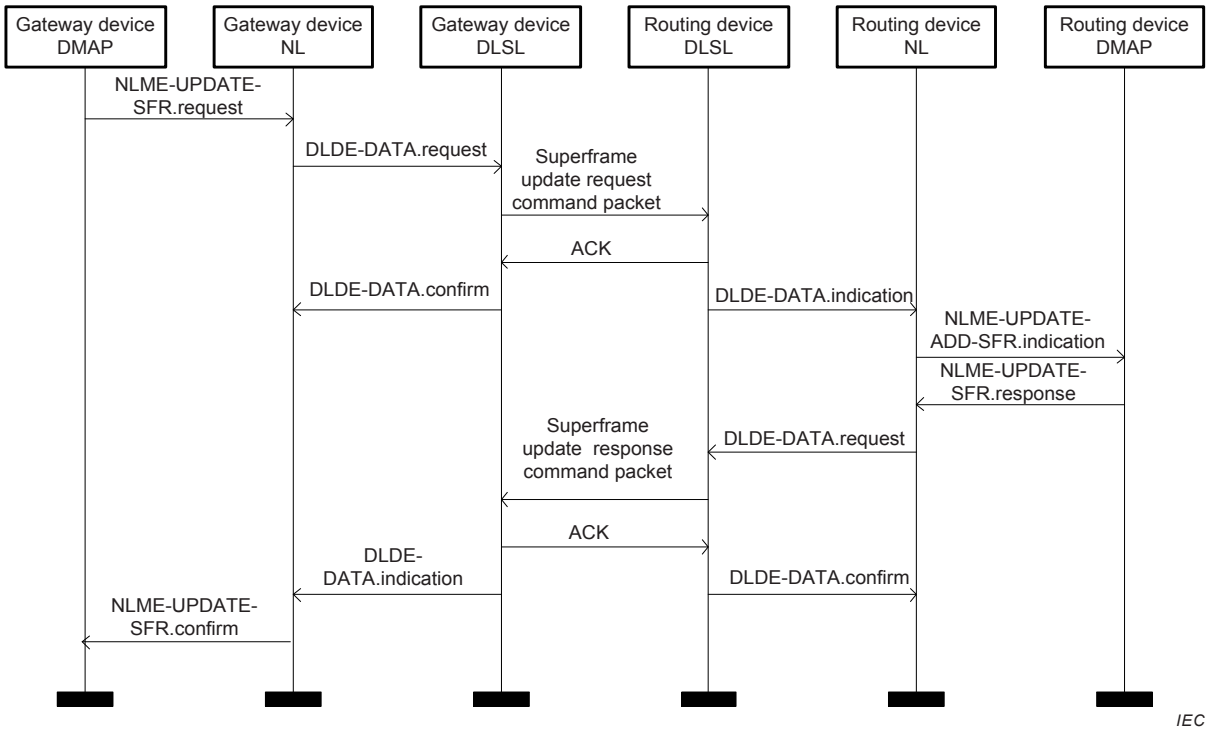

Figure 62 – Updating a superframe originating from gateway device to routing device

The time sequence for updating a superframe originating from a routing device to a field device is shown in Figure 63.

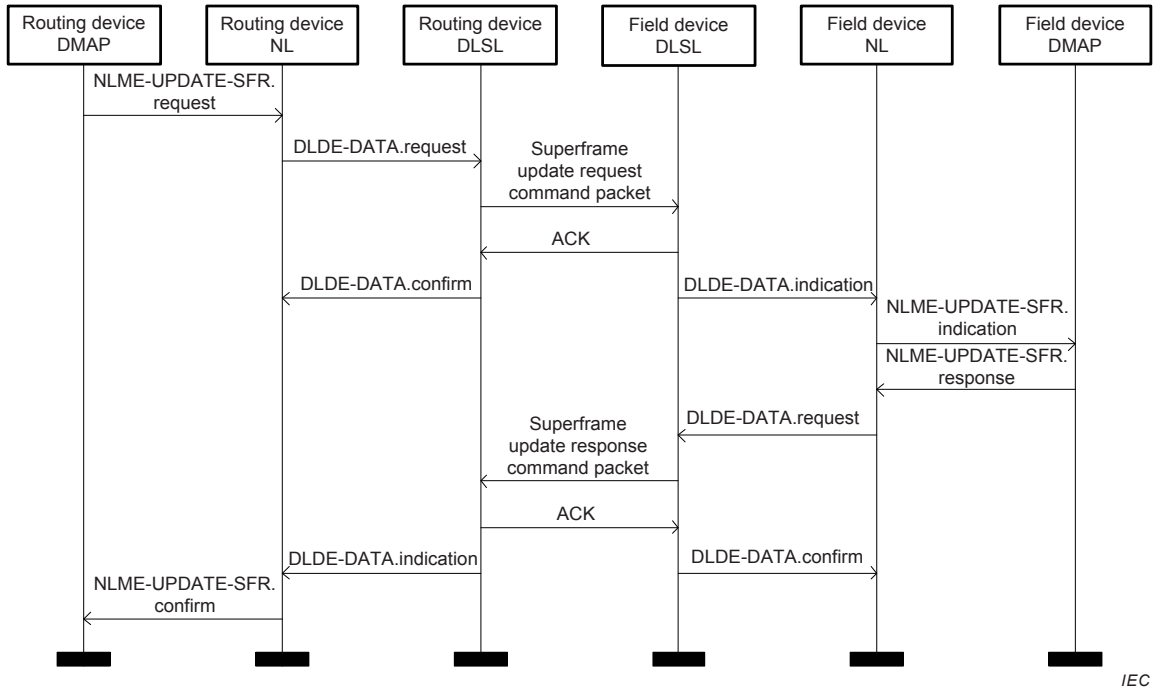

Figure 63 – Updating a superframe originating from routing device to field device

9.5.8.7 Superframe release services

9.5.8.7.1 NLME-RELEASE-SFR.request

NLME-RELEASE-SFR.request is used to delete a record of an existing superframe, which is originated either from the gateway device to a routing device or from a routing device to a field device.

The semantics of NLME-RELEASE-SFR.request are described as follows:

```
NLME-RELEASE-SFR.request (
                              DstAddr,
                              SuperframeID
                          )
```

Table 106 specifies the parameters for NLME-RELEASE-SFR.request.

Table 106 – NLME-RELEASE-SFR.request parameters

Name	Data type	Valid range	Descrition
DstAddr	Unsigned16	0 to 65 535	16-bit destination address
SuperframeID	Unsigned16	0 to 65 535	ID of the deleted superframe

9.5.8.7.2 NLME-RELEASE-SFR.confirm

NLME-RELEASE-SFR.confirm reports the result of NLME-RELEASE-SFR.request.

The semantics of NLME-RELEASE-SFR.confirm are described as follows:

```
NLME-RELEASE-SFR.confirm (
                              Status
                          )
```

Table 107 specifies the parameters for NLME-RELEASE-SFR.confirm.

Table 107 – NLME-RELEASE-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Results of superframe release request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.8.7.3 NLME-RELEASE-SFR.indication

NLME-RELEASE-SFR.indication is used to report to the DMAP that the device has successfully received a superframe release request packet.

The semantics of NLME-RELEASE-SFR.indication are described as follows:

```
NLME-RELEASE-SFR.indication    (
                                SuperframeID
                               )
```

Table 106 specifies the parameters for NLME-RELEASE-SFR.indication.

9.5.8.7.4 NLME-RELEASE-SFR.response

NLME-RELEASE-SFR.response is the response of NLME-RELEASE-SFR.indication.

The semantics of NLME-RELEASE-SFR.response are described as follows:

```
NLME-RELEASE-SFR.response (
                              Status
                           )
```

Table 107 specifies the parameters for NLME-RELEASE-SFR.response.

9.5.8.7.5 Time sequence for superframe release

The time sequence for releasing a superframe originating from the gateway device to a routing device is shown in Figure 64.

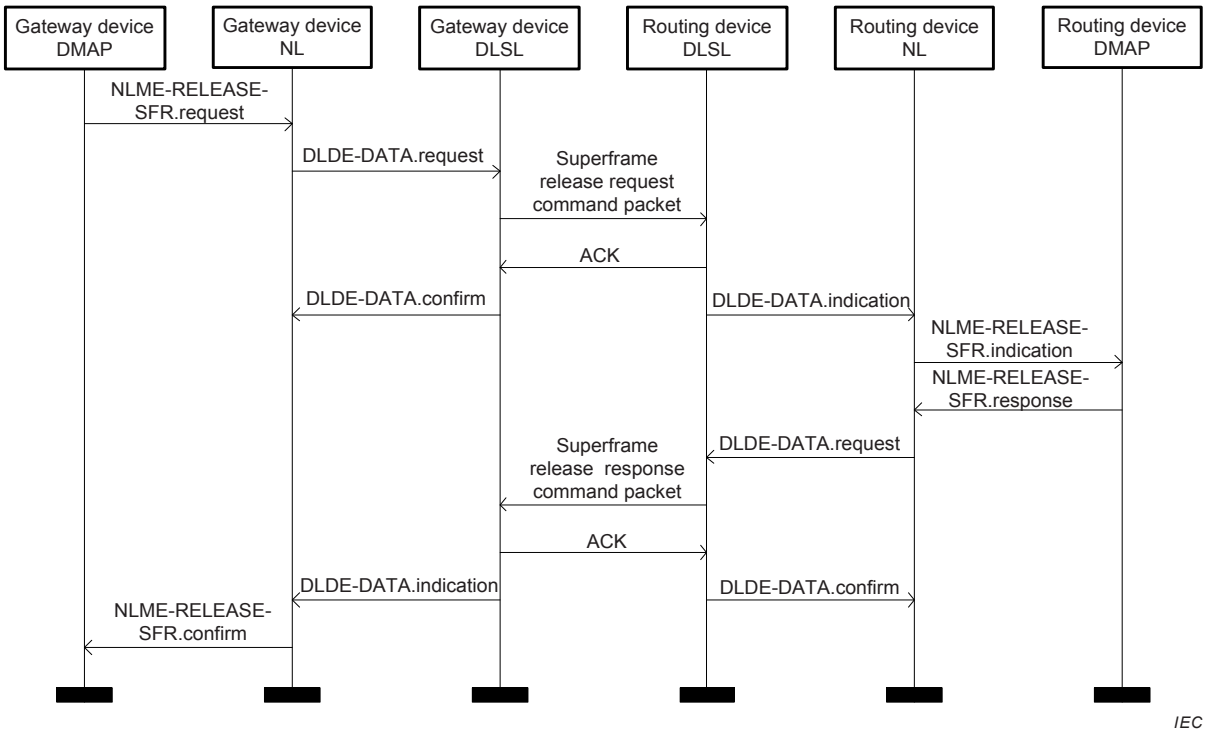

Figure 64 – Releasing a superframe originating from gateway device to routing device

The time sequence for releasing a superframe originating from a routing device to a field device is shown in Figure 65.

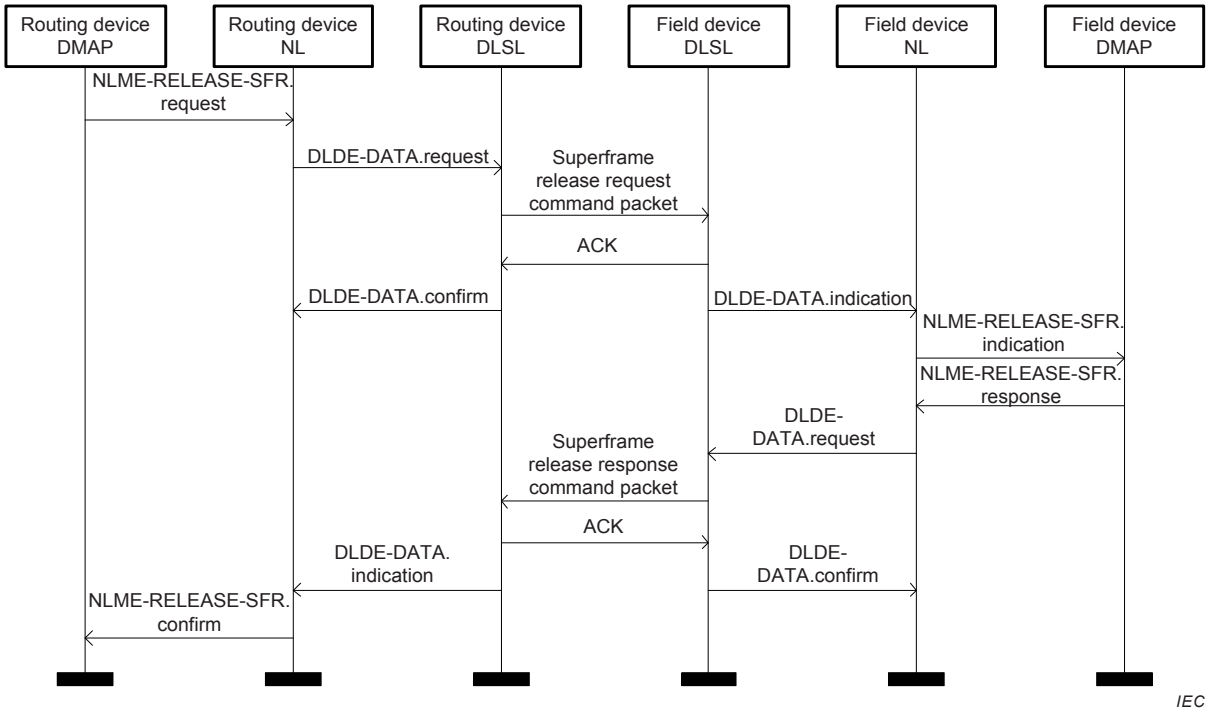

Figure 65 – Releasing a superframe originating from routing device to field device

9.5.9 Aggregation and disaggregation services

9.5.9.1 NLME-AGG.indication

NLME-AGG.indication is used for the NL to report to the DMAP that the received packets need to be aggregated.

The semantics of NLME-AGG.indication are described as follows:

```
NLME-AGG.indication (
                        SrcAddr,
                        NwkPayload
                    )
```

Table 108 specifies the parameters for NLME-AGG.indication.

Table 108 – NLME-AGG.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	ID of the aggregation VCR
NwkPayload	Octetstring		Payload of NL

9.5.9.2 NLME-AGO-SEND.request

NLME-AGO-SEND.request is used by the DMAP asking the NL to send the aggregated packet after the aggregation is completed.

The semantics of NLME-AGO-SEND.request are described as follows:

```
NLME-AGO-SEND.request (
                            GatewayAddr,
                            AggPacketFlag,
                            Priority,
                            AggNumber,
                            AGGRouteID,
                            PSFlag,
                            AggPayload
                        )
```

Table 109 specifies the parameters for NLME-AGO-SEND.request.

Table 109 – NLME-AGO-SEND.request parameters

Name	Data type	Valid range	Description
GatewayAddr	Unsigned16	0 to 65 535	Network address of the gateway device
AggPacketFlag	Unsigned8	0, 1	Indicating whether this packet is a data packet or aggregation packet: 0 = Data packet; 1 = Aggregation packet.
Priority	Unsigned8	0 to 15	Priority of process data
AggNumber	Unsigned8	0 to 255	Number of aggregated packets
AGGRouteID	Unsigned16	0 to 65 535	RouteID of the aggregation
PSFlag	Unsigned8	0, 1	Indicating whether this packet is P/S type: 0 = Not P/S type; 1 = P/S type.
AggPayload	Octetstring		DMAP aggregated packet

9.5.9.3 NLME-DAG.indication

NLME-DAG.indication is used to send the aggregated packet to the DMAP for disaggregating when the NL receives an aggregated packet.

The semantics of NLME-DAG.indication are described as follows:

NLME-DAG.indication (

AggNumber,
NwkPalyload
)

Table 110 specifies the parameters for NLME-DAG.indication.

Table 110 – NLME-DAG.indication parameter

Name	Data type	Valid range	Description
AggNumber	Unsigned8	0 to 255	Number of aggregated packets
NwkPayload	Octetstring		Payload of NL

9.5.10 Device status report services

9.5.10.1 NLME-DEVICE-STATUS.request

NLME-DEVICE-STATUS.request is used to report the device status either to the routing devices by a field device or to the gateway device by a routing device.

The semantics of NLME-DEVICE -STATUS.request are described as follows:

NLME-CHANNEL-DEVICE.request(

DstAddr,
DeviceCount,
DeviceConditionInfo
)

Table 111 specifies the parameters for NLME-DEVICE -STATUS.request.

Table 111 – NLME-DEVICE -STATUS.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	Destination address
DeviceCount	Unsigned8	0 to 255	Total number of reported devices
DeviceConditionInfo	DevConRep_Struct structure (see Table 22)		Information of device condition attributes

9.5.10.2 NLME-DEVICE-STATUS.indication

NLME-DEVICE-STATUS.indication is used to report the receipt of a device condition report command packet to the DMAP.

The semantics of NLME-DEVICE -STATUS.indication are described as follows:

NLME- DEVICE -STATUS.indication (

SrcAddr,
DeviceCount,
DeviceConditionInfo
)

Table 112 specifies the parameters for NLME-DEVICE -STATUS.indication.

Table 112 – NLME-DEVICE -STATUS.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	Source address
DeviceCount	Unsigned8	0 to 255	Total number of reported devices
DeviceConditionInfo	DevConRep_Struct structure (see Table 22)		Information of device condition attributes

9.5.10.3 NLME-DEVICE-STATUS.confirm

NLME-DEVICE-STATUS.confirm is used to return the results of NLME-DEVICE-STATUS.request.

The semantics of NLME-DEVICE-STATUS.confirm are described as follows:

NLME- DEVICE -STATUS.confirm (

Status
)

Table 113 specifies the parameters for NLME-DEVICE -STATUS.confirm.

Table 113 – NLME-DEVICE -STATUS.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of the channel condition report request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.10.4 Time sequence for device status reporting

The time sequence diagram for device status information is shown in Figure 66 and Figure 67.

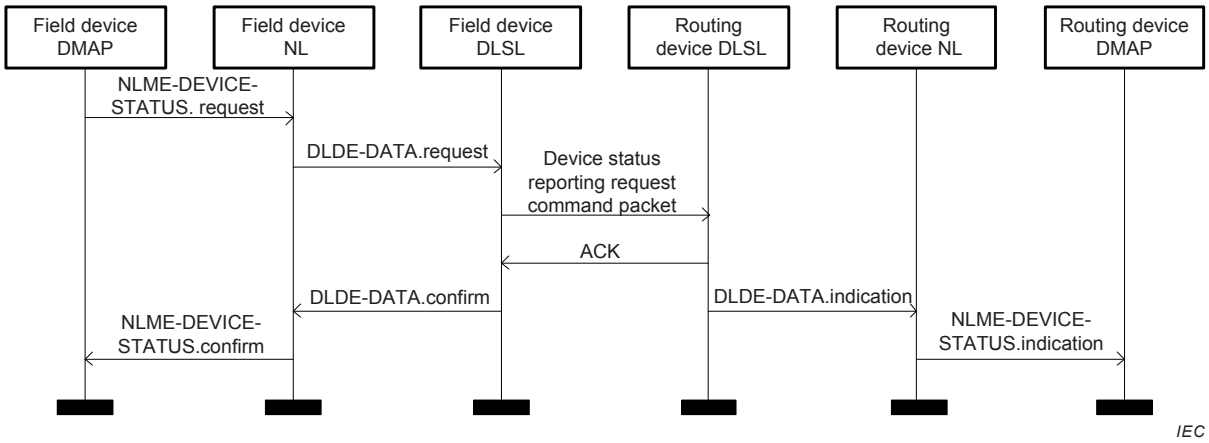

Figure 66 – Device status reporting process from field device to routing device

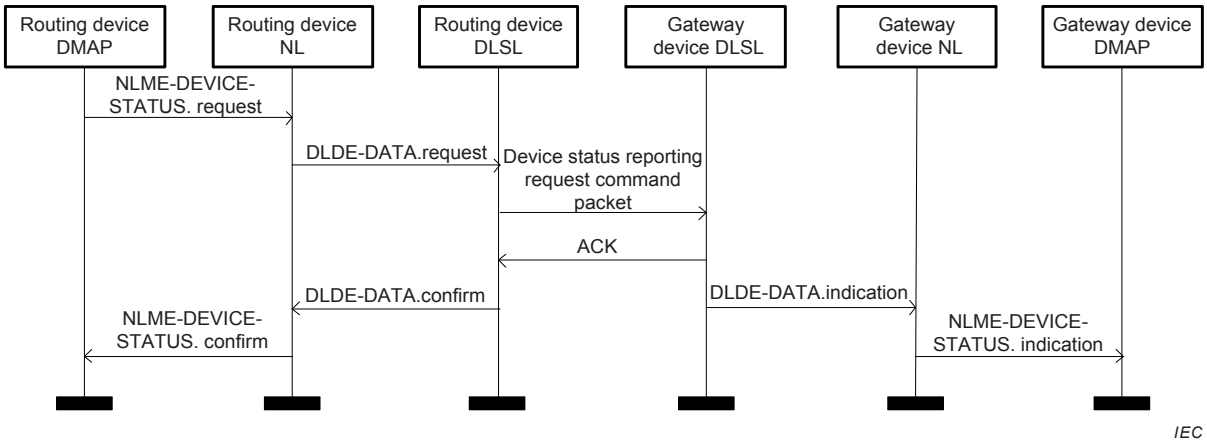

Figure 67 – Device status reporting process from routing device to gateway device

9.5.11 Channel condition report services

9.5.11.1 NLME-CHANNEL-CONDITION.request

NLME-CHANNEL-CONDITION.request is used to report the communication channel condition either to the routing device by a field device or to the gateway device by a routing device.

The semantics of NLME-CHANNEL-CONDITION.request are described as follows:

NLME-CHANNEL-CONDITION.request(

DstAddr,
ChannelCount,
ChannelConditionInfo
)

Table 114 specifies the parameters for NLME-CHANNEL-CONDITION.request.

Table 114 – NLME-CHANNEL-CONDITION.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	Destination address
ChannelCount	Unsigned8	0 to 255	Total number of channels
ChannelConditionInfo	ChanCon_Struct structure (See Table 19)		Information of channel condition attributes

9.5.11.2 NLME-CHANNEL-CONDITION.indication

NLME-CHANNEL-CONDITION.indication is used to report the receipt of a channel condition report command packet to the DMAP.

The semantics of NLME-CHANNEL-CONDITION.indication are described as follows:

NLME-CHANNEL-CONDITION.indication (

SrcAddr,
ChannelCount,
ChannelConditionInfo
)

Table 115 specifies the parameters for NLME-CHANNEL-CONDITION.indication.

Table 115 – NLME-CHANNEL-CONDITION.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	Source address
ChannelCount	Unsigned8	0 to 255	Total number of channels
ChannelConditionInfo	ChanCon_Struct structure (see Table 19)		Information of channel condition attributes

9.5.11.3 NLME-CHANNEL-CONDITION.confirm

NLME-CHANNEL-CONDITION.confirm is used to return the results of NLME-CHANNEL-CONDITION.request.

The semantics of NLME-CHANNEL-CONDITION.confirm are described as follows:

NLME-CHANNEL-CONDITION.confirm (

Status
)

Table 116 specifies the parameters for NLME-CHANNEL-CONDITION.confirm.

Table 116 – NLME-CHANNEL-CONDITION.confirm parameter

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	Result of the channel condition report request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.11.4 Time sequence for channel condition reporting

The time sequence diagram for channel condition information is shown in Figure 68 and Figure 69.

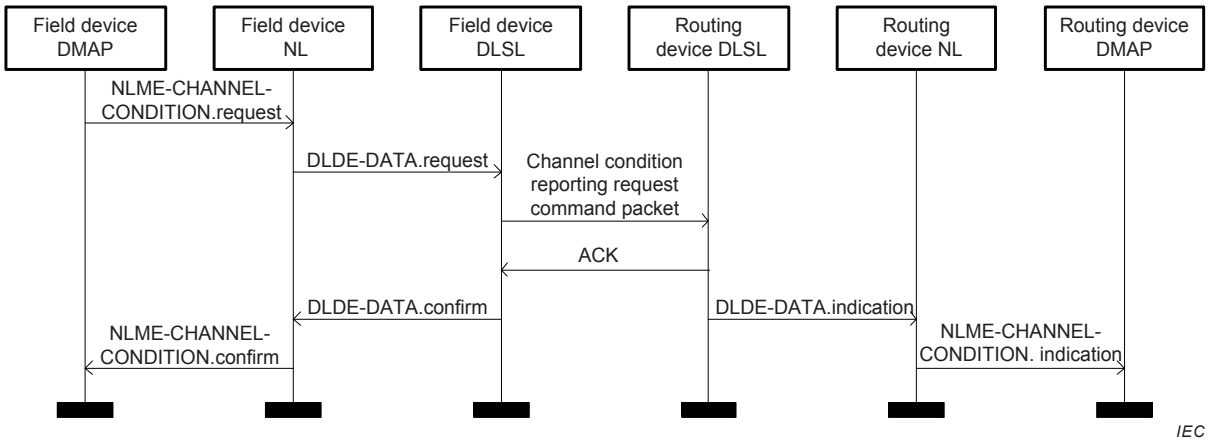

Figure 68 – Channel condition reporting process from field device to routing device

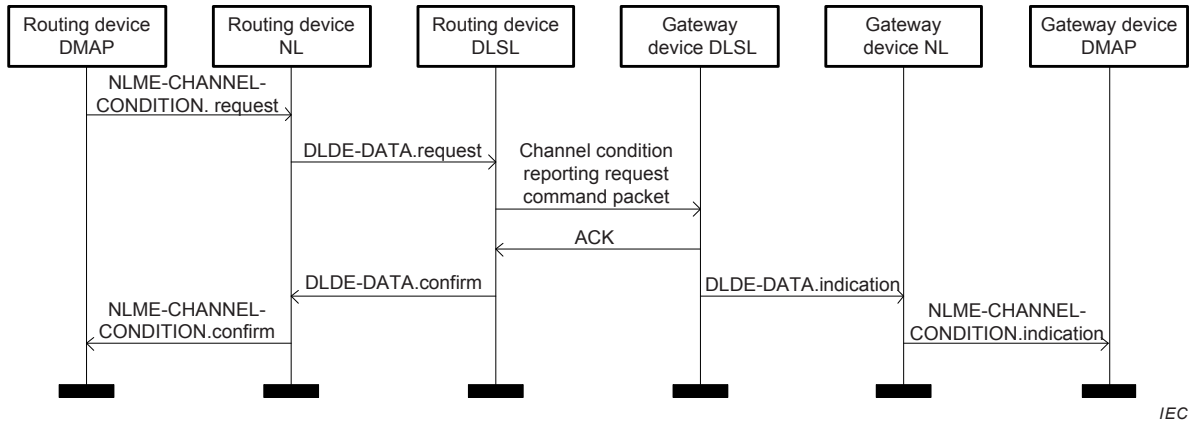

Figure 69 – Channel condition reporting process from routing device to gateway device

9.5.12 Failure path report services

9.5.12.1 General

All routing devices on a path record the value of RetryCounter (see Table 69) with next hop routing device. If RetryCounter value of the source routing device or an intermediate routing device exceeds the value of MaxEtoERetry, the source routing device reports the failure of related RouteID to the GW by using the redundant path and the intermediate reports the failure of related RouteID to the GW by using its pre-configured path (see 6.5.1 for redundant path).

9.5.12.2 NLME-PATH_FAILURE.request

NLME-PATH_FAILURE.request is used to report failed paths to the GW by routing devices.

The semantics of NLME-PATH_FAILURE.request are described as follows:

```
NLME-PATH_FAILURE.request (
                                SrcAddr,
                                DstAddr,
                                RouteID
                            )
```

Table 117 specifies the parameters for NLME-PATH_FAILURE.request.

Table 117 – NLME-PATH_FAILURE.request parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address of packets
DstAddr	Unsigned16	0 to 65 535	16-bit destination address of packets
RouteID	Unsigned16	0 to 65 535	Route ID of failing path

9.5.12.3 NLME-PATH_FAILURE.indication

NLME-PATH_FAILURE.indication is used by the NL to report to the DMAP that the device has successfully received a failed-path reporting request packet.

The semantics of NLME-PATH_FAILURE.indication are described as follows:

```
NLME-PATH_FAILURE.indication (
                                SrcAddr,
                                RouteID
                            )
```

Table 118 specifies the parameters for NLME-LEAVE.response.

Table 118 – NLME-PATH_FAILURE.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit source address of packets
RouteID	Unsigned16	0 to 65 535	Route ID of failing path

9.5.12.4 NLME-PATH_FAILURE.confirm

NLME-PATH_FAILURE.confirm is used to return the results of NLME-PATH_FAILURE.request.

The semantics of NLME-PATH_FAILURE.confirm are described as follows:

NLME-PATH_FAILURE.confirm (
Status
)

Table 119 specifies the parameters for NLME-PATH_FAILURE.confirm.

Table 119 – NLME-PATH_FAILURE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 to 255	The result of NLME-PATH_FAILURE.request: 0 = SUCCESS; 1 = FAILURE; Others are reserved.

9.5.12.5 Time sequence for failure path reporting

The time sequence diagram for failure path reporting is shown in Figure 70.

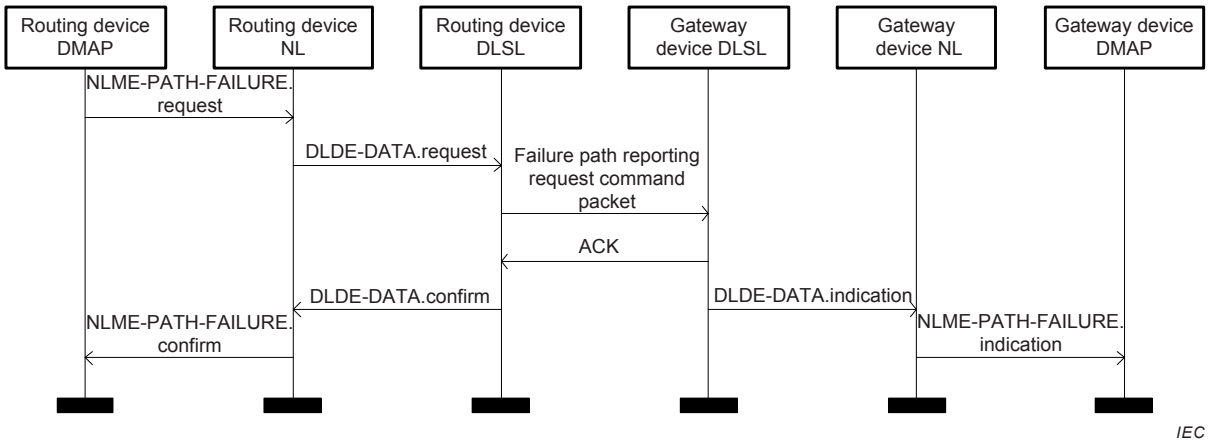

Figure 70 – Failure path reporting process

9.5.13 Network attribute getting services

9.5.13.1 NLME-INFO_GET.request

NLME-INFO_GET.request is used to remotely read the values of the attributes in the MIB.

The semantics of NLME-INFO_GET.request are described as follows:

NLME-INFO_GET.request (

DstAddr,
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count
)

Table 120 specifies the parameters for NLME-INFO_GET.request.

Table 120 – NLME-INFO_GET.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of destination
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; getting all attribute values from FirstValueStorIndex if Count = 0.

9.5.13.2 NLME-INFO_GET.indication

NLME-INFO_GET.indication is used to inform the DMAP of the successful receipt of an attribute getting request packet.

The semantics of NLME-INFO_GET.indication are described as follows:

NLME-INFO_GET.indication (

SrcAddr,
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count
)

Table 121 specifies the parameters for NLME-INFO_GET.indication.

Table 121 – NLME-INFO_GET.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of source
AttributeID	Unsigned8	0 to 255	The ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from FirstValueStorIndex if Count = 0.

9.5.13.3 NLME-INFO_GET.response

NLME-INFO_GET.response is used to respond to NLME-INFO_GET.request.

The semantics of NLME-INFO_GET.response are described as follows:

NLME-INFO_GET.response (

DstAddr,
Status,
AttributeID,
AttributeMemID,
FirstValueStorIndex,
Count,
AttributeValue
)

Table 122 specifies the parameters for NLME-INFO_GET.response.

Table 122 – NLME-INFO_GET.response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
Status	Unsigned8	0 to 255	Result of the attribute read request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = UNSUPPORTED_ATTRIBUTE_MEMBER; 3 = UNMATCHED_COUNT; Others are reserved.
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to read the structured MIB attributes. The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from FirstValueStorIndex if Count = 0.
AttributeValue	Octetstring		Value of the attribute to be read

If the operation of getting attributes is successful, the Status should be SUCCESS; if the MIB does not have the needed attributes or attribute members, the Status should be UNSUPPORTED_ATTRIBUTE or UNSUPPORTED_ATTRIBUTE_MEMBER; otherwise, if the number of attributes does not equal to the required number, the Status should be UNMATCHED_COUNT.

9.5.13.4 NLME-INFO_GET.confirm

NLME-INFO_GET.confirm is used to return the result of NLME-INFO_GET.request.

The semantics of NLME-INFO_GET.confirm are described as follows:

```
NLME-INFO_GET.confirm (
                        SrcAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                    )
```

Table 123 specifies the parameters for NLME-INFO_GET.confirm.

Table 123 – NLME-INFO_GET.confirm parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
Status	Unsigned8	0 to 255	Result of the attribute read request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = UNSUPPORTED_ATTRIBUTE_MEMBER; 3 = UNMATCHED_COUNT; Others are reserved.
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to read the structured MIB attributes The value 255 means that all attributes should be read.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to read the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes; all attribute values from FirstValueStorIndex if Count = 0.
AttributeValue	Octetstring		Value of the attribute to be read

9.5.14 Network attribute setting services

9.5.14.1 NLME-INFO_SET.request

NLME-INFO_SET.request is used by GW to remotely modify the MIB attribute values of routing devices or field devices.

The semantics of NLME-INFO_SET.request are described as follows:

```
NLME-INFO_SET.request (
                        DstAddr,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count,
                        AttributeValue
                      )
```

Table 124 specifies the parameters for NLME-INFO_SET.request.

Table 124 – NLME-INFO_SET.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	The ID of attribute member, which is used to write the structured MIB attributes The value 255 means that all attributes should be written
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.5.14.2 NLME-INFO_SET.indication

NLME-INFO_SET.indication is used to inform the DMAP of the successful receipt of an attribute setting request command packet.

The semantics of NLME-INFO_SET.indication are described as follows:

```
NLME-INFO_SET.indication (
                              SrcAddr,
                              AttributeID,
                              AttributeMemID,
                              FirstValueStorIndex,
                              Count,
                              AttributeValue
                          )
```

Table 125 specifies the parameters for NLME-INFO_SET.indication.

Table 125 – NLME-INFO_SET.indication parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
AttributeID	Unsigned8	0 to 255	ID of attribute in MIB
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes. The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
Count	Unsigned8	0 to 255	Number of attribute values or attributes member values, which is used to read the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.5.14.3 NLME-INFO_SET.response

NLME-INFO_SET.response is used to respond to NLME-INFO_SET.indication.

The semantics of NLME-INFO_SET.response are described as follows:

```
NLME-INFO_SET.response (
                        DstAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count
                    )
```

Table 126 specifies the parameters for NLME-INFO_SET.response.

Table 126 – NLME-SET.response parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0 to 65 535	16-bit short address of the destination
Status	Unsigned8	0 to 255	Result of the attribute setting request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = UNSUPPORTED_ATTRIBUTE_MEMBER; 3 = UNMATCHED_COUNT; Others are reserved.
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes. The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

If the operation of setting attributes is successful, the Status should be SUCCESS; if the MIB does not have the needed attributes or attribute members, the Status should be UNSUPPORTED_ATTRIBUTE or UNSUPPORTED_ATTRIBUTE_MEMBER; otherwise, if the number of attributes does not equal to the required number, the Status should be UNMATCHED_COUNT.

9.5.14.4 NLME-INFO_SET.confirm

NLME-INFO_SET.confirm is used to return the results of NLME-INFO_SET.request.

The semantics of NLME-INFO_SET.confirm are described as follows:

```
NLME-INFO_SET.confirm (
                        SrcAddr,
                        Status,
                        AttributeID,
                        AttributeMemID,
                        FirstValueStorIndex,
                        Count
                    )
```

Table 127 specifies the parameters for NLME-INFO_SET.confirm.

Table 127 – NLME-SET.confirm parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0 to 65 535	16-bit short address of the source
Status	Unsigned8	0 to 255	Result of the attribute setting request: 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = UNSUPPORTED_ATTRIBUTE_MEMBER; 3 = UNMATCHED_COUNT; Others are reserved.
AttributeMemID	Unsigned8	0 to 255	ID of attribute member, which is used to write the structured MIB attributes. The value 255 means that all attributes should be written.
FirstValueStorIndex	Unsigned16	0 to 65 535	The first storage index of multiple attribute values, which is used to write the structured MIB attributes
AttributeValue	Octetstring		Value of the attribute to be written

9.6 Network layer packet formats

9.6.1 Common packet format

The NL packet format is shown in Table 128.

Table 128 – Network layer common packet format

	Network layer common packet format										
	Network layer header										**Network layer payload**
Length in octet(s)	1	2	2	2	4	1	1	0/1	0/1	1	Variable length
Field name	Control	Destination address	Source address	Route-ID	Time-stamp	Priority	Sequence Number	Number of fragments	Fragment sequence number	Payload length	Network layer payload
		Routing									

The control field format is shown in Table 129.

Table 129 – Control field format

	Control field format				
Length in bit(s)	2	1	1	1	4
Subfield name	Packet type	Fragment flag	P/S flag	Authentication flag	Reserved

The NL header field has the following subfields:

a) Control subfield, which includes:
   – Packet type: this subfield has 2-bit length; the value 0 indicates a data packet, 1 indicates an aggregation packet, 2 indicates a command packet, and others are reserved;
   – Fragment flag: this subfield has 1-bit length and is used to indicate whether this packet is a fragment packet; the value 0 indicates no fragment and 1 indicates fragment;

- P/S flag: this subfield has 1-bit length, which is used to indicate whether the data in this packet is P/S type; the value 0 indicates non P/S type and 1 indicates P/S type;
- Reserved: this subfield has 4-bit length and is used for extension.

b) Routing field, which includes:
   - Destination address: final destination address of a packet (16 bits);
   - Source address: the address from which a packet originates (16 bits);
   - RouteID: unique identifier of the path (16 bits);

c) Timestamp: this subfield has 4-octet length and is labelled in seconds;

d) Priority: this subfield has 1-octet length and indicates the priority of a packet;

e) Sequence number: this subfield has 1-octet length and indicates the sequence number of the NL packet;

f) Number of fragments: this subfield has 0- or 1-octet length; it is used to indicate the number of fragments; if the packet is a fragment packet, this subfield has 1-octet length and its value is valid; otherwise, this subfield is 0-octet length.

g) Fragment sequence number: the fragment sequence number subfield is used to indicate the fragment sequence of the fragmentation packet. If the packet is a fragmentation packet, this field has 1-octet length and is valid; otherwise, this field is invalid.

h) Payload length: this field has 1-octet length and is used to indicate the length of the NL payload;

i) Network layer payload: this field has variable length and is used to fill in the NL data.

9.6.2 Data packet format

The data packet format is shown in Table 130.

Table 130 – Network layer data packet format

<table>
<tr><td rowspan="2"></td><td colspan="10">Network layer common packet format</td></tr>
<tr><td colspan="9">Network layer header</td><td>Network layer payload</td></tr>
<tr><td>Length in octet(s)</td><td>1</td><td>2</td><td>2</td><td>2</td><td>4</td><td>1</td><td>1</td><td>0/1</td><td>0/1</td><td>1</td><td>Variable length</td></tr>
<tr><td rowspan="2">Field name</td><td rowspan="2">Control field</td><td>Destina-tion address</td><td>Source address</td><td>RouteID</td><td rowspan="2">Time-stamp</td><td rowspan="2">Priority</td><td rowspan="2">Sequen-ce Number</td><td rowspan="2">Number of frag-ments</td><td rowspan="2">Frag-ment sequen-ce number</td><td rowspan="2">Payload length</td><td rowspan="2">Network layer payload</td></tr>
<tr><td colspan="3">Routing field</td></tr>
</table>

The value of packet type in control field of Network layer header = 0, which indicates a data packet.

The definitions of all the subfields are the same as those in Table 128.

9.6.3 Aggregated packet format

The aggregated packet format is described in Table 131.

Table 131 – Aggregated packet format

	Aggregated packet format								
	Network layer header	**Network layer payload**							
Length in octet(s)	14/16	1	2	1	Variable length	…	2	1	Variable length
Field name	Network layer header	Aggregated number	Source address 1	Data length1	Data1	…	Source address n	Data length n	Data n
			The first aggregated data			…	The n^{th} aggregated data		

The value of packet type in control field of Network layer header = 1, which indicates an aggregation packet.

The aggregated packet has the following fields or subfields:

a) Network layer header: the definition of the network layer header is the same as in Table 127;

b) Aggregated number: this subfield is used to record the number of packets that are aggregated;

c) First aggregated data: this subfield is used to fill in the first aggregated data, which includes:

   – Source address 1: this field has 2-octet length and indicates the 16-bit address of the first aggregated data;

   – Data length 1: this field has 1-octet length, which is used to fill in the length of the first aggregated data;

   – Data 1: this field has variable length and is used to fill in the first aggregated data.

d) n^{th} aggregated data: this subfield is used to fill in the n^{th} aggregated data, which is the same as the first aggregated data.

9.6.4 Command packet format

9.6.4.1 Common format of command packet

The NL command packets encapsulate the NL commands to accomplish some management functions. The format of the command packet is shown in Table 132.

Table 132 – Format of NL command packet

	Network layer common packet format										
	Network layer header										Network layer payload
Length in octet(s)	1	2	2	2	4	1	1	0/1	0/1	1	Variable length
Field name	Con-trol field	Destina-tion address	Source address	RouteI D	Time-stamp	Priority	Sequence Number	Number of fragment	Fragment sequence number	Payload length	Network layer payload
		Routing field									

The value of packet type in control field of Network layer header = 2, which indicates a command packet.

The data packet has the following fields or subfields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) The network layer payload field includes the following subfields:

- Command packet identifier: this subfield has 1-octet length and is used to indicate the identifier of the NL command packet (see Table 133);
- Command packet payload: this subfield has variable length and is used to fill in the payload of the NL command packet.

The NL command packets are listed in Table 133.

Table 133 – Network layer command packet

Command packet identifier	Command	User
0	Joining request	Routing device
1	Joining response	Gateway device
2	Communication status report request	Routing device
3	Leaving request	Routing device/ Gateway device
4	Leaving response	Gateway device/ Routing device
5	Cluster member report request	Routing device
6	Cluster member report response	Gateway device
7	Neighbour information report request	Routing device
8	Routeadding request	Gateway device
9	Routeadding response	Routing device
10	Routeupdate request	Gateway device
11	Route update response	Routing device
12	Route deleting request	Gateway device
13	Route deleting response	Routing device
14	Link adding request	Gateway device/ Routing device
15	Link adding response	Routing device / Field device
16	Link update request	Gateway device/ Routing device
17	Link update response	Routing device / Field device
18	Link release request	Gateway device/ Routing device
19	Link release response	Routing device / Field device
20	Superframe adding request	Gateway device/ Routing device
21	Superframe adding response	Routing device / Field device

Command packet identifier	Command	User
22	Superframe update request	Gateway device/ Routing device
23	Superframe update response	Routing device / Field device
24	Superframe release request	Gateway device/ Routing device
25	Superframe release response	Routing device / Field device
26	Device condition report request	Field device/Routing device
27	Channel status report request	Field device/Routing device
28	Failure path report request	Routing device
29	Attribute getting request	Gateway device/ Routing device
30	Attribute getting response	Routing device / Field device
31	Attribute setting request	Gateway device/ Routing device
32	Attribute setting response	Routing device / Field device
33 to 255	Reserved	Reserved

The execution results of command are shown in Table 134.

Table 134 – Execution results of commands

Command implementing result	Identifier	Description
SUCCESS	0	Command execution succeeds.
FAILURE	1	Command execution fails.

9.6.4.2 Joining request and response

The joining request packet is used by a routing device to forward the joining request. The format of the joining request packet is shown in Table 135.

Table 135 – Format of joining request packet

	Format of joining request packet				
Length in octet(s)	14/16	1	8	0/4	1
Field name	Network layer header	Command ID = 0	Physical address of the new device	Security material	Device type

The joining request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 0;

c) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that is joining the network;

d) Security material: this field has 0- or 4-octet length; if the authentication flag in the NL header is 0, the Security material is 0-octet long; otherwise, it is 4-octet long (see Clause 11 for the detailed information);

e) Device type: this field has 1-octet length and is used to indicate the type of the joining device. The value 0 indicates the gateway device, 1 indicates a routing device, 2 indicates a field device, and 3 indicates a handheld device; otherwise, this field is reserved.

The joining response packet is used to return the joining request result. The format of the joining response packet is shown in Table 136.

Table 136 – Format of joining response packet

	Format of joining response packet				
Length in octet(s)	14/16	1	1	8	2
Field name	Network layer header	Command ID = 1	Execution result	Physical address of the new device	Short address of the new device

The joining response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 1;

c) Execution result: this field has 1-octet length. The value 0 indicates that the joining process is SUCCESS; otherwise, 1 indicates that the joining process is FAILURE (see Table 134);

d) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that is joining the network;

e) Short address of the new device: this field has 2-octet length. If the value of the Execution result field is 0, this field is valid and the value is the newly allocated 16-bit short address for the joining device; otherwise, this field is invalid.

9.6.4.3 Communication status report request

The communication status report request packet is used by a routing device to forward the communication status report request. The format of the communication status report request packet is shown in Table 137.

Table 137 – Format of communication status report request packet

	Format of communication status report request packet				
Length in octet(s)	14/16	1	8	1	1
Field name	Network layer header	Command ID = 2	Physical address of the new device	Device type	Status

The communication status report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 2;

c) Physical address of the new device: this field is used to indicate the 64-bit physical address of the device that has just joined the network;

d) Device type: this field has 1-octet length and is used to indicate the type of newly joined device. The value 0 indicates the gateway device, 1 indicates a routing device, 2 indicates a field device, and 3 indicates a handheld device; otherwise, this field is reserved;

e) Status: this field has 1-octet length and is used to indicate the result of the joining process; the value 0 indicates that the joining process is successful and 1 indicates that the joining process fails.

9.6.4.4 Leaving request and response

The leaving request packet is used either by a routing device to request departure from the gateway device or by the gateway device to request a routing device to leave the network. The format of the leaving request packet is shown in Table 138.

Table 138 – Format of leaving request packet

	Format of leaving request packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 3	Leaving reason

The leaving request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 3;

c) Leaving reason: this field is 1 octet, the valid value of which is shown in Table 139.

Table 139 – Value of Leaving reason

Value	Description
0	Reserved
1	Passive leaving of a routing device or a field device
2	Active leaving of a routing device
3 to 255	Reserved

The leaving response packet is used to return the execution results. The format of the leaving response packet is shown in Table 140.

Table 140 – Format of leaving response packet

	Format of leaving response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 4	Execution result

The leaving response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 4;

c) Execution result: this field has 1-octet length and indicates the result of the departure process. If the departure process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.5 Cluster member report request and response

The cluster member report request packet is used by routing devices to report cluster member information to the gateway device. The format of the cluster member report request packet is described in Table 141.

Table 141 – Format of cluster member report request packet

	Format of cluster member report request packet			
Length in octet(s)	14/16	1	1	2
Field name	Network layer header	Command ID = 5	Cluster member modification flag	Cluster member address

The cluster member report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 5;

c) Cluster member modification flag: this field has 1-octet length and indicates the kinds of modification; the value 0 indicates the adding action, and 1 indicates the deleting action; otherwise, this field is reserved;

d) Cluster member address: this field has 2-octet length and indicates the 16-bit address of the modified cluster member.

The cluster member report response packet is used to return the execution results of the cluster member report request command. The format of the cluster member report response packet is described in Table 142.

Table 142 – Format of cluster member report response packet

	Format of cluster member report response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 6	Execution result

The cluster member report response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 6;

c) Execution result: this field has 1-octet length and indicates the result of the cluster member report process. If the report process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.6 Neighbour information report request

The neighbour information report request packet is used by routing devices to report its one-hop neighbour information to the NM. The format of the neighbour information report request packet is described in Table 143.

Table 143 – Format of neighbour information report request packet

	Format of neighbour information report request packet			
Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 7	Number of neighbours	Items of neighbour table

The neighbour information report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 7;

c) Number of neighbours: this field has 1-octet length and indicates the number of reported neighbours;

d) Items of neighbour table: this field has variable length; the structure of neighbour table item is shown in Table 15.

9.6.4.7 Route adding request and response

The route adding request packet is generated by the NM, and is sent to the destination routing device for adding a new routing record into its routing table.

The format of the route adding request packet is illustrated in Table 144.

Table 144 – Format of route adding request packet

	Format of route adding request packet		
Length in octet(s)	14/16	1	9
Field name	Network layer header	Command ID = 8	A record of routing table

The route adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 8;

c) A record of routing table: this field has 9-octet length; the detailed information is shown in Table 15.

The route adding response packet is used to return the execution results of the route adding request command.

The format of the route adding response packet is illustrated in Table 145.

Table 145 – Format of route adding response packet

	Format of route adding response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 9	Execution result

The route adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 9;

c) Execution result: this field has 1-octet length and indicates the results of the route adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.8 Route update request and response

The route update request packet is generated by the NM, and is sent to the destination routing device for modifying a routing table record in its routing table.

The format of the route update request packet is illustrated in Table 146.

Table 146 – Format of route update request packet

	Format of route update request packet		
Length in octet(s)	14/16	1	9
Field name	Network layer header	Command ID = 10	A record of routing table

The route update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 10;

c) A record of routing table: this field has 9-octet length; the detailed information is shown in Table 15.

The route update response packet is used to return the execution results of the route update request command.

The format of the route update response packet is illustrated in Table 147.

Table 147 – Format of route update response packet

	Format of route update response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 11	Execution result

The route update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 11;

c) Execution result: this field has 1-octet length and indicates the result of the route update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.9 Route deleting request and response

The route deleting request packet is generated by the NM, and is sent to the destination routing device for deleting a routing record from its routing table.

The format of the route deleting request packet is illustrated in Table 148.

Table 148 – Format of route deleting request packet

	Format of route update request packet		
Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 12	Route ID

The route update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 12;

c) Route ID: this field has 2-octet length and indicates the identifier of the deleted route.

The route deleting response packet is used to return the execution result of the Route deleting request command.

The format of the route deleting response packet is illustrated as Table 149.

Table 149 – Format of route deleting response packet

	Format of route update response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 13	Execution result

The route deleting response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 13;

c) Execution result: this field has 1-octet length and indicates the result of the route deleting response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.10 Link adding request and response

The link adding request packet is used by the NM to add one or more new link(s) to a routing device, and is also used by the routing device to add one or more new link(s) to a field device. After receiving this command packet, the routing device or field device should add a record to its link table.

The format of the link adding request packet is shown in Table 150.

Table 150 – Format of link adding request packet

	Format of link adding request packet					
Length in octet(s)	14/16	1	2	12	…	12
Field name	Network layer header	Command ID = 14	Number of links	Link table item 1	…	Link table item N

The link adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 14;

c) Number of links: this field has 2-octet length and indicates the number of added links;

d) Link table item 1 to N (N = Number of links): each of these link table items has 12-octet length, which indicates the detailed information of the added link. The structure of the link table item is shown in Table 17.

The link adding response packet is used to return the execution results of link adding. The format of the link adding response packet is shown in Table 151.

Table 151 – Format of link adding response packet

	Format of link adding response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 15	Execution result

The link adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 15;

c) Execution result: this field has 1-octet length and indicates the result of the link adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.11 Link update request and response

The link update request packet is used by the NM to update one or more existing link(s) of a routing device, and is also used by the routing device to update one or more existing link(s) to a field device. After receiving this command packet, the routing device or field device should update a record in its link table.

The format of the link update request packet is shown in Table 152.

Table 152 – Format of link update request packet

	Format of link update request packet					
Length in octet(s)	14/16	1	2	12	…	12
Field name	Network layer header	Command ID = 16	Number of links	Link table item 1	…	Link table item N

The link update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 16;

c) Number of links: this field has 2-octet length and indicates the number of updated links;

d) Link table item 1 to N (N = Number of links): each of these link table items has 12-octet length, which indicates the detailed information of the updated links. The structure of the link table item is shown in Table 17.

The link update response packet is used to return the execution results of link update. The format of the link update response packet is shown in Table 153.

Table 153 – Format of link update response packet

	Format of route update response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 17	Execution result

The link update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 17;

c) Execution result: this field has 1-octet length and indicates the result of the link update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.12 Link release request and response

The link release request packet is used by the NM to release one or more existing link(s) to a routing device, and is also used by the routing device to release one or more existing link(s) to a field device. After receiving this command packet, the routing device or field device should release a record in its link table.

The format of the link release request packet is shown in Table 154.

Table 154 – Format of link release request packet

	Format of link release request packet					
Length in octet(s)	14/16	1	1	2	…	2
Field name	Network layer header	Command ID = 18	Number of links	Link ID 1	…	Link ID N

The link release request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 18;

c) Number of links: this field has 2-octet length and indicates the number of released links;

d) Link ID 1 to N of link table (N = Number of links): each Link ID has 2-octet length, which indicates the identifiers of the released links.

The link release response packet is used to return the execution results of link release. The format of the link release response packet is shown in Table 155.

Table 155 – Format of link release response packet

	Format of link release response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 19	Execution result

The link release response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 19;

c) Execution result: this field has 1-octet length and indicates the result of the link release response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.13 Superframe adding request and response

The superframe adding request packet is used by the NM to add a new superframe to a routing device, and is also used by the routing device to add a new superframe to a field device. After receiving this command packet, the routing device or field device should add a record in its superframe table.

The format of the superframe adding request packet is shown in Table 156.

Table 156 – Format of superframe adding request packet

	Format of superframe adding request packet		
Length in octet(s)	14/16	1	12
Field name	Network layer header	Command ID = 20	Superframe table item

The superframe adding request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 20;

c) Superframe table item: this field has 12-octet length and indicates the detailed information of the added superframe (see Table 16).

The superframe adding response packet is used to return the execution results of superframe adding. The format of the superframe adding response packet is shown in Table 157.

Table 157 – Format of superframe adding response packet

	Format of superframe adding response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 21	Execution result

The superframe adding response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 21;

c) Execution result: this field has 1-octet length and indicates the result of the superframe adding response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.14 Superframe update request and response

The superframe update request packet is used by the NM to update an existing superframe to a routing device, and is also used by the routing device to update an existing superframe to a field device. After receiving this command packet, the routing device or field device should update a record in its superframe table.

The format of the superframe update request packet is shown in Table 158.

Table 158 – Format of superframe update request packet

	Format of superframe update request packet		
Length in octet(s)	14/16	1	12
Field name	Network layer header	Command ID = 22	Superframe table item

The superframe update request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 22;

c) Superframe table item: this field has 12-octet length and indicates the detailed information of the updated superframe (see Table 16).

The superframe update response packet is used to return the execution results of superframe update. The format of the superframe update response packet is shown in Table 159.

Table 159 – Format of superframe update response packet

	Format of superframe update response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 23	Execution result

The superframe update response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 23;

c) Execution result: this field has 1-octet length and indicates the result of the superframe update response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.15 Superframe release request and response

The superframe release request command frame is used by the NM to release an existing superframe to a routing device, and is also used by the routing device to release an existing superframe to a field device. After receiving this command packet, the routing device or field device should release a record in its superframe table.

The format of the superframe release request packet is shown in Table 160.

Table 160 – Format of superframe release request packet

	Format of superframe release request packet		
Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 24	Superframe ID

The superframe release request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 24;

c) Superframe ID: this field has 2-octet length and indicates the identifier of the superframe to be the released (see Table 16).

The superframe release response packet is used to return the execution results of superframe release. The format of the superframe release response packet is shown in Table 161.

Table 161 – Format of superframe release response packet

	Format of superframe release response packet		
Length in octet(s)	14/16	1	1
Field name	Network layer header	Command ID = 25	Execution result

The superframe release response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 25;

c) Execution result: this field has 1-octet length and indicates the result of the superframe release response process. If the response process is successful, the value of this field is 0; otherwise, the value of this field is 1 (see Table 134).

9.6.4.16 Device condition report request

The device condition report request packet is used to report the conditions of the devices remotely. The format of the device condition report request packet is described in Table 162.

Table 162 – Format of device condition report request packet

	Format of device condition report request packet			
Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 26	Number of devices	Items of device condition information

The device condition report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 26;

c) Number of devices: this field has 1-octet length and indicates the number of reported devices.

d) Items of device condition information: the detailed format of the device condition information field is described in Table 163. This field has N×11 octet length, where N = Number of devices.

Table 163 – Format of device condition information field

	Format of device condition information field						
Length in octet(s)	2	2	2	2	1	2	4
Field name	Device short address	Number of packets transmitted after last report	Number of packets terminated in this device after last report	Number of packets with MAC MIC failure after last report	Battery level	Restart count of device after last report	Uptime

The device condition information field includes the following subfields (see Table 22):

a) Device short address: this subfield has 2-octet length and indicates the short address of the reported device;

b) Number of packets transmitted after last report: this subfield has 2-octet length and indicates the number of transmitted packets since the last report of the reported device;

c) Number of packets terminated in this device after last report: this subfield has 2-octet length and indicates the number of received packets since the last report of the reported device;

d) Number of packets with MAC MIC failure after last report: this subfield has 2-octet length and indicates the number of packets with MAC MIC failure since the last report;

e) Battery level: this subfield has 1-octet length and indicates the residual power level;

f) Restart count of device after last report: this subfield has 2-octet length and indicates the number of restarts since the last report.

g) Uptime: this subfield has 4-octet length and indicates the uptime of a device.

9.6.4.17 Channel condition report request

The channel condition report request packet is used to report the quality status of the communication channels remotely. The format of the channel condition report request packet is described in Table 164.

Table 164 – Format of channel condition report request packet

	Format of channel condition report request packet			
Length in octet(s)	14/16	1	1	Variable length
Field name	Network layer header	Command ID = 27	Number of channels	Items of channel quality information

The channel condition report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 27;

c) Number of channels: this field has 1-octet length and indicates the number of reported channels;

d) Channel quality information: this field has variable length;

e) Items of channel quality information: the detailed format of the channel quality information is shown in Table 165. This field has N×5 octet length, where N = Number of channels.

Table 165 – Format of channel quality information field

	Format of channel quality information field			
Length in octet(s)	1	1	2	1
Field name	Channel index	Link quality	Packet loss rate	Number of retries

The channel quality information field includes the following subfields:

a) Channel index: this field has 1-octet length and indicates the sequence number of channels;

b) Link quality: this field has 1-octet length and indicates the LQI value of each channels;

c) Packet loss rate: this field has 2-octet length and indicates the packet loss rate of each channel;

d) Number of retries: this field has 1-octet length and indicates the number of retransmissions of each channel.

9.6.4.18 Path failure report request

The path failure report request packet is used by routing devices to report the failure of a path. The format of the path failure report request packet is described in Table 166.

Table 166 – Format of path failure report request packet

	Format of path failure report request packet		
Length in octet(s)	14/16	1	2
Field name	Network layer header	Command ID = 28	Route ID

The path failure report request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 28;

c) Route ID: this field has 1-octet length and indicates the identifier of the reported route.

9.6.4.19 Attribute getting request

The attribute getting request packet is used to ask a specified device to report the attributes of its MIB. The format of the attribute getting request packet is described in Table 167.

Table 167 – Format of attribute getting request packet

	Format of attribute getting request packet					
Length in octet(s)	14/16	1	1	1	2	1
Field name	Network layer header	Command ID = 29	Attribute ID	Attribute member ID	First storage index of multiple attribute values	Number of attributes

The attribute getting request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 29;

c) Attribute ID: this field has 1-octet length and indicates the index of the requested attribute;

d) Attributed member ID: this field has 1-octet length and indicates the index of the requested attribute member;

e) First storage index of multiple attribute values: this field has 2-octet length and indicates the first storage index of multiple attribute values;

f) Number of attributes: this field has 1-octet length and indicates either the number of attribute values or the number of attribute member values.

9.6.4.20 Attribute getting response

The attribute getting response packet is used to respond to the attribute getting request, which should return the values of the attributes. The format of the attribute getting response packet is described in Table 168.

Table 168 – Format of attribute getting response packet

	Format of attribute getting response packet							
Length in octet(s)	14/16	1	1	1	1	2	1	Variable length
Field name	Network layer header	Command ID = 30	Status	Attribute ID	Attribute member ID	First storage index of multiple attribute values	Number of attributes	Attribute value

The attribute getting response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 30;

c) Status: this field has 1-octet length and indicates the result of the attribute getting; the value 0 indicates the getting process is successful, and 1 indicates that the requested attribute is unsupported; otherwise, this field is reserved (see Table 122);

d) Attribute ID: this field has 1-octet length and indicates the index of the requested attribute;

e) Attributed member ID: this field has 1-octet length and indicates the index of the requested attribute member;

f) First storage index of multiple attribute values: this field has 2-octet length and indicates the first storage index of multiple attribute values;

g) Number of attributes: this field has 1-octet length and indicates either the number of attribute values or the number of attribute member values.

h) Attribute values: this field has variable length and indicates the requested attribute values.

9.6.4.21 Attribute setting request

The attribute setting request packet is used to ask a device to add or modify the attribute value in its MIB. The format of the attribute setting request packet is described in Table 169.

Table 169 – Format of attribute setting request packet

	Format of attribute setting request packet						
Length in octet(s)	14/16	1	1	1	2	1	Variable length
Field name	Network layer header	Command ID = 31	Attribute ID	Attribute member ID	First storage index of multiple attribute values	Number of attributes	Attribute value

The attribute setting request packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 31;

c) Attribute ID: this field has 1-octet length and indicates the index of the requested attribute;

d) Attributed member ID: this field has 1-octet length and indicates the index of the requested attribute member;

e) First storage index of multiple attribute values: this field has 2-octet length and indicates the first storage index of multiple attribute values;

f) Number of attributes: this field has 1-octet length and indicates either the number of attribute values or the number of attribute member values.

g) Attribute values: this field has variable length and indicates the attribute values to be set.

9.6.4.22 Attribute setting response

The attribute setting response packet is used to respond to the attribute setting request. The format of the attribute setting response packet is described in Table 170.

Table 170 – Format of attribute setting response packet

	Format of attribute setting response packet						
Length in octet(s)	14/16	1	1	1	1	2	1
Field name	Network layer header	Command ID = 32	Status	Attribute ID	Attribute member ID	First storage index of multiple attribute values	Number of attributes

The attribute setting response packet has the following fields:

a) Network layer header: the definition of the network layer header is the same as in Table 128;

b) Command ID: this field has 1-octet length and the value of it is 32;

c) Status: this field has 1-octet length and indicates the result of the attribute setting; the value 0 indicates the setting process is successful, and 1 indicates that the requested attribute is unsupported; otherwise, this field is reserved (see Table 126);

d) Attribute ID: this field has 1-octet length and indicates the index of the requested attribute;

e) Attributed member ID: this field has 1-octet length and indicates the index of the requested attribute member;

f) First storage index of multiple attribute values: this field has 2-octet length and indicates the first storage index of multiple attribute values;

g) Number of attributes: this field has 1-octet length and indicates either the number of attribute values or the number of attribute member values.

10 Application layer

10.1 Overview

10.1.1 General

The WIA-PA AL is comprised of UAP and ASL. AL defines application objects that interact with the industrial processes. It also defines communication services that support communications among multiple objects for the distributed industry applications.

The local operations among UAOs are not specified in this document.

10.1.2 AL structure

The AL structure is shown in Figure 71.

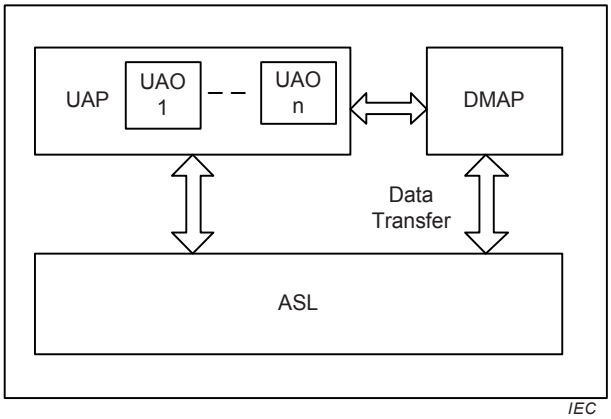

Figure 71 – AL structure

The distributed applications of industrial processes may be implemented by UAPs. A UAP is comprised of one or more UAOs. All UAOs in a device can interact with each other. The UAOs in different devices may interact with each other through the communication services that are provided by AL.

ASL provides transparent services of data transmission: between ASL and UAP.

10.1.3 Functions of UAP

AUAP has the following functions:

a) collecting process data, such as the process data of temperature, pressure, and flow;

b) processing collected process data, such as conversion, linearization, compensation, and filtration;

c) computing and generating output data based on the processed data of the device and that of other devices;

d) controlling the process;

e) generating and reporting alarm: UAP should report an alarm event, e.g. the process data is beyond the limits, the unexpected state of UAO changes, etc.; and

f) inter-operating among devices in WIA-PA network.

10.1.4 Functions of ASL

ASL provides data services and management services as follows:

a) Data transmission services: ASL provides end-to-end transparent data transmission services for UAP, and it supports C/S, P/S, and R/S data transmissions; and

b) ASL management services are not specified in this document.

10.2 UAP

10.2.1 General

UAP is a unit for processing information for specific applications and is used to implement the distributed applications within WIA-PA network.

A UAP may be constructed as an application process either according to IEC 61499-1 and IEC 61499-2 or IEC 61804-2. A UAP may also be an application process that is defined by users. The implementation of a UAP may be achieved either with function blocks or by other approaches, which is shown in Figure 72.

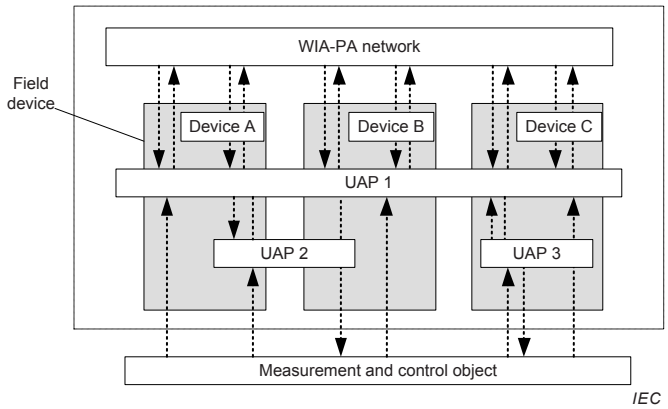

Figure 72 – User application process

A UAP may be composed of several UAOs within one device, which is shown as UAP 3 in Figure 72. A UAP may also be composed of several UAOs locating in different devices, which is shown as UAP 1 and UAP 2 in Figure 72.

10.2.2 UAO

10.2.2.1 General

UAO is defined according to different functions, which may be defined as function blocks by other approaches.

In order to support the interoperability among different devices, each UAO has its own attributes. The attributes of UAO are used to describe application profiles, which the application objects shall follow.

Each UAO shall be identified by object ID and instance ID. If some attributes of the UAO may be operated remotely, they shall be accessed further by the attribute identifier.

10.2.2.2 Functions of UAO

In WIA-PA, UAOs collect and process the data from industrial processes (such as temperature, pressure, and flow) by using different kinds of sensors, and control the process with actuators. The functions of a UAO are described in the following.

a) Range scaling

Range scaling is a linear transformation, which converts the process data from one unit to another unit in order to simplify further measurement and computation. For example, 4 mA ~ 20 mA signals may be converted into 0 Kpa to 10 Kpa signals. Range scaling may also convert the data with units into data without units, such as percentage.

b) Linearization

Sensor characteristics mostly are nonlinear and irregular. Therefore, linearization is necessary for dealing with the sensed data.

c) Compensation

Measured values of sensors may have linear drift due to the environment influence. Therefore, compensation is necessary.

d) Filtering

In industrial process, sensors may be interfered and result in a relatively abrupt jump in the measured values. In order to avoid this situation, filtering is used to eliminate the influence of the jump signals.

e) Storage

Storage involves copying the data to a temporary storage area so that it can be resent if the original data is lost.

f) Engineering unit conversion

Engineering unit conversion involves converting the units of input signals to other units that may be identified by I/O subsystems.

g) Measuring timestamp

The current time is added to the measured data as a timestamp.

UAO is not specified in this document. See Annex C for an example of UAO.

10.2.3 Method definition

10.2.3.1 General

Table 171 specifies the methods that are used to handle parameters of UAOs.

Table 171 – UAO method definition

Method	Method identifier	Scope		Description
		Object	Instance	
READ	request：0x00	Mandatory	Mandatory	Reading the attributes of UAOs
	response：0x80	Mandatory	Mandatory	Response of reading the attributes of UAOs
WRITE	request：0x01	Unsupport	Optional	Writing the attributes of UAOs
	response：0x81	Unsupport	Optional	Response of writing the attributes of UAOs
PUBLISH	request：0x02	Unsupport	Optional	Publishing attributes of UAOs
REPORT	request：0x03	Unsupport	Optional	Reporting alarm and event data
REPORT ACK	request：0x04	Unsupport	Optional	Alarm acknowledgement

10.2.3.2 READ format

The request format of the READ method is shown in Table 172.

Table 172 – Request format of READ method

	Request format of the READ method		
Length in octet(s)	1	1	1
Field name	Object identifier	Object instance identifier	Object attribute identifier

The request format of the READ method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute; when the object attribute identifier is 0, all attributes of the instance should be read.

The response format of the READ method is shown in Table 173.

Table 173 – Response format of READ method

	Response format of the READ method				
Length in octet(s)	1	1	1	1	Variable length
Field name	Object identifier	Object instance identifier	Object attribute identifier	Data length	Data

The response format of the READ method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute; if value the object attribute identifier is 0, all attributes of the instance should be read;

d) Data length: this field has 1-octet length and indicates the data length of the object attribute; the value 0 of Data length indicates the read operation fails.

e) Data: this field has variable length, which indicates the data of the object attribute. If the value of Data length is 0, this field shall be ignored.

10.2.3.3 WRITE format

The request format of the WRITE method is shown in Table 174.

Table 174 – Request format of WRITE method

	Request format of the WRITE method				
Length in octet(s)	1	1	1	1	Variable length
Field name	Object identifier	Object instance identifier	Object attribute identifier	Data length	Data

The request format of the WRITE method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute; when the value object attribute identifier is 0, all attributes of the instance should be written.

d) Data length: this field has 1-octet length and indicates the data length of the object attribute;

e) Data: this field has variable length, which indicates the data of the object attribute.

The response format of the WRITE method is shown in Table 175.

Table 175 – Response format of WRITE method

	Response format of the WRITE method			
Length in octet(s)	1	1	1	1
Field name	Object identifier	Object instance identifier	Object attribute identifier	Status

The response format of the WRITE method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute;

d) Status: this field has 1-octet length and indicates the result of the WRITE operation; the value 0 indicates the process is successful and 1 indicates that the process fails.

10.2.3.4 PUBLISH format

The format of the PUBLISH method is shown in Table 176.

Table 176 – Format of PUBLISH method

	Format of the PUBLISH method				
Length in octet(s)	1	1	1	1	Variable length
Field name	Object identifier	Object instance identifier	Object attribute identifier	Data length	Data

The format of the PUBLISH method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute; when the object attribute identifier is 0, all attributes of the instance should be published.

d) Data length: this field has 1-octet length and indicates the data length of the published object attribute;

e) Data: this field has variable length and indicates the data of the published object attribute.

10.2.3.5 REPORT format

The format of the REPORT method is shown in Table 177.

Table 177 – Format of REPORT method

	Format of the REPORT method				
Length in octet(s)	1	1	1	1	Variable length
Field name	Object identifier	Object instance identifier	Object attribute identifier	Data length	Data

The format of the REPORT method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;

b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;

c) Object attribute identifier: this field has 1-octet length and indicates the identifier of an object attribute; when the object attribute identifier is 0, all attributes of the instance should be reported.

d) Data length: this field has 1-octet length and indicates the data length of the reported object attribute;

e) Data: this field has variable length and indicates the data of the reported object attribute.

10.2.3.6 REPORT ACK format

The format of the REPORT ACK method is shown in Table 178.

Table 178 – Format of REPORT ACK method

	Format of REPORT ACK method			
Length in octet(s)	1	1	1	1
Field name	Object identifier	Object instance identifier	Object attribute identifier	Status

The format of the REPORT ACK method includes the following fields:

a) Object identifier: this field has 1-octet length and indicates the identifier of a UAO;
b) Object instance identifier: this field has 1-octet length and indicates the identifier of an object instance;
c) Object attribute identifier: this field has 1-octet length and indicates the identifier of the reported object attribute;
d) Status: this field has 1-octet length and indicates the result of the REPORT process; the value 0 indicates to clear the alarm event and 1 indicates to keep the alarm event.

10.3 Application sub-layer

10.3.1 General

ASL provides data communication services among UAOs in the WIA-PA network.

10.3.2 Application sub-layer data entity

10.3.2.1 General

ASL defines the ASLDE, which provides transparent interfaces for sending and receiving data between ASL and NL. ASLDE supports the transport of APDU between UAOs.

The ASL supports three communication modes: C/S mode, P/S mode, and R/S mode:

a) C/S mode supports unicast transmission of dynamic and aperiodic data;
b) P/S mode supports unicast and multicast transmission of the periodic data;
c) R/S mode supports unicast and multicast transmission of aperiodic data, such as alarms, alerts or events.

These communication modes are supported by corresponding types of VCRs.

The AL data services should be supported by three primitives: ASLDE-DATA.request, ASLDE-DATA.confirm, and ASLDE-DATA.indication.

10.3.2.2 ASLDE-DATA.request

The upper layer sends data to the ASL by using the ASLDE-DATA.request primitive.

The semantics of this primitive are as follows:

```
ASLDE-DATA.request (
                        VcrID,
                        Priority,
                        MethodIdentifier
                        AsduLength,
                        Asdu
                       )
```

Table 179 specifies the parameters of the ASLDE-DATA.request primitive.

Table 179 – ASLDE-DATA.request parameters

Name	Data type	Valid range	Description
VcrID	Unsigned16	0 to 65 535	VCR used for the transmission
Priority	Unsigned8	0 to 255	Priority of UAO data
MethodIdentifier	Unsigned8	0 to 255	The identifier of method (see Table 171)
AsduLength	Unsigned16	0 to 65 535	Number of octets consisting the ASDU to be transferred
Asdu	Octetstring		Set of octets comprising the ASDU to be transferred

When ASL receives a data request from the upper layer, it should encapsulate the ASDU according to the ASL packet format and send it to NL. After transporting the data, the ASL should issue ASLDE-DATA.confirm to the upper layer with the Status parameter set to the status value returned by the NL.

10.3.2.3 ASLDE-DATA.confirm

The ASLDE-DATA.confirm primitive is invoked to indicate the transmission result. When the data is transferred successfully, the Status parameter in this primitive should be set to SUCCESS. Otherwise, the Status parameter should be set to corresponding error code to indicate the reason for the failure.

The semantics of this primitive are as follows:

```
ASLDE-DATA.confirm (
                    VcrID,
                    Status
                    )
```

Table 180 specifies the parameters for the ASLDE-DATA.confirm primitive.

Table 180 – ASLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
VcrID	Unsigned16	0 to 65 535	VCR unique identifier used for this transmission
Status	Unsigned8	0 to 255	The status of the corresponding request: 0 = SUCCESS; 1 = INVALID_ADDRESS; 2 = FAIL; Others are reserved.

10.3.2.4 ASLDE-DATA.indication

After receiving packets from the NL, the ASL transfers the user data to the upper layer by using the ASLDE-DATA.indication primitive; the upper layer then performs the corresponding operation and processes the user data according to different service types.

The semantics of this primitive are as follows:

```
ASLDE-DATA.indication(
                      MethodIdentifier,
                      Priority,
                      AsduLength,
                      Asdu
                       )
```

Table 181 specifies the parameters for the ASLDE-DATA.indication primitive.

Table 181 – ASLDE-DATA.indication parameters

Name	Data type	Valid range	Description
MethodIdentifier	Unsigned8	0 to 255	The identifier of method (see Table 171)
Priority	Unsigned8	0 to 255	Priority of packet
AsduLength	Unsigned16	0 to 65 535	Number of octets comprising the ASDU received
Asdu	Octetstring		Set of octets comprising the ASDU received

10.3.2.5 ASLDE-AGG.request

The ASLDE-AGG.request primitive is invoked by the DMAP to request the ASL to send the aggregated packet.

The semantics of this primitive are as follows:

```
ASLDE-AGG.request(
                    VcrID,
                    SrcAddr,
                    GatewayAddr,
                    Priority,
                    AggPacketFlag,
                    AggRouteID,
                    AggPayload
                    )
```

Table 182 specifies the parameters for the ASLDE-AGG.request primitive.

Table 182 – ASLDE-AGG.request parameters

Name	Data type	Valid range	Description
VcrID	Unsigned16	0 to 65 535	VCR unique identifier used for this transmission
SrcAddr	Unsigned16	0 to 65 535	Network address of source
GatewayAddr	Unsigned16	0 to 65 535	Network address of gateway device
Priority	Unsigned8	0 to 255	Priority of process data
AggPacketFlag	Unsigned8	0 to 255	Indicating whether this packet is a data packet or aggregation packet: 0 = Data packet; 1 = Aggregation packet.
AggRouteID	Unsigned16	0 to 65 535	Route identifier for aggregation data
AggPayload	Octetstring		DMAP aggregated packet

10.3.2.6 ASLDE-DAG.indication

The ASLDE-DAG.indication primitive is invoked by the ASL to instruct the DMAP to disaggregate the packet.

The semantics of this primitive are as follows:

```
ASLDE-DAG.indication    (
                          AggPayload
                          )
```

Table 183 specifies the parameters for the ASLDE-DAG.indication primitive.

Table 183 – ASLDE-DAG.indication parameters

Name	Data type	Valid range	Description
AggPayload	Octetstring		Aggregated packet

10.3.2.7 Communication modes

10.3.2.7.1 General

The ASL provides three data communication modes for the data transmission services: C/S mode, P/S mode, and R/S mode.

10.3.2.7.2 Client/Server mode

The ASL provides the C/S mode to read or write UAOs and their attributes so as to support information getting or setting operations. The C/S mode is supported by the C/S VCRs.

The Client user application should send a read or write request to the Server user application. The Server UAO should perform corresponding operation on its attributes according to different service types and then return the results as a response to the requesting Client UAO.

The Client ASL should maintain a sequence number for each sending request, which is used for matching the response and acknowledge from the server.

The C/S communication process is shown in Figure 73.

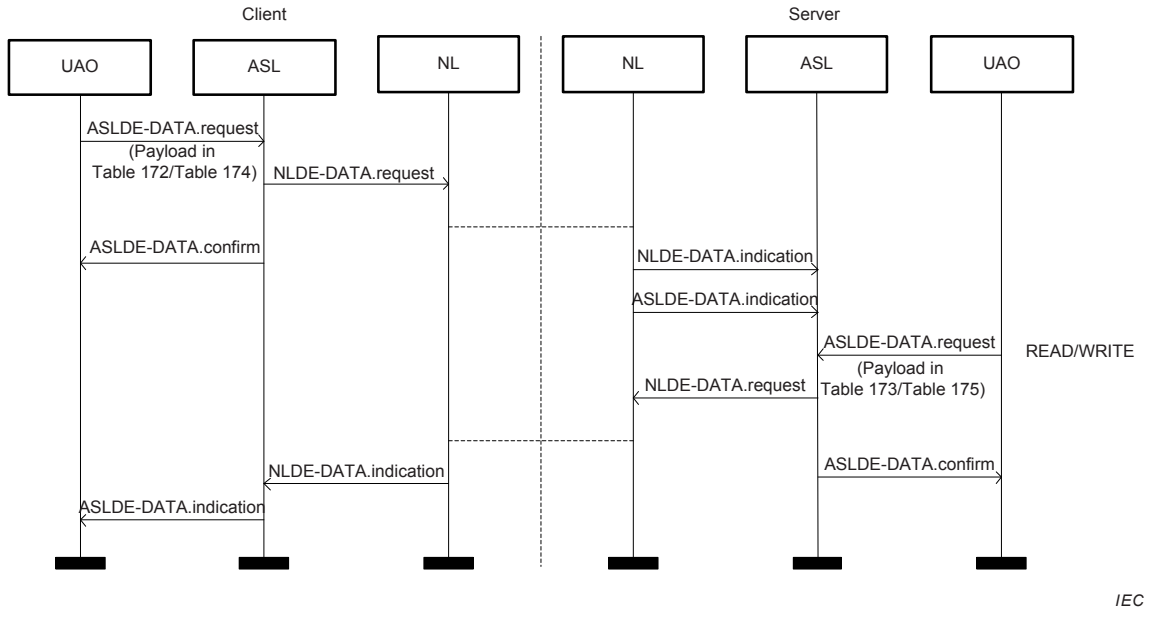

Figure 73 – C/S communication process

10.3.2.7.3 Publisher/Subscriber mode

The ASL provides the P/S mode to support the periodic data transmission of UAOs. The P/S mode is supported by P/S VCRs.

In Publisher, the corresponding VCRs have attributes to define the sending time for periodic data (DataUpdateRate), and the NM and the cluster head should allocate communication

resources to the device according to this attribute. In Subscriber, the corresponding VCRs have attributes including the address information of the Publisher data and the Subscriber UAOs, respectively. The published data should be transferred to the corresponding UAOs.

The P/S communication process is shown in Figure 74 and Figure 75. Figure 74 shows the P/S communication process when the devices disable the aggregation function. Figure 75 shows the P/S communication process when the devices enable the aggregation function.

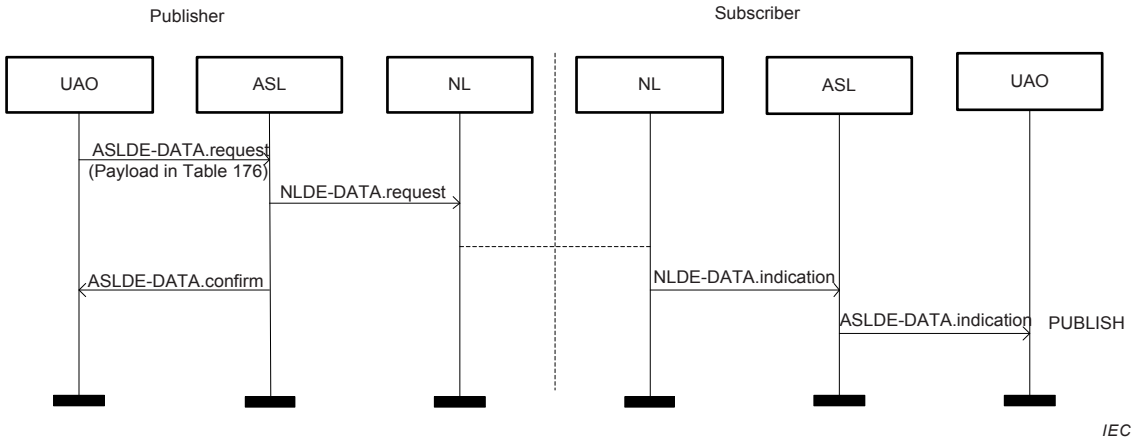

Figure 74 – P/S communication process (disable aggregation function)

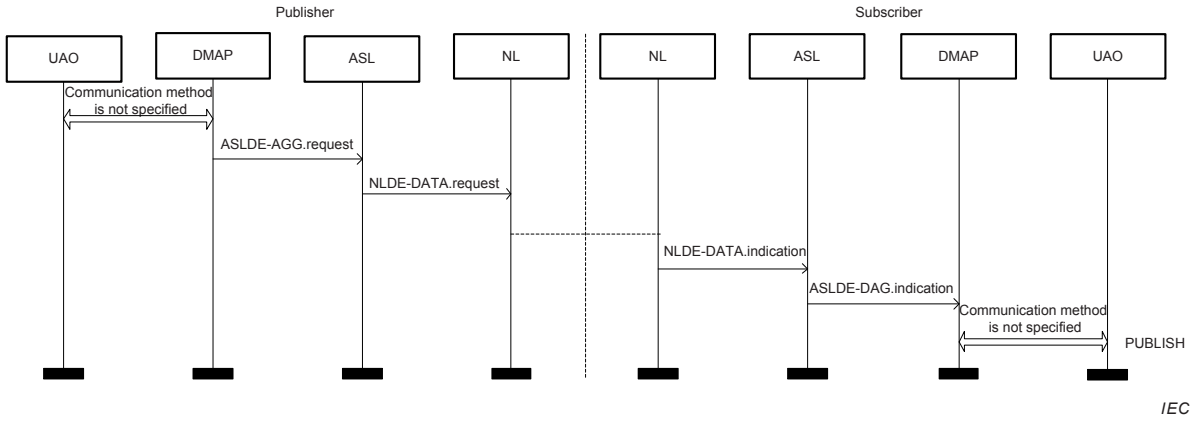

Figure 75 – P/S communication process (enable aggregation function)

10.3.2.7.4 Report source/Sink communication mode

The ASL provides the R/S mode to support the aperiodic alarm or event reports and to report the acknowledgement of the UAOs. The R/S mode is supported by the R/S VCRs.

In the R/S mode, the UAO should send the alarm or event produced by it through an allocated R/S VCR. In Report/Sink, the alarm or event received should be transferred to the corresponding UAOs.

The R/S communication process is shown in Figure 76.

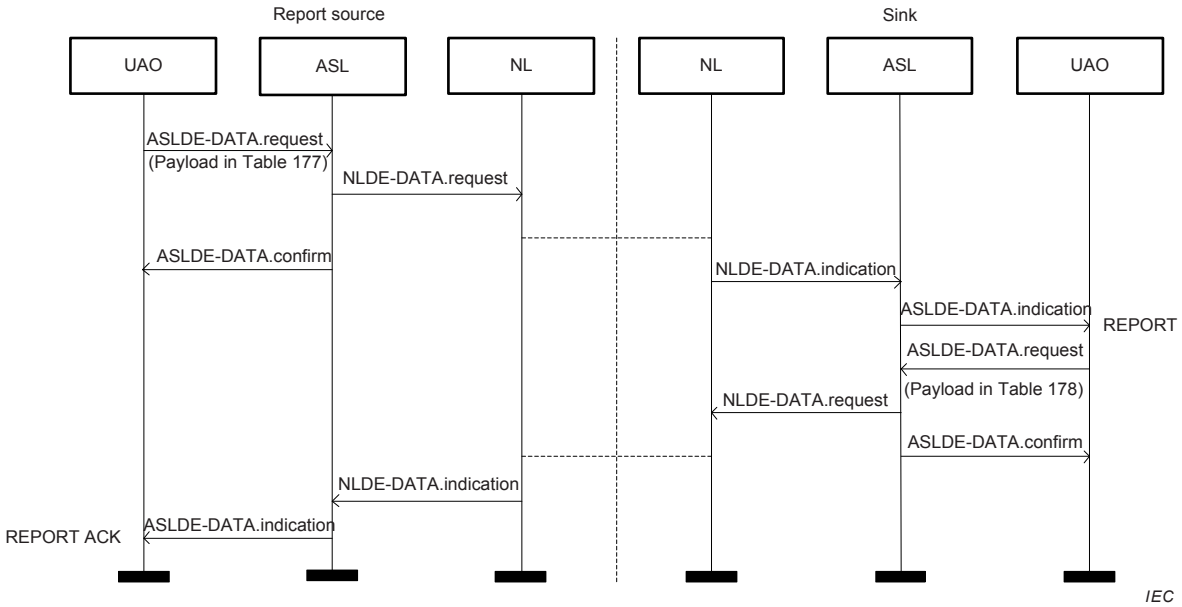

Figure 76 – R/S communication process

10.4 Application sub-layer packet formats

10.4.1 General

This part specifies the application sub-layer packet formats. Each application sub-layer packet includes the following two parts:

a) Application sub-layer packet header (ASL Header), which includes the packet control field, sequence number, method identifier, and payload length;

b) Application sub-layer packet payload (ASL Payload), whose length is variable.

10.4.2 ASL general packet format

10.4.2.1 General

The format of the ASL general packet is shown in Table 184.

Table 184 – Application sub-layer general packet format

	Format of ASL general packet				
	ASL header				**ASL payload**
Length in octet(s)	1	1	1	2	Variable length
Field name	Packet control	Sequence Number	Method identifier	Payload length	Payload

The ASL general packet includes the following fields or subfields:

a) ASL packet head, which includes the following subfields:
   - Packet control: this subfield has 1-octet length and is described in Table 185;
   - Sequence number: this subfield has 1-octet length and indicates the sequence number of the ASL packet;
   - Method identifier: this subfield has 1-octet length and indicates the identifier of the method (see 10.2.3);

– Payload length: this subfield has 2-octet length and indicates the length of the ASL payload;

b) ASL payload: this subfield has variable length and is used to fill in the ASL data.

10.4.2.2 ASL packet control field

10.4.2.2.1 General

The length of the packet control field is 8-bit. The format of the packet control field is shown in Table 185.

Table 185 – Packet control field format

	Packet control field format			
Length in bit(s)	2	1	1	4
Subfield name	Packet type	Security	ACK	Reserved

The packet control field includes the packet type, security, ACK, and reserved subfields, which should be described in the following sections.

10.4.2.2.2 Packet type subfield

The length of the packet type subfield is 2-bit. Its value is shown in Table 186.

Table 186 – Packet type subfield value

Packet type value b0b1	Packet type name
00	Data
01	Reserved
10	Acknowledgement
11	Aggregated packet

10.4.2.2.3 Security subfield

The length of the security subfield is 1 bit. The security subfield is used to control whether the AL uses the security function when transmitting the data. If the security subfield (namely b2) is set to 0, it indicates that the AL does not use security operation; if b2 is 1, it indicates that the AL uses the security operation to encrypt the data payload part of the application packet.

10.4.2.2.4 Acknowledgement subfield

The length of the acknowledgement subfield is 1 bit. It is used to specify whether application packet acknowledgement is needed. If the acknowledgement subfield (namely b3) is set to 0, it indicates that the acknowledgement is not needed; if the value of this bit is set to 1, it indicates that the acknowledgement is needed.

10.4.2.3 Sequence number field

The length of the sequence number field is 1 octet. The ASL should maintain the packet sequence number. If it transmits an AL packet, the sequence number should be increased by one. For the response data packet transmitted in the C/S communication mode, its sequence number is consistent with the corresponding request packet. The sequence number is used to match the request/response and acknowledgement packet. When the sequence number reaches its maximum value, it should be reset to 0.

10.4.2.4 Method identifier field

The length of the method identifier field is 1 octet (see Table 171).

10.4.2.5 Payload length field

The length of the payload length field is 1 octet. This field is used to indicate the length of the ASL payload in octets.

10.4.2.6 ASL payload

The length of the ASL payload is variable.

10.4.3 Packet formats

10.4.3.1 General

Two types of packet formats are defined in the ASL: ASL data packet and ASL acknowledgement packet.

10.4.3.2 ASL data packet

10.4.3.2.1 General

The format of the ASL data packet is shown in Table 187.

Table 187 – ASL data packet format

	ASL data packet format				
	ASL header				**ASL payload**
Length in octet(s)	1	1	1	2	Variable length
Field name	Packet control (Packet type = 0b00)	Sequence Number	Method identifier	Payload length	Data payload

The ASL header of the data packet includes the packet control, sequence number, method identifier, and payload length subfields. Detailed descriptions of these subfields are listed in 10.4.3.2.2 to 10.4.3.2.3.

10.4.3.2.2 ASL header field of data packet

The ASL header of the data packet includes the packet control field, sequence number field, method identifier, and payload length. The packet type subfield of the packet control field should be set to 0b00 in order to indicate that it is an ASL data packet. Other fields should be set according to the ASLDE-DATA.request of the ASL.

Once the AL transmits an application layer packet, its sequence number should increase by one. The sequence number of the response data packet that transmitted in the C/S communication mode is consistent with the corresponding request data packet.

10.4.3.2.3 ASL payload field

The ASL payload field includes the request data of the upper layer (see 10.3.2).

10.4.3.3 ASL acknowledgement packet

The acknowledgement is used to confirm whether or not the destination device has received the packet. If the acknowledgement is not returned after a certain period (configurable by EtoEACKTimeOut, see 6.9.1.2.1), the source device should retry the previous packet.

For the data packet sent by P/S mode, broadcast packet and acknowledgement packet, the ASL does not need to be acknowledged, and the acknowledgement subfield of the packet control field is set to 0.

The ASL indicates that the current packet needs to be acknowledged by setting the acknowledgement subfield of the packet control field to 1.

After receiving the AL packet that needs to be acknowledged, the local ASL should confirm it by returning an acknowledgement packet. The sequence number of the acknowledgement packet should be the same as that in the corresponding request packet.

The format of the acknowledgement packet is shown in Table 188.

Table 188 – ASL acknowledgement packet format

	ASL acknowledgement packet format			
	ASL header			
Length in octet(s)	1	1	1	2
Field name	Packet control (Packet type = 0b10)	Sequence Number	Method identifier	Payload length

The acknowledgement packet has only the ASL header, which includes the following subfields.

a) Packet control: this subfield has 1-octet length and is described in Table 185. The packet type subfield of the packet control field should be set to 0b10 in order to indicate that this packet is an acknowledgement packet. If using security, the security subfield should be set to 1; otherwise, the security subfield should be set to 0.

b) Sequence number: this subfield has 1-octet length and indicates the sequence number of the ASL acknowledgement packet; the sequence number subfield should be set to the same sequence number as that of in the corresponding received request packet.

c) Method identifier: this subfield has 1-octet length and indicates the identifier of the method (see 10.2.3); the method identifier of the acknowledgement packet is 0b04.

d) Payload length: this subfield has 2-octet length and indicates the length of the ASL payload; the payload length of the acknowledgement packet is 0.

11 Security

11.1 General

The security requirements, security principles and security objectives of the WIA-PA network should be analyzed before establishing the security strategies and security measures for the WIA-PA network. See Clauses A.1 to A.3 for the recommendation of the WIA-PA security risk analysis, security principles, and security objectives.

The WIA-PA MAC layer applies the security services defined in the IEEE STD 802.15.4-2011 MAC layer. The WIA-PA upper layers provide the following security services:

a) Data integrity. The data integrity check service is provided in the AL and the DLSL. A source device generates a check code with a check algorithm; the destination device verifies the check code and determines whether the received packet has been compromised.

b) Data confidentiality. The data confidentiality service is provided in the AL and the DLSL. A source device encrypts packets by using a symmetric key algorithm and sends the encrypted packets out. The destination device decrypts the encrypted packets using the same symmetric key algorithm.

c) Device authentication. The SM uses the security material to authenticate a device. The security material is generated by the Join Key and the long address. It should be irreversible.

The definition of the check algorithm and the symmetric key algorithm are not included in this document.

11.2 Security management framework

Considering the real-time requirement and the limited resources of devices, the WIA-PA network needs different security strategies and measures to be achieved in different layers. The recommendation of the graded and layered measures of the system and the graded measures of the secure packets are defined in Clause A.4. The security management framework of the WIA-PA network is shown in Figure 77.

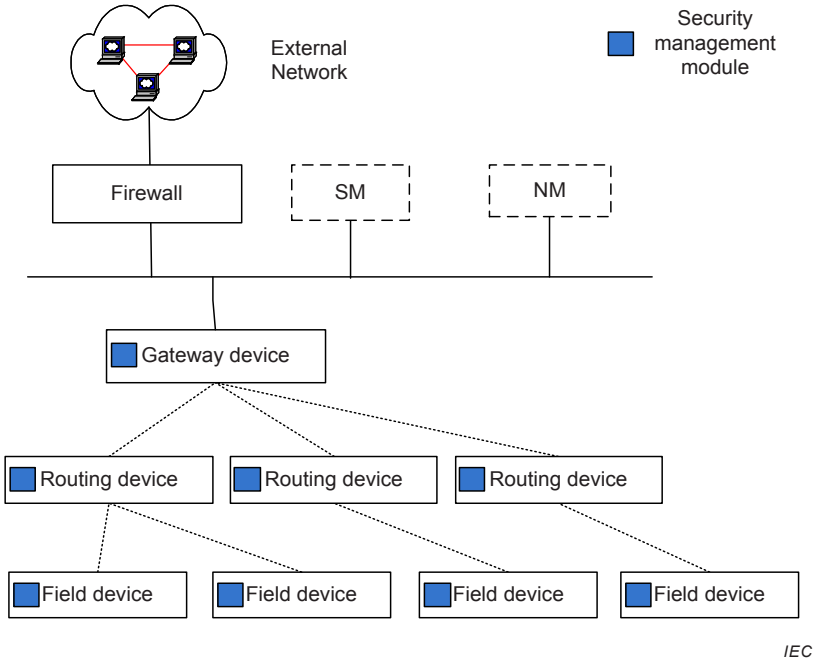

Figure 77 – Security management framework of WIA-PA network

As shown in Figure 77, the security management framework is comprised of an SM and security management modules in the gateway device, routing devices and field devices.

The security management framework interacts with external networks through the firewall. It protects the border of the whole WIA-PA network, and guarantees the normal operations of the WIA-PA network. The firewall is not specified in this document.

The SM should configure the security measures of the WIA-PA network, manage keys, and authenticate devices. It performs the following functions:

a) configuring the system security strategies and the security measures of the WIA-PA network according to specific applications;

b) configuring and monitoring the gateway device and routing devices according to specific applications and system security strategies;

c) authorizing the routing devices and field devices that are attempting to join in the WIA-PA network;

d) key management, including key generation, key distribution, key recovery, and key update.

The security management module of gateway device provides the following functions:

a) managing self security measures;

b) providing the functions of security management module in the routing devices.

The security management module in the routing device provides the following functions:

a) managing the keys used by the cluster and forwarding the keys allocated by the SM;

b) implementing the security measures of data encryption, decryption and data integrity check;

c) sending authentication request.

The security management module in the field device performs the following functions:

a) managing security measures of DLSL and ASL;

b) implementing the security measures of data encryption, decryption and data integrity check;

c) managing the keys.

The routing devices and field devices also provide alert reporting function. When the Attacked count reach to MaxAttackedCnt or the KeyAttackCnt reach MaxKeyAttackedCnt, the routing devices/field devices should report Alert to gateway device by using NLME_INFO_SET.

Redundant devices connect with primary devices by wires. The redundant device has the same KJ as the primary device. When security authentication is needed, the primary device uses the long address of the redundant device and the KJ to authenticate the latter.

11.3 Secure communication protocol stack

11.3.1 General

The secure communication protocol stack includes the security measures for each layer. Keys are stored in the MIB (see Table 23). Coordinating with SM, the security management module of the DMAP also provides management and maintenance of keys (see 11.4). DLSL and ASL deal with the secure transportation independently. The secure communication protocol stack is shown in Figure 78, where the grey modules are related to the security. The attributes related to security in MIB are defined as security management information base. The security management module uses the network attribute setting services to set security management information base.

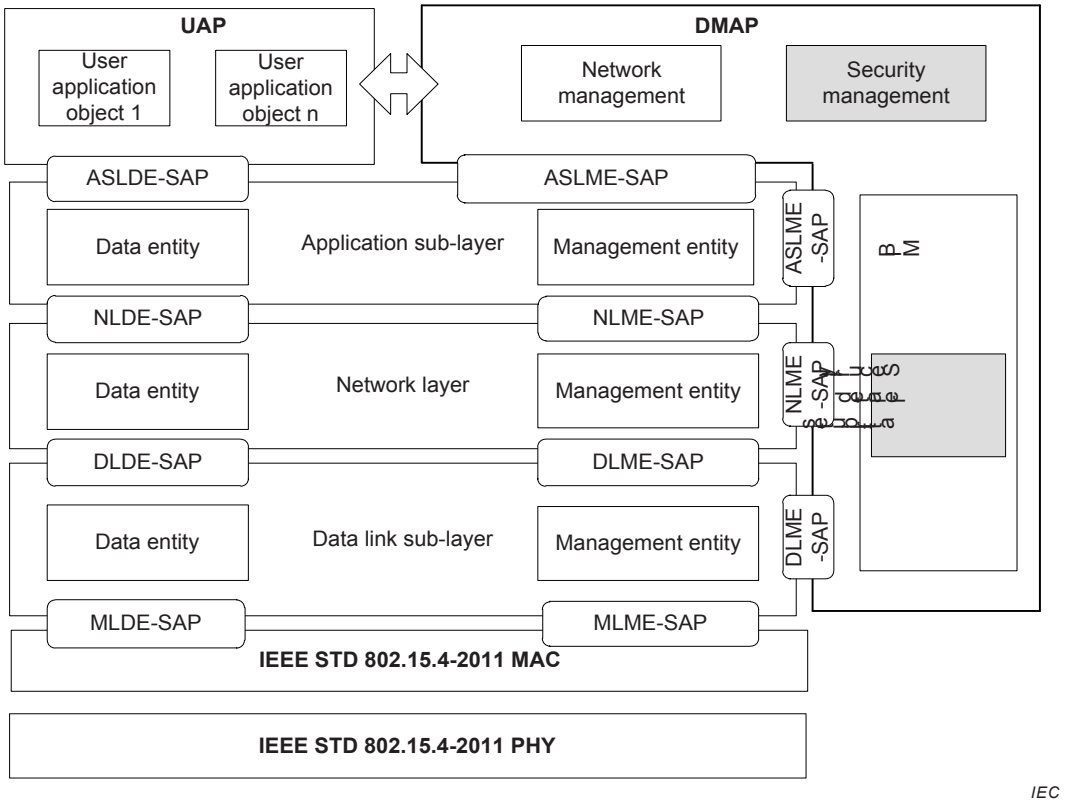

Figure 78 – Security communication protocol stack

11.3.2 Data link sub-layer security

11.3.2.1 General

The security measures are provided by DLSL in the WIA-PA network. The security of MAC layer is disabled. DLSL uses CCM MIC and the data encryption to guarantee point-to-point transportation security, in which the frame counter is replaced by ASN. The MIC is generated with the DLSL keys and the DLSL header and DLSL payload. It is used for message integrity check. The data encryption is used for data confidentiality. Encryption key getting is implemented by network attribute getting services (see 9.5.13).

The security manager can configure or read the key of devices by using NLME_INFO_SET or NLME_INFO_GET services.

11.3.2.2 Data link sub-layer frame security

The format of a security DLPDU is shown in Table 189.

Table 189 – Format of security DLPDU

	Format of security DLPDU			
Length in octet(s)	1	1	Variable length	0/4/8/16
Subfield name	DLSL frame control	DLSL security header	DLSL payload	MIC

If the security enabled subfield in the DLSL frame control field is set to 1, it should use the DLSL security header, as shown in Table 190.

Table 190 – Format of DLSL security header

	Format of DLSL security header		
Length in bit(s)	3	2	3
Subfield name	Security control field	Security material control field	Reserved

The size of the security control field is 3 bits, as described in Table 191. The security control field defines the security attributes.

Table 191 – Structure of security control field in DLSL security header

Security control field b2b1b0	Security attributes
000	None
001	MIC-32
010	MIC-64
011	MIC-128
100	ENC
101	ENC-MIC-32
110	ENC-MIC-64
111	ENC-MIC-128

If encryption and MIC check are both provided, the sender calculates MIC firstly and then encrypts the frames. Furthermore, the receiver should decrypt the frames and then check the MIC.

The security material control field defines packet type of the device security material service in Table 192.

Table 192 – Structure of security material control field in DLSL security header

Security material control field b1b0	Security attributes
00	Data packet
01	The device security material getting request packet
10	The device security material getting response packet
11	Reserve

11.3.3 Application sub-layer security

11.3.3.1 General

The ASL security implements end-to-end data transportation security. ASL uses CCM data encryption and MIC to guarantee end-to-end transportation security, in which the frame counter is replaced by ASN. The ASL establishes and maintains the keys in the security management information base. The MIC is generated with the ASL keys and ASL payload. It is used for message integrity check.

11.3.3.2 Application sub-layer packet security

The structure of a security APDU is shown in Table 193.

Table 193 – Security APDU structure

	Security APDU structure			
Length in octet(s)	5	1	Variable length	0/4
Field name	ASL header	ASL security header	ASL payload	MIC

If the security enabled subfield in the packet control of the ASL header is set to 1, it should use the ASL security header, as shown in Table 194.

Table 194 – Structure of ASL security header field

Bits: 0 to 1	2 to 7
00 = NONE 01 = ENC 10 = MIC-32 11 = ENC-MIC-32	Reserved

If encryption and MIC check are both provided, the sender calculates MIC firstly and then encrypts the frames. Furthermore, the receiver should decrypt the frames and then check the MIC.

11.4 Key management

11.4.1 Key type

The SM and the security management modules manage the keys used in the WIA-PA network. The following cryptographic keys are defined.

a) Join Key (KJ)

It is a temporary key used during the device joining. The KJ is established in the provisioning process. It is distributed by the security manager through the handheld devices. It is used to encrypt the 64-bit long address of the new device to produce the security material. After the device successfully joins the network, KJ is also used to protect the KEK when the KEK is distributed for the first time.

b) Share Key (KS)

It is a share key with unified value. The KS is established in the provisioning process. It is distributed by the security manager through the handheld devices. It is used to encrypt beacon and broadcast frame. During the joining process, the NM will return joining result, in the last hop, the KS is used to encrypt the joining response by the online routing device or gateway device,

c) Key Encryption Key (KEK)

KEK is distributed by the SM after a device has joined the WIA-PA network. It is used to encrypt other keys. When the SM distributes the KEK for the first time, the KEK is protected by KJ. After that, the new KEK is protected by the KEK which is used currently in its update period.

d) Data Encryption Key (KED)

KED is distributed by the SM after a device has joined the WIA-PA network, including the DLSL encryption key and the AL encryption key. It is used to protect data and check integrity during data transmission. The KED is protected by the KEK in its distribution and update period.

The key lifecycle is shown in Figure 79.

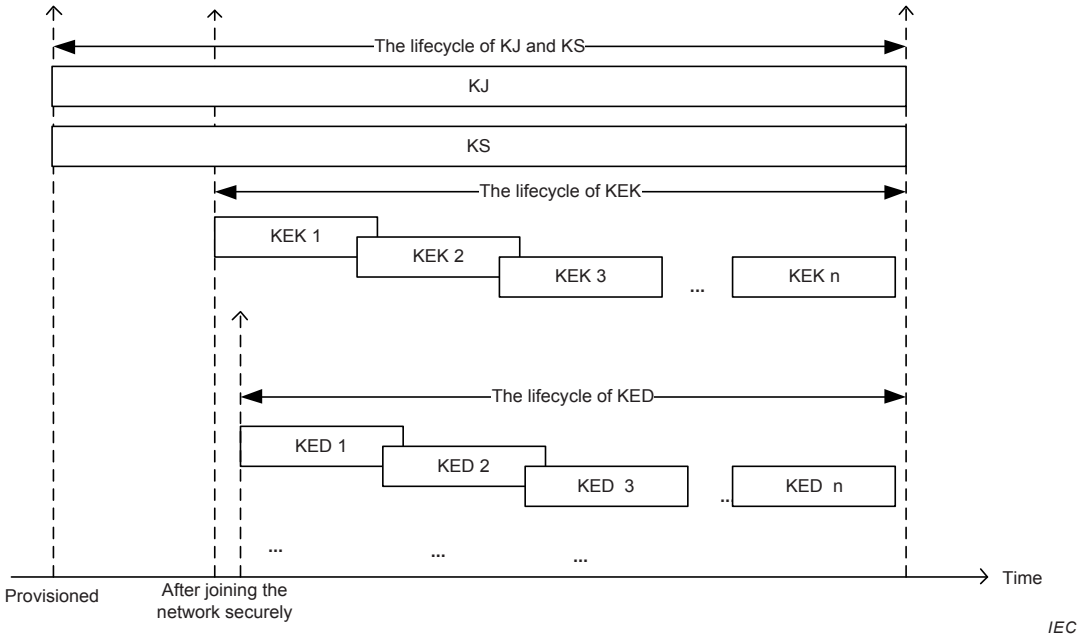

Figure 79 – Key lifecycle

The KEK and the KED need to be updated periodically by SM setting the key-related parameters in the MIB (see Table 23).

11.4.2 Key distribution

Before a device joins the network, the KJ should be distributed to the device. The KJ should be directly distributed by the handheld devices.

After the device has joined the WIA-PA network, KEK and KED should be distributed to the joined devices by SM. The key distribution is implemented by the network attribute setting services (see 9.5.14).

11.4.3 Key update

When the SM needs to update a device key, it should generate a new key. Then the new key is encrypted by the KEK and the encrypted new key is distributed to the WIA-PA device by using network attribute setting services (see 9.5.14).

Upon successful distribution of the new key, the device should decrypt the encrypted new key with its KEK and update its key.

The key update cycle is determined by users according to their requirements, and is recommended to be no more than 24 h in length.

The new key requires the following properties.

a) The new key should be different from keys used by other devices in the WIA-PA network.

b) The new key should be guaranteed to be the latest.

c) The new key should be sent to the related devices in time.

11.4.4 Key status

From its establishment to its abolition, a key should experience the following statuses:

a) BACKUP: this status is defined as when a key has been established but has not been used for normal operation;

b) USING: this status is defined as when a key is being used;

c) INVAILD: this status is defined as when a key is not available.

When the KeyActiveTime (see Table 23) of a key with BACKUP is reached, the key changes its status from BACKUP to USING. At the same time the key in USING status (if it exists) changes to INVAILD status.

11.5 Secure joining process

11.5.1 Secure joining process of a new WIA-PA device

When a new field device or a new routing device joins the WIA-PA network, SM should authenticate the new field device or the new routing device.

A new device should have the KJ and 64-bit long address before joining the WIA-PA network. KJ should be written by handheld devices

The secure joining process of a new WIA-PA device is shown in Figure 80.

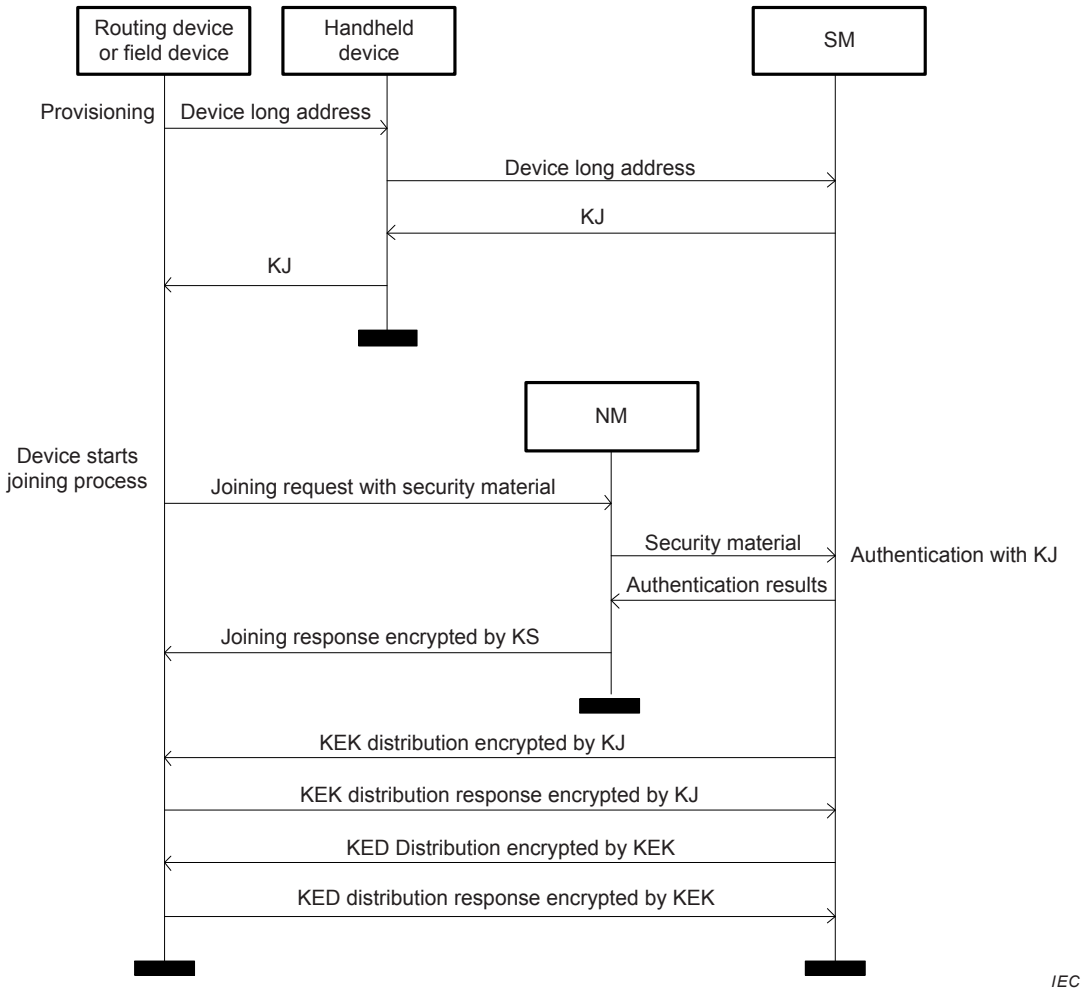

Figure 80 – Secure joining process of WIA-PA device

A new device that wants to join the WIA-PA network should proceed with the following steps.

a) Before the new device joins the WIA-PA network, the handheld device reads the 64-bit long address of the new device and transfers it to the SM. The SM generates the KJ for the new device and the handheld device transfers the KJ to the new device.

b) The new device with the KJ keeps scanning the available channels to get beacons from the online routing devices or the gateway device.

c) The new device chooses an online routing device or the gateway device as the cluster head, and synchronizes with the network according to the received beacon.

d) The new device generates the security material with the device long address and KJ, and sends it to the cluster head.The cluster head sends a joining request with security material to the NM for the new device.

e) After the NM receives the joining request, it should forward the security material to the SM. The SM verifies the security material. If the authentication fails, the NM will return failure response. If authentication succeeds, the NM will return the success response.

f) After the device completes the join process, the SM should generate the KEK for the new device and distribute the KEK encrypted by the KJ. Then the SM should generate the KED for the new device and distribute the KED encrypted by the KEK.

11.5.2 Device security material getting services

11.5.2.1 DLME-SEC.request

The routing devices use DLME-SEC.request to request the security material.

The semantics of DLME-SEC.request are described as follows:

```
DLME-SEC.request     (
                     DstPhyAddr
                     )
```

Table 195 specifies the parameters forDLME-SEC.request.

Table 195 – DLME-SEC.request parameters

Name	Data type	Valid range	Description
DstPhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device

11.5.2.2 DLME-SEC.indication

On receipt of device security material getting request packet, the DLSL of the new device will invoke DLME-SEC.indication and inform the DMAP.

The semantics of DLME-SEC. indication are described as follows:

```
DLME-SEC.indication (
                          DstPhyAddr
                           )
```

Table 196 specifies the parameters for DLME-SEC.request.

Table 196 – DLME-SEC.indication parameters

Name	Data type	Valid range	Description
DstPhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device

11.5.2.3 DLME-SEC.response

The DMAP of the new device invokes DLME-SEC.response to provide its security material.

The semantics of DLME-SEC.response are described as follows:

```
DLME-SEC.response   (
                        ProxyAddr,
                        PhyAddr,
                        SecMaterial,
                        Status
                        )
```

Table 197 specifies the parameters for DLME-SEC.response.

Table 197 – DLME-SEC.response parameters

Name	Data type	Valid range	Description
ProxyAddr	Unsigned16	0 to 65 535	The address of the routing device selected by the new device (Unicast address)
PhyAddr	Unsigned64	0 to (2^{64}-1)	64-bit physical address of the new device
SecMaterial	Unsigned32	0 to (2^{32}-1)	Device security material. It is generated with 64-bit address encrypted by KJ. Using bit 0-bit 31
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved

11.5.2.4 DLME-SEC.confirm

On receipt of the device security material getting response packet, the DLSL of the new device will invoke DLME-SEC.confirm and inform the DMAP.

The semantics of DLME-SEC.confirm are described as follows:

```
DLME-SEC.confirm (
                        PhyAddr,
                        SecMaterial,
                        Status
                    )
```

Table 198 specifies the parameters for DLME-SEC. confirm.

Table 198 – DLME-SEC.confirm parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 to $(2^{64}-1)$	The address of the routing device selected by the new device (unicast address)
SecMaterial	Unsigned32	0 to $(2^{32}-1)$	Device security material. It is generated by 64-bit address encrypted by the KJ. Using 0-31 bits.
Status	Unsigned8	0 to 255	Execution result of the request 0 = SUCCESS; 1 = FAILURE; Others are reserved.

11.5.2.5 Time sequence for device security joining

In the star and mesh architecture, if the SecEnableFlag = 1 (see 6.9.1.2.1), Figure 81 and Figure 82 illustrate examples of the security joining process of field devices.

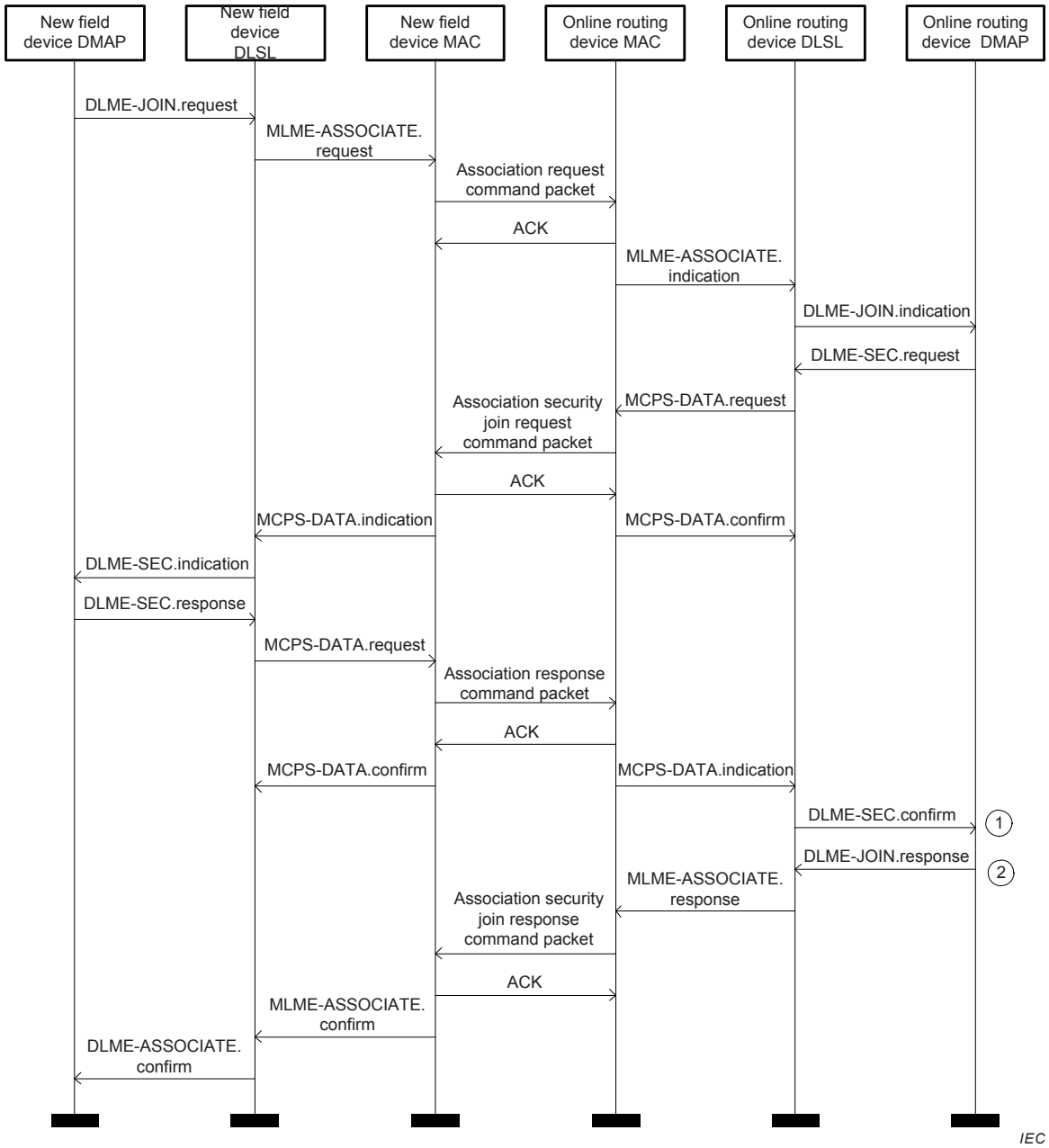

Figure 81 – Time sequence for field device joining (field device to routing device)

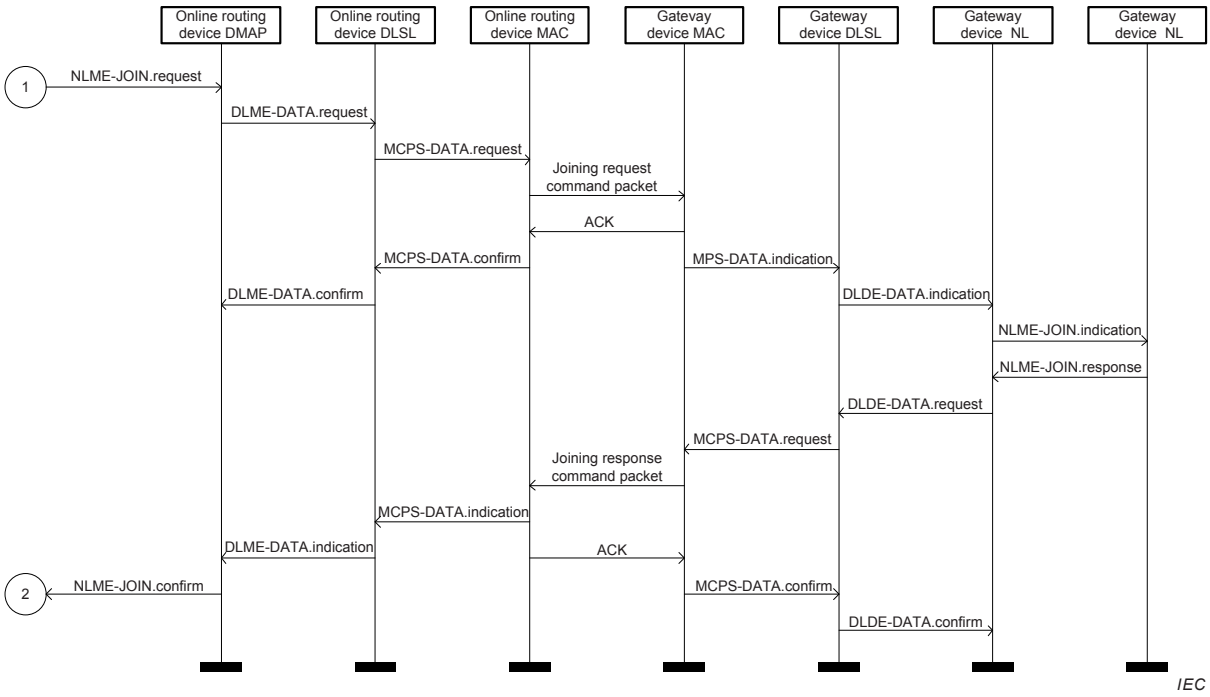

Figure 82 – Time sequence for field device joining (routing device to gateway device)

Figure 83, Figure 84, and Figure 85 illustrate examples of the join process of a routing device.

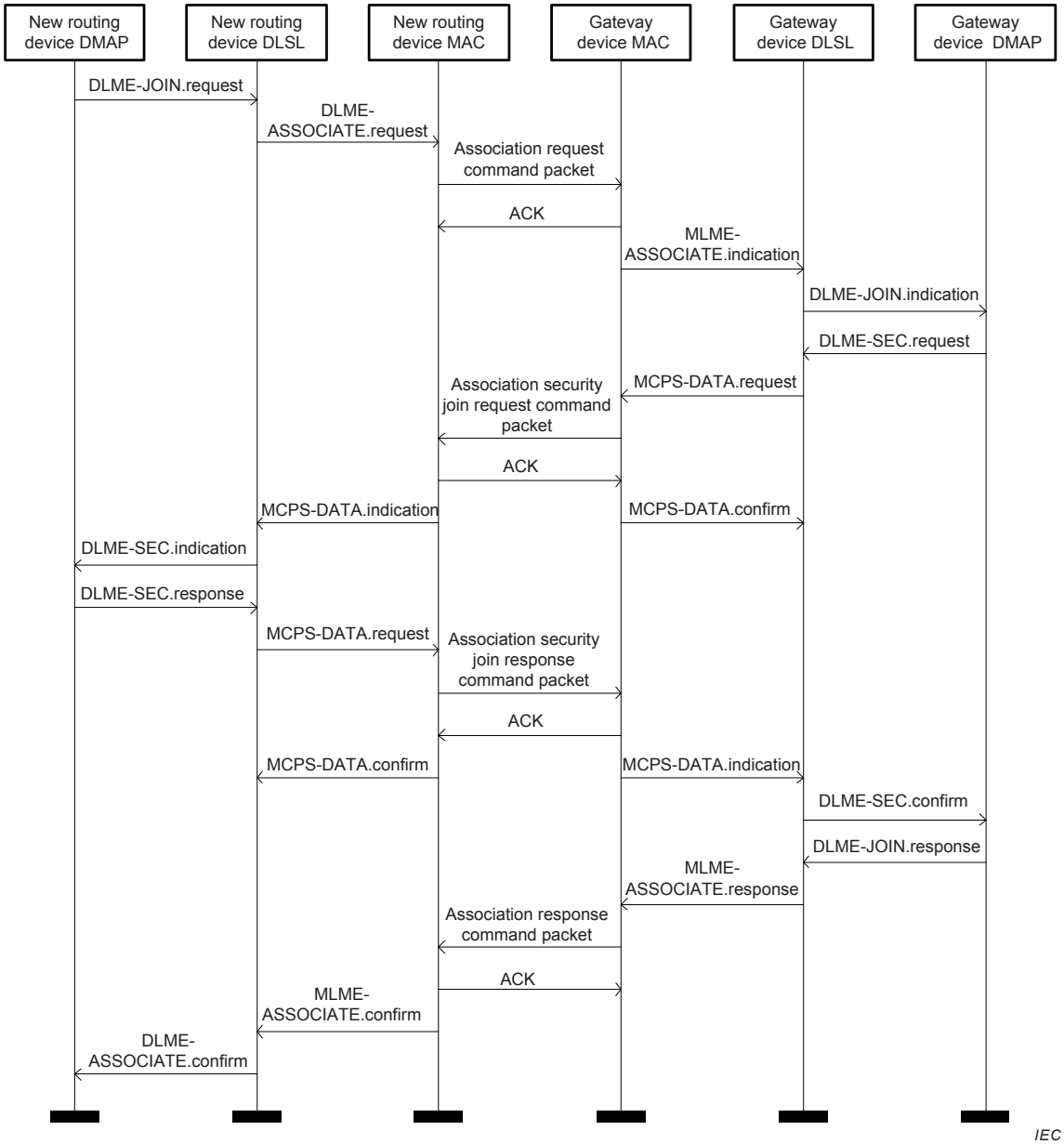

Figure 83 – One-hop joining process for routing device

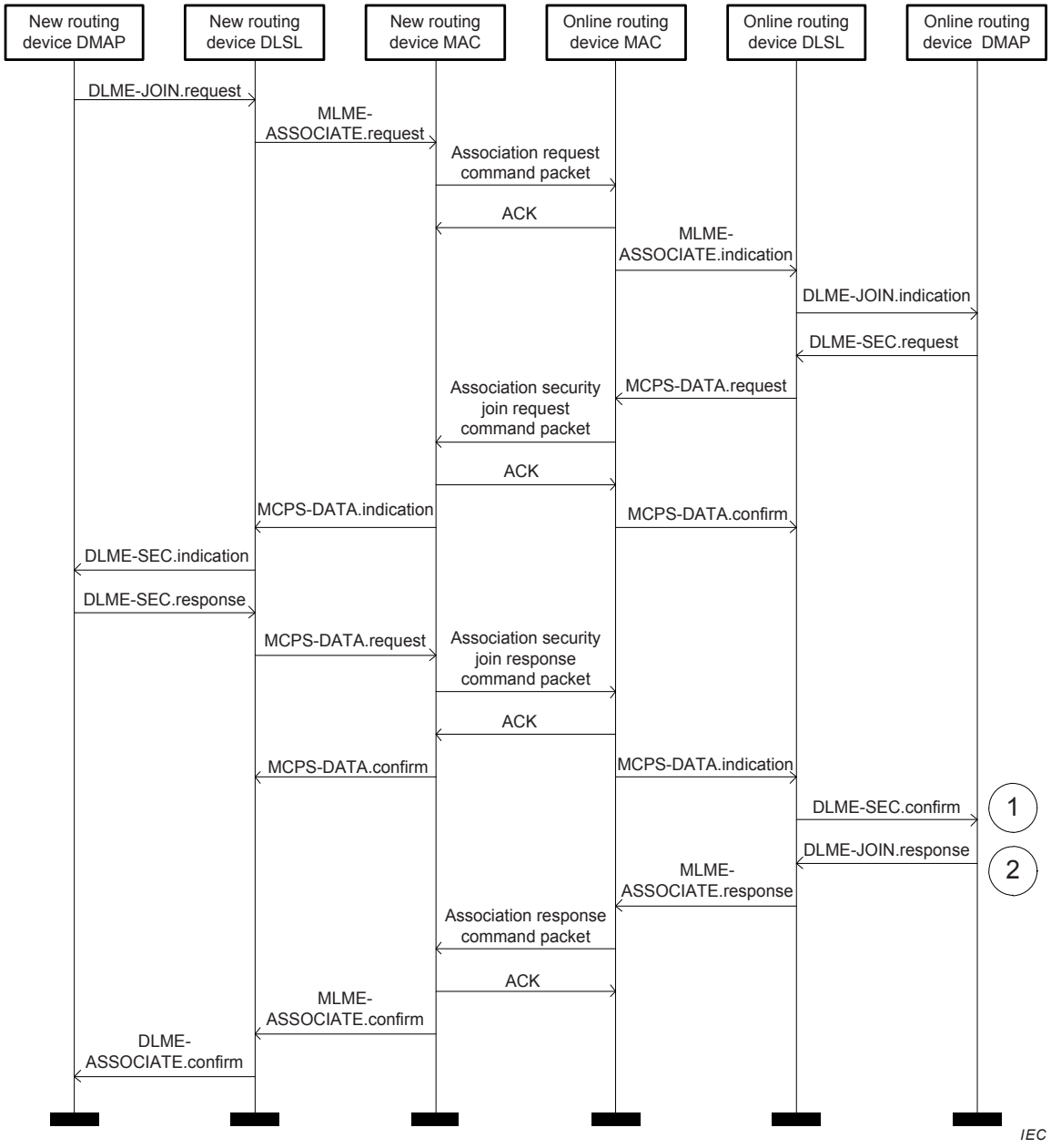

Figure 84 – Multi-hop join process of routing device (new routing device to routing device)

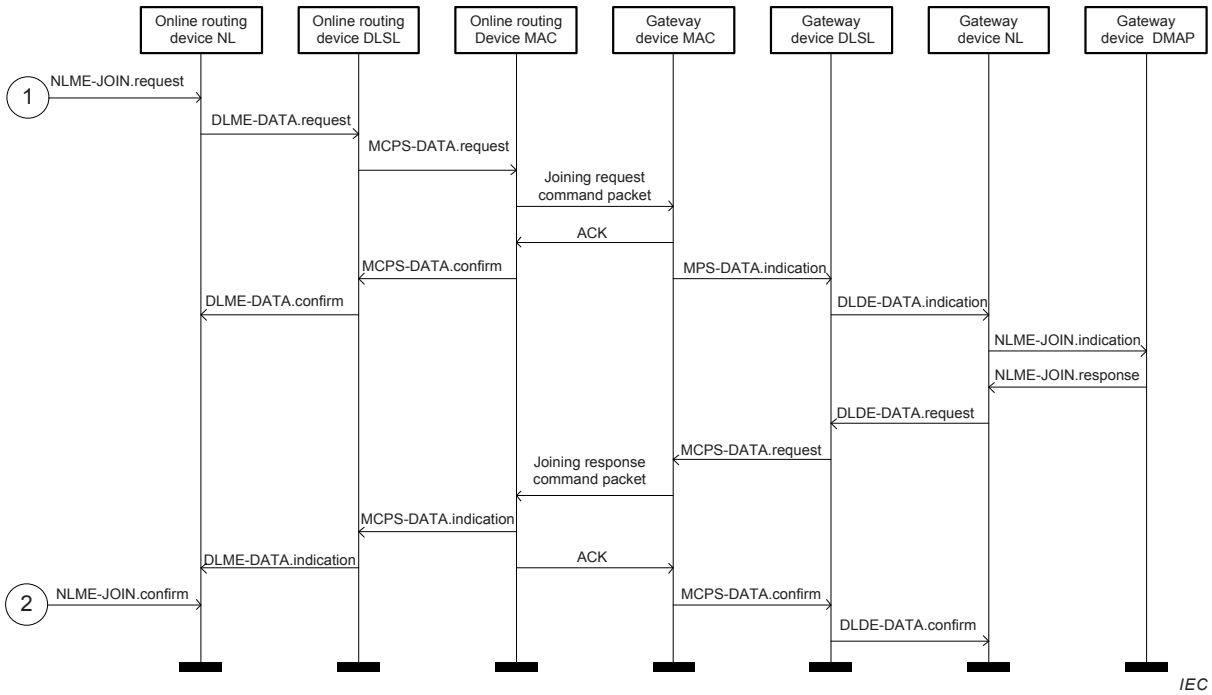

Figure 85 – Multi-hop join process of routing device (routing device to gateway device)

11.6 Secure transportation

11.6.1 Process of secure transportation from field device to host configuration computer

If the aggregation functions of routing devices and field device are not supported, the process of secure transportation from a field device to the host configuration computer includes the following steps.

a) According to the SecMode and SecLevel in the MIB, the security management module of the DMAP in the field device encrypts the ASL payload, gets the integrity MIC in AL and sends it to the NL. According to the SecMode and SecLevel in the MIB, the security management module of the DMAP in the field device encrypts the DLSL payload and acquires the integrity MIC. Then it is sent to the routing device.

b) When the routing device receives the packet, it decrypts the data frame and checks the integrity with the MICin the DLSL, and then uses the next hop DLSL key to get the integrity MIC and encrypt the DLSL payload. Then the packet will be forwarded to the gateway device.

c) When the gateway device receives the packet, the security management module will decrypt the packet and check the MIC on the DLSL and the ASL. Then the gateway device sends these data to the host configuration computer.

If the aggregation function of a field device is supported, the process of secure transportation from a field device to the host configuration computer includes the following steps.

a) If the field device supports the data aggregation, it encrypts the payloads of multiple process data from multiple UAOs. According to the SecMode and SecLevel in the MIB, the security management module of the DMAP in the field device encrypts the ASL payload, acquires the integrity MIC in AL and sends it to the NL. According to the SecMode and SecLevel in the MIB, the security management module of the DMAP in the field device encrypts the DLSL payload and acquires the integrity MIC. Then the encrypted packet is sent to the routing device.

b) If the aggregation function of a routing device is not supported, the routing device receives the packet. The routing device checks the MIC and decrypts the data packet on the DLSL. According to the SecMode and SecLevel in the MIB, the security management

module of the DMAP in the routing device encrypts the DLSL payload and gets the integrity MIC. Then the packet will be forwarded to the gateway device.

c) If the aggregation function of a routing device is supported, the NL informs the DMAP to aggregate these payloads and obtains a new packet. According to the SecMode and SecLevel in the MIB, the security management module of the DMAP in the routing device firstly encrypts the ASL payload and acquires the integrity MIC; then the security management module of the DMAP in the routing device encrypts the DLSL payload and acquires the integrity MIC in the DLSL. The encrypted packet will be forwarded to the gateway device.

d) When the gateway device receives the packet, the security management module checks the MIC and decrypts the packet in the DLSL and ASL. The packet is disaggregated and is sent to the host configuration computer.

11.6.2 Process of secure transportation from host configuration computer to field device

This process is the reverse transportation process from a field device to the host configuration computer.

Annex A
(informative)

Security strategy for WIA-PA network

A.1 Risk analysis for WIA-PA network

As an open system, the WIA-PA network has potential security risks. According to the sources, these risks may either be from inter-networks or other intra-networks of a company or be from the internal WIA-PA networks. The WIA-PA network may suffer some threats, such as invasion, data destruction, and replay attacks.

Therefore, some security measures should be taken to ensure the security operations by the WIA-PA users, to protect the internal resources, and to maintain the normal system operations. The main objectives of the WIA-PA network security are to protect the normal system operations, to find the attacks and adopt corresponding methods as soon as possible, to limit the damage to the lowest level, and to return the network to its normal state as soon as possible after being attacked.

A.2 Security principles for WIA-PA network

According to the characteristics of the WIA-PA network, the following security measures and security principles are recommended:

a) easy deployment and use;

b) be optional for the security services and security measures;

c) reduce the manual operation as much as possible;

d) reduce the transmission of security information as much as possible;

e) use encryption technologies based on hardware; and

f) use the existing encryption and authentication technologies in addition to the published standard.

A.3 Security objectives for WIA-PA network

The objectives of security in the WIA-PA network include:

a) border protection, which provides the border protection for the physical network;

b) system availability, which is used to access system resources when needed by legal users;

c) data integrity, which is used to ensure the consistency of the information and to prevent the illegal users in the system from modifying the data;

d) device authenticity, which is used to authenticate the device identity in the WIA-PA network;

e) confidentiality, which is used to ensure that the system hardware, software, and data can only be used by legal users;

f) key management, which is used to perform the functions of security key updating and management.

A.4 Graded and layered security system

Based on their business, service types, and service objects, users employ different security methods and measures to confront security risks, risk grades, and required security levels. As for the limitations of the computing resources and the computing abilities of field devices, more attention should be paid to the border security of the WIA-PA network.

The graded and layered security system is shown in Table A.1.

Table A.1 – Graded and layered security measures for WIA-PA network

WIA-PA network security level	Level 0	Level 1	Level 2
Border protection	Gateway of message filtering	Gateway of protocol translation and message filtering	Gateway of IPSec translation and message filtering
Security measure	Authentication Integrity check	Authentication Access control Integrity check XOR encryption	Authentication Access control Integrity check Encryption

Both the DLSL and the AL in the WIA-PA network have the ability to establish secure connections. When a routing device forwards data packets, the security levels of the original data packets cannot be lowered. The security levels of the data packets are set according to Table A.2.

Table A.2 – Security levels of data packets

Incoming security level	Outgoing security level			
	Uncheck and unencryption	Encryption and Uncheck	Check and unencryption	Encryption and check
Uncheck and unencryption	allowed	allowed	allowed	allowed
Encryption and uncheck	not allowed	allowed	not allowed	allowed
Check and unencryption	not allowed	not allowed	allowed	allowed
Encryption and check	not allowed	not allowed	not allowed	allowed

Annex B
(informative)

Format description

B.1 Time sequence diagram

Figure B.1 depicts the time sequence diagram. It is an interaction that focuses on the sequence of messages between stack layers of the WIA-PA network devices on the lifelines.

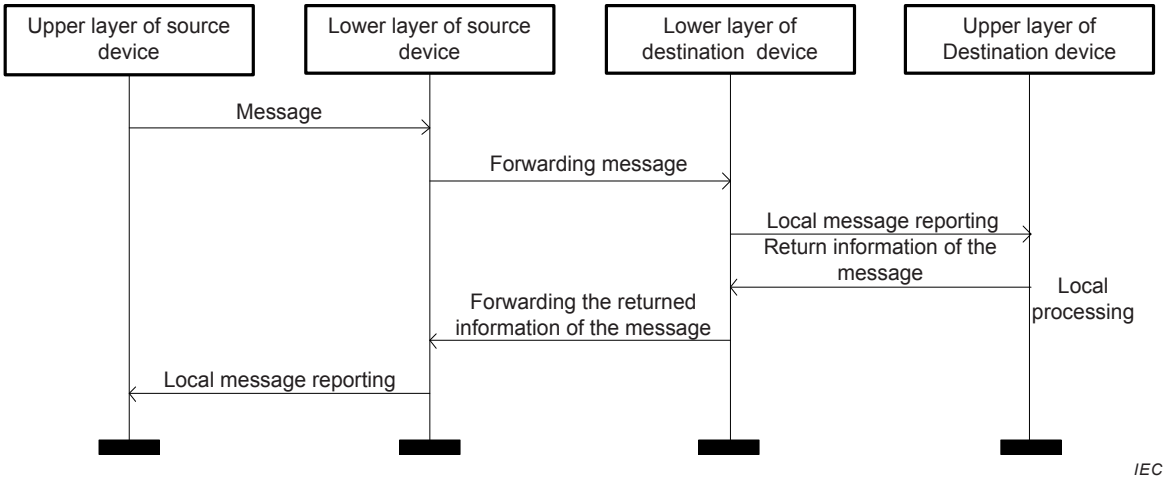

Figure B.1 – Time sequence diagram

The stack layers of the WIA-PA network devices are represented by rectangular blocks. Time is represented by vertical lines. A message is represented by an arrow. In this document, a full arrow is used to depict the general occurrence of a message independently of synchronous or asynchronous handling of the message. The sequence of messages is defined as the order of messages starting from the top of the diagram.

B.2 Packet or frame format

The definitions of a packet and a frame format have same structure. The main difference is their functional range. The packet format is used for NL and ASL, while the frame format is used for MAC and DLSL.

The field format in octet(s) of a packet or a frame is shown in Table B.1.

Table B.1 – Packet or frame format in octet(s)

	Packet or frame format in octet(s)			
Length in octet(s)	X_1	X_2	…	X_N
Field name	Field 1	Field 2	…	Field N

As shown in Table B.1, the value of X_i (i=1,2,…, N) describes the length in the octet(s) of the corresponding packet/frame field. The transmission of a packet/frame should be from the least significant octet to the most significant octet, i.e. the least significant octet of Field 1 should be transmitted first.

One field may have several lengths, depending on different contexts. In those cases, $X_{i1}/\ldots/X_{iM}$ should be used instead of X_i to specify the length of Field i.

One field may contain subfields. Table B.2 specifies the definition of the subfield format if a field with 1-octet length is further divided into several subfields in bit(s).

Table B.2 – Subfield format in bit(s)

	Subfield format in bit(s)			
Length in bit(s)	X_1	X_2	…	X_N
Subfield name	Subfield 1	Subfield 2	…	Subfield m

As shown in Table B.2, the value of X_i(i=1,2,…, N) describes how many bits are occupied by the corresponding subfield, which is set from 1 to 8. The sum of X_1, X_2, … and X_N is fixed at 8. The subfields from left to right are allocated from bit 0 to bit 7. The transmission of the subfield format should be from the least significant bit to the most significant bit, i.e. the least significant bit of Subfield 1 should be transmitted first.

Annex C
(informative)

Example of UAO

C.1 General

This document specifies UAOs by using an object-oriented approach. An UAO has its own attributes and methods. The attributes include the object name, attribute name, attribute identifier, and data type of the attribute. The methods provided by the UAOs are used to realize the operations of these attributes.

The UAOs may include Analog Input Object (AIO), Analog Output Object (AOO), Digital Input Object (DIO), and Digital Output Object (DOO). These objects should be defined in the application profiles. This document utilizes AIO as an example to specify the use of UAOs. Furthermore, DAGO, PAGO, and DGO are regarded as special UAOs.

C.2 Analog input object

C.2.1 Overview

An AIO is used to sample the industrial process and convert the measured value to the data value with physical dimensions. The methods supported by the AIO are shown in Table 171.

C.2.2 Class attribute of AIO

Table C.1 specifies the class attribute of AIO.

Table C.1 – AIO class attribute

Attribute name	Attribute identifier	Data type	Supporting method	Description
Object identifier	0	Unsigned8	Read	The unique identifier of an object that is used for object addressing.

C.2.3 Instance attribute of AIO

Table C.2 specifies the instance attributes of AIO.

Table C.2 – AIO instance attributes

Attribute identifier	Attribute name	Data type	Supporting method	Description
1	Instance ID	Unsigned8	READ	The unique identifier of an object instance
2	ProcessValue	Float	READ	Industrial process variable
3	OutValue	Float	READ PUBLISH REPORT	Output value based on ProcessValue
4	SimulateValue	Float	READ/WRITE	Value written by operator to simulate the OutValue
5	DefaultValue	Float	READ/WRITE	Default value for the OutValue if a sensor or sensor electronic fault is detected
6	HighWarning	Float	READ/WRITE	High limit value of warning
7	HighHighAlm	Float	READ/WRITE	High limit value of alarm
8	LowWarning	Float	READ/WRITE	Low limit value of warning
9	LowLowAlm	Float	READ/WRITE	Low limit value of alarm
10	EngineeringUnit	Unsigned16	READ/WRITE	Engineering unit for all the values
10	Action	Unsigned8	READ/WRITE	Start-up/stop measurement process
11	SamplePeriod	Unsigned32	READ/WRITE	Used for setting sampling cycle
12	Alarm	Unsigned8	READ	0 = Normal; 1 = HighWarning; 2 = LowWarning; 3 = HighAlarm; 4 = LowAlarm 5 to 255 = Reserved.
13 to 99	Reserved			Reserved for future usage
100 to 255	Manufacturer specific			Manufacturer specific attributes

Annex D
(informative)

Country-specific and region-specific provisions

The WIA-PA field medium employs a widely used and accepted physical interface and is therefore appropriate for use in many geographic regions of the world. However, some regions have special considerations that may impose different regulatory requirements. Even if a product is designed to meet those regulatory requirements, its use is often not legally permitted until it has undergone compliance testing and any required local permits or certificates have been issued by country-specific regulatory agencies.

This document is designed to support operation within a fixed geographic area that operates under uniform regulations. As such it is intended to support operation anywhere in the world, as discussed in 8.4.11.

This document does not specify what regulatory certifications or permit a product compliant to this document needs; this is the responsibility of the product manufacturer and the end user to determine. A product may have certifications for operation in multiple countries or regions.

Annex E addresses specific provisions that have been made in this document to support regulatory approval of fixed-locale equipment and systems.

Radio regulations often require devices to operate at constrained power levels at all times, including during over-the-air provisioning. Some identified EIRP thresholds are: 10 mW/MHz (Japan); 10 dBm (China); 10 dBm, 10 dBm/MHz, or 20 dBm (EC in the 2,4 GHz ISM-band); 36 dBm with at most 6 dBi antenna gain (US FCC).

In some countries, such as France, emission levels on certain channels may need to be attenuated. In other countries, such as Korea, the number and range of channels needs to be constrained.

Annex E
(informative)

Regional modification for compliance with ETSI standards

E.1 General

WIA-PA restricts the usage of the spectrum to the 2,4 GHz ISM band, see Clause 7.

Additional requirements apply in Europe to wide band radio frequency transmitting equipment operating in this 2,4 GHz ISM band. Some of these European requirements can be met by complying with two Harmonized Standards from ETSI, EN 300 328 and EN 300 440-2.

Annex E provides additional requirements for compliance of WIA-PA (IEC 62601) devices operating in the 2,4 GHz ISM band with these two ETSI standards:

- EN 300 440-2 is applicable for equipment (devices) with a maximum transmit power between 0 dBm and 10 dBm EIRP. (see Clause E.2);
- EN 300 328 is applicable for equipment (devices) with a maximum transmit power between 10 dBm and 100 dBm EIRP. (see Clause E.3).

NOTE In Annex E, the term “devices” refers to electronics with radios operating according to the appropriate standard; these include but are not limited to gateway devices, routing devices and field devices.

E.2 Compliance with EN 300 440-2

Table E.1 specifies the additional requirements which allow WIA-PA devices operating in the 2,4 GHz ISM band to satisfy the transmit power limitation requirements of EN 300 440-2.

NOTE EN 300 440-2 is listed as a Harmonized Standard under the R&TTE Directive 1999/5/EC which will be superseded by the Radio Equipment Directive 2014/53/EU.

Table E.1 – Applicable EN 300 440-2 requirements list

Parameter	EN 300 440-2 requirements	Additional requirements
Maximum Transmit Power	10 dBm (10 mW) EIRP	Devices shall be configured to emit less than 10 dBm EIRP, that means less than 10 dBm electrical power with an antenna gain of 0 dBi. If the antenna gain is different from 0 dBi, then the resulting EIRP shall be less than 10 dBm by an adequate electrical power adjustment. This requirement overwrites the requirement about electrical power given in 7.3.6.

E.3 Compliance with EN 300 328

Table E.2 specify the additional requirements which allow WIA-PA devices operating in the 2,4 GHz ISM band to satisfy various requirements of EN 300 328.

NOTE EN 300 328 V1.9.1 is listed as a Harmonized Standard under the R&TTE Directive 1999/5/EC which will be superseded by the Radio Equipment Directive 2014/53/EU.

Table E.2 – Applicable EN 300 328 requirements list

Parameter	EN 300 328 V1.9.1 requirements	Additional requirements
Maximum Transmit Power	20 dBm (100 mW) EIRP	Devices shall be configured to emit less than 10 dBm ± 3 dBm EIRP according to 7.3.6.

Parameter	EN 300 328 V1.9.1 requirements	Additional requirements
Maximum Power Spectral Density	10 mW/MHz EIRP	Due to the nature of DSSS (see IEEE 802.15.4-2011, Clause 10), a maximum EIRP of 13 dBm results in a spectral density below 10mW/MHz as required.
Duty Cycle, Tx-sequence, Tx-gap	For non-adaptive equipment with maximum EIRP above 10 dBm: The Duty Cycle shall be equal to or less than the maximum value declared by the supplier. Tx-sequence time <= 10ms. The minimum Tx-gap time following a Tx-sequence shall be equal to the duration of that proceeding Tx-sequence with a minimum of 3.5 ms	No additional requirement needed. The requirements are fulfilled by the following characteristics. The maximum Tx-sequence time for a WIA-PA packet is 4,256 ms in a 10 ms time slot according to Table E.3,TmaxPHY and the transmission time for a WIA-PA ACK is 352 μs according to Table E.3 TACK. The Maximum Tx-sequence Time for WIA-PA is identical to maximum transmission time; the Minimum Tx-gap Time for WIA-PA is identical to maximum transmission time.
Medium Utilization Limit	For non-adaptive equipment with maximum EIRP above 10 dBm: Maximum Medium Utilization factor = 10 %.	The Duty Cycle of a WIA-PA device cannot exceed 43 % based on a 10 ms timeslot and a maximum transmission time for a WIA-PA packet of 4,256 ms in a 10 ms time slot according to Table E.3. This results in a MU factor value of less than 10 % as required. MU = 13 dBm EIRP × 43 % < 9 %

Table E.3 – Timeslot timing definitions and calculations

Symbol	Time value	Calculation basis
T_{maxPHY}	4 256 μs	For 2 400 to 2 483, 5 MHz frequency band and 250 Kbps bit rate, the maximum time to transmit a PHY packet is 4 256 μs, which includes 128 μs Preamble (see Table 71 of IEEE STD 802.15.4-2011), 32 μs SFD (see Table 71 of IEEE STD 802.15.4-2011), and 4 096 μs PHR/PHY Payload (see Table 70 of IEEE STD 802.15.4-2011)
T_{ACK}	352 μs	WIA-PA devices shall use the IEEE STD 802.15.4-2011 ACK. The total length of ACK is 11 octets (see Figure 47 and Figure 67 in IEEE STD 802.15.4-2011,) and the time to transmit an ACK is 352 μs for 2 400 – 2 483,5 MHz frequency band and 250 Kbps bit rate
T_{Offset}	2 120 μs	Beginning of the timeslot to when the receiver shall be listening
T_{Delay}	1 000 μs	End of frame to start of Acknowledgment

Bibliography

ISO/IEC/IEEE 60559:2011, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

IEC 61804-2, *Function blocks (FB) for process control – Part 2: Specification of FB concept*

IEC 61499-1, *Function blocks – Part 1: Architecture*

IEC 61499-2, *Function blocks – Part 2: Software tools requirements*

IEC TR 62390:2005, *Common automation device – Profile guideline*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 9797-1:2011, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*

ISO/IEC 9797-2:2011, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

ISO 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework*

ISO 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework*

ISO 10181-5:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Confidentiality framework*

ISO 10181-6:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Integrity framework*

ISO/IEC 15408-1, *Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model*

ISO/IEC 15408-2, *Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional components*

ISO/IEC TR 10000-1:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework*

ISO/IEC 17799:2005[1], *Information technology – Security techniques – Code of practice for information security management*

ETSI EN 300 328 V1.8.1 (2012-06), *Electromagnetic compatibility and Radio spectrum Matters (ERM); Wideband transmission systems; Data transmission equipment operating in*

[1] Withdrawn.

the 2,4 GHz ISM band and using wide band modulation techniques; Harmonized EN covering essential requirements under article 3.2 of the R&TTE Directive

ETSI EN 300 440-2, V1.4.1 (2010-08), *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short range devices; Radio equipment to be used in the 1 GHz to 40 GHz frequency range; Part 2: Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive*

Directive 1999/5/EC of the European Commission, Radio equipment and telecommunication terminal equipment (R&TTE),
available at: <http://ec.europa.eu/enterprise/sectors/rtte/index_en.htm>

Directive 2014/53/EU of the European Parliament and of the Council, Radio Equipment (repealing Directive 1999/5/EC),
available at: <http://ec.europa.eu/enterprise/sectors/rtte/index_en.htm>

IEEE 64, *Guidelines for 64-bit Global Indentifier (EUI-64™)*

IEEE 754:2008, *Standard for floating-point arithmetic*

_____

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.