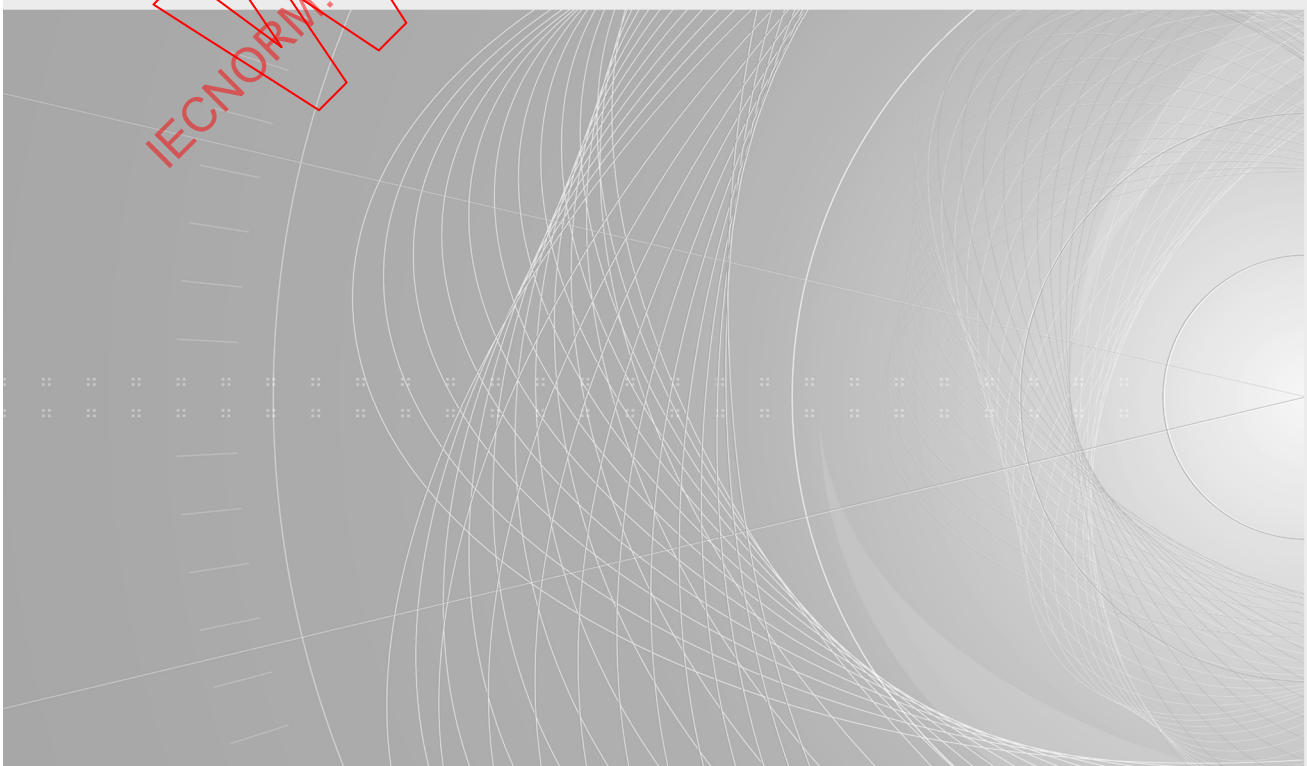
IEC/PAS 62601

Edition 1.0 2009-01

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Fieldbus specifications – WIA-PA communication network and communication profile



IEC/PAS 62601:2009(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

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

About IEC publications

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

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

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

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC/PAS 62601

Edition 1.0 2009-01

PUBLICLY AVAILABLE SPECIFICATION

PRE-STANDARD

Industrial communication networks – Fieldbus specifications – WIA-PA communication network and communication profile

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE XG

ICS 25.040.40; 35.100.05

ISBN 978-2-88910-814-5

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS – WIA-PA COMMUNICATION NETWORK AND COMMUNICATION PROFILE

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
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.

A PAS is a technical specification not fulfilling the requirements for a standard, but made available to the public.

IEC-PAS 62601 has been processed by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this PAS is based on the following document:	This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document
Draft PAS	**Report on voting**
65C/511/NP	65C/518/RVN

Following publication of this PAS, which is a pre-standard publication, the technical committee or subcommittee concerned may transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of 3 years starting from the publication date. The validity may be extended for a single 3-year period, following which it shall be revised to become another type of normative document, or shall be withdrawn.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS – WIA-PA COMMUNICATION NETWORK AND COMMUNICATION PROFILE

1 Scope

This PAS specifies WIA-PA system architecture and communication protocol for process automation based on IEEE 802.15.4.

WIA-PA network is used for industrial monitoring, measurement and control applications.

2 Normative references

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

IEC 61499 (all parts), *Function blocks*

IEC 61804 (all parts), *Function blocks (FB) for process control*

IEEE 802.15.4:2006, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*

3 Terms, definitions, abbreviated terms, acronyms, and conventions

3.1 Terms and definitions

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

3.1.1
active leaving
process by which an online field device is allowed to leave network through applying to its routing device, or by which an online routing device is allowed to leave network through applying to the gateway

3.1.2
adaptive frequency diversity

irregular change of transmit/receive frequency according to actual condition of channels for combating interference and fading

3.1.3
Aggregation
merging several packets into one

3.1.4
Application Sub-layer
a protocol sub layer which provides communication and management services for application layer

3.1.5
Beacon
a special frame broadcast by the routing device and gateway in the WIA-PA network. New routing device or end device join the WIA-PA network by listening to beacons first

3.1.6
Cluster
a logical group of devices which comprises a manager and many data sources

3.1.7
Cluster Head
a manager in a cluster

3.1.8
Cluster Member
a data source in a cluster

3.1.9
communication resource
channels and timeslots used to transport frame

3.1.10
Configuration software
software tools for configuring the network

3.1.11
data link sub-layer

upper layer of IEEE 802.15.4 MAC layer used to handle the aspects of network topology, link and communication resource in WIA-PA

3.1.12
Disaggregation
split-up the merged packet into original ones

3.1.13
Field device
the device which is connected to or controls the process and installed in the industrial field with sensor, actuators, etc

3.1.14
frequency hopping
change of transmit/receive frequency to combat interference and fading

3.1.15
Gateway Device
device connecting a WIA-PA network and other plant networks

3.1.16
Handheld device
a portable device with host application

3.1.17
Host Computer
users, maintenance/management person interact with a WIA-PA network through a host computer

3.1.18
lifecycle
maximum surviving time of each packet that specifies how long a packet can exist in a WIA-PA network before being discarded

3.1.19
link
communication parameters necessary to transport a frame between adjacent devices in the network. It includes source/destination address pairing, timeslot, channel, direction, and link type

3.1.20
mesh
a topology formed by routing devices in WIA-PA. One routing device has more than one connected routing device

3.1.21
Network Manager
responsible for configuration of the network, scheduling communication between routing devices, management of the routing tables and monitoring and reporting the health of the network. There must be one and only one network manager per WIA-PA network

3.1.22
packet lifecycle
maximal packet surviving time from generated to dropped

3.1.23
passive leaving
process by which an online field device is instructed to leave network by its routing device or an online routing device is instructed to leave network by the gateway

3.1.24
Routing device
Routing device forwards packets from one network device to another in a WIA-PA network

3.1.25
Security Manager
configure the security strategies of the whole network, manage keys， and authenticate devices

3.1.26
superframe
Collection of time slots repeating at a constant rate. It specifies the transmitting or receiving time of periodic communication

3.1.27
timeslot
basic time unit of data exchange. Its duration is configurable in WIA-PA

3.1.28
timeslot hopping
regular change of transmit/receive frequency per timeslot to combat interference and fading

3.1.29
traffic control
ensure that the data rate of transmitter is lower than that of receiver. Therefore, ensure the data integrity

3.1.30
user application process
the program that realizes some function specified by users

3.1.31
VCR
virtual communication relation identifying the communication resource between two user application objects

3.1.32
WIA-PA device
WIA-PA device includes gateway device, routing device, and field device in a WIA-PA network

3.2 Abbreviated terms and acronyms

ACK	Acknowledge
AFD	Adaptive Frequency Diversity
AGO	Aggregation object
AI	Analog Input
AIO	Analog Input Object
AL	Application Layer
AMIB	Application Layer Management Information Base
AO	Analog Output
AOO	Analog Output Object
ASL	Application Sub-layer
ASLDE	Application Sub-layer Data Entity
ASLME	Application Sub-layer Management Entity
ASLPDU	Application Sub-Layer Protocol Data Unit
CAP	Contention Access Period
CCA	Clear Channel Assessment
CFP	Contention-Free Period
C\S	Client-Server
CSMA	Carrier Sense Multiple Access
DAGO	Disaggregation Object
DLDE	Data Link Layer Data Entity
DLDE-SAP	Data Link Layer Data Entity Service Access Point
DLL	Data Link Layer
DLL-MIB	Data Link Layer Management Information Base
DLME	Data Link Layer Management Entity
DLME-SAP	Data Link Layer Management Entity Service Access Point
DLPDU	Data Link Layer Protocol Data Unit
DMAP	Device Management Application Process

DSMO	Device Security Management Object
EPR	Effective Radiated Power
EUI-64	Extended Unique Identifier (64 bits long)
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
GW	Gateway
IAT	International Atomic Time
ID	Identifier
IDS	Intrusion Detection Systems
KED	Data Encryption Key
KEK	Key Encryption Key
KJ	Join Key
KP	Provision Key
KS	Session Key
LIFS	Long InterFrame Spacing
LME-SAP	Layer Management Entity Service Access Point
LSB	Least Significant Bit
MAC	Medium Access Control
MHR	Medium Access Control Header
MIB	Management Information Base
MIC	Message integrity code
MSB	Most Significant Bit
NACK	Non-Acknowledge
NL	Network Layer
NLDE	Network Layer Data Entity
NLME	Network Layer Management Entity
NM	Network Manager
NMA	Network Management Agent
NL-MIB	Network Management Information Base
NPDU	Network Protocol Data Unit
NSDU	Network Service Data Unit
OUI	Organizationally Unique Identifier
P\S	Publisher-Subscriber
SAP	Service Access Point
SIFS	Short InterFrame Spacing
SM	Security Manager
SMA	Security Management Agent

S-MIB	Security Management Information Base
SMK	Symmetric Master Key
TDMA	Time Division Multiple Access
TH	Timeslot Hopping
UAO	User Application Object
UAO_ID	User Application Object Identifier
UAP	User Application Process
UTC	Coordinated Universal Time
VCR	Virtual Communication Relationship
VCR_ID	Virtual Communication Relationship Identifier
WIA-PA	Wireless Network for Industrial Automation – Process Automation

4 Definition of data types

4.1 Representation of Boolean type

Boolean ::= BOOLEAN --non-zero means TRUE

--o means FALSE

4.2 Representation of integer type

Int8 ::= INTEGER（-128..+127） -- integer range: $-2^7 <= i <= 2^7-1$

Int16 ::= INTEGER（-32768..+32767） -- integer range: $-2^{15} <= i <= 2^{15}-1$

Int32 ::= INTEGER -- interger range: $-2^{31} <= i <= 2^{31}-1$

Int64 ::= INTEGER -- integer range: $-2^{63} <= i <= 2^{63}-1$

4.3 Representation of unsigned integer type

Unsigned8 ::= INTEGER（0..255） -- integer range: $0 <= i <= 2^8-1$

Unsigned16 ::= INTEGER（0..65535） -- integer range: $0 <= i <= 2^{16}-1$

Unsigned32 ::= INTEGER -- integer range: $0 <= i <= 2^{32}-1$

Unsigned64 ::= INTEGER -- integer range: $0 <= i <= 2^{64}-1$

4.4 Representation of floating point number type

Real ::= BIT STRING SIZE（4） -- single precision

4.5 Representation of visible string type

VisibleString ::= VISIBLE STRING -- generally using

4.6 Representation of 8-bit byte type

Octetstring ::= OCTET STRING -- generally using

4.7 Representation of bit string type

BitString ::= BIT STRING -- generally using

4.8 Representation of time-of-day type

TimeOfDay ::= Octet6

4.9 Representation of binary date type

BinaryDate ::= Octet8

4.10 Representation of time difference type

TimeDifference ::= Octet6

5 WIA-PA overview

5.1 Device Type

There are five device types defined in WIA-PA.

- Host Computer

Users, maintenance/management person interact with a WIA-PA network through a *host computer*.

- Gateway Device

Gateway device connects a Host computer and a WIA-PA network. It provides interface between the WIA-PA network and other plant networks.

- Routing Device

Routing device forwards packets from one network device to another.

- Field device

Field Device is connected to or controls the process and installed in the industrial field with sensor, actuators, etc.

- Handheld Device

Handheld device is used in configuration and maintenance of a WIA-PA device. It can access the WIA-PA network temporarily.

5.2 Network Topology

A hierarchical network topology, hybrid star and mesh, is supported by WIA-PA. It is illustrated in Figure 1.

The first level network is mesh topology in which routing devices and gateway devices are deployed.

The second level network is star topology in which routing devices and field/handheld devices are deployed.

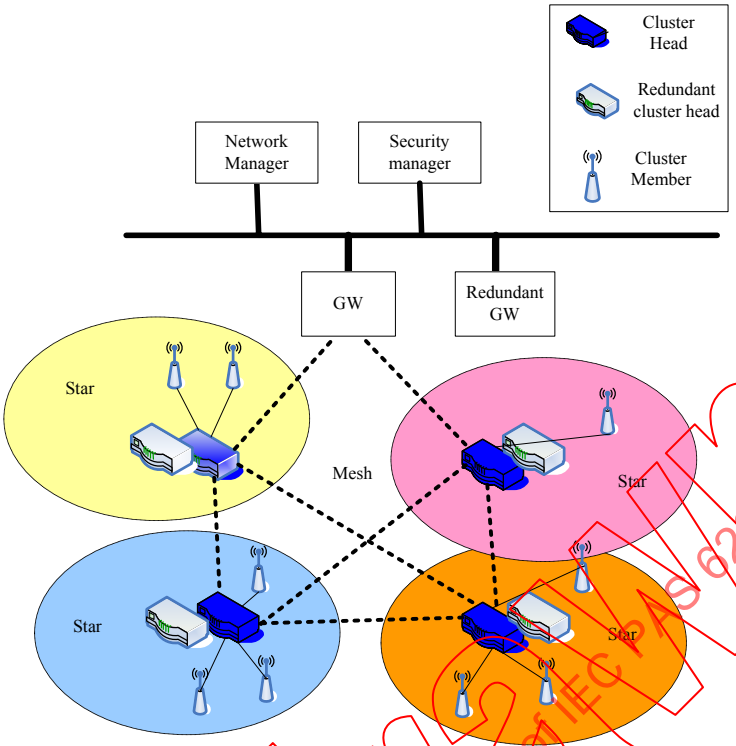


Figure 1 – Architecture of WIA-PA

Five logical roles are defined in WIA-PA:

- Network manager

Network manager is responsible for constructing the mesh network of routing devices and monitoring the performance of the whole network.

- Security manager

Security manager is responsible for security key management and security authentication of routing devices and field devices.

- Cluster head

Cluster head act as the agent of network manager, in charge of constructing the star network of field devices and monitoring the performance of the star network. Act as the agent of security manager, in charge of merging and forward packets of local cluster members and forward packets from other cluster heads.

- Redundant cluster head

Redundant cluster head is hot backup of the cluster head.

- Cluster member

Cluster member is responsible for collecting field data and sending the data to its cluster head.

A gateway device should act as the logical roles of network manager and security manager. A routing device should act as cluster head and redundant cluster head, but should not be both at the same time. A field device should act as cluster member only.

5.3 Stack

WIA-PA protocol stack is based on ISO/OSI 7-layer reference model. WIA-PA defines data link layer, network layer and application layer only. See Figure 2.

OSI Layer	Function	WIA-PA
Application	Provides the User with Network Capable Application	Provides the User with Network Capable Application
Presentation	Converts Application Data Between Network and Local Machine Formats	
Session	Connection Management Services for Applications	
Transport	Provides Network Independent, Transparent Message Transfer	
Network	End to End Routing of Packets. Resolving Network Addresses	Power-Optimized Redundant Path, Star and Mesh Networking
Data Link	Establishes Data Packet Structure, Framing, Error Detection, Bus Arbitration	Secure & Reliable, Time Synced, TDMA/ CSMA,Frequency Agile with ARQ
Physical	Mechanical / Electrical Connection. Transmits RawBit Stream	802.15.4 based radios

Figure 2 – OSI 7-Layer Communication Model mapped to WIA-PA

The protocol stack architecture of a WIA-PA network is illustrated in Figure 3. The purple block is the protocol layer entity. The yellow block is the functional component. The green block is the data and management interface between protocol layer entities.

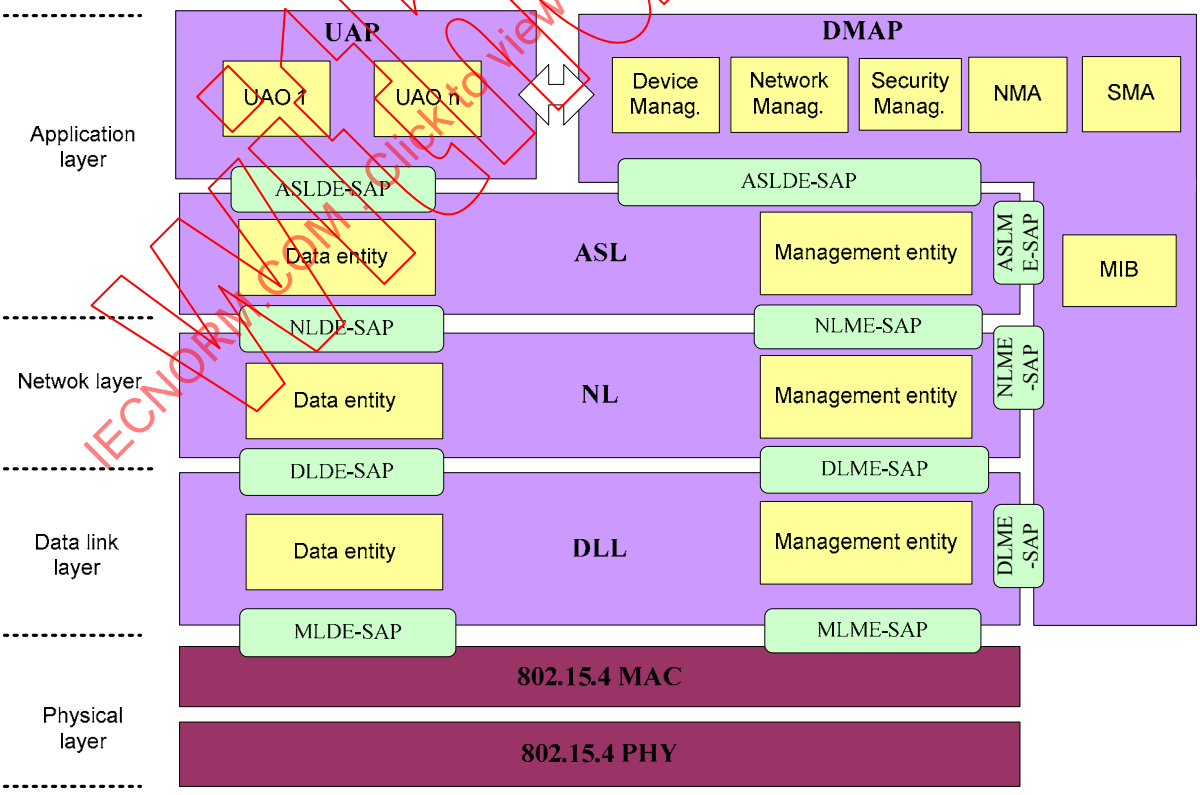


Figure 3 – WIA-PA protocol stack

5.4 Interconnection

WIA-PA realizes the network interconnection through a WIA-PA gateway. Besides the communication to the WIA-PA Network Manager and Security Manager, the WIA-PA gateway can communicate with other WIA-PA devices in order to exchange information between the devices. Meanwhile, a WIA-PA gateway can connect other networks, such as a wired fieldbus. The framework of a WIA-PA gateway is shown as Figure 4.

WIA-PA gateway includes the following components:

- WIA-PA access point

The WIA-PA access point physically connects the WIA-PA networks and transmits the state information and data of the WIA-PA routers and devices.

- Virtual devices

Virtual device defines a communication interfaces for other networks. This interface is used to map a data source from other networks into a WIA-PA device.

- Data disaggregation

This function of the WIA-PA gateway is used to disaggregate the packet which is aggregated in routing device.

- Data images

Data images store the data of device in the WIA-PA network, and provide an interface for access from other networks.

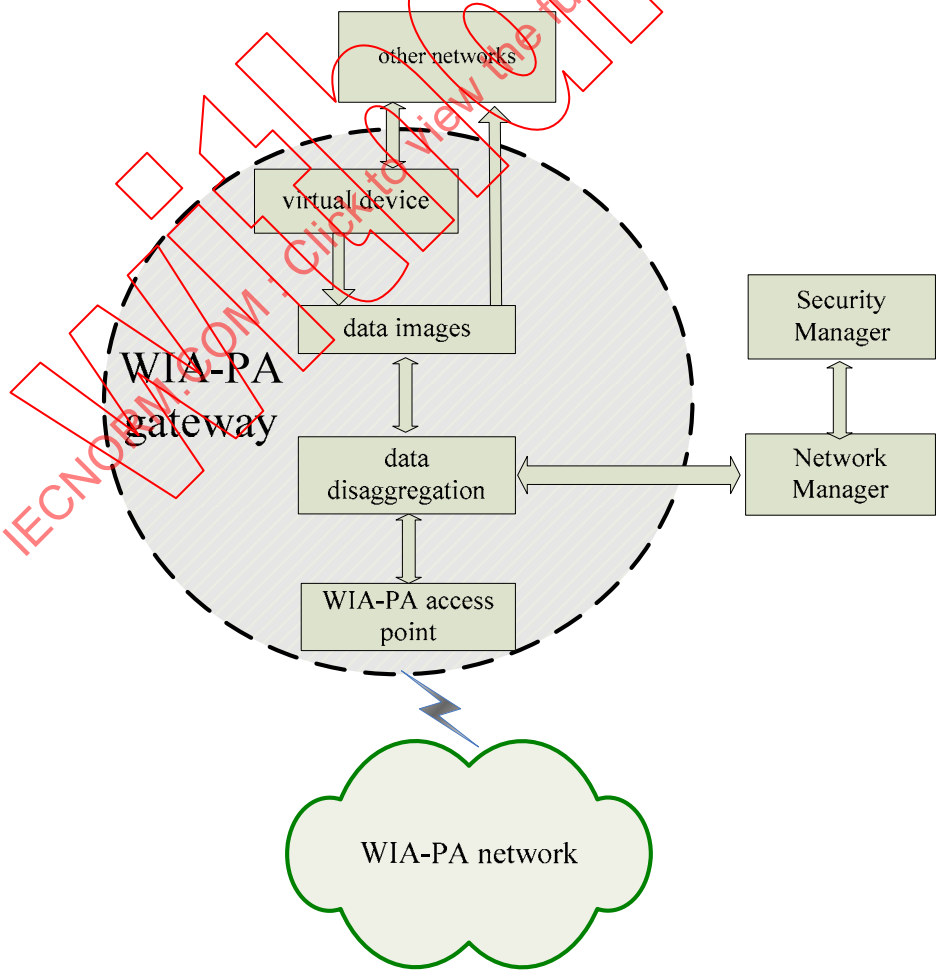


Figure 4 – The framework of a WIA-PA gateway

6 System Management

6.1 Overview

System management in WIA-PA includes network management and security management. The functions of system management are implemented by DMAP in each device.

Network management specifies the management of physical attributes of devices and attributes related to communicating and networking.

Network management functions include:

- Allocating network address

  Each device in a WIA-PA network has a EUI 64-bits address called long address and a 16-bits network address called short address. Long address of every device is assigned by vendor according to IEEE EUI–64 standard. The network address of the routing device is assigned by network manager. The network address of field devices is assigned by the routing device.

- Routing configuration

  Routing is implemented by the routing device. The routing table in each routing device is configured by network manager.

- Joining and leaving the network

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

- Communication resources configuration

  Communication resources of a WIA-PA network are divided into resource blocks. When a routing device successfully joins the network, it should be allocated a block of communication resources by network manager. After the routing device has established a star network, the communication resources of each field device are allocated from the communication resources of routing device.

- Configuration of clock source and system time services

  The WIA-PA network shall set one time source, which is usually acted by gateway. Devices in the network are usually relatively time synchronized with the gateway time in order to differentiate the orders of the events happening. As regards to the synchronization between the gateway time and the IAT, it is an optional function.

- Performance monitoring

  Performance monitoring is responsible for monitoring the performance of WIA-PA network.

- Firmware updating

  Firmware updating is responsible for updating protocol stack and user application.

- Layer management

  Layer management is responsible for maintaining the MIB in every layer.

The function of network management is achieved by DMAPs in network manager, cluster head and cluster members.

Security management specifies the management of attributes associated with network security. The function of security management is achieved by DMAPs in security manager, cluster head and cluster members.

For security management functions, see Clause 11.

DMAP is the entity that realizes the functions of system management in each device, its components include: network management, security management, network management agent, security management agent, device management and management information base. See Figure 5.

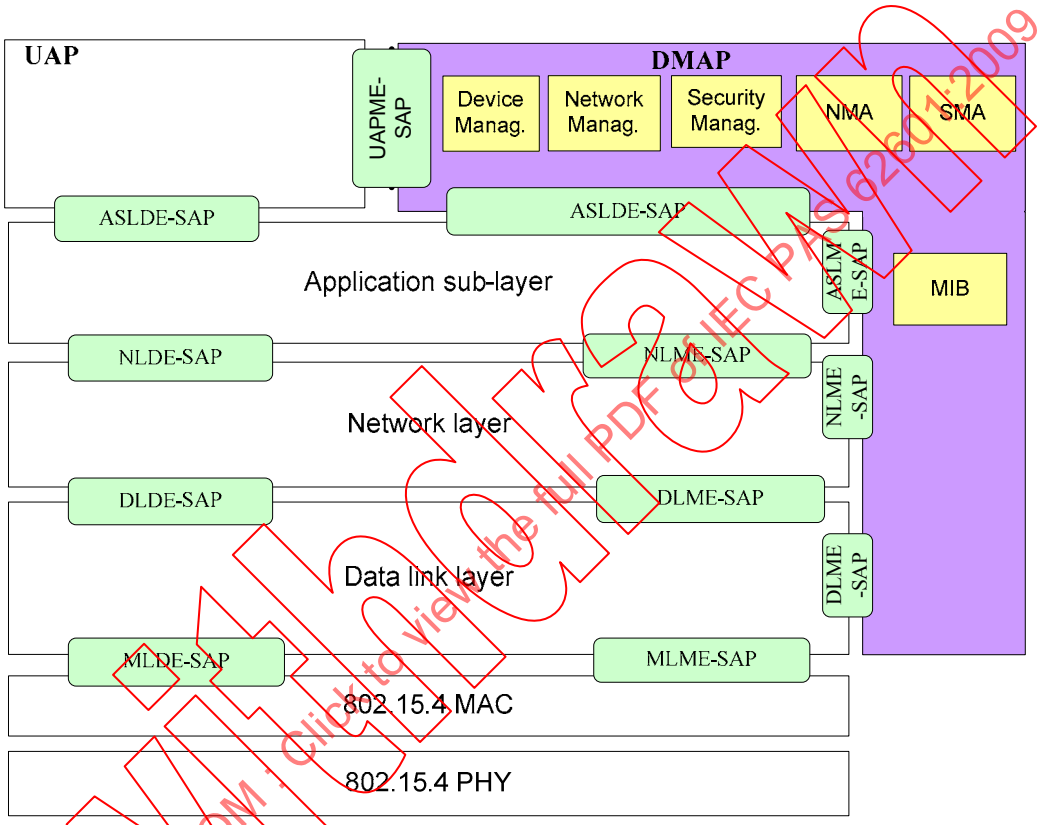


Figure 5 – DMAP in system management

6.2 Framework of system management

WIA-PA uses centralized management as well as distributed management scheme. See Figure 6.

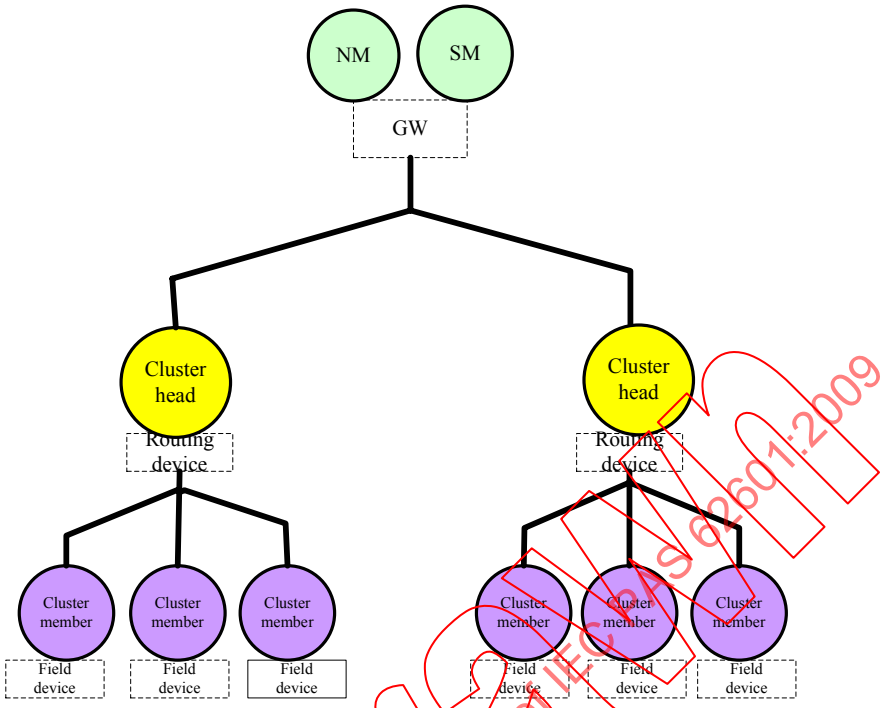


Figure 6 – System management with centralized and distributed approach

The centralized management is implemented by the network management component and security management component of DMAPs in network manager and security manager. The distributed management is implemented by the management agent component of DMAP in cluster head. See Figure 7.

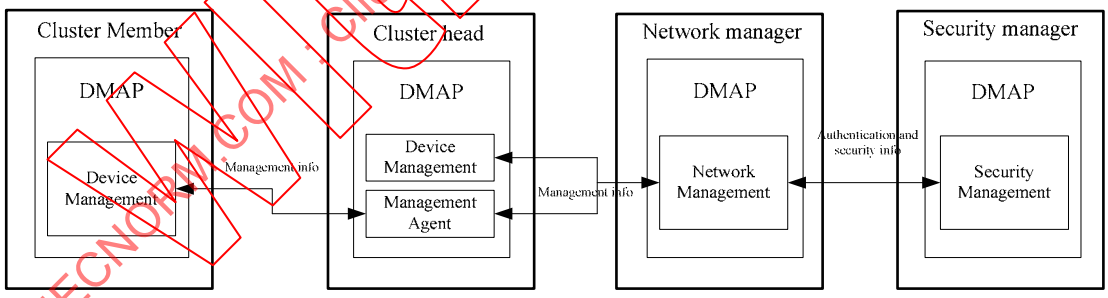


Figure 7 – System management flow

6.3 Virtual Communication Relationship

6.3.1 General

Virtual Communication Relationship (VCR) identifies the communication resource between two UAOs. The UAOs connected by VCR are called VCR endpoints. Each VCR is identified by VCR-ID.

To differentiate the communication resource of mesh network and star network, VCRs are classified into intra-cluster VCR and inter-cluster VCR.

To differentiate the usage, VCRs are also classified into management VCR and data VCR.

The Intra-cluster management services are accomplished by DMAPs in cluster head and cluster member through intra-cluster management VCR.

The Intra-cluster data services are accomplished by AGO in cluster head and UAO in cluster member through intra-cluster data VCR.

The Inter-cluster management services are accomplished by DMAPs in network manager and cluster head through inter-cluster management VCR.

The Inter-cluster data services are accomplished by DAGO in gateway and AGO in cluster head through inter-cluster data VCR.

The VCR_ID of management VCR is the default 0.

6.3.2 VCR structure

Table 1 – VCR structure

Name	Data type	Valid range	Description
VCR_ID	Unsigned16	0~0xFFFF	Identifier
VCR_TYPE	Unsigned8	0~1	Identifies Intra-cluster VCR and inter-cluster VCR
VCR_SRC_OBJ_ID	Unsigned16	0~0xFFFF	VCR Source node application object identifier
VCR_DES_OBJ_ID	Unsigned16	0~0xFFFF	VCR Destination Node application object identifier
VCR_STATUS	Unsigned8	0x00~0xFF	status
VCR_ACTIVATION_TIME	Unsigned64	0~0xFFFFFFFF FFFFFFFF	The activation time of VCR
SERVICE_TIME	Unsigned64	0~0xFFFFFFFF FFFFFFFF	The valid service times of VCR
SOURCE_CH_ADDRESS	Unsigned16	0~0xFFFF	VCR address at the cluster head
DESTINATION_ADDRESS	Unsigned16	0~0xFFFF	Destination address
VCR_PRIORITY	Unsigned8	0x00~0xFF	The priority of VCR
SECURITY_POLICY	Unsigned8	0x00~0xFF	Safe strategy
MAX_PDU_SIZE	Unsigned8	0x00~0xFF	The maximum size of PDU
AG_FLAG	Unsigned8	0~1	Indicates if the packet aggregation function is used or not
NOTE The time unit used in this table is the time slot used by superframe.			

6.3.3 VCR Establishment

Management VCR identifies the communication resource for the management services, and is established in the join process. If network manager has enough communication resource to accept a routing device to join, an inter-cluster management VCR is established between the joining routing device and network manager. If a routing device has enough communication resource to accept a field device, an intra-cluster management VCR is established between the joining field device and the routing device.

Data VCR identifies the communication resource for the data services, and is established after the applications of the network is configured. If the data update rate of a UAO in a field device is configured, the field device will send out an intra-cluster data VCR request to cluster head to ask for communication resource for this UAO. If cluster head does not have enough communication resource for this request, it sends out a negative response, else it will suspend this request, and ask for communication resource for relaying. The cluster head

starts up an inter-cluster data VCR request to network manager. If network manager has enough communication resource for this request, it will send out a positive response, and the inter-cluster data VCR is established. After the corresponding inter-cluster VCR is successfully established, the suspending intra-cluster VCR request is processed, and a positive response is sent to the requesting field device, and the intra-cluster data VCR is established.

6.3.4 VCR release

VCR is released when routing device or field device leaves the network. This process is the inverse process of the establishment.

6.4 Network management

6.4.1 General

Network management component only exists in DMAP of network manager. Its responsibility is to establish and manage the mesh network. The management services are accomplished through the inter-cluster management VCRs between network manager and routing devices.

6.4.2 Address assignment for the cluster head

In WIA-PA network, cluster head is acted by routing device. Each routing device in the network has a global unique 64bits "long address" and a 16 bits "short address". The long address is set by manufacturers according to IEEE EUI-64 standard.

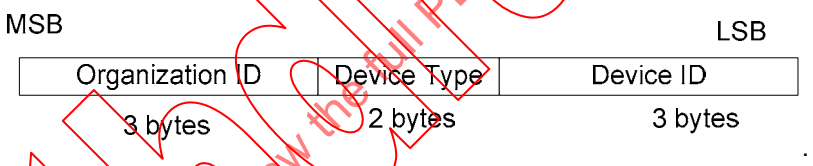


Figure 8 – Long address structure of routing device

The short address is two bytes long. MSB is cluster address which identifies the different clusters, and LSB is 0. The "short address" of cluster head is assigned by network manager.

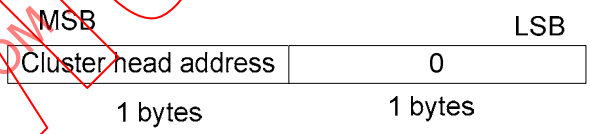


Figure 9 – Short address structure of routing device

6.4.3 Routing configuration

6.4.4 General

In WIA-PA network, routing is implemented by the routing device. The field device does not have routing function. The routing table in each routing device is configured by network manager.

6.4.5 Join process of routing device

When a routing device starts to join the network, it shall be provided with a cryptographic key.

The general progression that shall be followed for the routing device includes:

a) The routing device keeps scanning the available channels until it successfully receives a beacon from an online routing device or gateway.

b) The routing device chooses the online routing device or gateway as the temporal parent, and synchronizes with the network according to the received beacon.

c) The routing device sends a join request to its temporal parent which will forward the request to Network Manager.

d) When receiving a join request, network manager should communicate with security managers to complete authentication process. Network Manager returns a confirmed message.

e) The routing device receives the confirmed message relayed by its temporal parent. If it is a negative confirmation, the routing device will restart this join process, else, the routing device will request to establish an inter-cluster management VCR.

f) After an inter-cluster management VCR is established, Network manager configures the superframe and the routing table of the routing device, and the join process is finished.

g) The routing device begins to send out beacon.

These above steps are described Figure 10.

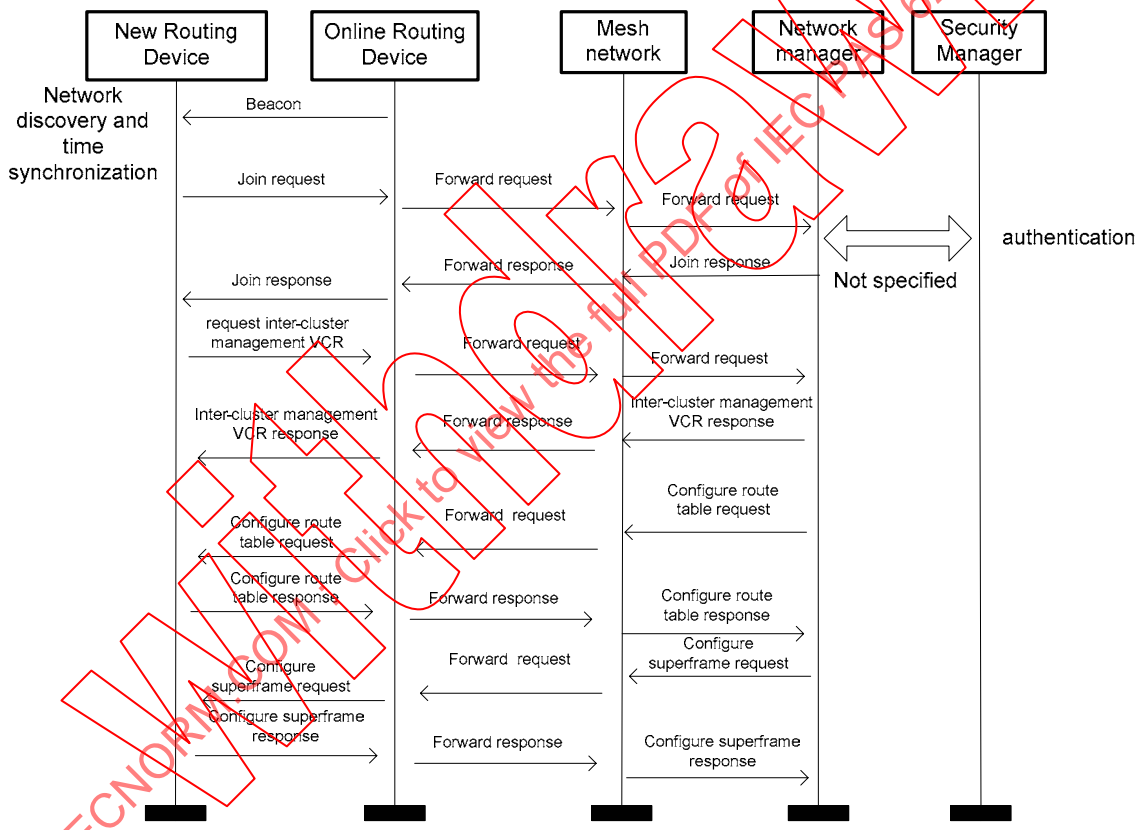


Figure 10 – Join process of routing device

6.4.6 Leaving process of the routing device

WIA-PA defines two leaving process for routing device, i.e., active leaving and passive leaving. In active leaving process, routing device notifies the gateway of its intent to leave the network. In passive leaving process, routing device is notified to leave the network.

The active leaving process of the routing device is as follow:

a) The routing device sends leaving request to network manager through the inter-cluster management VCR.

b) Network manager gives a response to the routing device.

c) After receiving the response, the routing device leaves the network.

d) Network manager releases network address and communication resources of the leaved routing device, and updates the network topology.

e) Network manager notifies routing devices which have allocated communication resource for leaved routing device to release the related communication resource.

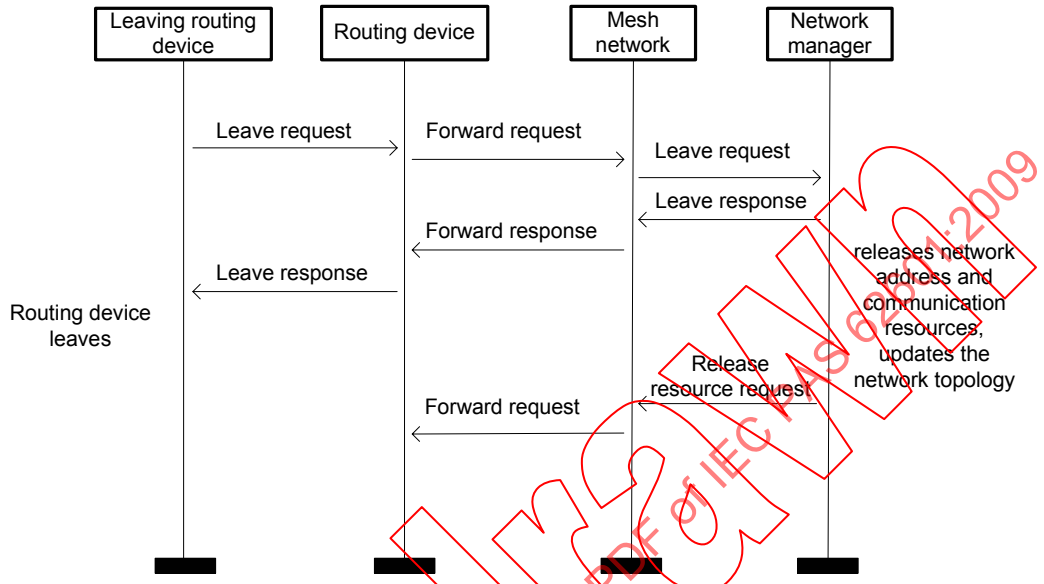


Figure 11 – Active leaving process of routing device

The passive leaving process of the routing device is as follow:

a) Network manager requests a routing device to leave through the inter-cluster management VCR.

b) The routing device gives a response to the network manager.

c) The routing device leaves the network.

d) After receiving the response from the routing device, Network manager releases network address and communication resources of the leaved routing device, and updates the network topology information.

e) Network manager notifies routing devices which have allocated communication resource for leaved routing device to release the related communication resource.

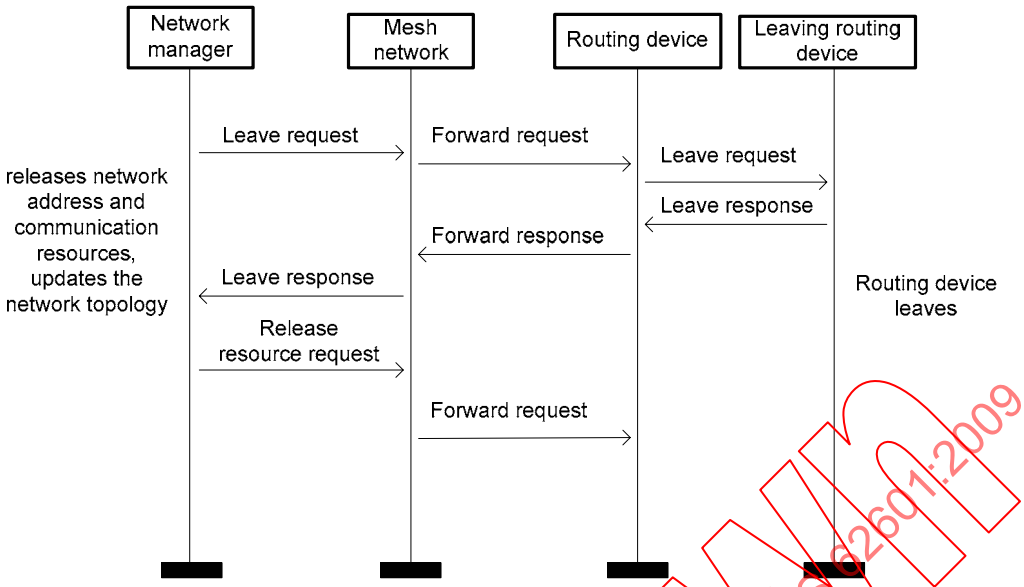


Figure 12 – Passive leaving process of routing device

6.4.7 Allocation of communication resources for routing device

WIA-PA applies a hybrid of centralized and distributed management scheme. Each routing device is assigned a block of communication resource by network manager in the join process. This block of communication resource is organized in the form of superframe. After joining the network, each routing device broadcasts its superframe architecture through a beacon.

6.4.8 Time source and time synchronization

A time source shall be configured in WIA-PA network. Usually gateway device may act as time source. Devices in the network should synchronize with the gateway. Whether gateway device synchronizes with the IAT or not is optional in WIA-PA.

To guarantee the time accuracy, the maximal synchronization interval should be set.

6.4.9 Network performance monitoring

The cluster head should report the network's health information to network manager periodically.

Network manager should optimize the performance of the network and react to the changing of the network. In addition, network manager should set the alarm limit, such as energy insufficiency, link broken and so on.

6.4.10 Firmware update

The firmware is updated by network manager through the following steps.

- Reading the attributes in device MIB to determine if a previous updating process is in progress. If yes, the present updating process should be cancelled. Otherwise, the next.
- Initializing the firmware downloading process.
- Monitoring the status of the device, until the device status changes into updating enable state.
- Downloading firmware.
- Activating the new firmware by writing an update request command to the device. Wait for the device to activate the new firmware by monitoring the state attribute until it transitions into an idle state for as long as the download activation time of the device.

- If switch time is written in device DMAP, then device will activate the new firmware at switch time.

6.5 Network Management Agent

6.5.1 General

Network management agent only exists in DMAP of cluster head. Its responsibility is to establish and manage the star network.

6.5.2 Cluster member address assignment

In WIA-PA network, cluster member is acted by field device. Each field device in the network has a global unique 64 bits “long address” and a 16 bits “short address”. The “long address” is set by manufacturers according to IEEE EUI-64 standard.

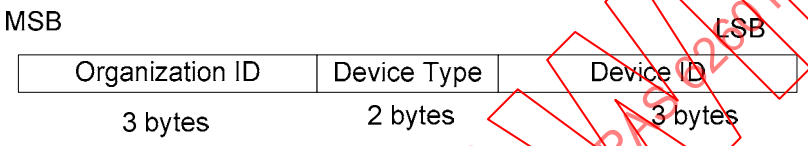


Figure 13 – Long address structure

The short address is two bytes long. MSB is cluster address, and LSB is the cluster member address. Cluster member address is assigned by cluster head.

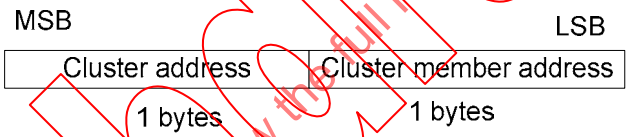


Figure 14 – Short address structure

6.5.3 Joining process of cluster member

When a field device starts to join the network, it shall be provided with a cryptographic key.

The joining process that shall be followed for the field device includes:

a) The field device keeps scanning the available channels until successfully receive a beacon from an online routing device.

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

c) The field device sends a join request to cluster head.

d) When receiving a join request, cluster head returns a confirmed message according its available communication resource.

e) The field device receives the confirmed message from cluster head. If it is a negative confirmation, the field device will restart this join process, else, the field device will request to establish an intra-cluster management VCR.

f) After an intra-cluster management VCR is established, the field device reports its UAO list to cluster head.

g) Cluster head updates its cluster member list and UAO list. The new cluster member list and UAO list is reported to network manager by cluster head.

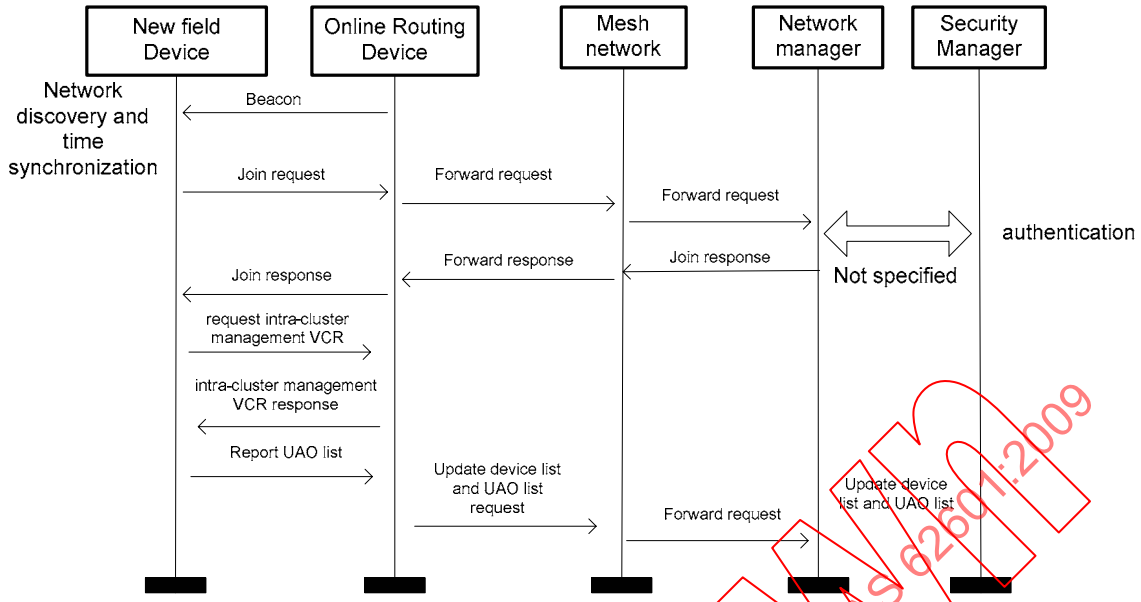


Figure 15 – Joining process of field device

6.5.4 Leaving process of cluster member

WIA-PA defines two leaving process for field device, i.e., active leaving and passive leaving. In active leaving process, field device notifies the gateway of its intent to leave the network. In passive leaving process, field device is notified to leave the network.

The active leaving process of field device includes:

a) The field device sends leaving request to cluster head through the intra-cluster management VCR.

b) Cluster head gives a response to the field device.

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

d) Cluster head releases network address and communication resources of the leaved field device, and updates cluster member list and UAO list.

e) Cluster head report the updated cluster member list and UAO list to network manager.

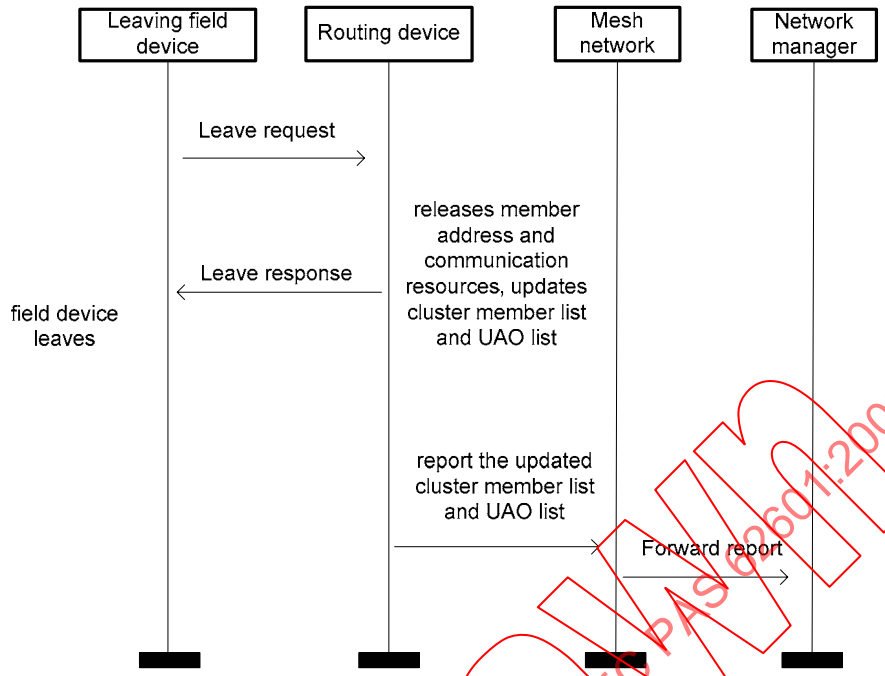


Figure 16 – Active leaving process of field device

The passive leaving process of field device includes:

a) Network manager sends leaving request to cluster head through inter-cluster management VCR. This request is forward by cluster head to the field device through the intra-cluster management VCR.

b) Field device gives a response to the gateway through cluster head.

c) The field device leaves the network.

d) Cluster head releases network address and communication resources of the leaved field device, and updates cluster member list and UAO list.

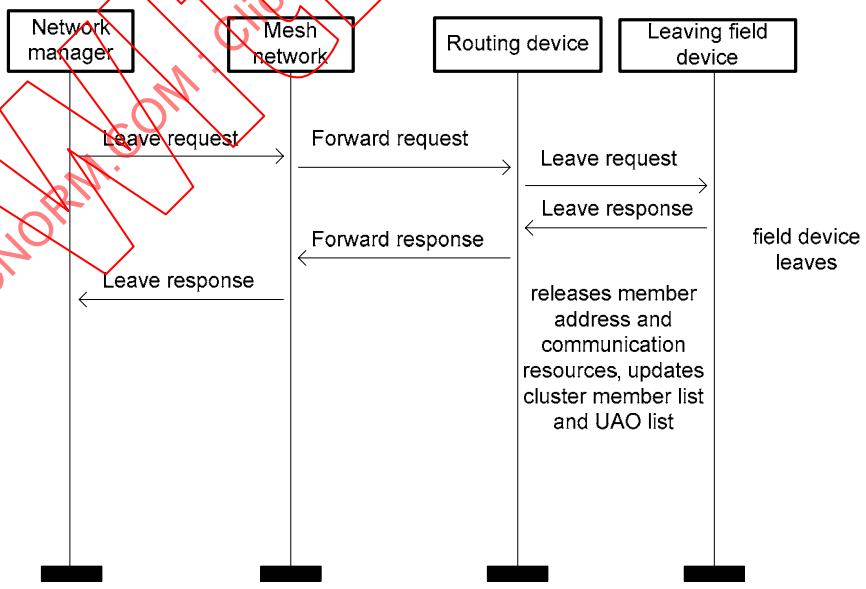


Figure 17 – Passive leaving process of field device

6.5.5 Cluster message aggregation

WIA-PA provides a packet aggregation option. If this option is chosen, cluster head will aggregate the packets from its cluster members to one packet before relaying. The combined packet will be de-aggregated in gateway.

The aggregation function is implemented by the aggregation object (AGO) in cluster head, and the de-aggregation function is implemented by the de-aggregation object (DAGO) in gateway. The operation parameters of AGO is configured by the aggregation management object in network manager. The attributes of aggregation object is shown in Table 2.

Table 2 – Aggregation Object Attribute

Name	Data type	Valid range	Default value	Description
MAX_PDU_SIZE	Unsigned16	0~0xFFFF	40	The largest length of aggregation message
AGG_PERIOD	Usigned8	0~0xFF	1	Aggregation period
NOTE The unit of message length is byte, the unit of aggregation period is second, the message aggregation object configuration uses GET and SET primitive.				

The aggregation configuration process is listed as follows.

- After the network is established, the configurator in host device first sends the configuration information to all UAOs in field device, and the aggregation management object in the network manager.
- After receiving the configuration information, UAOs in field device begin to establish intra-cluster data VCR with AGO.
- In the process of intra-cluster data VCR establishment, AGO establishes the corresponding inter-cluster data VCR with DAGO.
- After all data VCRs are established, the aggregation management object in the network manager configures the aggregation period of AGO to the minimal update rate of aggregating UAOs.
- AGO and DAGO begin to run.

6.5.6 An example for aggregation configuration process

The example below illustrates the aggregation configuration process. The example is based on a network which comprises two field devices and one routing device. As shown in Figure 18, there are two UAOs called a1 and a2 reside in device A, and two UAOs called b1 and b2 reside in device B.

A configurator that resides in the host device is responsible to configure the data update rate of each UAO. In this example, the data update rate of a1 and a2 are configured to 1 second, and the data update rate of b1 and b2 are configured to 4 seconds. To illustrate the process of packet aggregation, the packets from a1, b1 are configured to be aggregated and the packets from a2, b2 are configured to be aggregated.

There are two aggregation objects, AGO1 and AGO2, reside in the routing device, which are responsible to aggregate packets from a1, b1 and a2, b2. There are two de-aggregation objects, DAGO1 and DAGO2, reside in gateway, which are responsible to de-aggregate data from AGO1 and AGO2 in routing device. Network manager is responsible to configure the aggregation period according to the update rate of a1, a2, b1, b2 through an aggregation management object.

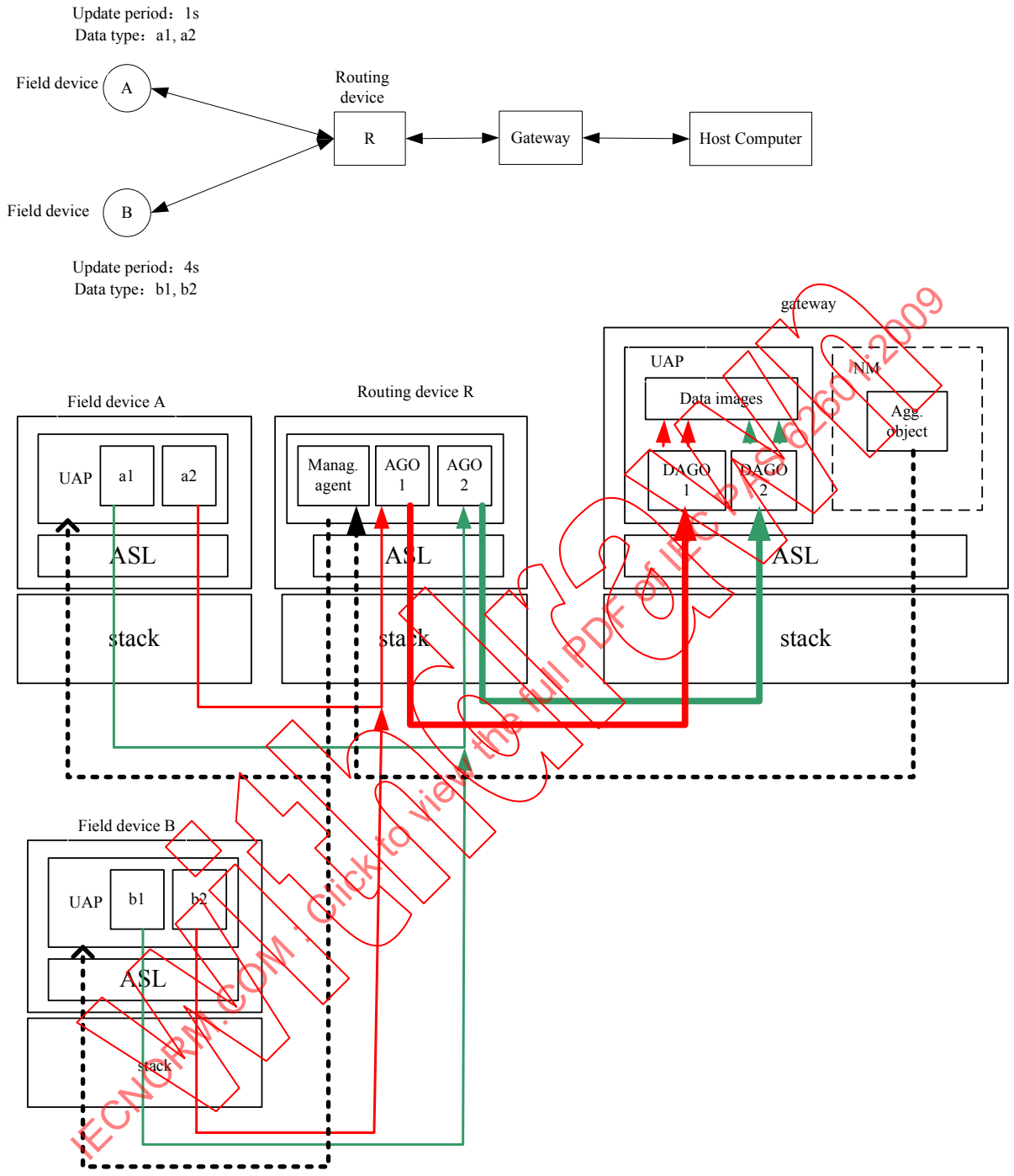


Figure 18 – An example of packet aggregation

The packet aggregation process is listed below.

- After the network is established, the configurator in host device first sends the configuration information to a1, a2, b1, b2, and the aggregation management object in the network manager. The data update rate of a1, a2 are set to 1 second, and the data update rate of b1, b2 are set to 4 seconds.
- After receiving the configuration information, a1, a2, b1, b2 start up to establish intra-cluster data VCR. In this process, four intra-cluster data VCRs, A_vcr1, B_vcr1, A_vcr2, B_vcr2 are established. The endpoints of A_vcr1 are a1 and AGO1. The endpoints of B_vcr1 are b1 and AGO1. The endpoints of A_vcr2 are a2 and AGO2. The endpoints of B_vcr2 are b2 and AGO2.

- In the process of intra-cluster data VCR establishment, two inter-cluster data VCRs, VCR1 and VCR2, are established between AGO1, DAGO1 and AGO2, DAGO2. The endpoints of VCR1 are AGO1 and DAGO1. The endpoints of VCR2 are AGO2 and DAGO2.
- After all data VCRs are established, the aggregation management object in the network manager starts up to configure the aggregation frequency. For the minimal update rate of a1 and b1 is 1 second, the aggregation frequency of AGO1 is configured to 1 second. For the minimal update rate of a2 and b2 is 1 second, the aggregation frequency of AGO2 is configured to 1 second.

6.5.7 Allocation of communication resource for field device

Routing devices allocate time slots and channels to field device from its avaliable communication resource, when the field device requests to establish a intra-cluster data VCR.

6.5.8 Cluster performance monitoring

The field device should report its health information to routing device periodically.

6.6 Device management

6.6.1 General

Device management component exists in DMAP of all devices. It acts as a server and is responsible to response all operation request initiated by network management component in network manager and network management agent component of cluster head.

6.6.2 Device attributes management

Device attributes are listed in Table 3. The attributes can be expanded later easily.

Table 3 – Device Attribute

Name	Data type	Valid range	Classification	Access ibility	Description
Long address	Unsigned64	N/A	Constant	R	64 bits global unique address As for the Device_type field as defined in 6.4.2 and 6.5.2: 0: gateway 1: routing device 2: field device
Short address	Unsigned16	0~0xfffe	static	R/W	Shore address of WIA-PA device, high 8bits as cluster address, low 8bits intra-cluster address
Manufacturer_ID	VisibleString	N/A	Static	R	Identifier of manufacturer ,set by manufacturer
Device_Serial_Number	Unsigned64	N/A	Constant	R	Device serial number, set by manufacturer
Power_Supply_Status	Unsigned8	0-10	Dynamic	R/W	0:on-line supply 1-10: indicates the power level
Device_state	Unsigned8	0-2	Static	R/W	Device state: 0=Inactive 1=Active 2=Failed

Name	Data type	Valid range	Classification	Access ibility	Description
Join_Status	Unsigned8	0-2	Dynamic	R	0=not joined/connected 1=attempting to join/connect 2=unsecure connection 3=full connection
Restart_Count	Unsigned16	0-0xffff	Dynamic	R	The number of device restart, restart cause maybe battery instead, software restart, update, connect error, hardware failure
Uptime	Unsigned32	0-0xffffffffffffffff	Dynamic	R	Time from the last restart, count by second
Device_Memory_total	Unsigned32	N/A	Constant	R	Total memory in a device
Device_Used_Memory	Unsigned32	N/A	Dynamic	R	Memory used by device in bytes
UTC_time	Unsigned32	N/A	Dynamic	R	Present time
Clock_Master_role	Boolean	0-1	Dynamic	R/W	The device is clock master or not: 0=no 1=yes
Clock_update	Unsigned32	0-0xffffffff	Dynamic	R	The last adjustment of clock in second
Firm_ware_version	Unsigned32	0-0xffffffff	Dynamic	R/W	The version of firmware loaded in the device
VCR_Table	Array of VCR structure	N/A	Dynamic	R/W	All the VCR used

6.6.3 Layer management

Every layer management is visited through nLME-SAP. See management services in each layer.

6.6.4 Security management

See the security management in Clause 11.

6.6.5 VCR mapping management

A UAO in application layer are binded with a data VCR. This mapping relation is stored in a VCR mapping table. The VCR mapping table is maintained by VCR mapping object. A UAO should first get its corresponding VCR_ID before startup a data transmission.

6.7 Management Information Base (MIB)

6.7.1 Attribute and Method

Items stored in MIB are called attributes. These attributes can be accessed by DMAP and network manager.

Attributes are classified to provide guidance regarding their change frequency and their behaviour when a reset command is issued. Attributes are classified as

- Constant,
- Static,

- Dynamic.

A constant attribute is unchangeable throughout time. An example of a constant attribute is the serial number of a wireless device. Constant attributes are set when the device leaves the factory. The values of these attributes shall be preserved when the device undergoes a warm restart / power-fail.

A static attribute changes its value infrequently. An example of a static attribute is an alarm limit. The values of these attributes shall be preserved when the device undergoes a warm restart / power-fail.

A dynamic attribute may be changed spontaneously by the object and without external stimulation from the wireless network. A dynamic attribute is not required to survive when the device goes through a warm restart / power-fail or when the device resets to factory defaults.

6.7.2 MIB services

The operations of reading and writing the attributes use nLME-GET and nLME-SET service.

6.8 Interaction with plant operations or maintenance personnel

Network manager should provide an interface that allows plant operations and maintenance personnel to observe and control the functioning of the network and devices. However, the definition of this interface is out of the scope of the current WIA-PA PAS.

7 Physical Layer

WIA-PA PHY is based on IEEE 802.15.4 physical layer.

8 Data link layer

8.1 General

The WIA-PA Data Link Layer (DLL) functionality is designed to guarantee communication between network devices in a reliable, safe and real-time way. The WIA-PA DLL is fully compliant to the IEEE 802.15.4 superframe structure and extends it. WIA-PA network allows devices according with the IEEE 802.15.4 specification to join WIA-PA network. The WIA-PA DLL supports certain key functions, including frequency hopping mechanism, retries, TDMA and CSMA hybrid channel access mechanism. These mechanisms are used to guarantee the reliability and real-time transmission in communication. The WIA-PA DLL is designed to use Message Integrity Code (MIC) mechanism and encryption technology to guarantee the integrity and confidentiality in the communication.

8.2 Stack structure

The WIA-PA Data Link Layer function is designed to leverage the IEEE 802.15.4 to meet the requirements of process automation. The WIA-PA DLL is based on IEEE 802.15.4 MAC layer and extends it. DLL stack structure is shown in Figure 19.

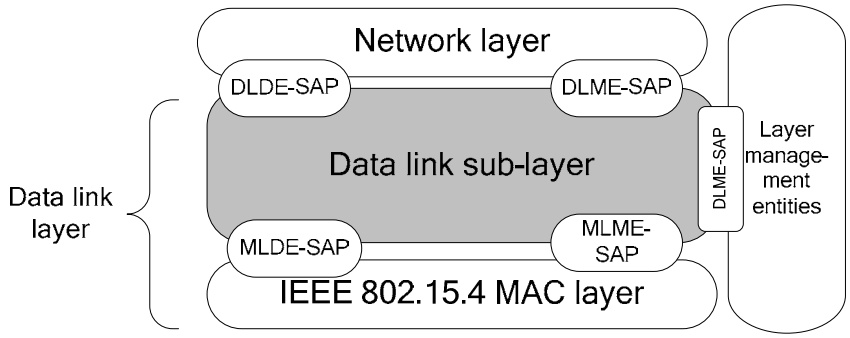


Figure 19 – WIA-PA data link layer stack structure

The WIA-PA DLL includes the following parts.

- The IEEE 802.15.4 MAC. This handles the mechanisms of sending and receiving individual data frames.
- The Data Link Sub-layer. This handles the aspects of mesh network topology, link and communication resource.

NOTE The content of IEEE 802.15.4 MAC is not stated in this PAS. Unless specifically noted, DLL means the Data Link Sub-layer in this PAS.

The DLL provides an interface between the NL and the MAC. The DLL conceptually includes a management entity called the DLME. This entity provides the service interfaces through which layer management functions may be invoked. The DLME is also responsible for maintaining a database of pertaining to the DLL. This database is referred to as the DLL-MIB.

Figure 20 depicts the components and interfaces of the DLL.

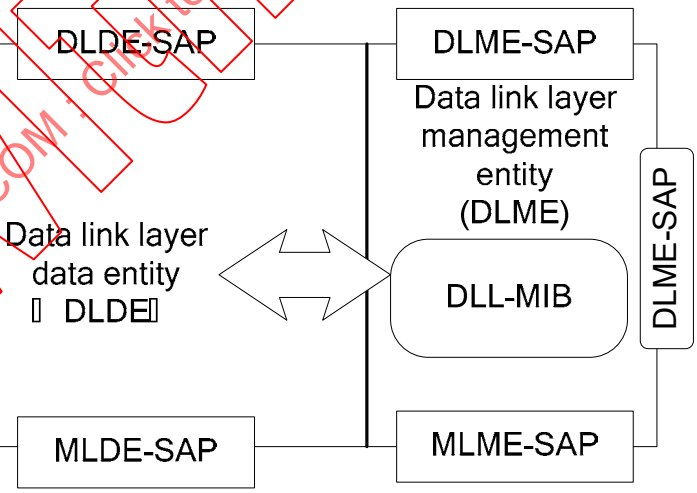


Figure 20 – WIA-PA DLL reference model

The DLL provides two services, accessed through two SAPs:

– the DLL data service, accessed through the DLL data entity SAP (DLDE-SAP), and

– the DLL management service, accessed through the DLL management entity SAP (DLME-SAP).

8.3 Functional description

8.3.1 General

In WIA-PA, the main function of DLL is allocating communication resource between competitive users in order to avoid collision, improve throughput and bandwidth utilization. The main concepts of DLL are timeslot, superframe and link.

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

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

- 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 link and the receiving link shall only allow designated device to exchange packets.

8.3.2 Compatibility and coexistence

WIA-PA network must consider the following compatibility and coexistence strategies.

- WIA-PA network is fully compliant to the IEEE 802.15.4 superframe structure and extends it. WIA-PA network allows devices according with the IEEE 802.15.4 specification to join WIA-PA network.

- The WIA-PA DLL together with the network manager realizes coexistence with other users of frequency band. The WIA-PA DLL incorporates several strategies that are used simultaneously to optimize coexistence:

  – timeslot communication;

  – low duty-cycle;

  – multi-channel;

  – self-adaptive frequency hopping;

  – collision avoidance.

8.3.3 Time synchronization

In order to guarantee the reliability of the TDMA communication mode, devices in a network must synchronize with the time source. Devices in WIA-PA network only synchronize with the Gateway in order to distinguish the event occurrence sequence.

Time synchronization is divided into two parts:

- In the mesh network, Gateway is the time source. All routing devices synchronize with Gateway.
- In the star network, every routing device is the time source. All field devices synchronize with their routing devices.

NOTE 1 For details of time synchronization, see IEEE 802.15.4.

NOTE 2 In this PAS, the maximum synchronization error should be less than 10 % of the basic timeslot length in the maximum superframe duration.

8.3.4 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 delayed. The timeslot length of WIA-PA DLL is fully compliant to the timeslot length of IEEE 802.15.4.

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

8.3.5 WIA-PA superframe

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

NOTE See IEEE 802.15.4 for more information of the IEEE 802.15.4 superframe structure.

The WIA-PA superframe structure based on IEEE 802.15.4 superframe structure is shown in Figure 21.

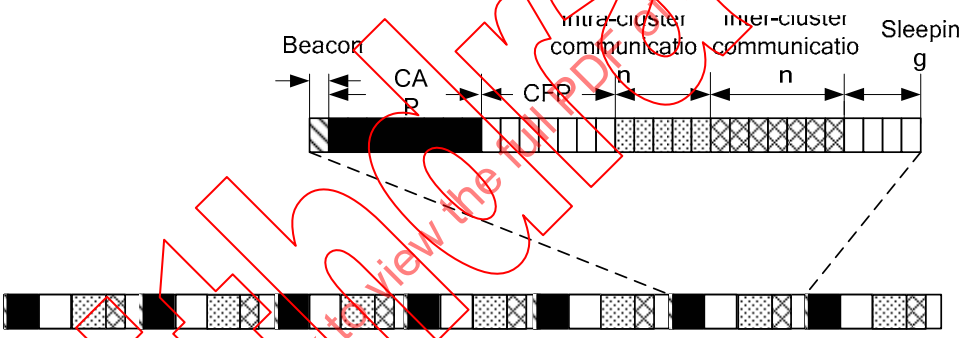


Figure 21 – WIA-PA superframe

- The CAP period defined in the IEEE 802.15.4 superframe is used for device joining, intra-cluster management and retry in the WIA-PA superframe.
- The CFP period defined in the IEEE 802.15.4 superframe is used for mobile devices and intra-cluster communication in the WIA-PA superframe.
- The inactive period defined in the IEEE 802.15.4 superframe is used for intra-cluster communication, inter-cluster communication and sleeping in the WIA-PA superframe.

The network manager is responsible for generating a WIA-PA superframe.

Because the inactive period defined in the IEEE 802.15.4 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 32 timeslots. The duration of WIA-PA superframe is as 2^N (N is a natural integer) times as the WIA-PA basic superframe duration.

Supposing that there have three routing devices, R1, R2 and R3 are in WIA-PA network. The superframe lengths of R1, R2 and R3 are respectively one, two and four WIA-PA basic superframe duration(s), shown in Figure 22.

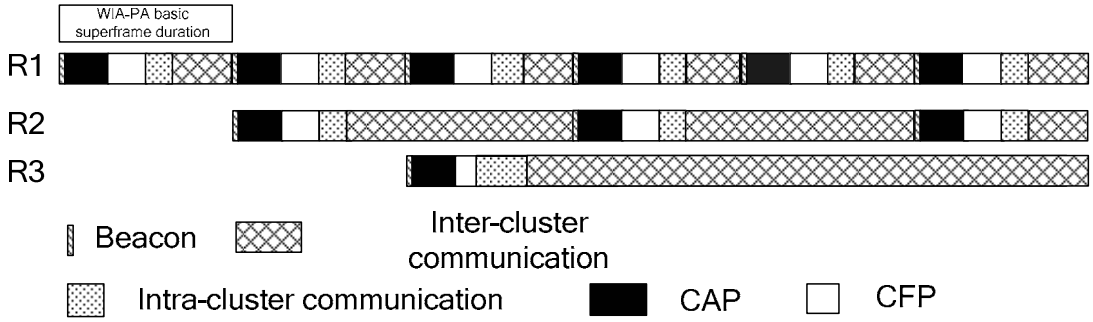


Figure 22 – R1, R2 and R3 superframe structures

8.3.6 Frequency hopping

Frequency hopping in WIA-PA includes two mechanisms: AFD and TH.

Adaptive Frequency Diversity (AFD): irregularly change communication channel according to actual channel condition. That is to say, bad channel condition, which can be measured with packet drop rate or resend time, triggers the operation of changing channel.

Timeslot Hopping (TH): regularly change of transmit/receive frequency per timeslot to combat interference and fading.

In the WIA-PA superframe, Beacon, CAP and CFP use the same channel in same superframe cycle, and use AFD mechanism in different superframe cycle; timeslot hopping is performed during inter- and intra-cluster communication in the inactive period.

WIA-PA network supports timeslot hopping. The hopping sequence is configured by the network manager. The structure is: <timeslot 1, channel 1> <timeslot 2, channel 2>… <timeslot i, channel i>.

The specific hopping mechanisms in WIA-PA are shown in Table 4.

Table 4 – Hopping mechanisms

IEEE 802.15.4	WIA-PA	Basic MAC mechanism	Hopping mechanism
Beacon	Beacon	TDMA	AFD
CAP	CAP	CSMA + FDMA	
CFP	CFP	TDMA + FDMA	
Inactive	Intra-cluster period	TDMA + FDMA	
	Inter-cluster period	TDMA	TH
	Sleeping	------	------

8.3.7 Communication resource allocation

The communication resources include timeslots and channels. The communication resource allocation should consider both.

The process of communication resource allocation includes:

- In the mesh network, the network manager allocates a block of communication resources to the routing devices. These resources include:

– resources used for communication between routing devices in the mesh network;

– resources used for field devices.

- In the star network, the routing device distributes part of communication resources allocated by network manager to its field devices.

An example of the resource allocation is shown in Figure 23.

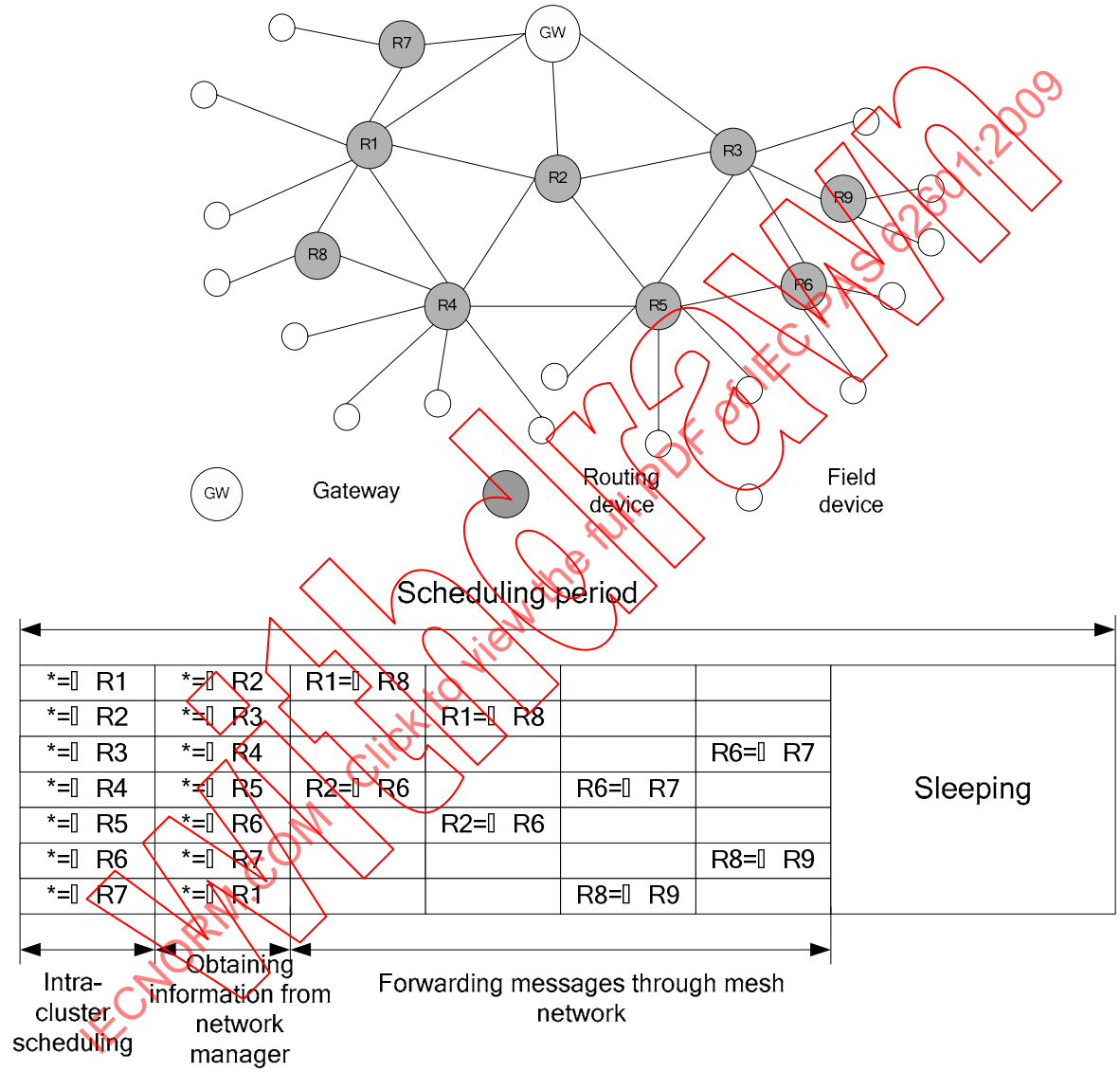


Figure 23 – An example of resource allocation

8.3.8 DLPDU priority and scheduling rules

There are four priority levels of DLPDUs defined:

- Command (highest priority)

  Any packet containing a payload with network-related diagnostics, configuration, or control information shall be classified with a priority of “Command”.

- Process data

Any packet containing process data shall be classified as secondary priority level “Process Data”.

- Normal

DLPDUs not meeting the criteria for “Command”, “Process Data”, or “Alarm” shall be classified as “Normal” priority.

- Alarm (lowest priority)

Packets containing only alarm and event payload shall assume a priority of “Alarm”. Devices shall buffer no more than one DLPDU having “Alarm” priority.

Where there are multiple packets that can be transmitted in one timeslot, the scheduling rules shall be used to select the packet. These scheduling rules are:

- allocating channels to the beacon and inactive period in priority;
- allocating timeslots to the devices with fastest update rate in priority;
- allocating resources to the packet with earliest generating time in multi-hop situation in priority;
- allocating resources to the highest priority packet prior to other packets.

8.3.9 Retry strategy

The network manager in the mesh network and the routing devices in the star network should allocate some timeslots for retries.

This PAS supports the following retry strategies.

- If first transmission fails and resources are enough, retry is executed over the same channel.
- If first retry fails and resources are enough, retry is executed over another channel.
- If second retry fails and resources allocated to redundant path exist, retry is executed over this redundant path.
- If third retry fails, retries are executed in the CAP period of next superframe cycle until the retry times reach the retry threshold. Then the frame is discarded.

Retry supports the AFD mechanism.

NOTE The number of retry timeslots is bounded by the constant “aMaxFrameRetries” in the IEEE STD 802.15.4: 2006 MAC PIB.

8.3.10 Subnet discovery

DLL provides specific services for subnet discovery. For details see 8.6.

8.3.11 Management service

WIA-PA DLL provides management services to the upper layer and DMAP with DLME-SA. The services include network joining and leaving, resource allocation and operations of management database attributes.

8.3.12 Radio link control and quality measurement

Link quality information may be accumulated at the DLL and reported through the DMAP. Control of the emission power level of field devices is also supported.

Generally, the DLL may be configured to accumulate the following types of performance data per neighbour.

- Received signal strength indication (RSSI)
- Link quality indication (LQI)
- Count of ACKs
- For transmission, counts of attempted messages, backoffs and packet errors
- For reception, counts of CRC failures

8.3.13 Security

WIA-PA DLL is designed to use Message Integrity Code (MIC) mechanism and encryption technology to guarantee the integrity and confidentiality in the communication. See the Clause 11 for details.

8.3.14 DLL state machine

The WIA-PA DLL state machine is shown in Figure 24.

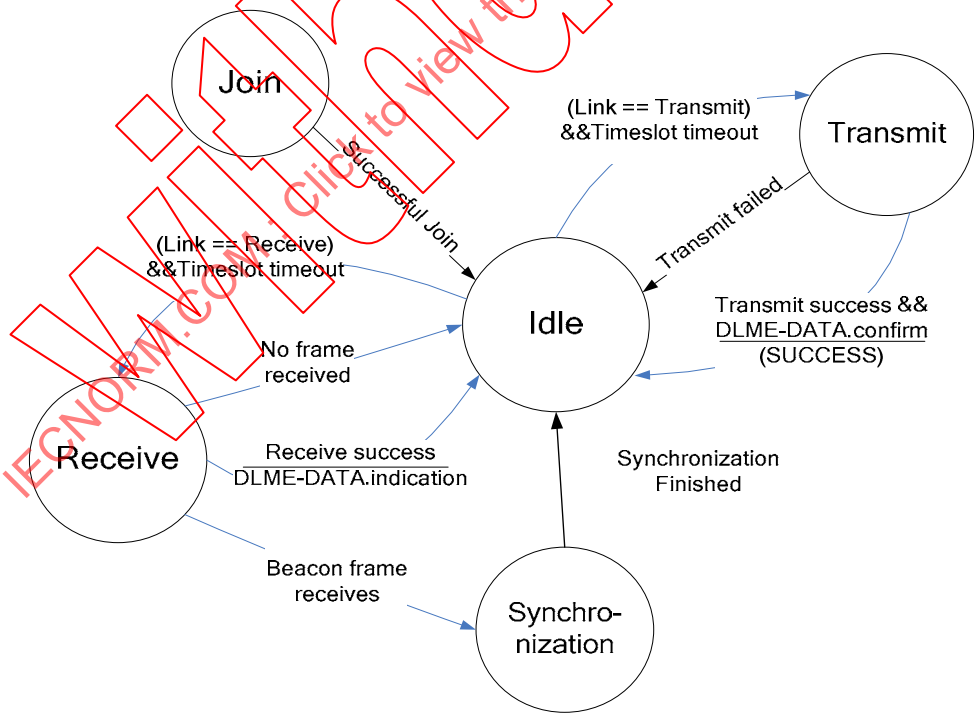


Figure 24 – DLL state machine

a) Join

This state handles the joining procedure of a device.

After a device joining the network, DLL enters “Idle” state.

b) Idle state

The following transitions can occur while in “Idle” state.

- When the timeslot arrives, DLL enters “Transmit” state or “Receive” state according to the link options (transmitting or receiving).
- When receiving management command frames and primitives, DLL enters “Management” state.

c) Transmit state

When the timeslot arrives and the link option is transmit link, DLL enters the “Transmit” state. The following results can occur while in the “Transmit” state.

- Successful propagation of a frame with broadcast/multicast destination address occurs as soon as the frame is transmitted. After that the frame’s buffer is released.
- Successful propagation of a frame with unicast destination address occurs when a validated, successful confirm is received from the local MAC layer. This indicates that message propagation has completed successfully and the frame’s buffer is released.
- If a response with an error or no response is received, then the frame will be retried.

d) Receive state

The functions of the “Receive” state are receiving, checking and processing the frame.

The following transitions can occur while in the “Receive” state.

- If there is no frame received, DLL will evaluate the link and return to the “Idle” state.
- If a beacon frame is captured, then the “Synchronization” state is entered.
- Upon successful frame reception, DLL will process the frame, and then the device returns directly to the “Idle” state.

e) Synchronization

In this state, time synchronization is executed after receiving a beacon frame. Then DLL enters “Idle” state.

8.4 DLL frame formats

8.4.1 General frame format

Table 5 – General frame format

<table>
<tr><td colspan="4">PPDU</td></tr>
<tr><td rowspan="3">PPDU header</td><td colspan="3">MPDU</td></tr>
<tr><td rowspan="2">IEEE 802.15.4 MAC header(MHR)</td><td colspan="2">DPDU</td></tr>
<tr><td>DLL frame control</td><td>DLL payload</td></tr>
</table>

The DLL general frame format is illustrated in Table 5. The DLL frame format is composed of:

- IEEE 802.15.4-2006 MAC Header (MHR). See IEEE 802.15.4.
- DLL frame control.
- DLL payload.

NOTE The content of security can see the “DLL security” clause for more information.

The WIA-PA DLL frame control is shown in Table 6.

Table 6 – DLL frame control

Bits: 0	1	2	3~4	5~7
Frame type 0=Data 1=Command	Clock recipient 0=No 1=Yes	Security enable 0= Disable 1= Enable	MIC options 0=MIC-32 1=MIC-64 2~3=reserved	reserved

8.4.2 Date frame format

Table 7 – Date frame format

Octets: 1	variable	0/4/8
DLL frame control	Data payload	MIC

The frame type of data frame is 0.

8.4.3 MAC beacon format

WIA-PA network uses IEEE 802.15.4-2006 MAC beacon payload to distribute superframe information. For the MAC beacon frame format, see 7.2.2.1 in IEEE 802.15.4:2006.

The beacon payload is shown in Table 8.

Table 8 – Beacon payload

Octets: 1	2~7	9
ClusterID	Absolute timeslot number	Channel used to transmit next beacon

- Channel map array: This is an array of 64 bits starting from the least significant bit. Each bit corresponds to a channel. If the bit is set the corresponding channel is in use. Its specific structure is decided by the physical layer. The first 27 bits in channel map array, numbered from bit 0 to bit 26, are assigned to IEEE 802.15.4. Among them:
  - bit 0 to bit 15 are assigned to the 16 channels of 2,4 G frequency band;
  - other bits are assigned according to the frequency band in practical environment. If the amount of channels is less than 64, unused bits are filled with 0.

8.4.4 Command frame format

8.4.4.1 General command frame format

Table 9 – General command frame format

Octets: 1	2	variable
DLL frame control	Command frame identifier	Command payload

8.4.4.2 DLL command frame

The DLL command frames are used for resource allocation and state maintenance.

Table 10 – DLL command frame

Command frame identifier	Command name	User	Description
1	Link adding request	Network manager/ network manager agent	Request to add a link
2	Link update request	Network manager/ network manager agent	Request to update a link
3	Link release request	Network manager/ network manager agent	Request to release a link
4	Superframe adding request	Network manager/ network manager agent	Request to add a superframe
5	Superframe update request	Network manager/ network manager agent	Request to update a superframe
6	Superframe release request	Network manager/ network manager agent	Request to delete a superframe
7	Keep-alive notification	Network manager/ network manager agent	Indicating an existing device

8.4.4.3 Link adding request command frame

Link adding request command frame is used by the network manager to add a new link to gateway or a routing device, and also used by the network management agent on a routing device to add a new link to a field device. After receiving this command frame, the gateway, routing device or field device adds a record to its link table.

Link adding request command frame format is shown in Table 11.

Table 11 – Link adding request command frame format

Octets: 1	1	12
DLL frame control	Command frame identifier	Link table item

- Command frame identifier is 1.
- Link table item is shown in 8.7.3.2.

8.4.4.4 Link update request command frame

Link update request command frame is used by the network manager to update an existed link to gateway or a routing device, and also used by the network management agent on a routing device to update an existed link to a field device. After receiving this command frame, the gateway, routing device or field device updates a record to its link table.

Link update request command frame format is shown in Table 12.

Table 12 – Link update request command frame format

Octets: 1	1	12
DLL frame control	Command frame identifier	Link table item

- Command frame identifier is 2.
- Link table item is shown in 8.7.3.2.

8.4.4.5 Link release request command frame

Link release request command frame is used by the network manager to release an existed link to gateway or a routing device, and also used by the network management agent on a routing device to release an existed link to a field device. After receiving this command frame, the gateway, routing device or field device releases a record to its link table.

Link release request command frame format is shown in Table 13.

Table 13 – Link release request command frame format

Octets: 1	1	12
DLL frame control	Command frame identifier	Link table item

- Command frame identifier is 3.
- Link table item is shown in 8.7.3.2.

8.4.4.6 Superframe adding request command frame

Superframe adding request command frame is used by the network manager to add a new superframe to gateway or a routing device, and also used by the network management agent on a routing device to add a new superframe to a field device. After receiving this command frame, the gateway, routing device or field device adds a record to its superframe table.

Superframe adding request command frame format is shown in Table 14.

Table 14 – Superframe adding request command frame format

Octets: 1	1	11
DLL frame control	Command frame identifier	Superframe table item

- Command frame identifier is 4.
- Superframe table item is shown in 8.7.3.1.

8.4.4.7 Superframe update request command frame

Superframe update request command frame is used by the network manager to update an existed superframe to gateway or a routing device, and also used by the network management agent on a routing device to update an existed superframe to a field device. After receiving this command frame, the gateway, routing device or field device updates a record to its superframe table.

Superframe update request command frame format is shown in Table 15.

Table 15 – Superframe update request command frame format

Octets: 1	1	11
DLL frame control	Command frame identifier	Superframe table item

- Command frame identifier is 5.
- Superframe table item is shown in 8.7.3.1.

8.4.4.8 Superframe release request command frame

Superframe release request command frame is used by the network manager to release an existed superframe to gateway or a routing device, and also used by the network management agent on a routing device to release an existed superframe to a field device. After receiving this command frame, the gateway, routing device or field device releases a record to its superframe table.

Superframe release request command frame format is shown in Table 16.

Table 16 – Superframe release request command frame format

Octets: 1	1	11
DLL frame control	Command frame identifier	Superframe table item

- Command frame identifier is 6.
- Superframe table item is shown in 8.7.3.1.

8.4.4.9 Keep-alive notification command frame

Keep-alive notification command frame is used for connection maintenance between neighbour devices.

Keep-alive notification command frame format is illustrated in Table 17.

Table 17 – Keep-alive notification command frame format

Octets: 1	1
DLL frame control	Command frame identifier

- Command frame identifier is 7.

8.5 Data link layer data services

8.5.1 General

Data link layer data service access point (DLDE-SAP) supports the point-to-point transmission of DLPDUs between devices. The primitives supported by Data link layer data services include DLDE-DATA.request primitive, DLDE-DATA.confirm primitive and DLDE-DATA.indication primitive.

8.5.2 DLDE-DATA.request

DLDE receives the payload from network layer through DLDE-DATA.request primitive and adds it to the message queue of the DLL.

The semantics of the DLDE-DATA.request primitive is as follows:

DLDE-DATA.request (

SrcAddrMode,
SrcAddr,
DstAddrMode,
DstAddr,
Priority,
Type,
VCR_ID
PayloadLength,
Payload,
PayloadHandle

)

Table 18 specifies the parameters for DLDE-DATA.request.

Table 18 – DLDE-DATA.request parameters

Name	Data type	Valid range	Description
SrcAddrMode	Unsigned8	0~3	Mode of source address: 0=no address; 1=reserved; 2=16-bit short address; 3=64-bit long address.
SrcAddr	Unsigned16/64	0~65535 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	Decided by address mode	Mode of destination address: 0=no address; 1=reversed; 2=16-bit short address; 3=64-bit long address.
DstAddr	Unsigned16/64	0~65535 or (2^{64}-1)	Destination address, 16- or 64-bit.
Priority	Unsigned8	0~15	Priority of the payload.
Data type	Unsigned8	0 ~ 1	0= intra-cluster transmission; 1= inter-cluster transmission.

Name	Data type	Valid range	Description
VCR_ID	Unsigned16	0 ~ 65535	VCR ID of the payload.
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octets	------	Payload
PayloadHandle	Unsigned8	0 ~ 255	Handle allocated when call DLDE-DATA.request primitive.

8.5.3 DLDE-DATA.confirm

DLDE-DATA.confirm (

PayloadHandle,
Status

)

Table 19 specifies the parameters for the DLDE-DATA.confirm primitive.

Table 19 – DLDE-DATA.confirm parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0 ~ 255	Handle allocated when call DLDE-DATA.confirm primitive
Status	Unsigned8	0 ~ 15	Result of the data transmission of DLL: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

Table 20 – Status table

ID	Value	Description
0	SUCCESS	The requested operation was completed successfully. For a transmission request, this value indicates a successful transmission.
1	TRANSACTION_OVERFLOW	No enough space for storing transactions.
2	TRANSACTION_EXPIRED	Transaction has expired and its information is discarded.
3	NO_ACK	No acknowledgement message was received after *aMaxFrameRetries*.
4	CHANNEL_ACCESS_FAILURE	Can not transmit due to channel access failure.
5	UNAVAILABLE_KEY	No valid key in access control list.
6	FAILED_SECURITY_CHECK	Received packet makes a security checking error in security mode.
7	INVALID_PARAMETER	A parameter in the primitive is out of value range.
8	READ-ONLY	The parameter is read-only, can not be write
9	UNSOPPORTED-ATTRIBUTE	Unsupported PIB attribute in SET/GET request primitive.
10	NO-BEACON	A scan operation failed to find any network beacons.
11-15	reserved	--

8.5.4 DLDE-DATA.indication

DLDE-DATA.indication (

ScrAddrMode,
SrcAddr,
Type,
Priority,
PayloadLength,
Payload,
PayloadLinkQuality,
SecurityUse

)

Table 21 specifies the parameters for the DLDE-DATA.indication primitive.

Table 21 – DLDE-DATA.indication parameters

Name	Data type	Valid range	Description
SrcAddrMode	Unsigned8	0~3	The source address mode. This value can take one of the following values: 0=no address; 1=reserved; 2=16-bit short address; 3=64-bit long address.
SrcAddr	Unsigned16/64	0~65535 or (264-1)	Source address, using 64-bit long address only in device join process, and using 16-bit short address generally.

Name	Data type	Valid range	Description
Type	Unsigned8	0 ~ 1.	0=intra-cluster transmission; 1=inter-cluster transmission
Priority	Unsigned8	0~15	Priority of payload.
PayloadLength	Unsigned8	≤MaxMACFrameSize	Length of payload
Payload	Octets	--------	Payload
PayloadLinkQuality	Unsigned8	0~255	LQ value measured during reception of the DPDU. Lower values represent lower LQ.
SecurityUse	Unsigned8	0 ~ 1	Indicate whether received frame used security mode. If the security enable field of WDHR is 1, the value sets 1, else 0.

8.5.5 Time sequence of the DLL data service

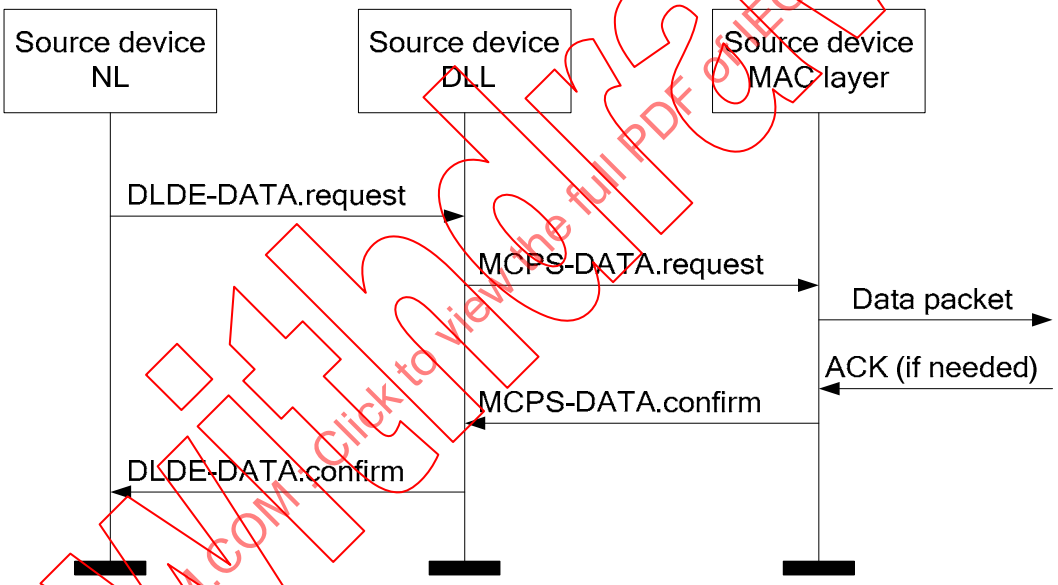


Figure 25 – Time sequence of the data service (source device side)

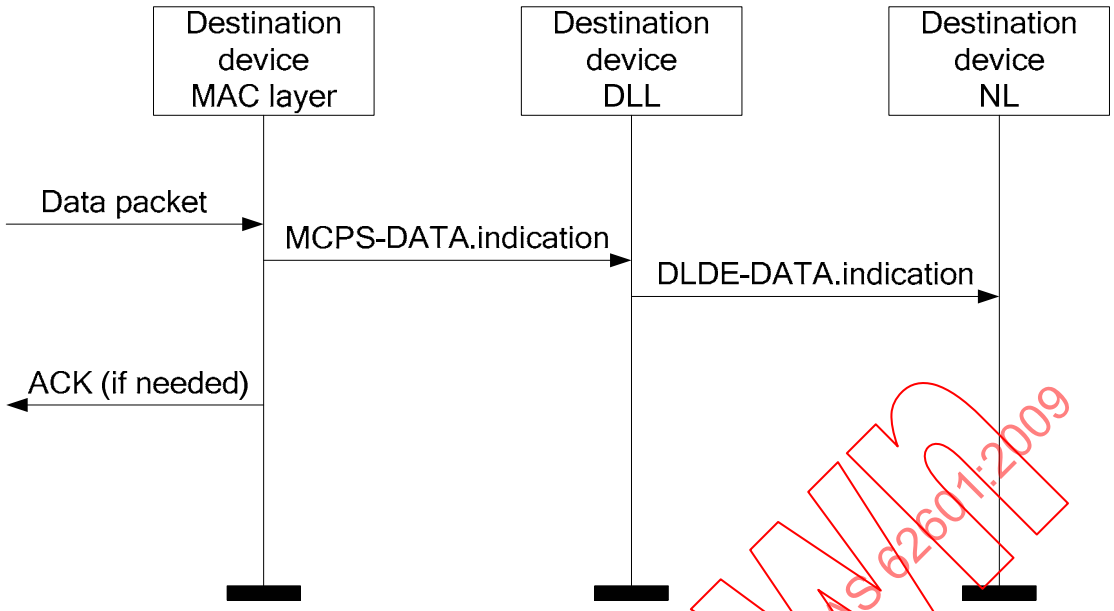


Figure 26 – Time sequence of the data service (destination device side)

Figure 25 and Figure 26 show the basic procedures of the frame sending and receiving. The DLDE-DATA.request primitive is generated by a local NLDE when a data NPDU is to be transferred to a peer NLDE. On receipt of the DLDE-DATA.request primitive, DLDE begins the transmission of the supplied DLPDU.

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

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

8.6 Data link layer management services

8.6.1 General

Upper layer uses DLL management entity service access point (DLME-SAP) to send management commands to the DLL. The DLL management services mainly include subnet discovery, device joining and leaving, resource allocation and DLL-MIB maintenance.

8.6.2 Sub-network discovery services

The DLL subnet discovery services are used to scan channels over a given list of communication channels. One device can use the subnet discovery services to measure channel energy, search cluster heads (or gateway) sending beacon frames within its communication scope. The DLL subnet discovery services provide DLME-DISCOVERY.request primitive and DLME- DISCOVERY.confirm primitive.

8.6.2.1 DLME-DISCOVERY.request primitive

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

DLME-DISCOVERY.request (

ScanChannels,
ScanDuration

)

Table 22 specifies the parameters for DLME-DISCOVERY.request primitive.

Table 22 – DLME-DISCOVERY.request parameters

Name	Data type	Valid range	Description
ScanChannels	Unsigned64	64-bit Bits-Map	First 27 bits are distributed to IEEE 802.15.4 channels of 2.4G frequency band; others are set up based on real field frequency channel (if less than 64-bit, the rest bits are filled with 0.)
ScanDuration	Unsigned8	0 ~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 of *ScanDuration* parameter.
NOTE The definition and value set of *aBaseSuperframeDuration* refer to parameters of PIB in the IEEE STD 802.15.4: 2006 specification.			

8.6.2.2 DLME- DISCOVERY.confirm primitive

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

DLME-DISCOVERY.confirm(

Status,
NetworkCount,
NetworkDescriptor

)

Table 23 specifies the parameters for DLME- DISCOVERY.confirm primitive.

Table 23 – DLME- DISCOVERY.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0~15	Scan results: SUCCESS; NO_BEACON; INVALID_PARAMETER.
NetworkCount	Unsigned8	0~255	The count of active network founded during scan.
NetworkDescriptor	list	0~ NetworkCount	Network Descriptor list of every founded network, refer to Table 24.

Table 24 – Network Descriptor list

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0~63	Logic channel used for joining, chosen from valid channels supported by PHY.
BeaconOrder	Unsigned8	0-15	The frequency sending beacon frame.
SuperframeOrder	Unsigned8	0-15	Active period length of the superframe
PermitJoining	Unsigned8	0/1	Whether routing device permits field device to join: 0= at least one device is permitted to join 1=no permit

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

The time sequence for subnet discovery is showed in Figure 27.

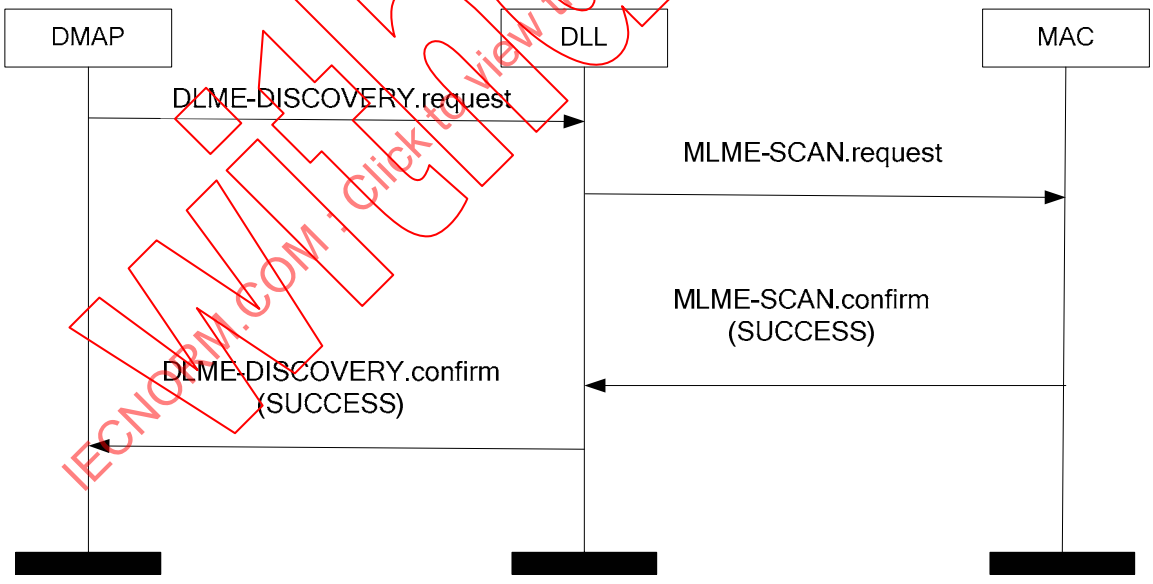


Figure 27 – Time sequence of subnet discovery

8.6.3 Device joining services

8.6.3.1 General

There are two cases for device joining services: (1) New field device joins the star network. (2) New routing device joins the mesh network. Device joining services provide DLME-JOIN.request primitive, DLME-JOIN.indication primitive, DLME-JOIN.response primitive, and DLME-JOIN.confirm primitive.

8.6.3.2 DLME-JOIN.request primitive

DLME-JOIN.request primitive is used for device joining a network (star or mesh).

DLME-JOIN.request (

LogicalChannel,
JoinAddr,
PhyAddr,
DeviceType

)

Table 25 specifies the parameters for DLME-JOIN.request primitive.

Table 25 – DLME-JOIN.request parameters

Name	Data type	Valid range	Description
LogicalChannel	Unsigned8	0~63	Logic channel used for joining, chosen from valid channels supported by PHY.
JoinAddr	Unsigned16	0 ~ 65535	The address of routing device that accepts joining request.
PhyAddr	Unsgned64	0 ~ (2^{64}-1)	Physical address of the new device waiting for joining.
DeviceType	Unsigned8	0 ~ 1	Type of the new device waiting for joining; 0=field device; 1=routing device

8.6.3.3 DLME- JOIN.indication primitive

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

DLME- JOIN.indication (

PhyAddr,
DeviceType

)

Table 26 specifies the parameters for DLME-JOIN.indication primitive.

Table 26 – DLME-JOIN.indication parameters

Name	Data type	Valid range	Decription
PhyAddr	Unsigned64	0 ~ (2^{64}-1)	Address of new device waiting for joining
DeviceType	Unsigned8	0 ~ 1	Type of device waiting for joining: 0=field device; 1=routing device

8.6.3.4 DLME-JOIN.response primitive

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

DLME-JOIN.response (

PhyAddr,
ShortAddr,
TimeSource,
Status

)

Table 27 specifies the parameters for DLME-JOIN.response primitive.

Table 27 – DLME-JOIN.response parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	0 ~ (2^{64}-1)	Address of device waiting for joining
ShortAddr	Unsigned16	0 ~ 65535	Short address allocated by the network manager to device waiting for joining
TimeSource	Unsigned8	0 ~ 1	Whether this device is set as time source: 0= not time source; 1=time source
Status	Unsigned8	0 ~ 1	Result of joining request 0=FAILURE; 1=SUCCESS.

8.6.3.5 DLME-JOIN.confirm primitive

DLME-JOIN.confirm primitive reports the joining result to the DLL upper layer.

DLME-JOIN.confirm (

ShortAddr,
Status

)

Table 28 specifies the parameters for DLME-JOIN.confirm primitive.

Table 28 – DLME-JOIN.confirm parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 ~ 65535	Short address allocated by network manager for device waiting for joining.
Status	Unsigned8	0 ~ 1	Result of joining request: 0=FAILURE; 1=SUCCESS.

8.6.3.6 Time sequence for device joining in the network

The joining process includes the processes of joining the mesh network for a routing device and joining the star network for a field device.

Time sequence for a routing device joining the mesh network is shown in Figure 28 and Figure 29.

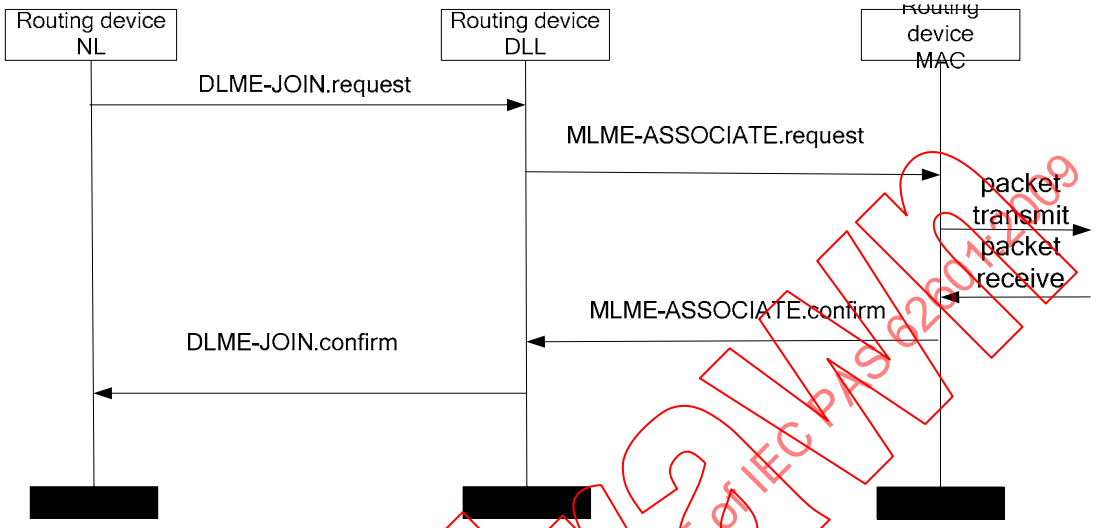


Figure 28 – Time sequence of routing device joining (routing device side)

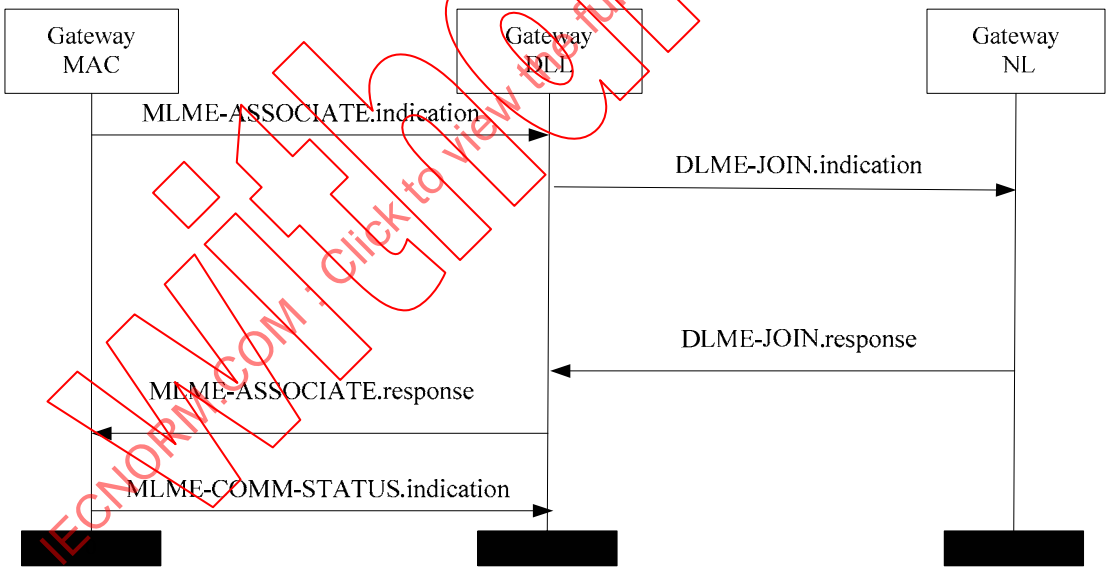


Figure 29 – Time sequence of routing device leaving (gateway side)

Time sequence for a field device joining the star network is shown in Figure 30 and Figure 31.

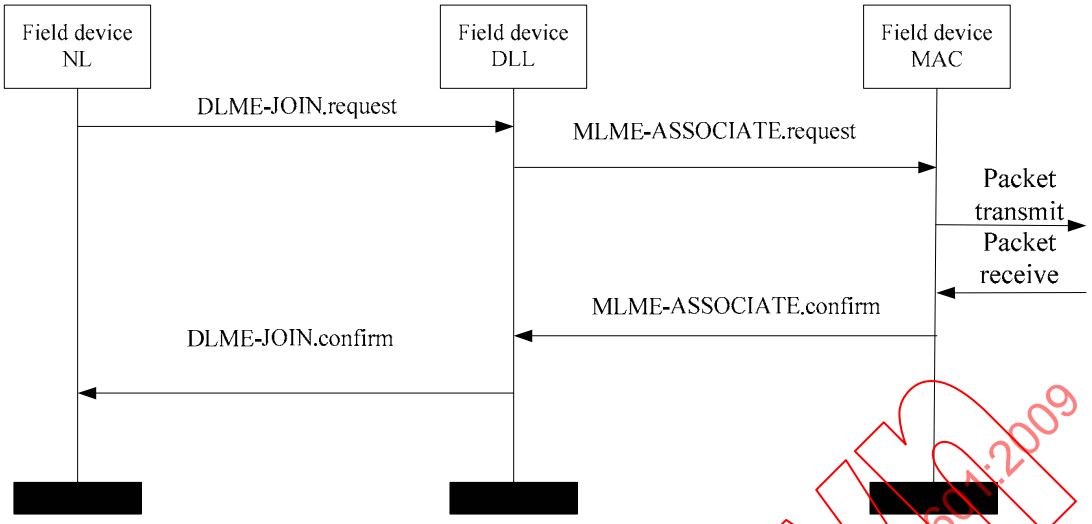


Figure 30 – Time sequence of field device joining (field device side)

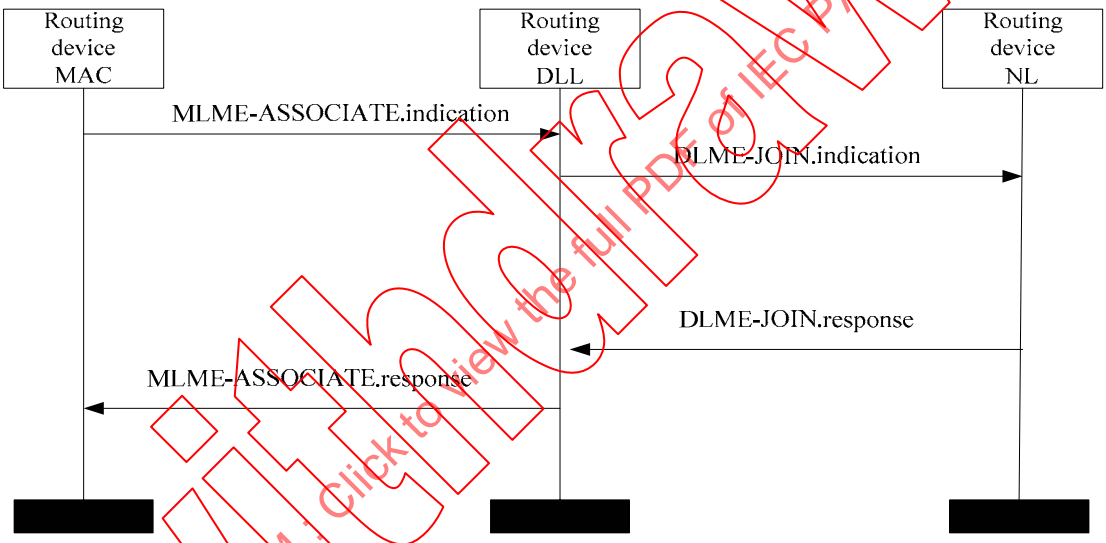


Figure 31 – Time sequence of field device joining (routing device side)

8.6.4 Device leaving services

8.6.4.1 General

There are two cases for device leaving services: (1) Field device leaves the star network (2) Routing device leaves the mesh network. Device leaving services provide DLME-LEAVE.request primitive, DLME-LEAVE.indication primitive, DLME-LEAVE.response primitive and DLME-LEAVE.confirm primitive.

8.6.4.2 DLME-LEAVE.request primitive

The DLME-LEAVE.request primitive is used for an existed field device to notify its routing device or for an existed routing device to notify the gateway of its intent to leave the network, which is called active leaving.

The DLME-LEAVE.request primitive is also used for gateway to instruct an existed routing device or for routing device to instruct an existed field device to leave the network, which is called passive leaving.

DLME-LEAVE.request (

ShortAddr

)

Table 29 specifies the parameters for the DLME-LEAVE.request primitive.

Table 29 – DLME-LEAVE.request parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 ~ 65535	The short address of the device asking for leaving.

8.6.4.3 DLME-LEAVE.indication primitive

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

DLME-LEAVE.indication (

ShortAddr

)

Table 30 specifies the parameters for the DLME-LEAVE.indication primitive.

Table 30 – DLME-LEAVE.indication parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0 ~ 65535	The short address of the device asking for leaving.

8.6.4.4 DLME-LEAVE.confirm primitive

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

DLME-LEAVE.confirm (

Status

)

Table 31 specifies the parameters for the DLME-LEAVE.confirm primitive.

Table 31 – DLME-LEAVE.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~ 1	Result of leaving request: 0=FAILURE; 1=SUCCESS.

8.6.4.5 Time sequence for device leaving

8.6.4.5.1 Time sequence for routing devices leaving the mesh network

Routing devices connect to both the mesh network and the star network. Therefore, the routing device will inform its leaving to both gateway and field devices.

- Active leaving: routing device preparing to leave will transmit a request to gateway. After acknowledged by gateway, it will leave. Meanwhile, routing device will inform its leave to field devices.

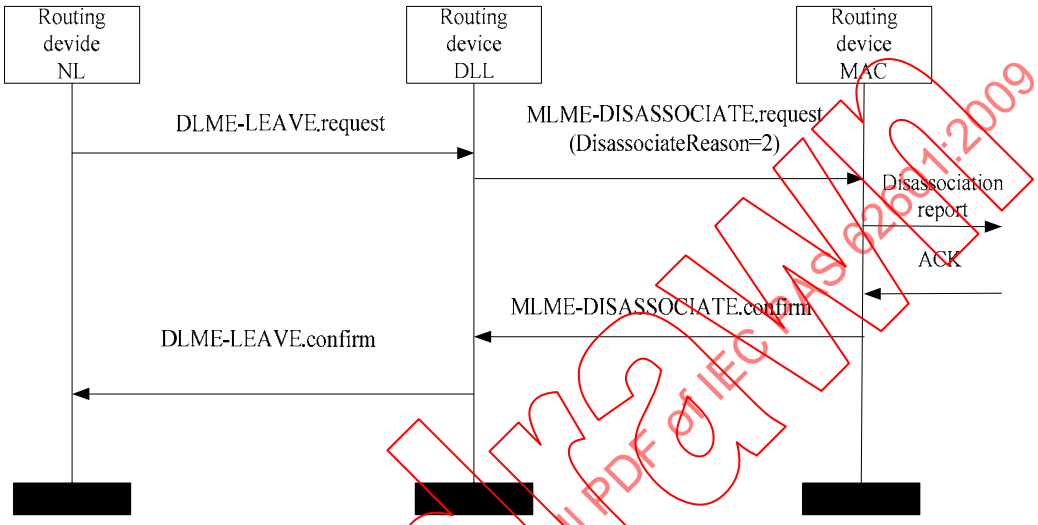


Figure 32 – Time sequence of routing device active leaving (routing device side)

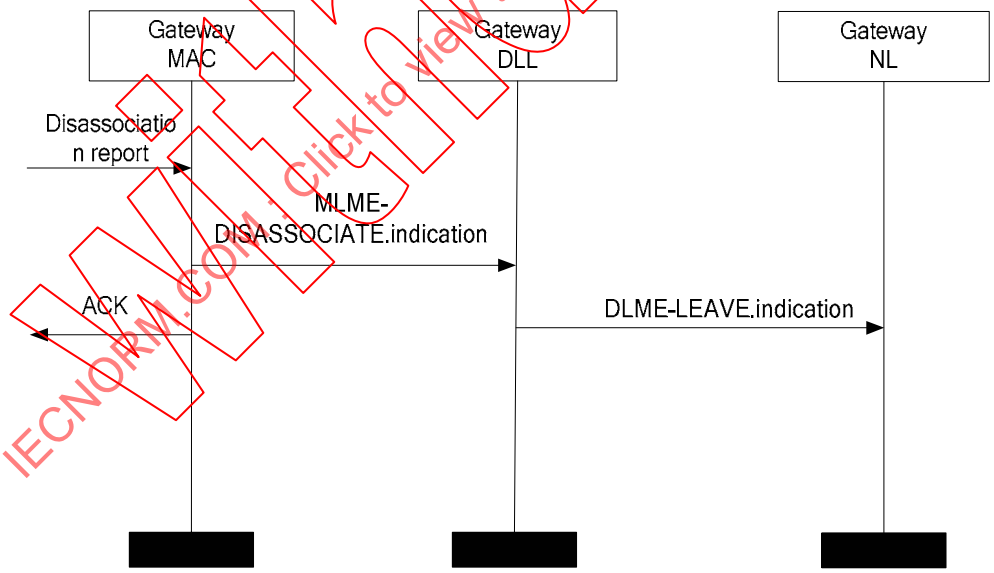


Figure 33 – Time sequence of routing device active leaving (gateway side)

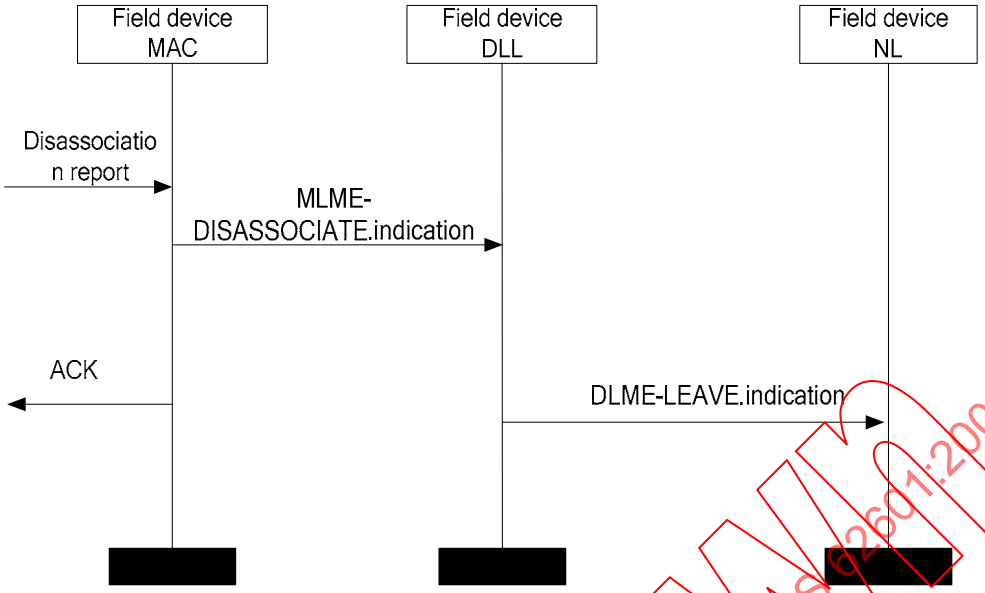


Figure 34 – Time sequence of routing device active leaving (field device side)

- Passive leave: if gateway asks routing devices to leave, it will transmit a leaving request to the routing device. After returning ACK to gateway, the routing device will leave the network. Meanwhile, the routing device will inform its leaving to the field devices.

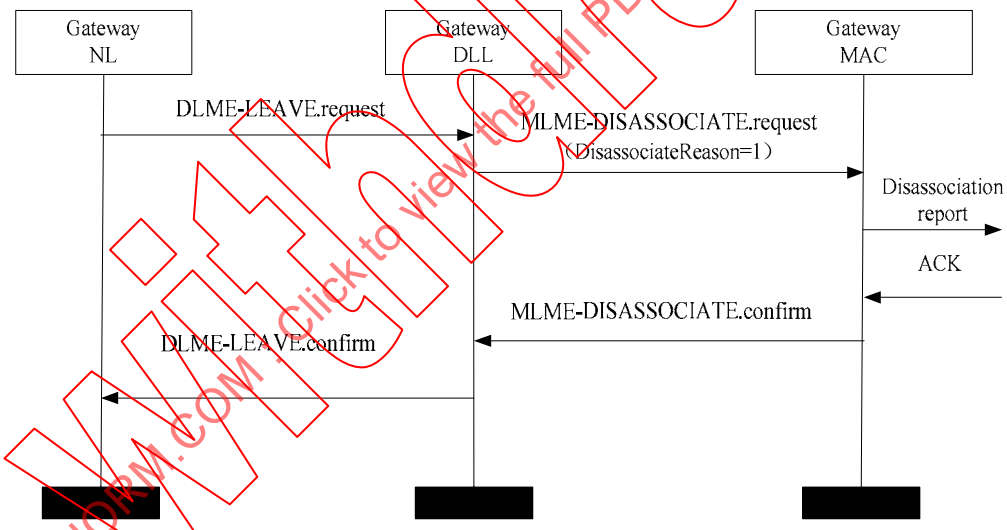


Figure 35 – Time sequence of routing device passive leaving (gateway side)

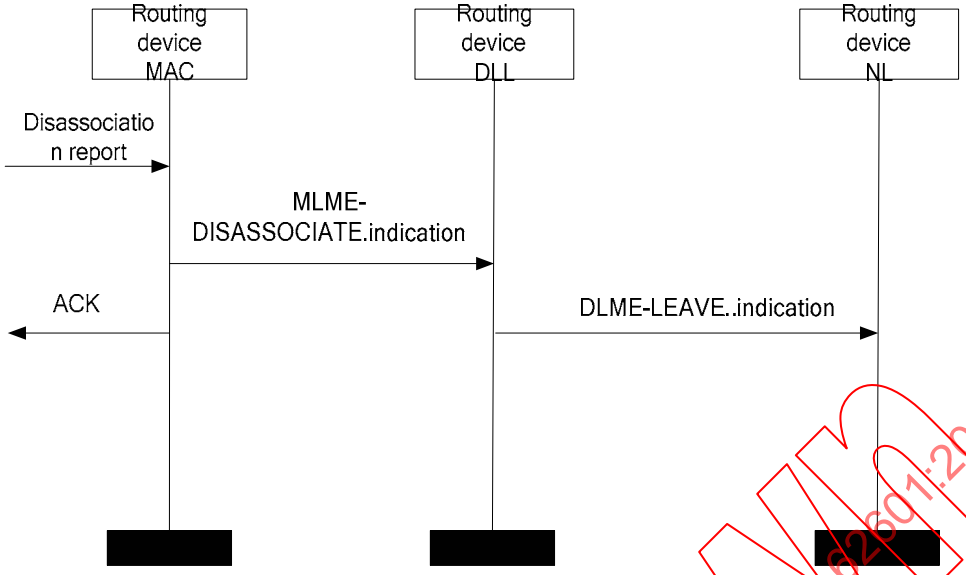


Figure 36 – Time sequence of routing device passive leaving (routing device side)

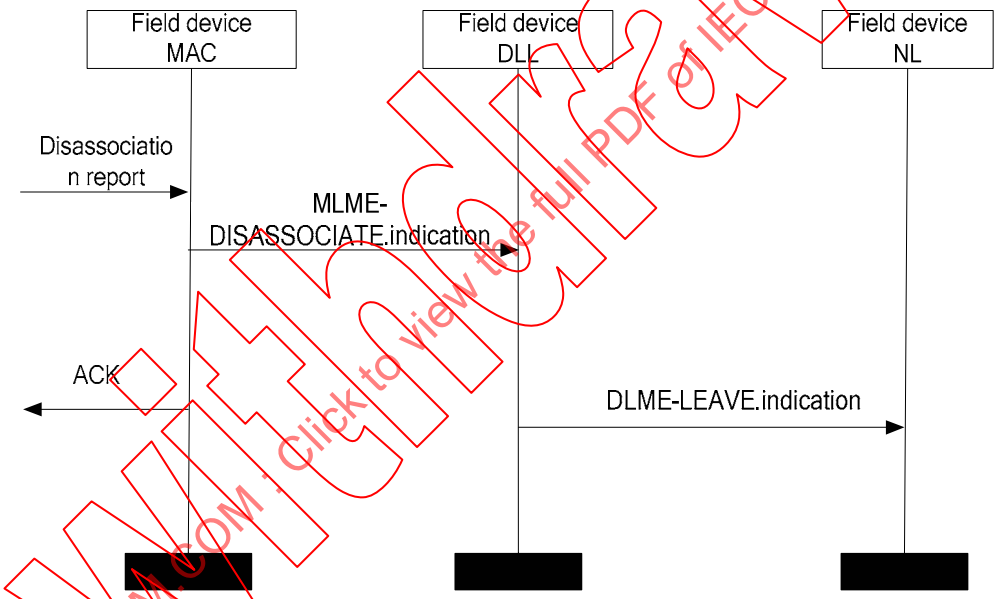


Figure 37 – Time sequence of routing device passive leaving (field device side)

8.6.4.5.2 Time sequence for field devices leaving the star network

Field devices leaving the star network include:

- Active leaving: field device preparing to leave will transmit a leaving request to its routing device. After acknowledged by routing device, field device will leave the star network.

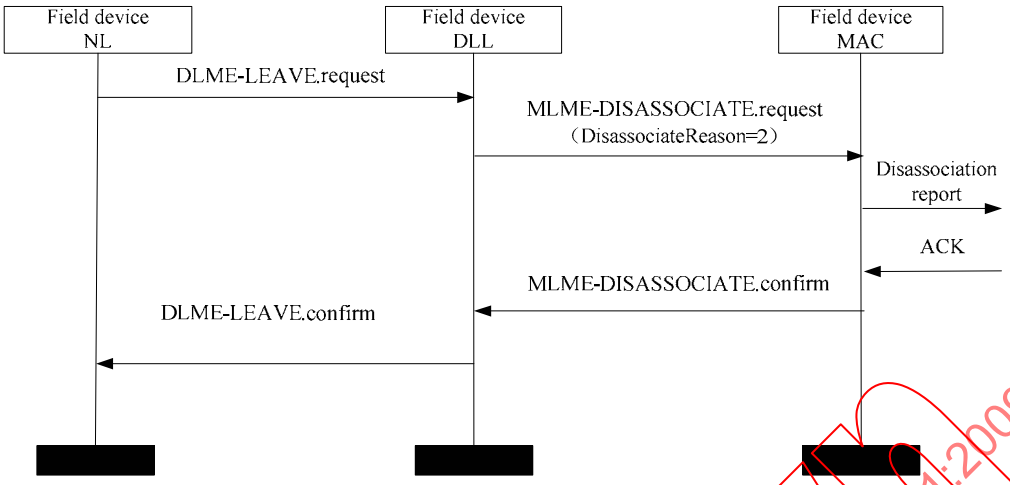


Figure 38 – Time sequence of field device active leaving (field device side)

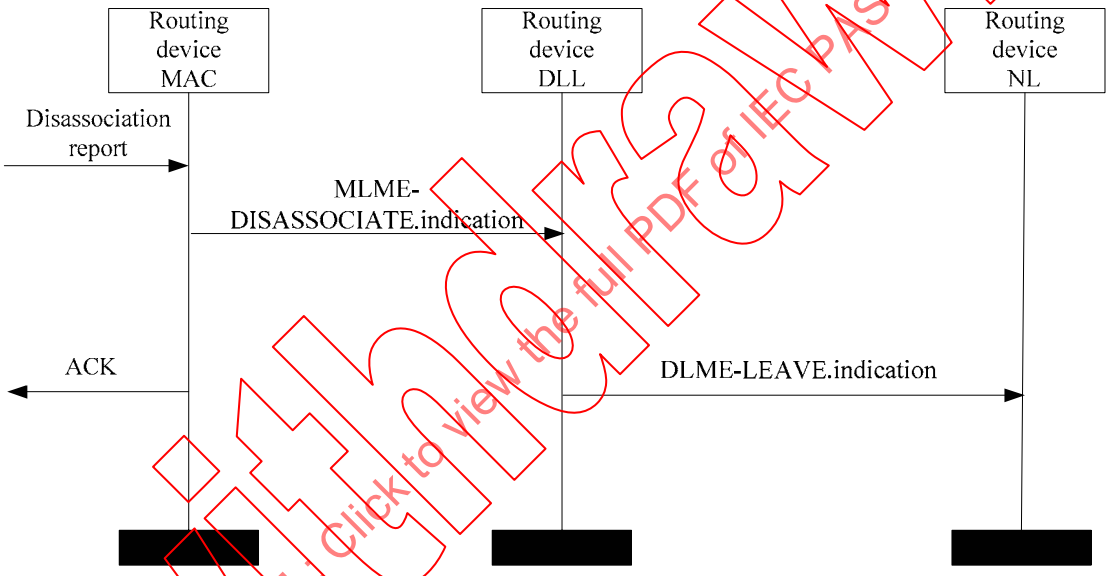


Figure 39 – Time sequence of field device active leaving (routing device side)

- Passive leaving: if routing device asks field device to leave, it will transmit a leaving request to the field device. After returning ACK to routing device, the field device will leave the network.

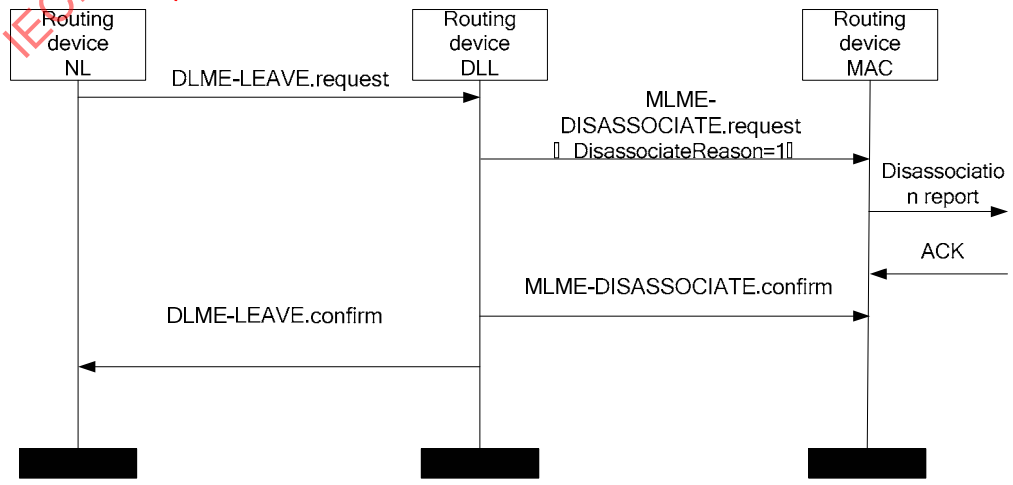


Figure 40 – Time sequence of field device passive leaving (routing device side)

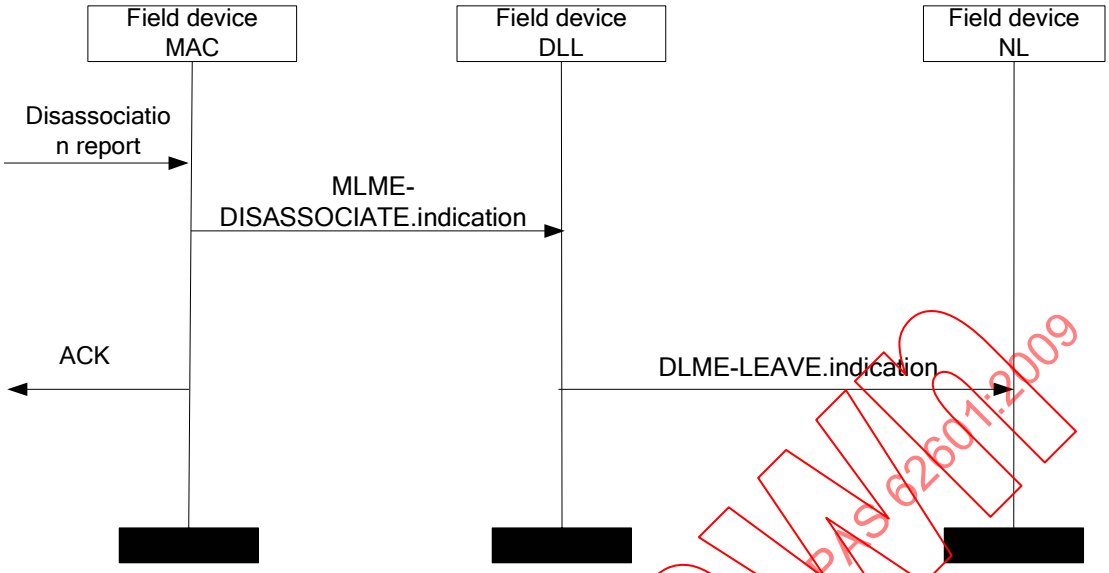


Figure 41 – Time sequence of field device passive leaving (field device side)

8.6.5 Communication resource allocation services

8.6.5.1 General

Communication resource allocation includes the allocation of link and superframe. These services provide following primitives:

- Link adding services:
  - DLME-ADD-LINK.request primitive;
  - DLME-ADD-LINK.confirm primitive;
- Link update services:
  - DLME-UPDATE-LINK.request primitive;
  - DLME-UPDATE-LINK.confirm primitive;
- Link release services:
  - DLME-RELEASE-LINK.request primitive;
  - DLME-RELEASE-LINK.confirm primitive;
- Superframe adding services:
  - DLME-ADD-SFR.request primitive;
  - DLME-ADD-SFR.confirm primitive;
- Superframe update services:
  - DLME-UPDATE-SFR.request primitive;
  - DLME-UPDATE-SFR.confirm primitive;
- Superframe release services:
  - DLME-RELEASE-SFR.request primitive;
  - DLME-RELEASE-SFR.confirm primitive.

8.6.5.2 Link adding services

8.6.5.2.1 DLME-ADD-LINK.request primitive

DLME-ADD-LINK.request primitive is used to add a record of a new link, which is originated from gateway to routing device or from routing device to field device.

DLME-ADD-LINK.request (

DstAddr,
LinkCount,
LinkStructure

)

Table 32 specifies the parameters for the DLME-ADD-LINK.request primitive.

Table 32 – DLME-ADD-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0~65535	16-bit.dstination address.
LinkCount	Unsigned16	0~65535	Count of added links to support adding multiple links each time
LinkStructure[]	Link_Struct	---------	The information of the link attribute
NOTE The data type of Link_Struct is struct, see 8.7.3.2 for details.			

8.6.5.2.2 DLME-ADD-LINK.confirm primitive

DLME-ADD-LINK.confirm primitive reports the result of DLME-ADD-LINK.request.

DLME-ADD-LINK.confirm (

Status

)

Table 33 specifies the parameters for the DLME-ADD-LINK.confirm primitive.

Table 33 – DLME-ADD-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~ 15	Results of link adding request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The detail of the results is shown in the Table 20.

8.6.5.2.3 Time sequence for link adding

Time sequence for adding a link originated from gateway to routing device is shown in Figure 42.

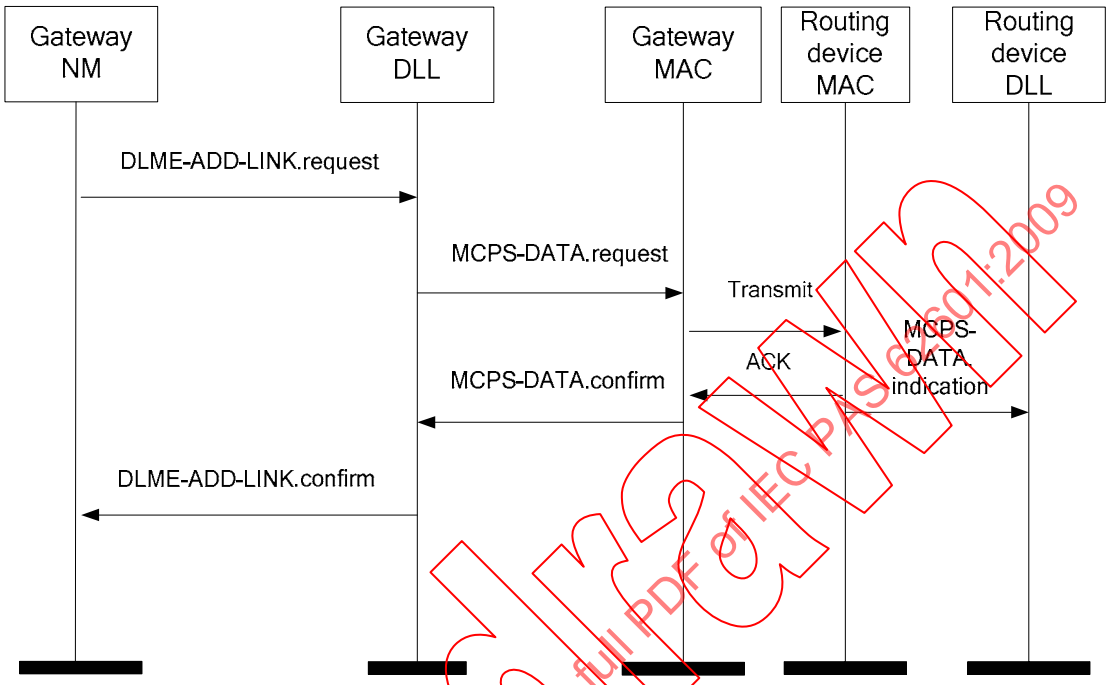


Figure 42 – Adding a link originated from gateway to routing device

Time sequence for adding a link originated from routing device to field device is shown in Figure 43.

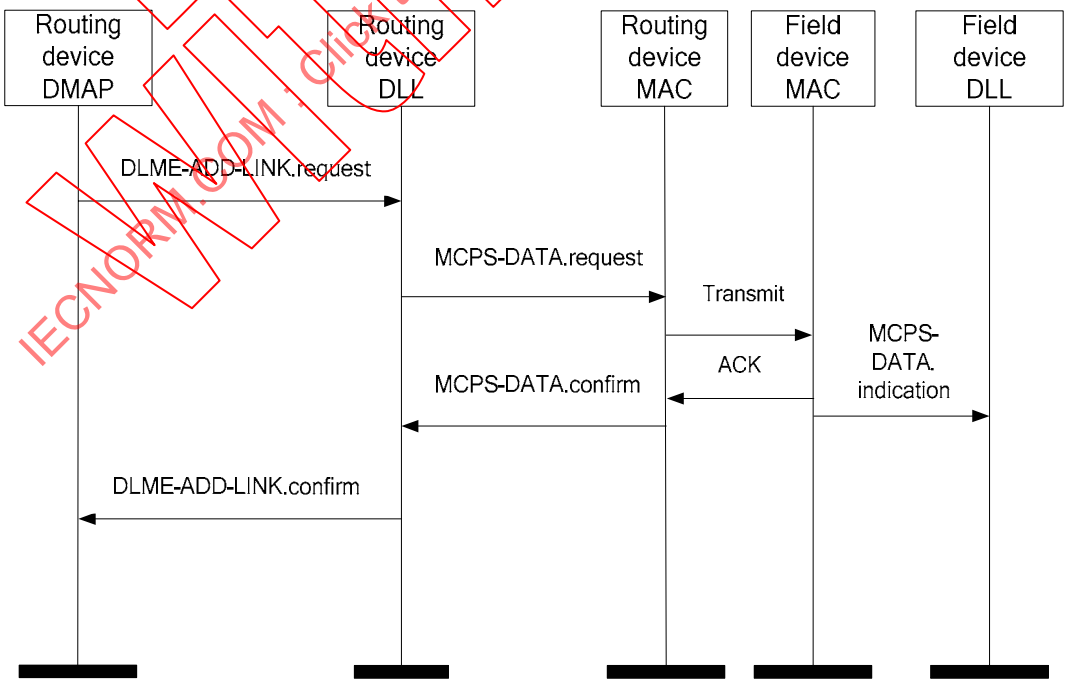


Figure 43 – Adding a link originated from routing device to field device

8.6.5.3 Link update services

8.6.5.3.1 DLME-UPDATE-LINK.request primitive

DLME-UPDATE-LINK.request primitive is used to update a record of an existed link, which is originated from gateway to routing device or from routing device to field device.

DLME-UPDATE-LINK.request (

DstAddr,
LinkCount,
LinkStructure

)

Table 34 specifies the parameters for the DLME-UPDATE-LINK.request primitive.

Table 34 – DLME-UPDATE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0~65535	16-bit.dstination address.
LinkCount	Unsigned16	0~65535	Count of added links to support updating multiple links each time
LinkStructure[]	Link_Struct	---------	The information of the link attribute
NOTE The data type of Link_Struct is struct, see 8.7.3.2 for details.			

8.6.5.3.2 DLME-UPDATE-LINK.confirm primitive

DLME-UPDATE-LINK.confirm primitive reports the result of DLME-UPDATE-LINK.request.

DLME-UPDATA-LINK.confirm (

Status

)

Table 35 specifies the parameters for the DLME-UPDATE-LINK.confirm primitive.

Table 35 – DLME-UPDATE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~ 15	Results of the link update request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The details of the results are shown in Table 20.

8.6.5.3.3 Time sequence for link update

Time sequence for updating a link originated from gateway to routing device is shown in Figure 44.

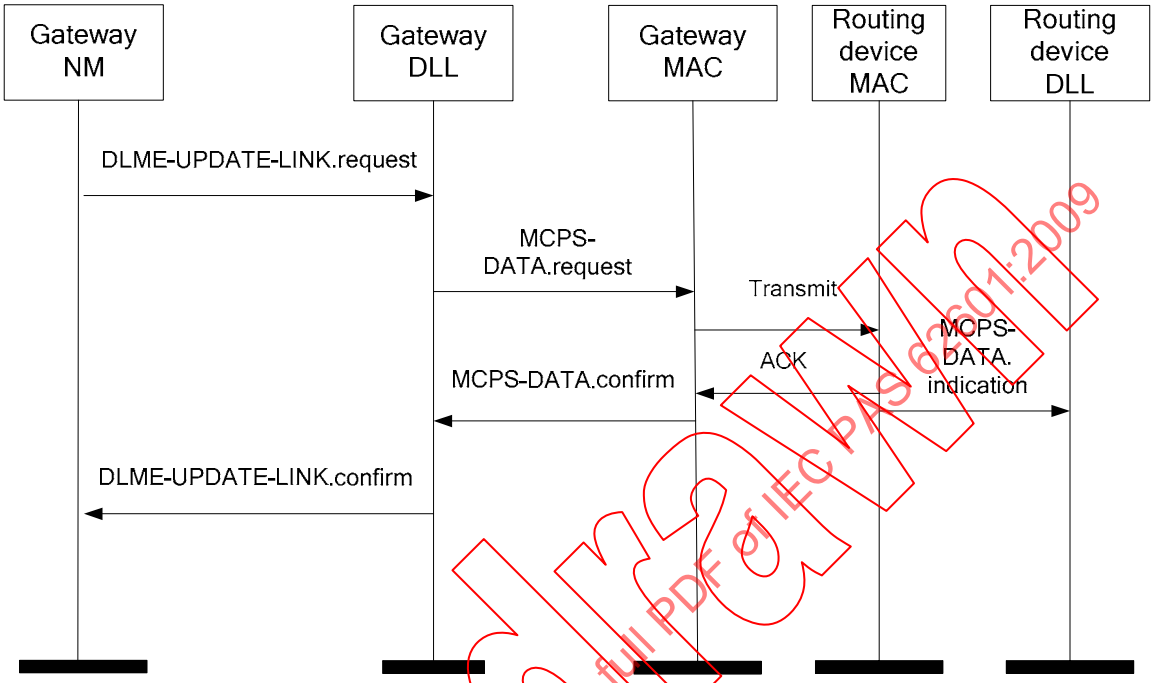


Figure 44 – Updating a link originated by gateway to a routing device

Time sequence for updating a link originated from routing device to field device is shown in Figure 45.

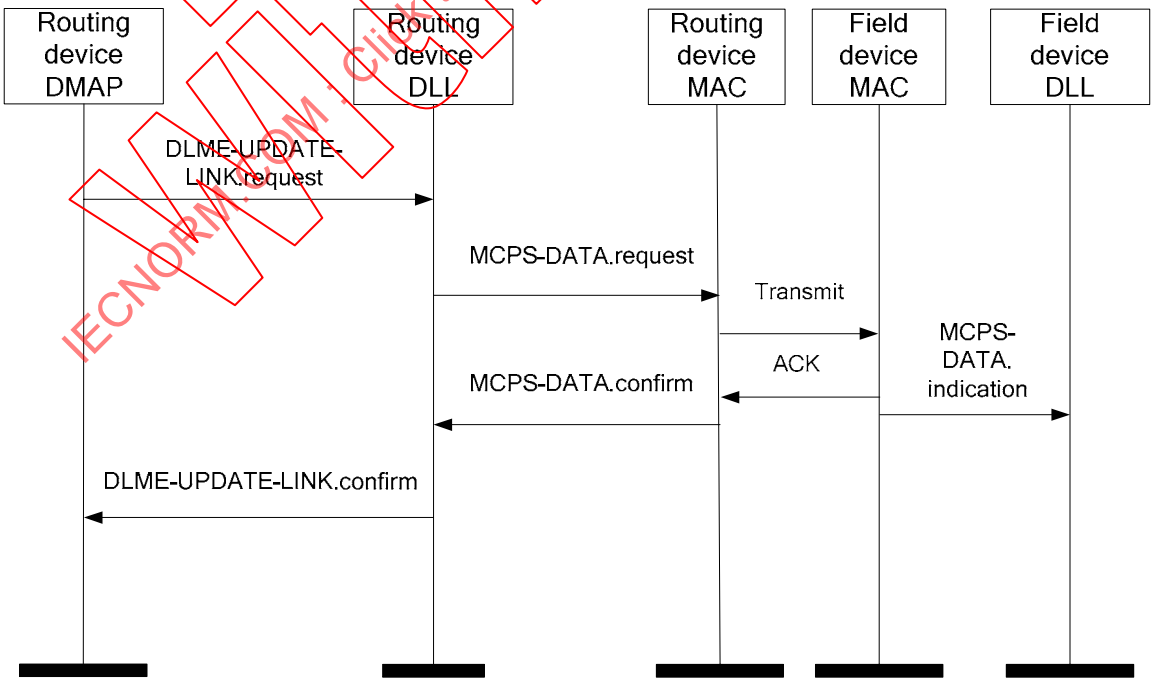


Figure 45 – Updating a link originated from routing device to field device

8.6.5.4 Link release services

8.6.5.4.1 DLME-RELEASE-LINK.request primitive

DLME-RELEASE-LINK.request primitive is used to delete an existed link, which is originated from gateway to routing device or from routing device to field device.

DLME-RELEASE -LINK.request (

DstAddr,
LinkCount,
LinkStructure

)

Table 36 specifies the parameters for DLME-RELEASE-LINK.request primitive.

Table 36 – DLME-RELEASE-LINK.request parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0~65535	16-bit dstination address.
LinkCount	Unsigned16	0~65535	Count of added links to support releasing multiple links each time.
LinkStructure[]	Link_Struct	----------	The information of the link attribute.
NOTE The data type of Link_Struct is struct, see 8.7.3.2 for details.			

8.6.5.4.2 DLME-RELEASE-LINK.confirm primitive

DLME-RELEASE-LINK.confirm primitive reports the result of DLME-RELEASE-LINK.request.

DLME-RELEASE-LINK.confirm (

Status

)

Table 37 specifies the parameters for the DLME-RELEASE-LINK.confirm primitive.

Table 37 – DLME-RELEASE-LINK.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~ 15	Results of the link release request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The details of the results are shown in Table 20.

8.6.5.4.3 Time sequence for link release

Time sequence for releasing a link originated from gateway to routing device is shown in Figure 46.

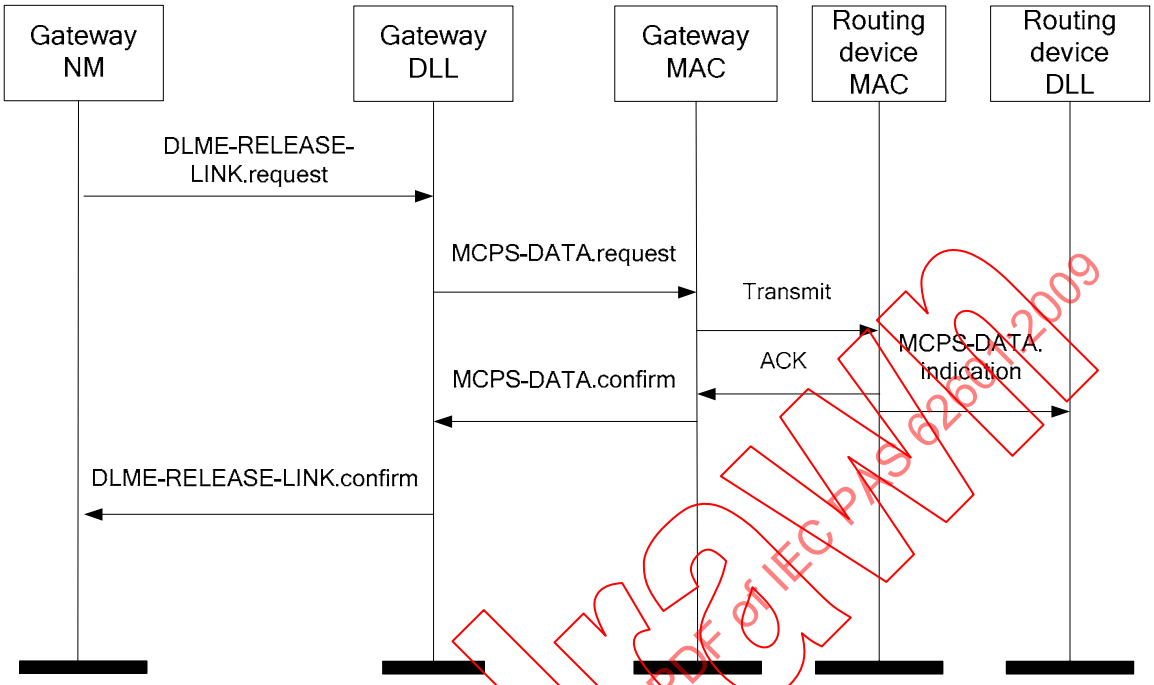


Figure 46 – Releasing a link originated from gateway to routing device

Time sequence for releasing a link originated from routing device to field device is shown in Figure 47.

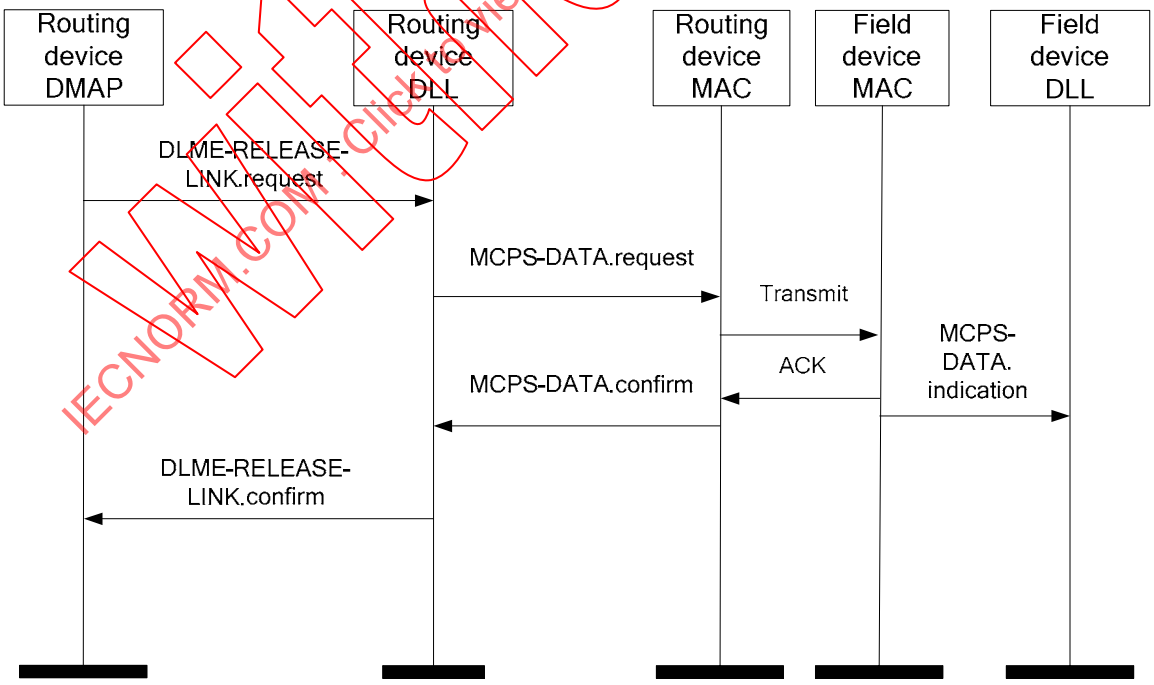


Figure 47 – Releasing a link originated from routing device to field device

8.6.5.5 Superframe adding services

8.6.5.5.1 DLME-ADD-SFR.request primitive

DLME-ADD-SFR.request primitive is used to add a record of a new superframe, which is originated from gateway to routing device or from routing device to field device.

```
DLME-ADD-SFR.request (
                        DstAddr,
                        SuperframeStructure
                    )
```

Table 38 specifies the parameters for the DLME-ADD-SFR.request primitive.

Table 38 – DLME-ADD-SFR.request parameters

Name	Data type	Valid range	Attribute description
DstAddr	Unsigned16	0~65535	16 bit.dstination address.
SuperframeStructure	Superframe_Struct	--------	The information of Superframe attribute.
NOTE The data type of Superframe_Struct is struct, see 8.7.3.1 for details.			

8.6.5.5.2 DLME-ADD-SFR.confirm primitive

DLME-ADD-SFR.confirm reports the result of DLME-ADD-SFR.request.

```
DLME-ADD-SFR.confirm (
                        Status
                    )
```

Table 39 specifies the parameters for the DLME-ADD-SFR.confirm primitive.

Table 39 – DLME-ADD-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~15,	Results of superframe release request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The details of the results are shown in Table 20.

8.6.5.5.3 Time sequence for superframe adding

Time sequence for adding a superframe originated from gateway to routing device is shown in Figure 48.

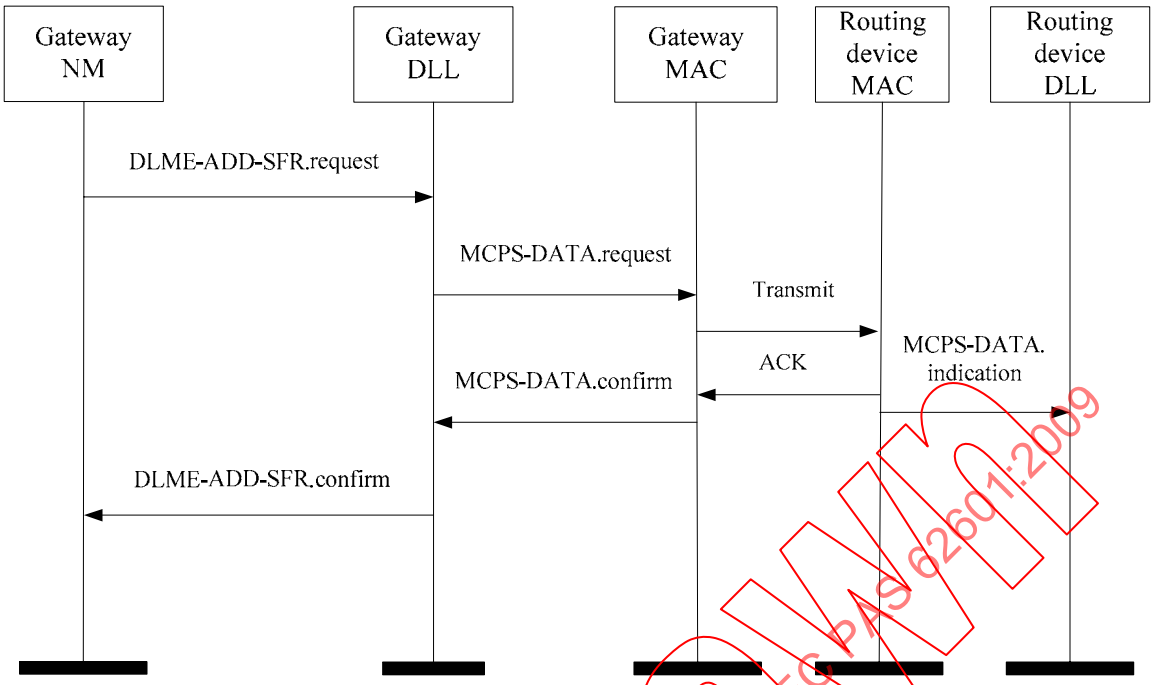


Figure 48 – Adding a superframe originated from gateway to routing device

Time sequence for adding a superframe originated from routing device to field device is shown in Figure 49.

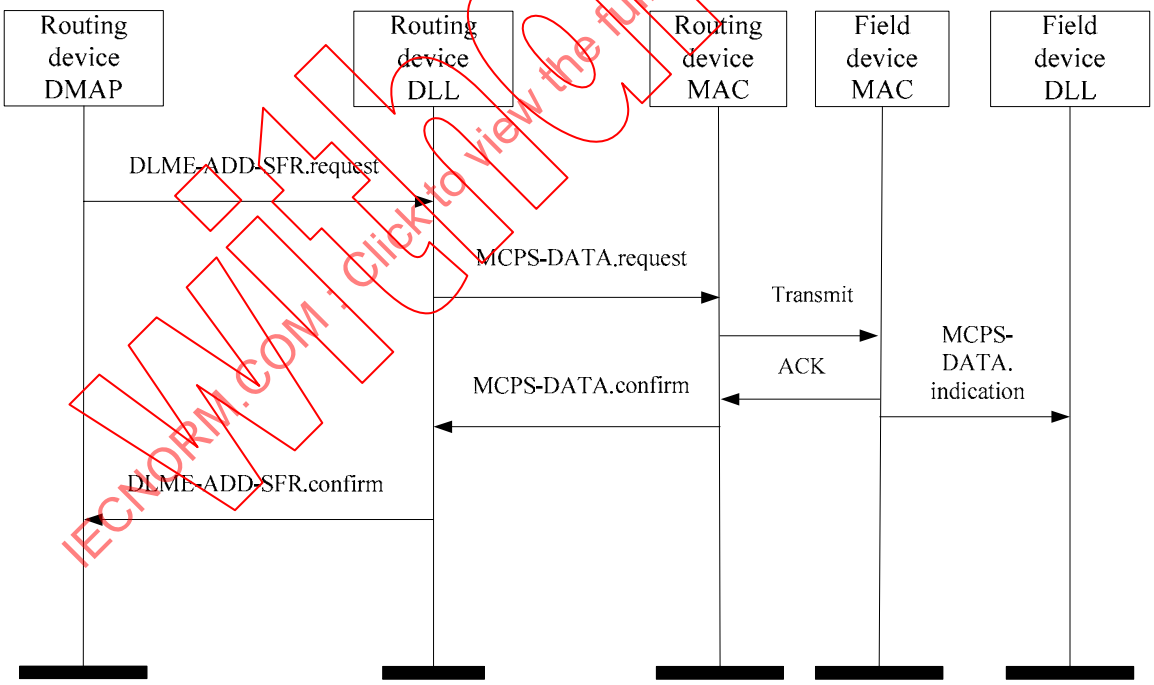


Figure 49 – Adding a superframe originated from routing device to field device

8.6.5.6 Superframe update services

8.6.5.6.1 DLME-UPDATA-SFR.request primitive

DLME-UPDATA-SFR.request primitive is used to modify a record of an existed superframe, which is originated from gateway to routing device or from routing device to field device.

DLME-UPDATA-SFR.request (

DstAddr,

SuperframeStructure

)

Table 40 specifies the parameters for DLME-UPDATA-SFR.request primitive.

Table 40 – DLME-UPDATA-SFR.request parameters

Name	Data type	Valid range	Attribute description
SuperframeID	Unsigned16	0 ~ 65535	Unique identifier of the superframe, supplied by the network manager.
SuperframeStructure	Superframe_Struct	--------	The information of Superframe attribute.
NOTE The data type of Superframe_Struct is struct, see 8.7.3.1 for details.			

8.6.5.6.2 DLME-UPDATE-SFR.confirm primitive

DLME-UPDATE-SFR.confirm reports the result of DLME-UPDATA-SFR.request.

DLME-UPDATE-SFR.confirm (

Status

)

Table 41 specifies the parameters for DLME-UPDATE-SFR.confirm primitive:

Table 41 – DLME-UPDATE-SFR.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~15	Results of superframe update request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The details of the results are shown in Table 20.

8.6.5.6.3 Time sequence for superframe update

Time sequence for updating a superframe originated from gateway to routing device is shown in Figure 50.

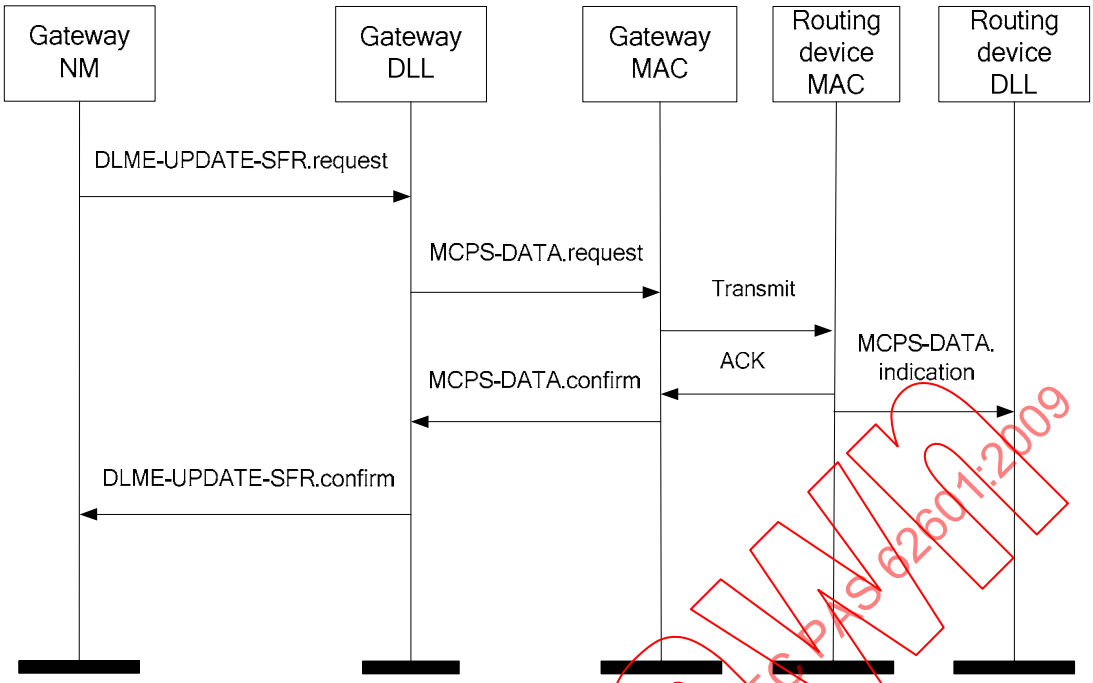


Figure 50 – Updating a superframe originated from gateway to routing device

Time sequence for updating a superframe originated from routing device to field device is shown in Figure 51.

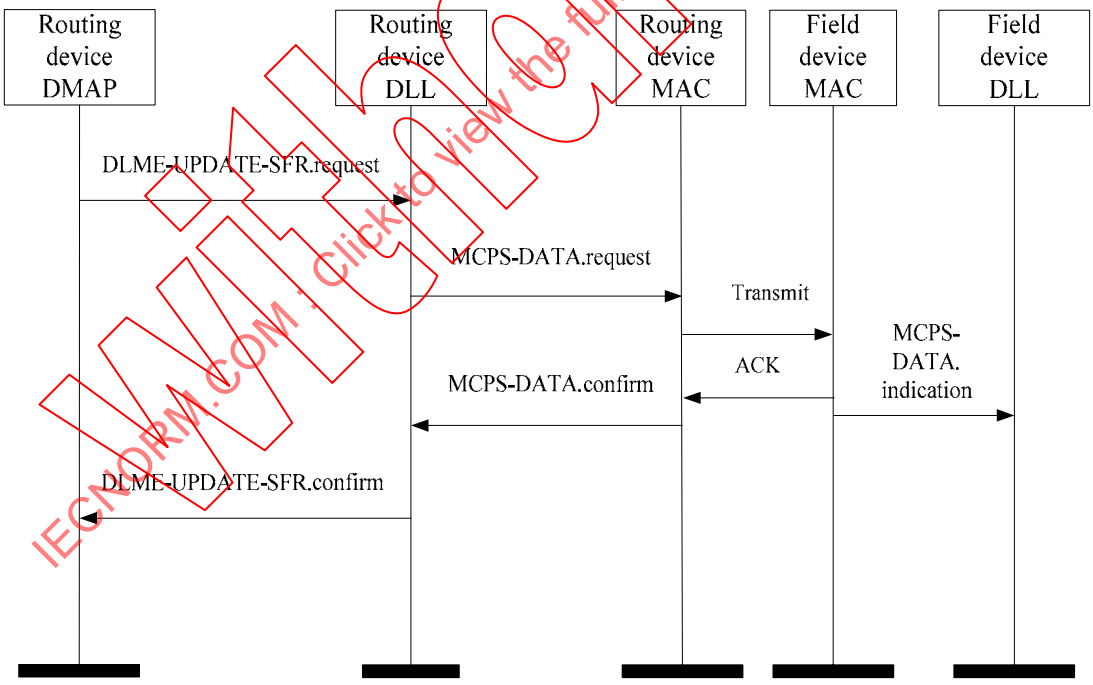


Figure 51 – Updating a superframe originated from routing device to field device

8.6.5.7 Superframe release services

8.6.5.7.1 DLME-RELEASE-SFR.request primitive

DLME-RELEASE-SFR.request is used to delete a record of an existed superframe, which is originated from gateway to routing device or from routing device to field device.

DLME-RELEASE-SFR.request (

DstAddr,

SuperframeStructure

)

Table 42 specifies the parameters for the DLME-RELEASE-SFR.request primitive:

Table 42 – DLME-RELEASE-SFR.request primitive parameters

Name	Data type	Valid range	Description
DstAddr	Unsigned16	0~65535	16-bit.dstination address.
SuperframeStructure	Superframe_Struct	--------	The information of Superframe attribute
NOTE The data type of Superframe_Struct is struct, see 8.7.3.1 for details.			

8.6.5.7.2 DLME-RELEASE-SFR. confirm primitive

DLME-RELEASE-SFR.confirm primitive reports the result of DLME-RELEASE-SFR.request.

DLME-RELEASE-SFR. confirm (

Status

)

Table 43 specifies the parameters for the DLME-RELEASE-SFR. confirm primitive:

Table 43 – DLME-RELEASE-SFR. confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~15,	Results of superframe release request: SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, UNAVAILABLE_KEY, FAILED_SUCURITY_CHECK, INVALID_PARAMETER

The details of the results are shown in Table 20.

8.6.5.7.3 Time sequence for superframe release

Time sequence for releasing a superframe originated from gateway to routing device is shown in Figure 52.

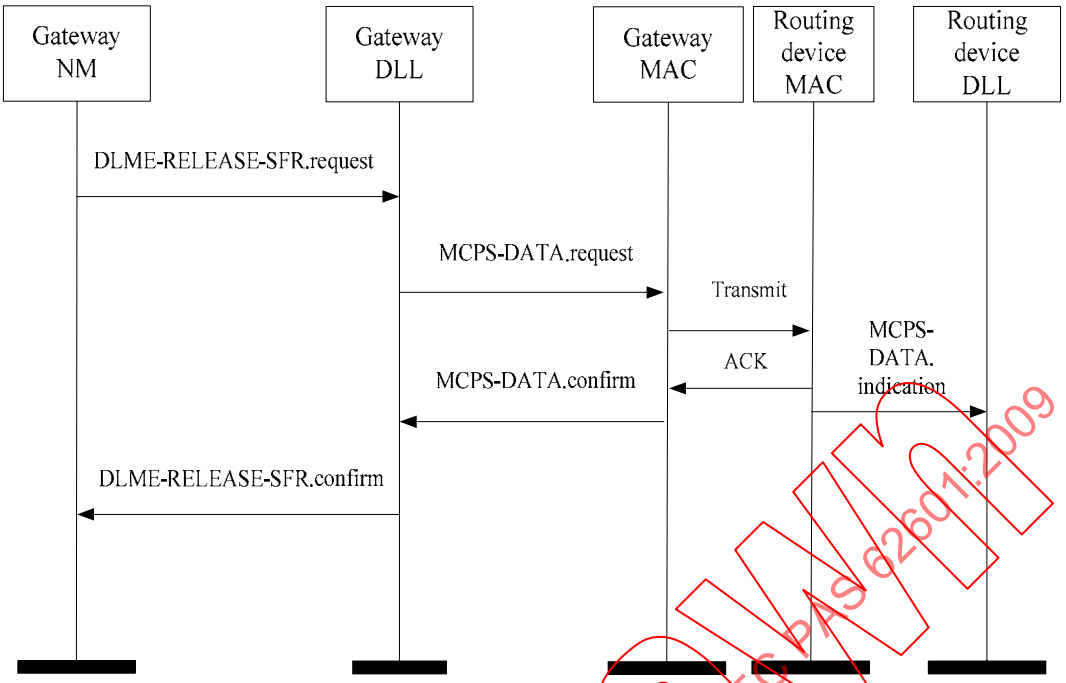


Figure 52 – Releasing a superframe originated from gateway to routing device

Time sequence for releasing a superframe originated from routing device to field device is shown in Figure 53.

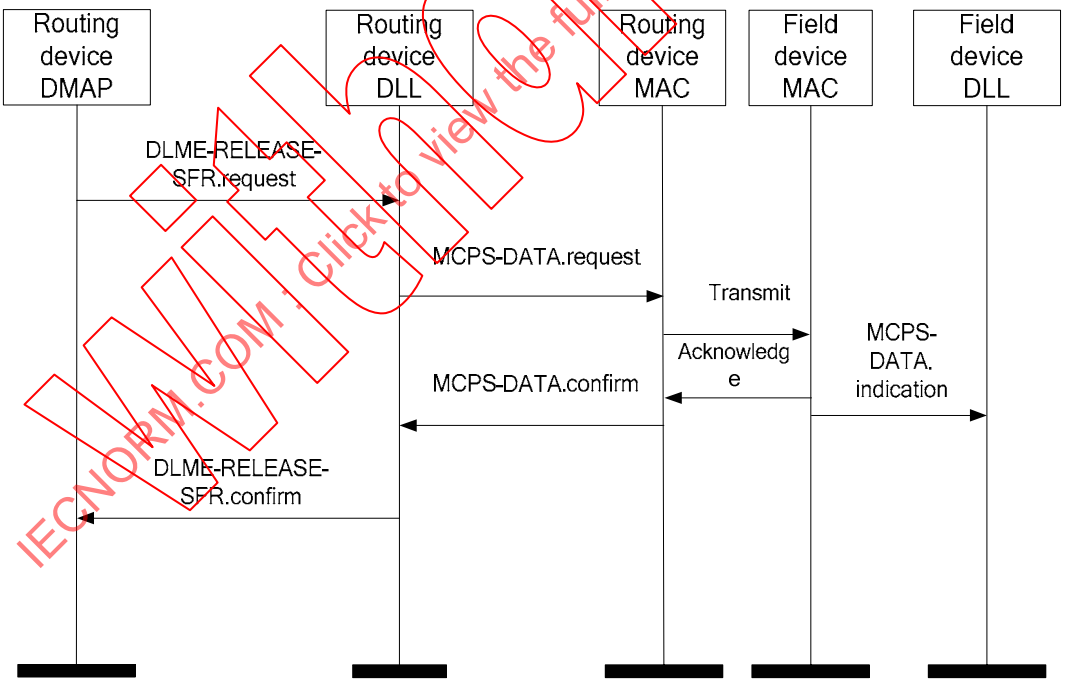


Figure 53 – Releasing a superframe originated from routing device to field device

8.6.6 Management information attribute getting services

8.6.6.1 DLME-GET.request primitive

DLME-GET.request primitive is used to request information about a given DLL-MIB attribute.

DLME-GET.request (

DllMIBAttribute_ID

)

Table 44 specifies the parameters for the DLME-GET.request primitive:

Table 44 – DLME-GET.request primitive parameters

name	Data type	Valid range	Description
DllMIBAttribute_ID	Unsigned8	0 ~ 255	Attribute Identifier

The DLME-GET.request primitive is generated by upper layer and issued to its DLME to obtain information from the DLL-MIB.

8.6.6.2 DLME-GET.confirm primitive

DLME-GET.confirm primitive reports the result of requesting an attribute value from the DLL-MIB.

DLME-GET.confirm (

Status,
DllMIBAttributeID,
DllMIBAttributeLength,
DllMIBAttributeValue

)

Table 45 specifies the parameters for the DLME-GET.confirm primitive.

Table 45 – DLME-GET.confirm parameters

Name	Data type	Valid range	Description
status	Unsigned8	0 ~ 15	Results of requesting a attribute: SUCCESS; UNSUPPORTED_ATTRIBUTE.
DllMIBAttibuteID	Unsigned8	0 ~ 255	Attribute identifier
DllMIBAttributeValue	Octets	------	The attribute value. The value is 0 when status is UNSUPPORTED_ATTRIBUTE.

The DLME-GET.confirm primitive is generated by the DLME and issued to upper layer in response to a DLME-GET.request primitive. This primitive returns a status of either SUCCESS, indicating that the request to read a DLL-MIB attribute was successful, or an error code of UNSUPPORTED_ATTRIBUTE. When an error code of UNSUPPORTED_ATTRIBUTE is returned, the DllMIBAttibuteLength will be set to zero.

8.6.7 Management information attribute setting services

8.6.7.1 DLME-SET.request primitive

DLME-SET.request primitive attempts to write the given value to the indicated DLL-MIB attribute.

DLME-SET.request (

DllMIBAttribute_ID,
DllMIBAttributeValue

)

Table 46 specifies the parameters for the DLME-SET.request primitive:

Table 46 – DLME-SET.request parameters

Name	Data type	Valid range	Description
DllMIBAttribute_ID	Unsigned8	0 ~ 255	Attribute Identifier in the DLL-MIB
DllMIBAttributeValue	Octets	-------	Attribute value that will be set. The value is 0 when status is UNSUPPORTED_ATTRIBUTE.

The DLME-SET.request primitive is generated by upper layer and issued to its DLME to write the indicated DLL-MIB attribute.

8.6.7.2 DLME-SET.confirm primitive

DLME-SET.confirm primitive reports the result of an attempt to write a value to a DLL-MIB attribute.

DLME-SET.confirm(

Status,
DllMIBAttribute_ID

)

Table 47 specifies the parameters for the DLME-SET.confirm primitive:

Table 47 – DLME-SET.confirm parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0 ~ 15	Results of attribute value set request: SUCCESS; READ_ONLY; UNSUPPORTED_ATTRIBUTE; INVALID_PARAMETER.
DllMIBAttribute_ID	Unsigned8	0 ~ 255	Attribute Identifier that is modified

The DLME-SET.confirm primitive is generated by the DLME and issued to upper layer in response to a DLME-SET.request primitive. The DLME-SET.confirm primitive returns a status of either SUCCESS, indicating that the requested value has been written to the indicated DLL-MIB attribute, or the appropriate error code. The details of the results are shown in Table 20.

8.7 Data link layer management information base (DLL-MIB)

8.7.1 General

The DLL-MIB comprises the attributes required to manage the DLL of a device.

These attributes include unstructured attributes and structured attributes.

8.7.2 Unstructured attribute

Table 48 lists the unstructured attribute maintained by DLL.

Table 48 – Unstructured attribute

Identifier	Name	Data type	Range	Access	Default	Description
17	StatisticsDuration	Unsigned16	0 ~ 65535	R/ W	0	Configuring the cycle of the statistic collection. After this duration, devices update the statistics in the neighbour tables.

8.7.3 Structured attributes

DLL defines three structured attributes, including superframe, link and neighbour. Each structured attribute has special data structure. DLL maintains three tables of communication information, which respectively comprise records of the corresponding data structure.

The identifiers of structured attributes are shown in Table 49.

Table 49 – Identifier of the structured attributes

Identifier	Attribute Name	Data type
18	Superframe	Superframe_Struct
19	Link	Link_Struct
20	Neighbour	Neighbour_Struct

Table 50 presents the general information of these tables.

Table 50 – Three tables of the structured attributes

Name	Description
Superframe table	Describes the superframe attributes
Link table	Describes the link attributes
Neighbour table	Describes the neighbour attributes

Figure 54 shows the relationship among superframe, link and neighbour attributes.

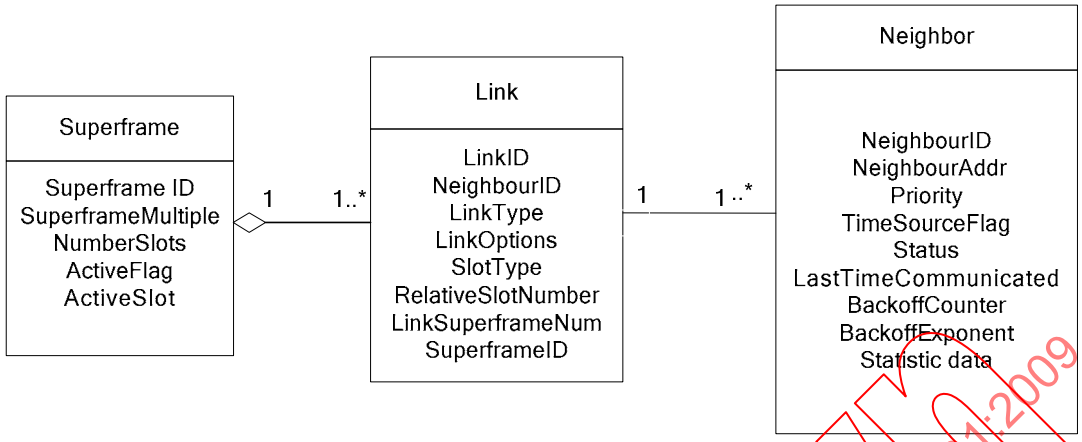


Figure 54 – Relationship of the DLL structured attributes

8.7.3.1 Superframe attribute

Superframe attribute maintains the information of superframes.

Table 51 shows the structure of Superframe attribute.

Table 51 – Superframe attribute structure

Name	Data type	Valid range	Attribute description
SuperframeID	Unsigned16	0 ~ 65535	Unique identifier of the superframe, supplied by the network manager.
SuperframeMultiple	Unsigned8	0 ~ 255	The multiples of the basic WIA-PA superframe. It is used for restricting the WIA-PA superframe length. It is used for processing the long period data transmission.
NumberSlots	Unsigned16	0 ~65535	Superframe size (counts of timeslots)
ActiveFlag	Unsigned8	0~1	Superframe active flag 0 = Inactive; 1 = Active;
ActiveSlot	Unsigned48	0 ~(2^{48}-1)	Absolute timeslot number when a superframe begins active.

8.7.3.2 Link attribute

Link attribute maintains the information of links. Link attribute refers to Superframe attribute and Neighbour attribute.

Table 52 shows the structure of Link attribute.

Table 52 – Link attribute structure

Name	Data type	Valid range	Description
LinkID	Unsigned16	0 ~ 65535	Unique identifier of the Link
NeighborID	Unsigned16	0 ~ 65535	Reference to a Neighbour table entry.
LinkType	Unsigned8	0 ~3	0= Unicast; 1= Broadcast; 2= Multicast.
LinkOptions	Unsigned8	0 ~ 3	The character of a link: 0=transmitting; 1=transmit-shared; 2=receiving; 3=reserve.
SlotType	Unsigned8	0 ~ 3	The type of a timeslot: 0=data timeslot; 1=management timeslot; 2~3=reserved.
RelativeSlotNumber	Unsigned16	0 ~ 65535	Relative timeslot number.
LinkSuperframeNum	Unsigned8	0 ~ 255	The multiples of the *basic WIA-PA superframe* in the *WIA-PA superframe* when this link happens. It is used for processing the long period data transmission.
SuperframeID	Unsigned16	0 ~ 65535	Reference to a Superframe table entry.

Long period data is defined that its data update rate is larger than the maximum superframe length of IEEE 802.15.4.

TransmitFlag is defined as follows:

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

The process of long period data transmission is described as follows.

If *TransmitFlag* is 0, the frame is transmitted in the timeslot of the current superframe cycle; otherwise the frame is not transmitted in current superframe cycle. The definition of *AbsoluteSlotNumber* refers to 8.4.3, and *NumberSlots*, *ActiveSlot* and *SuperframeMultiple* refer to 8.7.3.1. Figure 55 shows an example of dealing with long period frame.

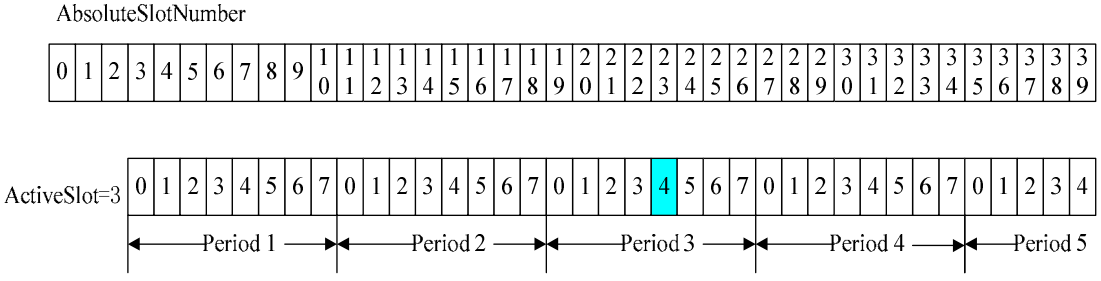


Figure 55 – An example of long period data transmission

In Figure 55, *NumberSlots* = 8, *SuperframeMultiple* = 3, and the slot numbered 4 of the third superframe is assigned for a long period data transmission.

If the current absolute slot of Beacon is 19, *TransmitFlag* is calculated as follows:

TransmitFlag = ⌈(19-3+1)/ 8(%3 = 0,

Then, the packet is sent at current period.

8.7.3.3 Neighbour attribute

Neighbour attribute maintains the information of neighbours of the device.

‼ REF _Ref204511271 \r \h ┌¶Table 53┴ shows the structure of neighbour attribute.

Table 53 – Neighbour attribute structure

Name	Data type	Valid range	Description
NeighborID	Unsigned16	0 ~65535	Neighbour node ID
NeighborAddr	Unsigned16	0 ~ 65535	The short address of neighbour node.
Priority	Unsigned8	0 ~15	The priority of join in of neighbour node.
TimeSourceFlag	Unsigned8	0 ~ 1	Indicating that whether the neighbour node is a time source: 0=no; 1=yes.
Status	Unsigned8	0 ~ 127	The status of neighbour node, such as link invalid etc.
BackoffCounter	Unsigned8	0 ~ 31	Count of backoff.
BackoffExponent	Unsigned8	0 ~15	Backoff exponent.
LastTimeCommunicated	Unsigned56	0 ~(2^{56}-1)	Time when last communicated with the neighbour node.
AveRSL	Unsigned8	0 ~ 255	The average level of signals received from the neighbour node in *StatisticsDuration* (see 8.7.2).
PacketsTransmitted	Unsigned16	0 ~ 65535	The number of un-broad-cast frames sends to the neighbour node in *StatisticsDuration* (see 8.7.2).
AckPackets	Unsigned16	0 ~ 65535	The number of expected ACK/NACK packets received in *StatisticsDuration* (see 8.7.2).
PacketsReceived	Unsigned16	0 ~ 65535	The number of good packets received from the neighbour node *StatisticsDuration* (see 8.7.2).
BroadcastPackets	Unsigned16	0 ~ 65535	The number of good broadcast packets received from the neighbour node in *StatisticsDuration* (see 8.7.2).

9 Network Layer

9.1 General

The WIA-PA network layer transports packets over networks, provides interfaces to application sub-layer, and carries out network layer management.

9.2 Stack structure

The structure of WIA-PA network layer is shown in Figure 56.

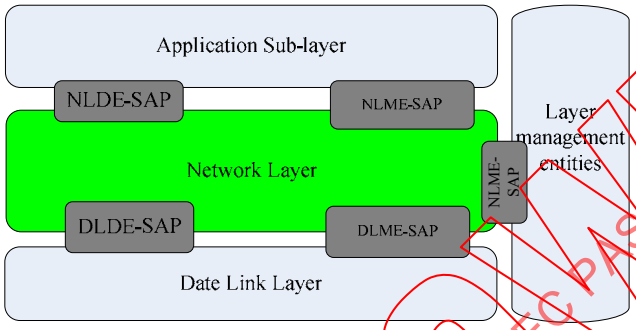


Figure 56 – WIA-PA network layer stack structure

Network layer defines Network Layer Data Entity (NLDE) and Network Layer Management Entity (NLME).

NLDE provides the service interface through which the application sub-layer transmits and receives data.

NLME provides the service interfaces through which layer management functions may be invoked. The NLME is also responsible for maintaining a database of pertaining to the network layer. This database is referred to as the NL-MIB.

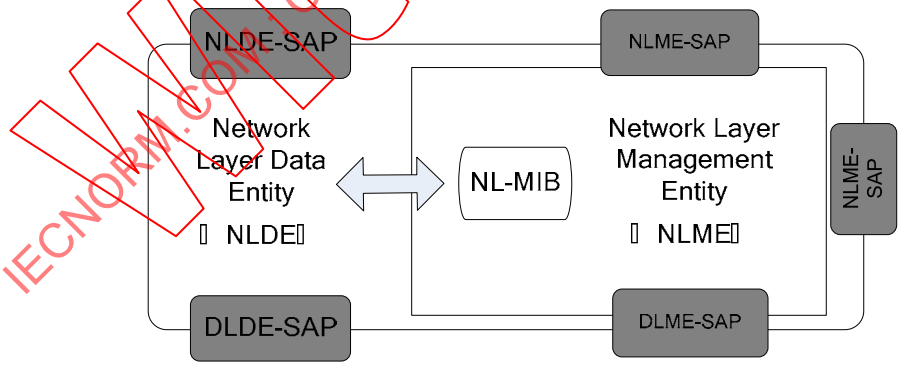


Figure 57 – WIA-PA Network Layer reference model

The network layer provides two services, accessed through two SAPs:

– the network layer data service, accessed through the network layer data entity SAP (NLDE-SAP), and

– the network layer management service, accessed through the network layer management entity SAP (NLME-SAP).

9.3 Function description

9.3.1 General

The network layer in WIA-PA is designed to perform the following functions.

a) Addressing
b) Routing
c) Packet lifecycle management
d) Management of Device joining and leaving network
e) End-to-end network performance monitoring
f) Fragmentation and reassembly

9.3.2 Addressing

Each device has a unique 64-bit physical address (EUI-64) and a 16-bit network address. The 16-bit network address can be denoted as x.x, where x is an 8-bit integer. The most-significant 8-bit in the 16-bit network address is the cluster address, and the least-significant 8-bit is the intra-cluster address. The range of the cluster address is 1~254, and the range of the intra-cluster address is 0~254. The number 255 is used for broadcast address. Routing device's intra-cluster address is 0.

The classifications of the network addresses are as follows:

a) Unicast addresses – Each device's address is the unicast address, including the following types:
   - Gateway address is 0.x, where x is 0~254;
   - Routing device address is x.0, where x is 1~254;
   - Field device address is x.x, where x is 1~254.
b) Broadcast address – According to the different broadcast domains, there is 4 types broadcast addresses:
   - Intra-cluster broadcast address is x.255, where x is 1~254;
   - Global broadcast address is 255.255;
   - Mesh network broadcast address is 255.0.
c) Multicast address – Reserved 255.1~255.254 as multicast address.

9.3.3 Routing

9.3.3.1 General

WIA-PA network supports static routing algorithms, which are configured by the network manager.

After getting the neighbour information from each routing device, network manager generates the connection relationship of all the routing devices. On the basis of the connection relationship, network manager generates and writes the routing information to each routing device in the form of routing table. Each route in the table is assigned a Route ID. To improve reliability, there may be more than one route corresponding to one data VCR.

9.3.3.2 Routing table

Each routing device maintains a routing table, which is generated by the network manager. The routing table is used for the path selection in the mesh network. The table have five items. Route_ID is the identifier of a route. Destination Address is the address of the endpoint of a route. Next hop is the address of the next hop relayer in a route. VCR_ID identifies which

VCR a route is serving for. RetryCounter records the times of end-to-end retry in a route which reflects the status of the route.

A routing table example is shown in Table 54.

Table 54 – Routing table

Route_ID	Destination Address	Next hop	VCR_ID	RetryCounter
5	N1	N3	4	0
8	GW	N2	6	0
NOTE GW in the table denotes a gateway address. N1, N2 and N3 denote routing devices' addresses.				

9.3.4 Packet lifecycle management

Each packet has a lifecycle in the WIA-PA networks. The lifecycle is expressed as a maximal surviving time. The network layer records packets' generating time by using timestamps. Surviving time is computed according to generating time. When the surviving time of a packet exceeds its lifecycle, then the packet is discarded.

9.3.5 Joining and leaving network of device

WIA-PA network layer supports device joining and leaving network.

9.3.6 End-to-end network performance monitoring

WIA-PA network layer monitors each path health status. The NLME records path failures in management information base. If there is a path failure, the NLME sends an indication to the DMAP.

9.3.7 Fragmentation and reassembly

Fragmentation and reassembly are handled at the network layer. If the length of a NPDU is larger than the maximum DLL payload, the NPDU will be fragmented at the network layer of sender. When the fragmented NPDUs reach the receiver, they are reassembled at the network layer.

9.3.8 Network layer state machine

The network layer defines the following states, see Table 55.

Table 55 – Network layer states

State	Description
Idle	Do nothing, wait for events occurring
Transmitting	Pack the packets, and pass them to the DLL
Receiving	Unpack the packets, and pass them to the ASL

a) Idle state

The following transitions can occur while in "Idle" state:

- When DLL invokes DLDE-DATA.indication primitives, NL enters "receiving" state.
- When ASL invokes ASLDE-DATA.request primitives, NL enters "Transmitting" state.

b) Transmitting state

In the “Transmitting” state, if receiving NLDE-DATA.confirm primitives, NL enters “Idle” state.

c) Receiving state

In the “Receiving” state, NL enters “Idle” state after NL issues NLDE-DATA.indication primitive to the upper layer.

States transitions are shown in Figure 58:

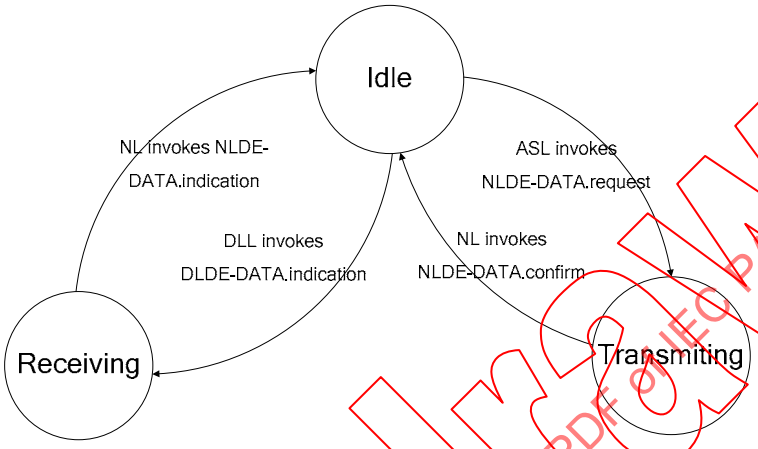


Figure 58 – Network layer state machine

9.4 Network layer packet formats

9.4.1 Common packet format

Network layer packet format is shown in Figure 59.

Octet: 1	4		2	4	2	Variable
Control field	Destination address	Source address	Route_ID	Timestamp	Payload length	Network layer payload
	Routing filed					
Network layer packet header						Network layer payload

Figure 59 – Network layer packet format

Control field format is shown in Figure 60.

Bits: 0-1	2-3	4-7
Type	Routing type	Reserved

Figure 60 – Control field

Control field includes:

- Type, 2 bits. 00 denotes data packet, 11 denotes command packet, 01 and 10 is reserved;
- Routing type, 2 bits;
- Reserved, 4 bits.

Routing field includes:

- Destination address: final destination address of packet(16 bits);
- Source address: the address from where packet originates (16 bits);
- Route_ID: unique identifier of the path (16 bits).

9.4.2 Data packet

Data packet format is shown in Figure 61.

<table>
<tr><td>Octets: 1</td><td colspan="2">4</td><td>2</td><td>4</td><td>2</td><td>variable</td></tr>
<tr><td rowspan="2">Control field</td><td>Destination address</td><td>Source address</td><td>Route_ID</td><td rowspan="2">Timestamp</td><td rowspan="2">Payload length</td><td rowspan="2">Network layer payload</td></tr>
<tr><td colspan="3">Routing field</td></tr>
<tr><td colspan="5">Network layer header</td><td colspan="2">Network layer payload</td></tr>
</table>

Figure 61 – Network layer data packet format

9.4.3 Command packet

9.4.3.1 Common command packet format

Network layer command packets encapsulate network layer commands to accomplish some management functions.

Command packet format is shown in Figure 62.

<table>
<tr><td>Octets: 1</td><td colspan="2">4</td><td>2</td><td>4</td><td>2</td><td>1</td><td>variable</td></tr>
<tr><td rowspan="2">Control field</td><td>Destination address</td><td>Source address</td><td>Route_ID</td><td rowspan="2">Timestamp</td><td rowspan="2">Payload length</td><td>Network layer command identifier</td><td>Network layer command payload</td></tr>
<tr><td colspan="3">Routing field</td><td colspan="2">Network layer payload</td></tr>
<tr><td colspan="6">Network layer header</td><td colspan="2">Network layer payload</td></tr>
</table>

Figure 62 – Network layer command packet format

Network layer command packets are listed in Table 56.

Table 56 – Network layer command packet

Command packet identifier	command
0	Adding route request
1	Adding route response
2	Updating route request
3	Updating route response
4	Deleting route request
5	Deleting route response
6-255	reserved

Executing results of Command are shown in Table 57.

Table 57 – Executing results of Commands

Command implementing result	identifier	description
SUCCESS	0	Command execution successes
NO_DST_ADDRESS	1	No destination address
PACKET_TOO_LONG	2	Packet length is longer than NPDU
INVALID_PARAMETER	3	Parameters are invalid

9.4.3.2 Adding route request and response

Command packet of adding route request is generated by network manager and sent to destination routing device for adding a new routing record into its routing table.

The format of this request command packet is illustrated as Figure 63.

Octets: 1	9
Command identifier: 1	A routing table record
Network layer payload	

Figure 63 – Command packet of adding route request

Command packet of adding route response is used to return the executing result of the adding route request command.

The format of this response command packet is illustrated as Figure 64.

Octets: 1	1
Command identifier: 2	Result
Network layer payload	

Figure 64 – Command packet of adding route response

NOTE Executing results are shown in Table 57.

9.4.3.3 Updating route request and response

Command packet of updating route request is generated by network manager and sent to destination routing device for modifying a routing table record in its routing table.

The format of the request command packet is illustrated as Figure 65.

Octets: 1	9
Command identifier: 3	A routing table record
Network layer payload	

Figure 65 – Command packet of updating route request

Command packet of updating route response is used to return the executing result of the updating route request command.

The format of the response command packet is illustrated as Figure 66.

Octets: 1	1
Command identifier: 4	Result
Network layer payload	

Figure 66 – Command packet of updating route response

NOTE Executing results are shown in Table 57.

9.4.3.4 Deleting route request and response

Command packet of deleting route request is generated by the network manager and sent to destination routing device for deleting a routing record from its routing table.

The format of the request command packet is illustrated as Figure 67.

Octets: 1	2
Command identifier: 5	Route_ID
Network layer payload	

Figure 67 – Command packet of deleting route request

Command packet of deleting route response is used to return the executing result of the deleting route request command.

The format of the response command packet is illustrated as Figure 68:

Octets: 1	1
Command identifier: 6	Result
Network layer payload	

Figure 68 – Command packet of deleting route response

NOTE Executing results are shown in Table 57.

9.5 Network layer data services

9.5.1 General

Network layer data service access point (NLDE-SAP) is used to transport NPDUs. Network layer data services provide NLDE-DATA.request primitive, NLDE-DATA.confirm primitive and NLDE-DATA.indication primitive.

9.5.2 NLDE-DATA.request primitive

NLDE receives the payload from application sub-layer through NLDE-DATA.request primitive and adds it to the message queue of the NL.

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

NLDE-DATA.request (

VCR_ID,
DstAddr,
SrcAddr,
Priority,
PayloadLength,
Payload,
PayloadHandle

)

Table 58 specifies the parameters for the NLDE-DATA.request primitive.

Table 58 – NLDE-DATA.request primitive parameters

Name	Data type	Valid range	Description
VCR_ID	Unsigned16	0~65535	Virtual Communication Relationships(VCR) ID
DstAddr	Unsigned16	0~65535	The 16-bit short address of the NSDU's destination.
SrcAddr	Unsigned16	0~65535(Unicast address)	The 16-bit short address of the NSDU's source.
Priority	Unsigned8	0~15	Priority of this NSDU.
PayloadLength	Unsigned16	0~65535	The length of NSDU to be transmitted.
Payload	Octets	—	NSDU
PayloadHandl	Unsigned8	0~255	The handle associated with the NSDU to be transmitted.

On the occasion of having data to send to the network layer, the application sub-layer invokes the NLDE-DATA.request primitive to request the network layer to transmit data. On receipt of the NLDE-DATA.request primitive, the network layer hands the data packets to the DLL for transmission.

9.5.3 NLDE-DATA.confirm primitive

The NLDE-DATA.confirm primitive reports the result of a NLDE-DATA.request primitive and informs the data in the sending queue having been sent.

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

NLDE-DATA.confirm (

PayloadHandle,
Status

)

Table 59 specifies the parameters for the NLDE-DATA.confirm primitive.

Table 59 – NLDE-DATA.confirm primitive parameters

Name	Data type	Valid range	Description
PayloadHandle	Unsigned8	0~255	The handle of the NSDU.
Status	Unsigned8	0=SUCCESS, 1=FAIL, 2~255=reserved	The result of NLDE-DATA.request primitive.

The network layer generates the NLDE-DATA.confirm primitive in response to NLDE-DATA.request primitive. The NLDE-DATA.confirm primitive returns a status of either SUCCESS or FAIL as soon as the request is executed.

9.5.4 NLDE-DATA.indication primitive

The NLDE-DATA.indication primitive informs the application sub-layer when the network layer receives a packet.

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

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

Table 60 specifies the parameters of the NLDE-DATA.indication primitive.

Table 60 – NLDE-DATA.indication primitive parameters

Name	Data type	Valid range	Description
SrcAddr	Unsigned16	0~65535(Unicast Address)	The 16-bit short address of the NSDU's source.
Priority	Unsigned4	0~15	Priority of this NSDU.
NSDULength	Unsigned16	0~65535	The length of the NSDU.
NSDU	octet	—	The data of the NSDU.

The network layer will generate the NLDE-DATA.indication primitive when receiving an appropriately addressed data frame from the local data link layer successfully. The network layer invokes the NLDE-DATA.indication primitive to inform the application sub-layer.

9.5.5 Time sequence of the NL data service

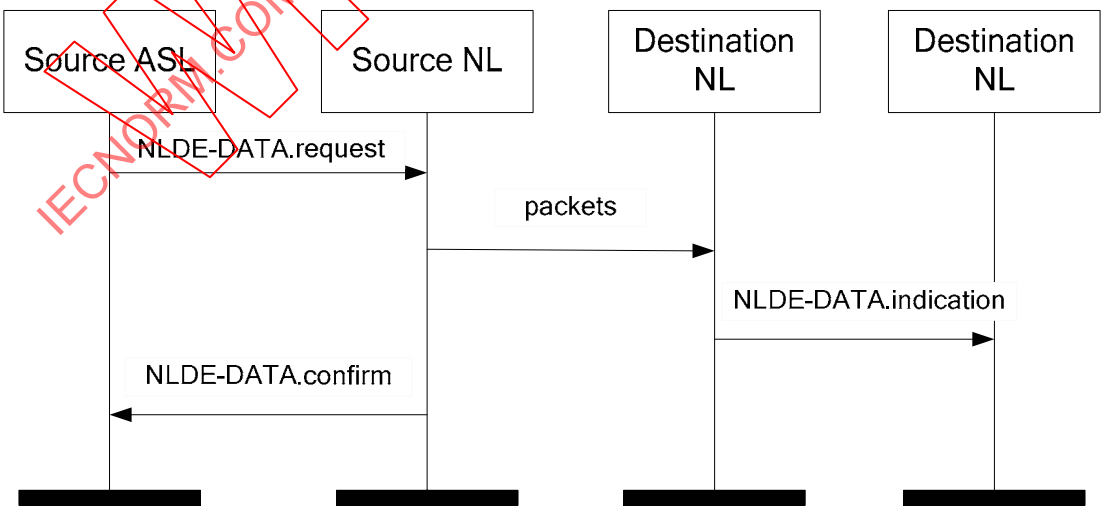


Figure 69 – Time sequence of network layer data services

Figure 69 shows the basic procedures of the packet sending and receiving. The NLDE-DATA.request primitive is generated by a local ASLDE when a data ASLPDU is to be

transferred to a peer NLDE. On receipt of the NLDE-DATA.request primitive, NLDE begins the transmission of the NPDU.

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

The NLDE-DATA.indication primitive is generated by the NLDE of destination device and issued to the ASLDE on receipt of a data packet at the local NLDE that passed the appropriate message filtering operations.

9.6 Network layer management services

9.6.1 General

The DMAP uses the interface supplied by the NLME-SAP to configure and control the network layer's operation.

9.6.2 NL-MIB attribute getting services

The upper layer or DMAP invokes the MIB attribute getting primitives to get the values of attributes in MIB.

9.6.2.1 NLME-GET.request primitive

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

```
NLME-GET.request (
        NIMIBAttribute_ID
)
```

Table 61 specifies the parameters for the NLME-GET.request primitive.

Table 61 – NLME-GET.request primitive parameters

Name	Data type	Valid range	Description
NIMIBAttribute_ID	Unsigned8	0~255	MIB attribute ID

9.6.2.2 NLME-GET.confirm primitive

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

```
NLME-GET.confirm (
        NIMIBAttribute_ID,
        Status,
        NIMIBAttributeValue
)
```

Table 62 specifies the parameters for the NLME-GET.confirm primitive.

Table 62 – NLME-GET.Confirm primitive parameters

Name	Data type	Valid range	Description
NIMIBAttribute_ID	Unsigned8	0~255	MIB attribute ID
Status	Unsigned8	0=SUCCESS, 1=UNSUPPORTED_ATTRIBUTE, 2~255=reserved	The result of the request
NIMIBAttributeValue	Octet	—	The value of the attribute

The network layer generates the NLME-GET.confirm primitive in response to a NLME-GET.request primitive. The NLME-GET.confirm primitive returns the executing result of request to the upper layer.

- SUCCESS: the attribute value has been successfully read.
- UNSUPPORTED_ATTRIBUTE: the requested attribute value does not exist.

9.6.3 NL-MIB attribute setting services

9.6.3.1 NLME-SET.request primitive

The upper layer invokes the NLME-SET.request primitive to set the value of a MIB attribute.

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

NLME-SET.request (

NIMIBAttribute_ID,
NIMIBAttributeValue

)

Table 63 specifies the parameters for the NLME-SET.request primitive.

Table 63 – NLME-SET.request primitive parameters

Name	Data type	Valid range	Description
NIMIBAttribute_ID	Unsigned8	0~255	MIB attribute ID
NIMIBAttributeValue	Octet	—	The value of the attribute

9.6.3.2 NLME-SET.confirm primitive

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

NLME-SET.confirm (

NIMIBAttribute_ID,
Status

)

Table 64 specifies the parameters for the NLME-SET.confirm primitive.

Table 64 – NLME-SET.confirm primitive parameters

Name	Data type	Valid range	Description
NIMIBAttribute_ID	Unsigned8	0~255	MIB attribute ID
Status	Unsigned8	0=SUCCESS, 1=UNSUPPORTED_ATTRIBUTE 2=INVALID_PARAMETER, 3=READ_ONLY, 4~255=reserved	The result of the request

The network layer generates the NLME-SET.confirm primitive in response to a NLME-SET.request primitive. The NLME-GET.confirm primitive returns a status of SUCCESS or the appropriate error code to the upper layer.

a) If the attribute is read-only, the NLME-SET.confirm primitive will return a “READ_ONLY” status.

b) If the attribute value is out of range, the NLME-SET.confirm primitive will return an “INVALID_PARAMETER” status.

c) If there is no such attribute, the NLME-SET.confirm primitive will return an “UNSUPPORTED_ATTRIBUTE” status.

9.6.4 Routing services

9.6.4.1 Route adding services

9.6.4.1.1 NLME-ADD_ROUTE.request primitive

The NLME-ADD_ROUTE.request primitive is used to add a record to the routing table at the network layer of routing devices.

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

```
NLME-ADD_ROUTE.request (
        DstAddress,
        Routing_table_record
)
```

Table 65 specifies the parameters for the NLME-ADD_ROUTE.request primitive.

Table 65 – NLME-ADD_ROUTE.request primitive parameters

Name	Data type	Valid range	Description
DstAddress	Unsigned16	0~65535	The 16-bit short address of the routing device.
Routing_table_record	Structure NLRoute_Tbl		A routing table item, see Table 82.

The network manager invokes the NLME-ADD_ROUTE.request primitive to add a record to the routing table at the network layer of routing devices.

9.6.4.1.2 NLME-ADD_ROUTE.confirm primitive

The NLME-ADD_ROUTE.confirm primitive reports the result of a NLME-ADD_ROUTE.request primitive.

The semantics of NLME_ADD-ROUTE.confirm primitive are described as follows:

```
NLME-ADD_ROUTE.confirm（
        Status
)
```

Table 66 specifies the parameters for the NLME-ADD_ROUTE.confirm primitive.

Table 66 – NLME-ADD_ROUTE.confirm primitive parameters

Name	Data type	Valid range	Description
Status	Unsigned16	0=SUCCESS, 1=INVALID_PARAMETER, 2~255=reserved	The result of a NLME-ADD_ROUTE.request primitive.

On receipt of the NLME-ADD_ROUTE.request primitive, the network layer will transmit an adding-route-request command packet to the routing device and will return a NLME-ADD_ROUTE.confirm primitive to report the result.

9.6.4.1.3 Time sequence for route adding

Time sequence diagram for adding a record to the routing table of routing devices is shown in Figure 70.

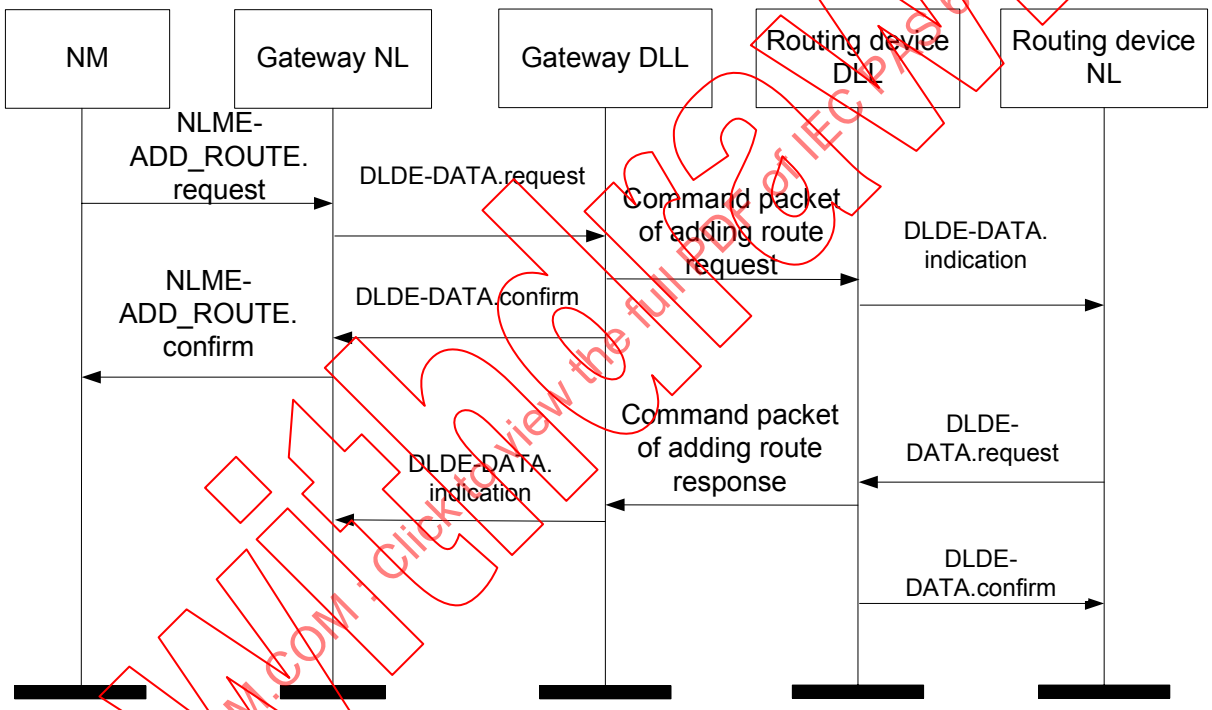


Figure 70 – Time sequence for route adding

9.6.4.2 Route updating services

9.6.4.2.1 NLME-UPDATE_ROUTE.request primitive

The NLME-UPDATE_ROUTE.request primitive updates a record in the routing table at the network layer of routes devices.

The semantics of the NLME-UPDATE_ROUTE.request primitive are described as follows:

```
NLME-UPDATE_ROUTE.request (
        DstAddress,
        Routing_table_record
)
```

Table 67 specifies the parameters for the NLME-UPDATE_ROUTE.request primitive.

Table 67 – NLME-UPDATE_ROUTE.request primitive parameters

Name	Data type	Valid range	Description
DstAddress	Unsigned16	0~65535	The 16-bit short address of routing device.
Routing_table_record	Structure NLRoute_Tbl		A routing table item, see 9.3.3.2.

The network manager invokes the NLME-UPDATE_ROUTE.request primitive to update a record in the routing table at the network layer of routing devices.

9.6.4.2.2 NLME-UPDATE_ROUTE.confirm primitive

The NLME_UPDATE-ROUTE.confirm primitive reports the executing result of a NLME-UPDATE_ROUTE.request primitive.

The semantics of NLME_UPDATE-ROUTE.confirm primitive are described as follows:

NLME-UPDATE_ROUTE.confirm (

Status

)

Table 68 specifies the parameters for the NLME-UPDATE-ROUTE.confirm primitive.

Table 68 – NLME-UPDATE_ROUTE.confirm primitive parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0=SUCCESS, 1=INVALID_PARAMETER, 2~255=reserved	The result of a NLME-UPDATE_ROUTE.request primitive.

On receipt of the NLME-UPDATE_ROUTE.request primitive, the network layer will transmit a command packet of updating route request to the routing device and return a NLME-UPDATE_ROUTE.confirm primitive to report the result.

9.6.4.2.3 Time sequence for route updating

Time sequence diagram for updating a record to the routing table of routing devices is shown in Figure 71.

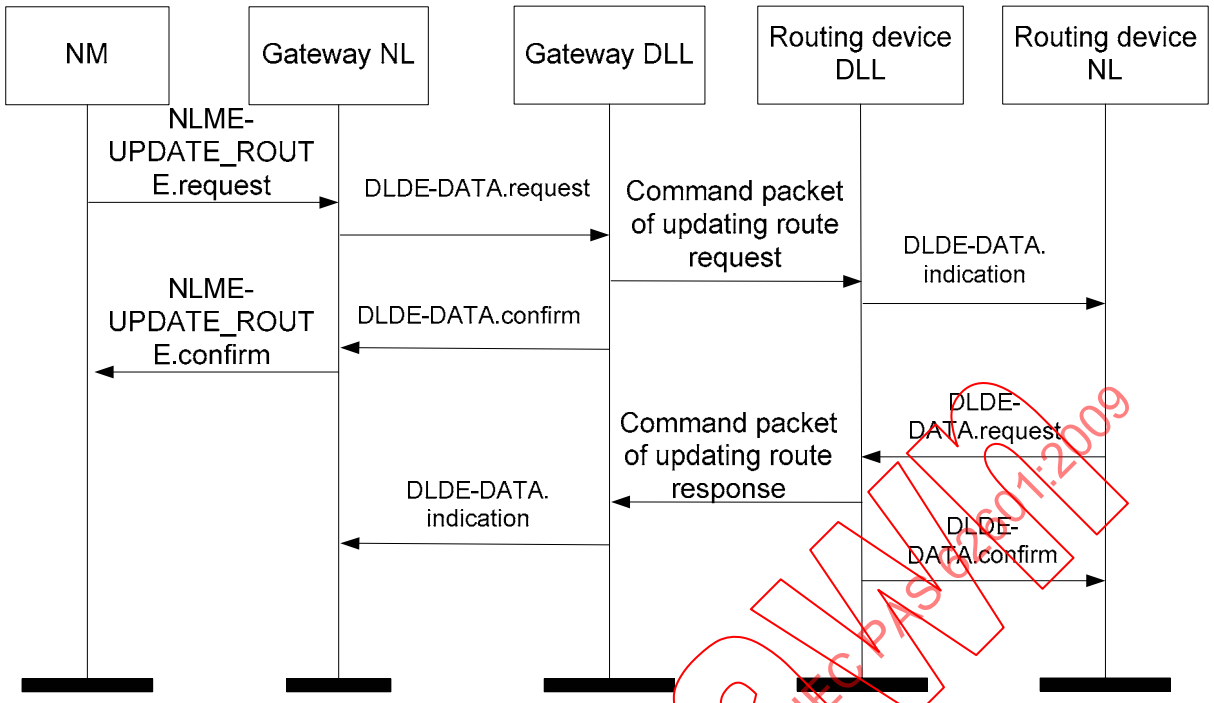


Figure 71 – Time sequence for route updating

9.6.4.3 Route deleting services

9.6.4.3.1 NLME-DELETE_ROUTE.request primitive

The network manager invokes the NLME-DELETE_ROUTE.request primitive to delete a record in the routing table of routing devices.

The semantics of NLME_DELETE-ROUTE confirm primitive are described as follows:

NLME-DELETE_ROUTE.request (

DstAddress,
Route_ID

)

Table 69 specifies the parameters for the NLME-DELETE-ROUTE.request primitive.

Table 69 – NLME-UPDATE_ROUTE.request primitive parameters

Name	Data type	Valid range	Description
DstAddress	Unsigned16	0~65535	The 16-bit short address of the destination routing device of packets.
Route_ID	Unsigned16	0~65535	Route ID

9.6.4.3.2 NLME-DELETE_ROUTE.confirm primitive

The NLME_DELETE-ROUTE.confirm primitive reports the executing result of the NLME-DELETE_ROUTE.request primitive.

The semantics of NLME_DELETE-ROUTE.confirm primitive are described as follows:

NLME-DELETE_ROUTE.confirm (

Status

)

Table 70 specifies the parameters for the NLME-DELETE-ROUTE.confirm primitive.

Table 70 – NLME-DELETE_ROUTE.confirm primitive parameters

Name	Data type	Valid range	Description
Status	Unsigned8	0=SUCCESS, 1=INVALID_PARAMETER, 2~255=reserved	The result of a NLME-DELETE_ROUTE.request primitive.

On receipt of the NLME-DELETE_ROUTE.request primitive, the network layer will transmit a command packet of deleting route request to the routing device and return a NLME-DELETE_ROUTE.confirm primitive to report the result.

9.6.4.3.3 Time sequence for route deleting

Time sequence diagram for deleting a record to the routing table of routing devices is shown in Figure 72.

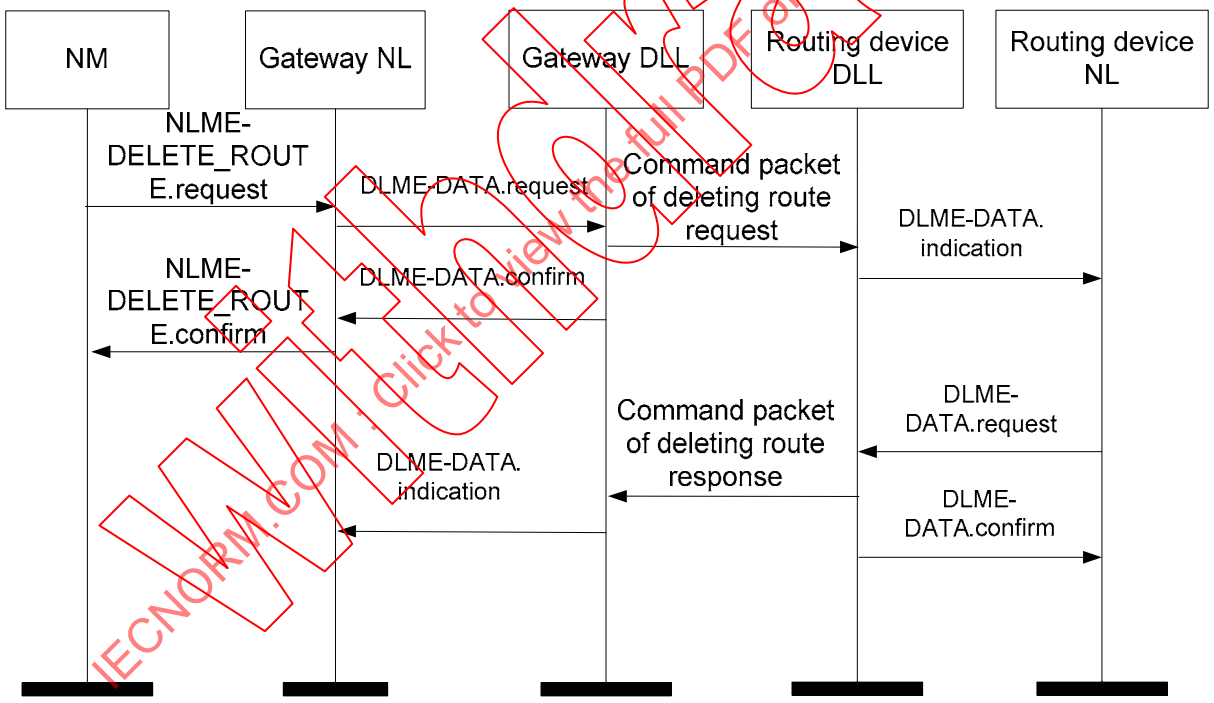


Figure 72 – Time sequence for route deleting

9.6.5 Joining Network services

9.6.5.1 NLME-JOIN.request primitive

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

NLME-JOIN.request(

JoinAddr,
PhyAddr,
DeviceType

)

Table 71 specifies the parameters for the NLME-JOIN.request primitive.

Table 71 – NLME-JOIN.request primitive parameters

Name	Data type	Valid range	Description
JoinAddr	Unsigned16	0~65535(Unicast address)	The address of the routing device selected by the new device.
PhyAddr	Unsigned64	$0\sim2^{64}-1$	64-bit physical address of the new device.
DeviceType	Unsigned8	0=routing device, 1=field device, 2~255=reserved	Device type: a routing device or a field device.

When a new device wants to join the network, the upper layer invokes the NLME-JOIN.request primitive and requests its network layer to start the joining process.

9.6.5.2 NLME-JOIN.confirm primitive

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

NLME-JOIN.confirm (

ShortAddr,
Status

)

Table 72 specifies the parameters for the NLME-JOIN.confirm primitive.

Table 72 – NLME-JOIN.confirm primitive parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0~65535(Unicast address)	The device's network address
Status	Unsigned8	0=SUCCESS, 1=FAILURE, 2~255=reserved	Executing result of the request

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

9.6.5.3 NLME-JOIN.indication primitive

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

NLME-JOIN.indication (

PhyAddr,
DeviceType

)

Table 73 specifies the parameters for the NLME-JOIN.indication primitive.

Table 73 – NLME-JOIN.indication primitive parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	$0\sim2^{64}-1$	Device's physical address.
DeviceType	Unsigned8	0=routing device, 1=field device, 2~255=reserved	Device type: a routing device or a field device.

On receipt of the NLME-JOIN.request primitive, the network layer of routing device or gateway will invoke a NLME-JOIN.indication primitive and inform the upper layer.

9.6.5.4 NLME-JOIN.response primitive

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

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

Table 74 specifies the parameters for the NLME-JOIN.response primitive.

Table 74 – NLME-JOIN.response primitive parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	$0\sim2^{64}-1$	Device physical address
ShortAddr	Unsigned16	0~65535(Unicast address)	Network address
Status	Unsigned8	0=SUCCESS, 1=FAILURE, 2~255=reserved	Executing result of the request

The upper layer invokes the NLME-JOIN.response primitive to assign a network address to the new device.

9.6.5.5 Time sequence for device joining

When a field device wants to join the star network, the ASL of the new field device instructs the network layer to transmit a joining request to a routing device. The network management agent of the routing device receives the request and assigns a network address to the field device. The network layer of field device invokes the DLL joining process.

Figure 740 illustrates an example of the joining process of a field device.

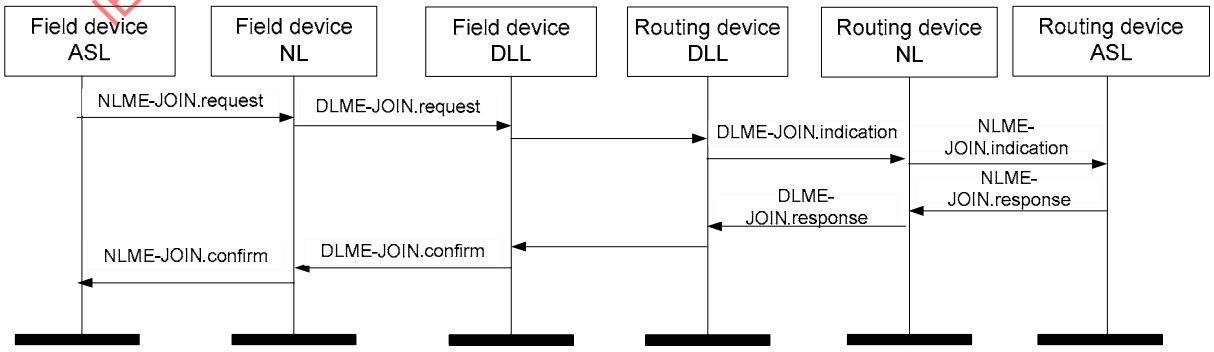


Figure 73 – Time sequence for field device joining

When a routing device wants to join the mesh network, the ASL of the new routing device instructs the network layer to transmit a joining request to the gateway. The network manager receives the request and assigns a network address to the routing device. The network layer of routing device invokes the DLL joining process.

Figure 74 illustrates an example of the join process of a routing device.

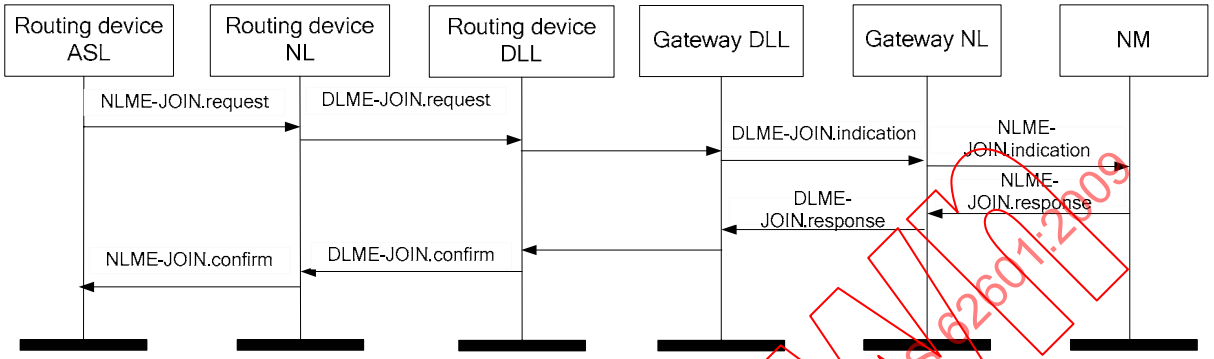


Figure 74 – Time sequence for routing device joining

9.6.6 Leave Network services

9.6.6.1 NLME-LEAVE.request primitive

The NLME-LEAVE.request primitive is used for an existing field device to notify its routing device or for an existing routing device to notify the gateway of its intent to leave the network, which is called active leaving.

The NLME-LEAVE.request primitive is also used for gateway to instruct an existed routing device or for routing device to instruct an existed field device to leave the network, which is called passive leaving.

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

NLME-LEAVE.request (

ShortAddr

)

Table 75 specifies the parameters for the NLME-LEAVE.request primitive.

Table 75 – NLME-LEAVE.request primitive parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0~65535(Unicast address)	The network address.

When a new device wants to leave the network, the upper layer invokes the NLME-LEAVE.request primitive and requests its network layer to start the leaving process.

9.6.6.2 NLME-LEAVE.confirm primitive

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

NLME-LEAVE.confirm (
ShortAddr,
Status

)

Table 76 specifies the parameters for the NLME-LEAVE.confirm primitive.

Table 76 – NLME-LEAVE.confirm primitive parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0~65535(Unicast address)	The network address
Status	Unsigned8	0=SUCCESS, 1=FAILURE, 2~255=reserved	Executing result of the request

The NLME-LEAVE.confirm primitive is generated by the network layer in response to an NLME-LEAVE.request primitive. The NLME-LEAVE.confirm primitive returns a status of either SUCCESS indicating that the request to transmit was successful, or FAILURE indicating that the request to transmit was failed.

9.6.6.3 NLME-LEAVE.indication primitive

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

NLME-LEAVE.indication (

ShortAddr

)

Table 77 specifies the parameters for the NLME-LEAVE.indication primitive.

Table 77 – NLME-LEAVE.indication primitive parameters

Name	Data type	Valid range	Description
ShortAddr	Unsigned16	0~65535 (Unicast address)	The network address

On receipt of the NLME-LEAVE.request primitive, the network layer will invoke a NLME-LEAVE.indication primitive and inform the upper layer.

9.6.6.4 NLME-LEAVE.response primitive

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

NLME-LEAVE.response (

PhyAddr,
Status

)

Table 78 specifies the parameters for the NLME-LEAVE.response primitive.

Table 78 – NLME-LEAVE.response primitive parameters

Name	Data type	Valid range	Description
PhyAddr	Unsigned64	$0\sim2^{64}-1$	Physical address of the device
Status	Unsigned8	0=SUCCESS, 1=FAILURE, 2~255=reserved	Executing result of the request

NLME-LEAVE.response primitive is used to indicate the leaving device whether the leaving request has been accepted.

9.6.6.5 Time sequence for device leaving

9.6.6.5.1 Time sequence for field devices leaving the star network

Field device leaving includes the following two cases:

- Active leaving: For field device, ASL invokes NLME-LEAVE.request to send a leaving request to the routing device which is its cluster head. After receiving the DLME-LEAVE.confirm from DLL, NL of the field device transmits NLME-LEAVE.confirm to ASL. For routing device, after receiving the DLME-LEAVE.indication from DLL, NL of the routing device transmits NLME-LEAVE.indication to ASL. ASL invokes NLME-LEAVE.response to send a leaving response to the field device.

Figure 75 illustrates time sequence of field device active leaving.

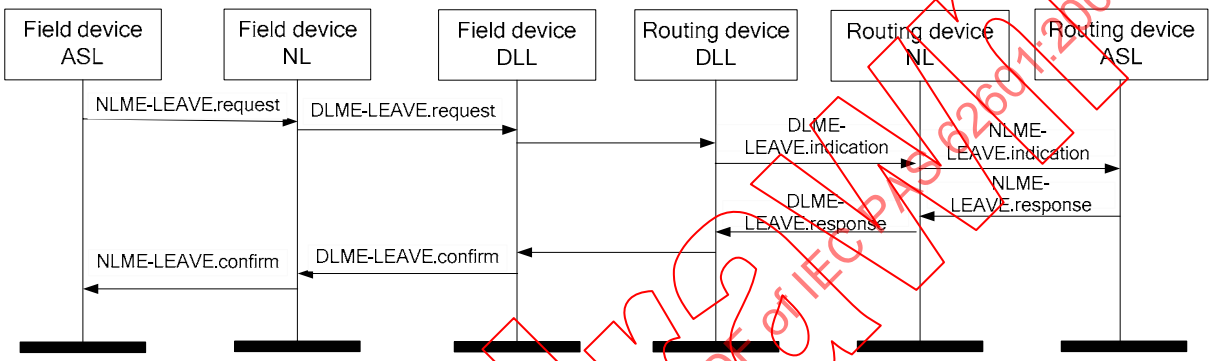


Figure 75 – Time sequence of field device active leaving

- Passive leaving: For routing device, ASL invokes NLME-LEAVE.request to send a leaving request to the field device. After receiving the DLME-LEAVE.confirm from DLL, NL of the routing device transmits NLME-LEAVE.confirm to ASL. For field device, after receiving the DLME-LEAVE.indication from DLL, NL of the field device transmits NLME-LEAVE.indication to ASL. ASL invokes NLME-LEAVE.response to send a leaving response to the routing device.

Figure 76 illustrates time sequence of field device passive leaving.

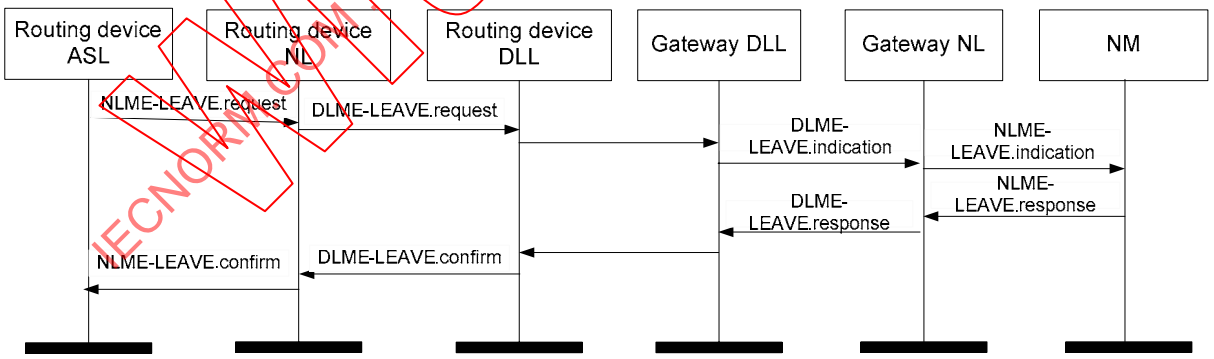


Figure 76 – Time sequence of field device passive leaving

9.6.6.5.2 Time sequence for routing devices leaving the mesh network

Routing devices connect to both the mesh network and the star network. Therefore, the routing device will inform its leaving to both gateway and field devices.

Routing devices leaving gateway include:

- Active leaving: For routing device, ASL invokes NLME-LEAVE.request to send a leaving request to the gateway. After receiving the DLME-LEAVE.confirm from DLL, NL of the gateway transmits NLME-LEAVE.confirm to network manager. For gateway, after

receiving the DLME-LEAVE.indication from DLL, NL of the gateway transmits NLME-LEAVE.indication to network manager. Network manager invokes NLME-LEAVE.response to send a leaving response to the routing device.

Figure 77 illustrates time sequence of routing device active leaving.

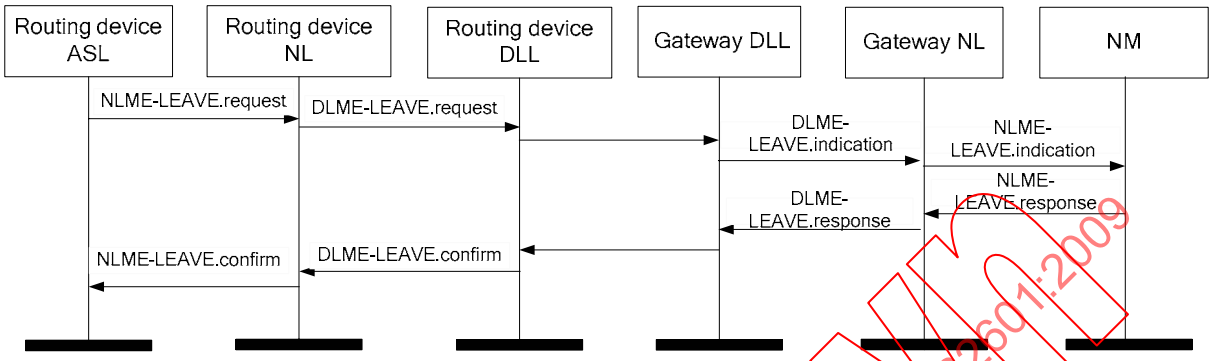


Figure 77 – Time sequence of routing device active leaving

- Passive leave: For gateway, network manager invokes NLME-LEAVE.request to send a leaving request to the routing device. After receiving the DLME-LEAVE.confirm from DLL, NL of the gateway transmits NLME-LEAVE.confirm to network manager. For routing device, after receiving the DLME-LEAVE.indication from DLL, NL of the routing device transmits NLME-LEAVE.indication to ASL. ASL invokes NLME-LEAVE.response to send a leaving response to the gateway.

Figure 78 illustrates time sequence of routing device passive leaving.

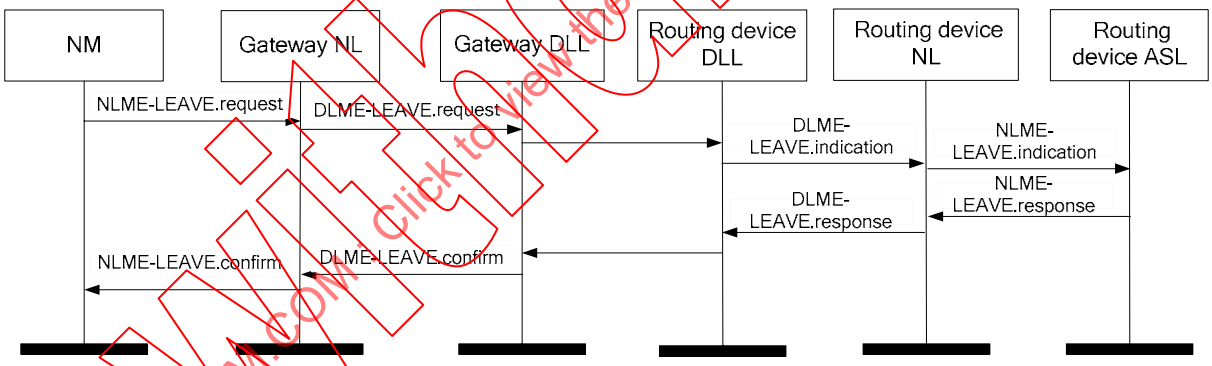


Figure 78 – Time sequence of routing device passive leaving

The leaving of routing device will trigger the passive leaving of all its field devices.

9.6.7 NLME-PATH_FAILURE.indication primitive

The semantics of NLME-PATH_FAILURE.indication primitive is described as follows:

NLME-PATH_FAILURE.indication (

Route_ID,
SrcAddr,
DstAddr

)

Table 79 specifies the parameters for the NLME-LEAVE.response primitive.

Table 79 – NLME-PATH_FAILURE.indication primitive parameters

Name	Data type	Valid range	Description
Route_ID	Unsigned16	0~65535	Route ID of failing path
SrcAddr	Unsigned16	0~65535(Unicast address)	The 16-bit source address of packets.
DstAddr	Unsigned16	0~65535	The 16-bit destination address of packets.

In WIA-PA networks, path status is evaluated to indicate the path health condition using the RetryCounter attribute of the NL-MIB. When a path fails, the network layer invokes the NLME-PATH_FAILURE.indication primitive to report it to the application sub-layer.

9.7 Network layer management information Base (NL-MIB)

9.7.1 General

The NL-MIB is used to store network layer attributes.

9.7.2 Unstructured attributes

Table 80 specifies the unstructured attributes in the management information base.

Table 80 – Unstructured attributes

Identifier	Name	Data type	Range	Access	Default	Description
0	Router_Capable	Unsigned8	0=routing device, 1=field device, 2~255=reserved	R	According to the type of the device	A flag to indicate if the device is a routing device or not.
1	Network_Address	Unsigned16	0~65535(Unicast address)	R/W	N/A	The 16-bit network address of the device, assigned by the network manager or by the network management agent.
2	Max_NSDU_Size	Unsigned16	0-65535	R	N/A	Maximum service data unit size supported by network layer.
3	AckTimeOut	Unsigned32	$0\sim2^{32}-1$	R/W	N/A	Defined by the network manager, if time out, retransmission the packet.
4	Net_Address_Assign	Unsigned8	0=no, 1=yes, 2~255=reserved	R/W	0	A flag indicating whether the network address has been assigned.
5	PacketsFromDLL	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from DLL.
6	PacketsFromDLLRejected	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from DLL rejected by NL.
7	PacketsFromDLLAccepted	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from DLL accepted by NL.
8	PacketsFromAL	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from ASL.
9	PacketsFromALDropped	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from ASL dropped by NL.
10	PacketsOutToDLL	Unsigned32	$0\sim2^{32}-1$	R/W	0	Count of the packets from ASL forwarded by NL.

9.7.3 Structured attributes

Table 81 specifies the unstructured attributes in the management information base.

Table 81 – Unstructured attributes

Identifier	Name	Data type	Range	Access	Default	Description
11	Route_Table	NLRoute_Tbl structure		R/W	N/A	Routing table used by routing device, consisted of Route ID, destination address, next hop address and VCR ID, see in Table 82

Every device in the network maintains a routing table and a path status table to control the procedure of communication.

The description of the routing table refers to 9.3.3.2.

Table 82 specifies the NLRoute_Tbl structure of routing table.

Table 82 – NLRoute_Tbl structure

Name	Data type	Valid range	Attribute description
*Route_ID	Unsigned16	0~65535	A Unique ID identified a routing.
Destination_Address	Unsigned16	0~65535	Destination address of a packet.
Next_Hop	Unsigned16	0~65535	Next hop address of a packet.
VCR_ID	Unsigned16	0~65535	A Unique ID identified a VCR.
RetryCounter	Unsigned8	0~255	A counter to record end-to-end retries.
NOTE * indicates a key field.			

10 Application Layer

10.1 Overview

10.1.1 General

WIA-PA Application Layer (AL) is composed of the User Application Process (UAP) and the Application Sub-layer (ASL). The AL defines software objects that interact with the industrial processes. It also defines communication services to support the communications between multiple objects for the distributed applications in the industry field environment.

WIA-PA defines the structure of distributed applications; it does not include the specific implementation. In addition, the local operations between the user application objects are not defined in WIA-PA.

10.1.2 AL structure

The structure of AL is shown in Figure 79.

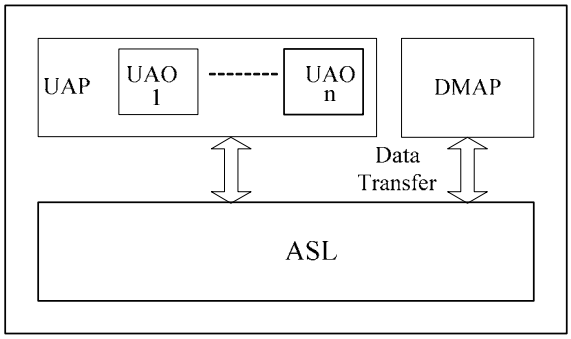


Figure 79 – Structure of application layer

The distributed applications of industrial field can be implementated by UAP. A UAP consists of one or more user application objects (UAOs). In one device, the local UAOs can interact with each other. The UAOs in different devices can interact with each other through the communication services provided by the AL.

ASL provides transparent data transmission services for UAP and DMAP.

10.1.3 Functions of UAP

The UAP has the following functions.

- UAP collects physical data information, such as the process data information of temperature, pressure and flow from the industrial filed. After processing these data, such as conversion, linearization, compensation, filtering, UAP will compute and generate output based on the data or the data from other devices. It completes the control operations through the actuators.
- UAP produces and publishes alarm: UAP should send out alarm when the physical data is beyond the limits or the state switch of UAO.
- The interoperability with other fieldbus devices could be achieved through UAP.

10.1.4 Functions of ASL

ASL provides data transfer services and management services.

- Data transfer services: ASL provides end-to-end (intra-cluster/inter-cluster) transparent data communication services for the user application process, it supports client/server, publisher/subscriber and report/sink data transmissions.
- Management services: ASL supports local and remote device management. AL MIB can be remotely modified/configured through the AL management services.

10.2 UAP

10.2.1 General

According to the definition of ISO/OSI reference model, UAP is the element processing information for a specific application; it is a part of the distributed applications that resides in the WIA-PA device.

The UAP can be constructed by the “Industrial process measurement and control systems function modules” defined in IEC 61499 or can be constructed by the “Function blocks for process controlling” defined in IEC 61804. It can also be the application processes defined by the users. That is to say, the construction of UAP can be achieved through function blocks or other approaches.

A UAP can be implemented by two methods, as shown in Figure 80.

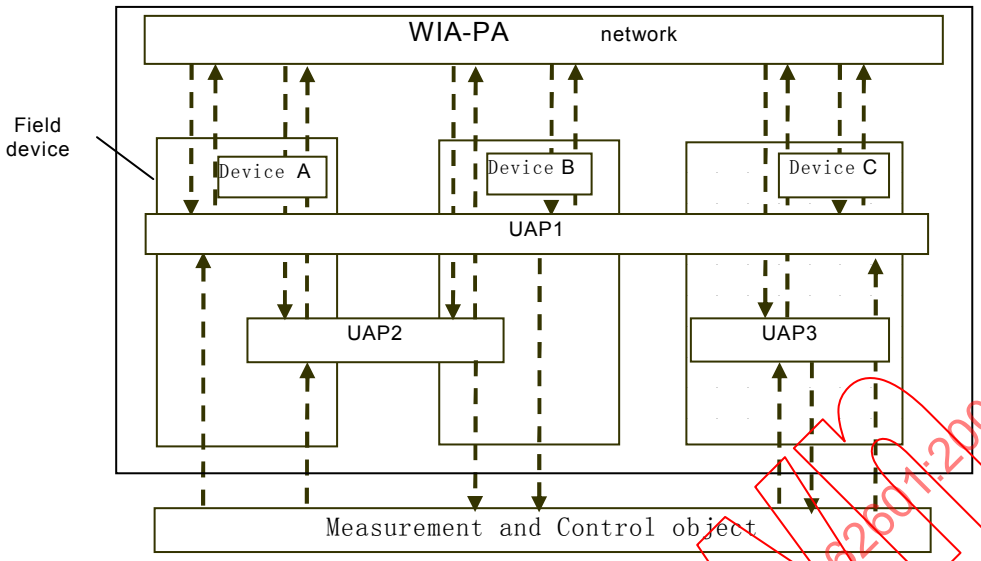


Figure 80 – User application process

The UAP can be constructed by several UAOs within one device, shown as UAP3 in Figure 80. It can also be constructed by several UAOs which reside in different devices, shown as UAP1 and UAP2 in Figure 80.

10.2.2 UAO

10.2.2.1 General

UAP consists of the user application objects. Each user object can be defined according to different functions. They can be defined by the function block technique or by other approaches.

To support the interoperability between different devices, each UAO in every device has its own description information. UAO description information is used to describe the profile that the object follows, the object structure and object parameters.

The addressable information of each application object is the Object ID. If some parameters can be operated remotely, they can also have the addressable attributes, such as using the index to represent the parameters.

10.2.2.2 Functions of UAO

UAO collects and processes the data from industrial process (such as temperature, pressure, flow) through different sensors.

The functions of UAO are described as follows.

- Range conversion

Range conversion is a linear transformation, which converts a signal data from one unit scale to the other unit scale in order to perform the following measurement and computation. For example, 4 mA to 20 mA signals will be converted into 0 kPa to 10 kPa. Range conversion can also convert the data with unit into data with no unit, such as percentage, and vice versa.

- Linearization

Because the characteristics of many sensors are not linear or even irregular, it is necessary to do linearization when dealing with sensor data. Linearization is to make the calculation

value being close to the real value as much as possible by using the formula or more data points.

- Compensation

The measurement value of a sensor may appear as a linear migration owing to the impact of the environment. So, it is necessary to make a compensation for this excursion. For example, the voltage of thermocouple is compensated by the value of the reference end. COMPENS_PAMAR defines the compensation type. Cold-end compensation can be internal (the device measures the temperature of the reference end through the internal installed sensor) or external.

- Filtering

In the industrial field, sensors may be interfered with, which results in a relatively big jump in the measurement values. In order to avoid this case, filtering is adopted in the measurement process to eliminate the bad influence of the jump signals.

- Storage

Storage is to copy the data to a temporary storage area for re-sending when the data is lost.

- Unit conversion

Unit conversion is to convert the unit of input signals to the unit that can be identified by I/O subsystems.

- Measuring timestamp

The current system time is added to the measurement data as a timestamp.

10.2.3 Instance of UAO

10.2.3.1 General

This PAS specifies using an object-oriented approach to define the AL UAO. UAO has its own attributes and methods. The defined contents include object name, attribute name, attribute identifier, data type of the attribute and the supported methods.

10.2.3.2 Analogy input object

Analogy Input Object (AIO) is used to sample the industrial process and convert the measured physical value to the data value with physical dimension. Table 83 shows the AIO object.

Table 83 – AIO object

Object Name		AIO		
Attribute Name	**Attribute Identifier**	**Data Type**	**Supporting Method**	**Description**
Object Identifier	Key Identifier	Unsigned8	Read	The unique identifier of an Object that is used for object addressing.
Process Value	0	Float	Read	Industrial process variable.
Out Value	1	Float	Read /Publish	Output value based on PV.
ASN	2	Unsigned48	Read /Write	Absolute slot number.
Scale	3	Scale	Read /Write	The measurement range for the conversion of actual measured physical value.
Action	4	Unsigned8	Read /Write	Start-up /stop measurement process.
Sample Period	5	Unsigned32	Read /Write	Used for setting sampling cycle.
High Limit	6	Float	Read /Write	Alarm high limit.
Low Limit	7	Float	Read /Write	Alarm low limit.

10.2.3.3 Analogy Output Object

Analogy Output Object (AOO) is used to transform output values to the values that hardware channels require, as shown in Table 84.

Table 84 – AOO object

Object Name		AOO		
Attribute Name	**Attribute Identifier**	**Data Type**	**Supported Method**	**Description**
Object Identifier	Key Identifier	Unsigned8	Read	The unique identifier of an object, used for object addressing.
Out Value	0	Float	Read /Write /Publish	The output data to the actuator based on the input data.
Read Back Value	1	Float	Read	Feedback data got from actuator.
Input Value	2	Float	Read /Write /Subscribe	Input data coming from other UAOs or manually input.
Scale	3	Scale	Read /Write	The measurement range for the conversion of actual measured physical value.
Action	4	Unsigned8	Read /Write	Start-up/stop output.
Output Period	5	Unsigned32	Read /Write	Used for setting output cycle.
High Limit	6	Float	Read /Write	Alarm high limits.
Low Limit	7	Float	Read /Write	Alarm low limits.

10.2.3.4 Method format

The data transmitted between UAP has fixed formats and can be identified through different methods.

For the read request data format, see Table 85.

Table 85 – Read request data format

Byte Length	1	1	1
Field name	Method identifier	Object identifier	Object parameter index

For the read response data format, see Table 86.

Table 86 – Read response data format

Byte Length	1	1	1	1	Variable
Field name	Method identifier	Object identifier	Object parameter index	data length	data

For the write request data format, see Table 87.

Table 87 – Write request data format

Byte Length	1	1	1	1	Variable
Field name	Method identifier	Object identifier	Object parameter index	Data length	Data

For the write response data format, see Table 88.

Table 88 – Write response data format

Byte Length	1	1	1	1
Field name	Method identifier	Object identifier	Object parameter index	Status

For the publish data format, see Table 89.

Table 89 – Publish data format

Byte Length	1	1	1	1	Variable
Field name	Method identifier	Object identifier	Object parameter index	data length	data

For the report data format, see Table 90.

Table 90 – Report data format

Byte Length	1	1	1	1	Variable
Field name	Method identifier	Object identifier	Object Parameter index	Data length	Data

For the report Ack data format, see Table 91.

Table 91 – Report ack data format

Byte Length	1	1	1	1
Field Name	Methods identifier	Object identifier	Object parameter index	Status

10.3 Application Sub-layer

10.3.1 General

ASL provides data communication service between two or more application objects in the WIA-PA network.

- Application Sub-layer Data Entity （ASLDE）is defined in ASL, it supports data communication service. It provides transparent interfaces for data sending and receiving for the upper layer.
- Application Sub-layer Management Entity (ASLME）is defined in ASL. It provides management services of AL management information maintenance and configuration.

10.3.2 Application sub-layer data entity

10.3.2.1 General

ASLDE provides data communication services for devices in the WIA-PA network, including sending data to the network layer and receiving data from the network layer. It supports the transport of the application data unit between two application entities.

ASL supports three communication modes: Client-Server mode、Publisher-Subscriber mode and Report-Sink mode.

- Client-Server mode supports unicast transmission of dynamic, aperiodic information.
- Publisher-Subscriber mode supports transmission of preconfigured periodic information.
- Report-Sink mode supports transmission of aperiodic information transmission, such as alert and event.

Application layer data transfer services include three primitives: data transmit request primitive, data transmit confirm primitive and data transmit indication primitive.

10.3.2.2 ASLDE-DATA.request

The upper layer sends data to the WIA-PA network by issuing the ASLDE-DATA.request primitive to the ASL. The upper layer must provide enough information in order to encode the AL data frame header according to the format of application layer frame.

ASLDE-DATA.request (

VCR_ID,

TransMode,

AsduLength,

Asdu

)

Table 92 lists the parameters for ASLDE-DATA.request primitive.

Table 92 – ASLDE-DATA.request parameters

Name	Type	Byte Length	Description
VCR_ID	Unsigned16	2	The VCR used for the transmission.
TransMode	Bitmap	1	The communication mode for transmitting the data: 0b0001:Client—Server mode, request; 0b1001: Client—Server mode, response; 0x0010:Publisher—Subscriber mode; 0x0100:Report—Sink
AsduLength	Unsigned8	1	The number of octets comprising the ASDU to be transferred.
Asdu	Set of octets	—	The set of octets comprising the ASDU to be transferred.

TransMode parameter value 0b0001 denotes the data is transferred using Client-Server mode. In this mode, the destination address cannot be 0Xffff; otherwise ASL will issue ASLDE-DATA.confirm primitive to the upper layer with Status parameter set to INVALID_ADDRESS to indicate an invalid address.

TransMode parameter value 0x0010 denotes data is transferred using Publisher—Subscriber mode. In this mode, destination address is 0xffff indicates that the destination is all the devices in the WIA-PA network, and the data should be broadcasted to every device in the network. When the destination parameter is not 0xffff, ASL will issue ASLDE-DATA.confirm primitive to the upper layer with Status parameter set to INVALID_ADDRESS to indicate an invalid address.

TransMode parameter value 0x0100 indicates data is transferred using Report—Sink mode. In this mode, the destination address cannot be 0Xffff; otherwise the ASL sub-layer will issue ASLDE-DATA.confirm to the upper layer with Status parameter set to INVALID_ADDRESS to indicate invalid address.

When ASL sub-layer receives the data request primitive from the upper layer, it encodes the ASDU according to the standard frame format of the ASL sub-layer, and transports it to the network layer. After transporting the data frame, the ASL should issue ASLDE-DATA.confirm primitive to the upper layer with the Status parameter set to SUCCESS.

10.3.2.3 ASLDE-DATA.confirm

The ASLDE-DATA.confirm primitive is issued to upper layer to indicate the transmission result. When the data is transferred successfully, this Status parameter in this primitive should be set to SUCCESS. Otherwise, the Status parameter should be set to corresponging error code to indicate failure reason.