INTERNATIONAL STANDARD

ISO/IEC 17821

First edition
2015-04-15

Information technology — Specification of low power wireless mesh network over channel-hopped TDMA links

Technologies de l'information — Spécification des réseaux maillés sans fil à faible puissance par liens AMRT à saut de canaux

Reference number
ISO/IEC 17821:2015(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

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

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

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

Introduction

This working draft defines the protocol for the low-power wireless mesh network over channel-hopped TDMA links (LPWMN). The objective of LPWMN is to define a wireless mesh network specification, which is relatively lightweight compared to the legacy WPAN network specifications and can maximize usefulness of the channel-hopped TDMA link which is followed by the deterministic and synchronous multichannel extension (DSME) MAC specified in IEEE 802.15.4e-2012 amendment.

In recent years, there is market demand for applying the low-energy short range communication of the WSN to the networks for mission-critical services or real-time services including remote monitoring and alarming of health devices or medicine equipment, sensing and actuation control of process automation, and voice service over the low-energy short range networks. To provide more reliable link and deterministic delay for the mission-critical services, enhancements of the IEEE 802.15.4-2006 MAC specification was started in March 2008 and the draft of the amendment is approved in February 2012. In the IEEE 802.15.4e-2012 amendment, three types of TDMA and two types of channel diversity function are added as optional MAC features.

These new MAC features introduce two attributes to be managed in the network layer, which includes time slots and radio channels. To employ the new enhanced MAC for the reliable and real-time services in the low-cost, low-power, short-range communication network, the network protocol needs to be designed for managing the network resources including time slots and radio channels. The LPWMN is a network specification over the DSME of IEEE 802.15.4e-2012 amendment. The LPWMN is applicable to industrial applications that require a loss sensitive large wireless network guaranteeing deterministic end-to-end delay with low-power resource-constrained communication nodes.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning the functional procedure and message structure of LPWMN given in Clauses 5, 6, 7, and 8, and a patent concerning the authentication and key establishment protocols given in Annex B.

ISO and IEC take no position concerning the evidence, validity, and scope of this patent right.

The holder of this patent right has assured the ISO and IEC the willingness to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the following:

— Patent holder: Electronics and Telecommunications Research Institute (ETRI)

  Address: 138 Gajeongno, Yuseong-gu, Daejeon, 305-700, Korea;

— Patent holder: China IWNCOMM Co., LTD

  Address: A201, QinFengGe, Xi'an Software Park, No. 68 Keji 2rd, Xi'an Hi-Tech Industrial Development Zone, Xi'an Shaanxi, P.R.China. 710075.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain online databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up-to-date information concerning patents.

Information technology — Specification of low power wireless mesh network over channel-hopped TDMA links

1 Scope

This International Standard defines the network specification for devices, which are operated on IEEE Std. 802.15.4-2011 PHY, capable to support the channel-hopped TDMA links of the DSME MAC of IEEE Std. 802.15.4e-2012, to provide low-cost communication network that allows reliable, deterministic-latency, and scalable wireless mesh connectivity.

This International Standard provides the following:

— DSME MAC link control;

— unbalanced cluster-tree based network formation;

— directional multiple grades mesh connection;

— link-path routing and data forwarding;

— link and link-path maintenance.

2 Normative references

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

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

IEEE Std 802.15.4e-2012, *IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC sublayer*

3 Terms and definitions

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

3.1
coordinator
device in a low-rate wireless personal area network (LR WPAN) that provides synchronization services to other devices in the LR WPAN

3.2
device
any entity containing an implementation of the IEEE 802.15.4e-2012 DSME medium access control and physical interface to the wireless medium

Note 1 to entry: A device may be a reduced-function device or a full-function device.

3.3
frame
format of aggregated bits from a medium access control sublayer entity that are transmitted together in time

3.4
full-function device
FFD
device capable of operating as a coordinator

3.5
personal area network (PAN) coordinator
coordinator that is the principal controller of a PAN

Note 1 to entry: An IEEE 802.15.4 network has exactly one PAN coordinator.

3.6
reduced-function device
RFD
device that is not capable of operating as a coordinator

4 Abbreviations

The following acronyms are used in this document.

AKEP	Authentication and Key Establishment Protocols
CSMA-CA	Carrier Sense Multiple Access – Collision Avoidance
DLCE	DSME MAC Link Control sublayer Entity
DLIB	DSME MAC Link network Information Base
DLNE	DSME MAC Link Network sublayer Entity
DLPDU	DSME MAC Link network sublayer Protocol Data Unit
DLSDU	DSME MAC Link network sublayer Service Data Unit
DSME	Deterministic and Synchronous Multi-channel Extension
FFD	Full Function Device
GTS	Guaranteed Time Slot
ID	Identification
IDAP	ID-based Authentication Protocol
IEEE	Institute of Electrical and Electronics Engineers
LPWMN	Low-Power Wireless Mesh Network
LSAP	Light-weight Shared-key Authentication Protocol
MAC	Media Access Control
MCPS	MAC Common Part Sublayer
MIB	MAC Information Base
MLME	MAC Layer Management Entity
MSDU	MAC Service Data Unit
PAN	Personal Area Network

PHY	Physical Layer
PTK	Pairwise Temporal Key
QoS	Quality-of-Service
RF	Radio Frequency
RFD	Reduced Function Device
SAP	Service Access Point
TDMA	Time Division Multiple Access
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

5 General description

5.1 General

The LPWMN is a low cost communication network that allows reliable, deterministic-latency, and scalable wireless mesh connectivity with low-power and low-rate WPAN devices. The main objectives of an LPWMN are optimal utilization of the channel-hopped TDMA links, cluster-tree based mesh network formation, and multiple grade of path selection, while maintaining a simple procedure and a lightweight protocol.

Some of the capabilities provided by this standard are as follows:

— resource management of the DSME MAC links

— beacon-enabled multi-hop network formation

— unbalanced cluster-tree based address assignment

— directional multiple grades mesh connection

— address based mesh routing

— unicast data forwarding with deterministic-latency

— load balanced path maintaining

This standard defines the link network specification for devices that are operated on IEEE Std. 802.15.4-2011 PHY capable to support the channel-hopped TDMA links of the DSME MAC of IEEE Std. 802.15.4e-2012.

5.2 Components of the LPWMN

The LPWMN consists of a gateway router, routers, and devices, which follow the IEEE Std. 802.15.4-2011 and have the DSME MAC sublayer of the IEEE Std. 802.15.4e-2012. The gateway router starts a network by configuring the attributes of the link network and forming the link network topology, and interconnects the LPWMN to an external network. The routers join a link network and forward frames through the DSME MAC links. The devices are reduced-function device (RFD) or full-function device (FFD) of the IEEE Std. 802.15.4-2011 and perform applications.

The LPWMN operates in the star topology and the peer-to-peer topology, as shown in Figure 5.1. A device typically establishes a star topology to routers and is either the initiation point or the termination point for link network communications. The gateway router is the primary controller of the LPWMN. The gateway router and routers are capable to establish a peer-to-peer topology. All devices operating on a link network have unique 64bit MAC addresses and 16bit link network addresses. The link network address is allocated by the cluster group.

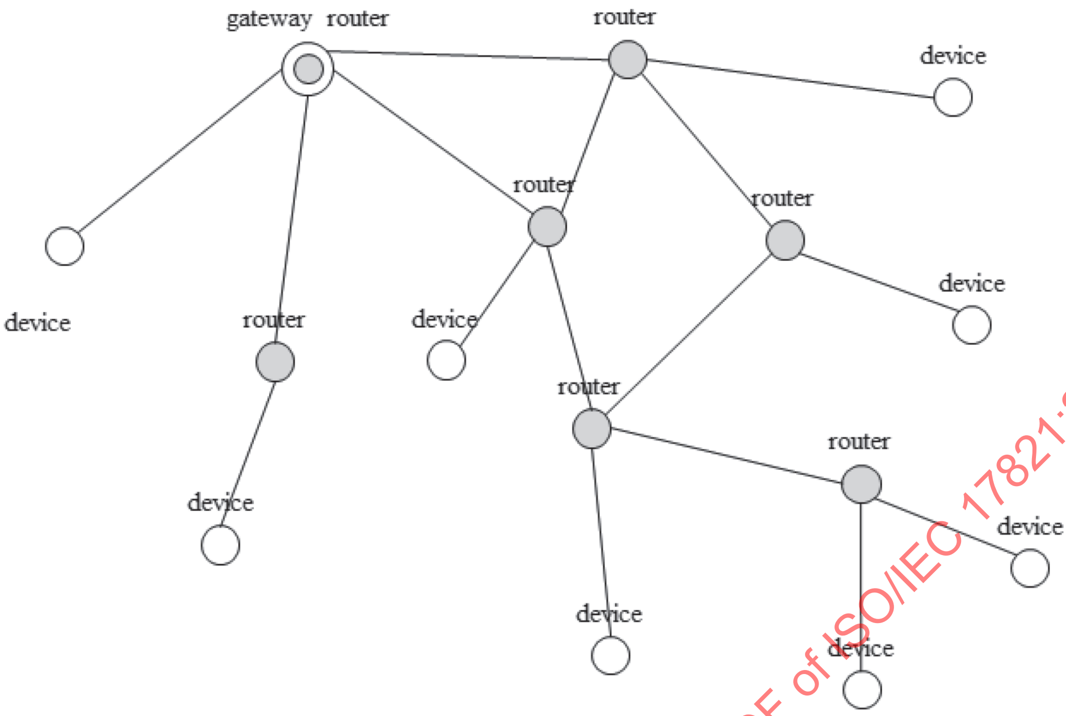

Figure 5.1 — An LPWMN topology

5.3 Architecture of the LPWMN

The reference architecture of the LPWMN is defined in terms of layers, as illustrated in Figure 5.2. The LPWMN provides link control layer interfaces to the next higher layer and link network layer interfaces to the next higher or to the application layer. The LPWMN operates over the IEEE Std 802.15.4e-2012 DSME MAC Sublayer and the IEEE Std 802.15.4-2011 PHY. The LPWMN offers services to the next higher layer.

The LPWMN consists of two sublayers: DSME MAC Link Control sublayer (DLC) and DSME MAC Link Network sublayer (DLN). The LPWMN DLC sublayer entity (DLCE) provides services to the next higher layer via the DSME MAC Link Control sublayer Service Access Point (DLC-SAP). The LPWMN DLN sublayer entity (DLNE) provides services to the next higher layer via the DSME MAC Link Network sublayer Service Access Point (DLN-SAP).

The DLC sublayer provides link connection control, link management services, data transmission on a link, and relaying frames with supporting of the DSME MAC sublayer via the MCPS-SAP and the MLME-SAP.

The DLN sublayer consists of link network formation block, addressing block, routing block, and link network management block, and data processing block which provides receiving function and forwarding function for the frames received from the MAC layer or for the data received from the next higher layer.

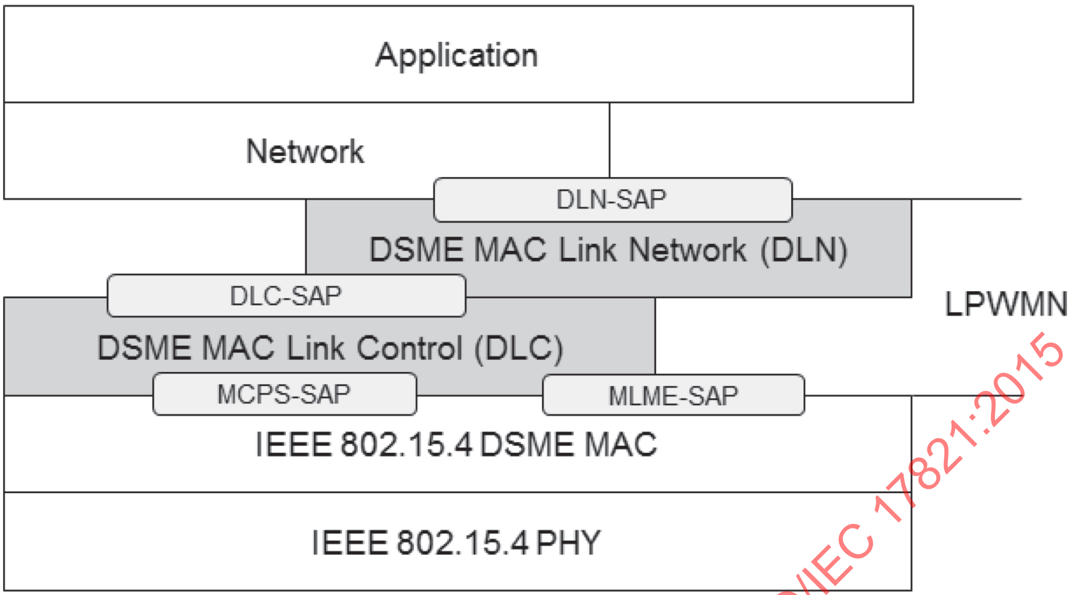

Figure 5.2 — Reference architecture of the LPWMN

5.4 Functional overview

5.4.1 Link network formation

The LPWMN establishes a link between two neighbour devices and a virtual link between two devices multi-hopped apart, as shown in the Figure 5.3. The LPWMN provides a routed link-path, link network, which is constituted of links and virtual links from a source to a destination device.

The link connects the two neighbour devices: link network router 2 and device 3 are neighbour and a link connects two devices. The virtual link is the multi-hop link connection through the routers which perform frame relaying instead of routed forwarding: network router 2 and device 6 are connected through router 4 and router 6 which relay the frames inward or outward. The routed link-path is provided by the DLN sublayer's routing functions: for connecting between device 4 and device 8, the LPWMN provides several routed link-paths which are the path 1 (device 4 - router 4 - link network router 2 - gateway router - link network router 3 - device 8), path 2 (device 4 - router 4 - link network router 2 - link network router 3 - device 8), and path 3 (device 4 - router 4 - router 5 - link network router 3 - device 8).

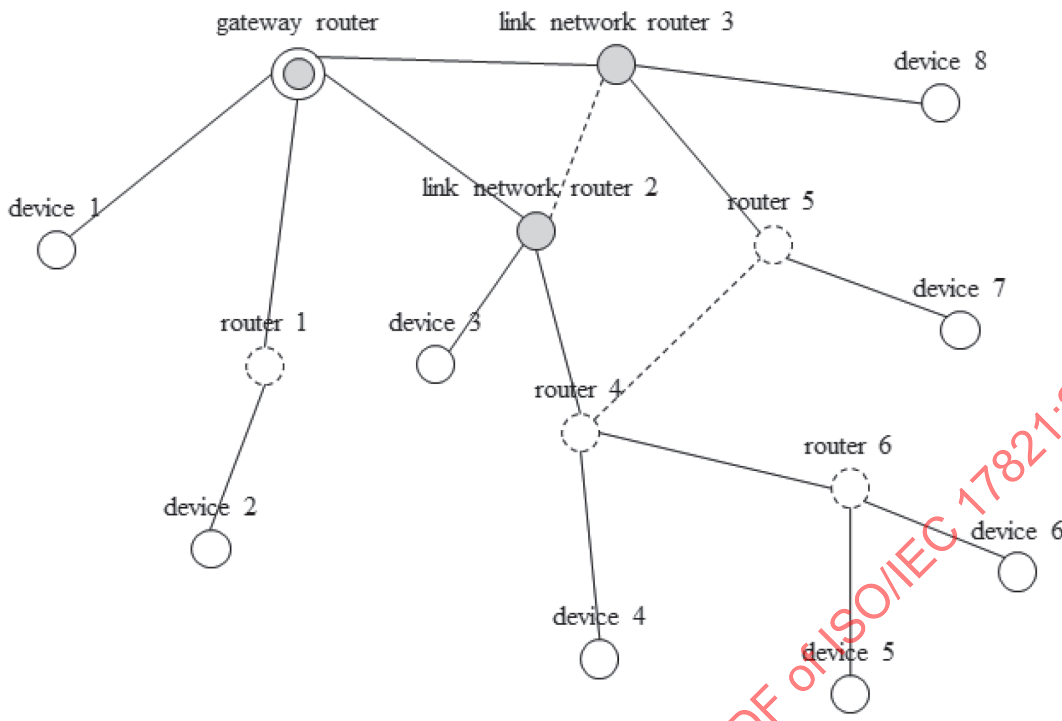

Figure 5.3 — Link, virtual link, routed link-path in the LPWMN

The link network formation of the LPWMN is performed in two stages: link connection and link network routing. In the link connection phase, the DLC sublayer entity of the LPWMN scans the neighbour devices, selects the links to the neighbour devices, establishes connections, and maintains the links. In the link network routing phase, according to the request of the next higher layer of the LPWMN, a set of virtual link is established and/or a set of routed path is established.

5.4.2 Link connection

The LPWMN is a link network that connects two devices by switching the time slots. The DSME MAC sublayer provides the contention access period (CAP) and guaranteed timeslots (GTS) in a beacon interval for communicating, as illustrated in Figure 5.4.

A CAP link is active periodically and is used for the bidirectional data transmission. A GTS link in the LPWMN is established by assigning the offset of channel hopping sequence and the DSME timeslots between two devices. To allocate the DSME timeslots between devices, the LPWMN issues MLME-DSME-GTS.request primitive with attribute values on the link which are the direction of DSME-GTSs, the number of slots to be allocated, preferred slots, and current information on the slot availability in one hop neighbourhood of the requesting device.

The LPWMN GTS link can be established not only between two directly connected devices, but also between two devices which can be connected with multi-hop routed links. The LPWMN provides a primitive for establishing a routed link connection. To form a link-path from a source device to a destination device, each device on the path requests to allocate DSME timeslots. As a result of the success of the routed link connection, the link connection identifier will be provided to the next higher layer of the source device.

The LPWMN GTS link can be used as shared link or dedicated link. The default shared link is established during joining a network. An inward rx timeslot and an inward tx timeslot of a link to the inner router are assigned as the default shared link and the default path to the gateway router is established. If the next higher layer of the LPWMN needs a dedicated link, it is required to setup a link connection before sending data. To establish a dedicated link connection, link network layer routing is performed through the routers on the routed link-path from a source device to a destination device. To establish a bidirectional dedicated link, it needs to establish an inward dedicated link and an outward dedicated link separately. The link setup procedure from the source device and the link setup procedure from the destination device are required.

The dedicated link transmits a frame received only at the source device of the link. The shared link may transmit a frame received at the routers on the link-path to the destination device of the link.

When the next higher layer of the LPWMN requests to transmit data, according to the required quality of transmission, it selects a type of data transmission, which specifies the type of link for communications, the recovery procedure, and flow control. The LPWMN provides six types of data transmission.

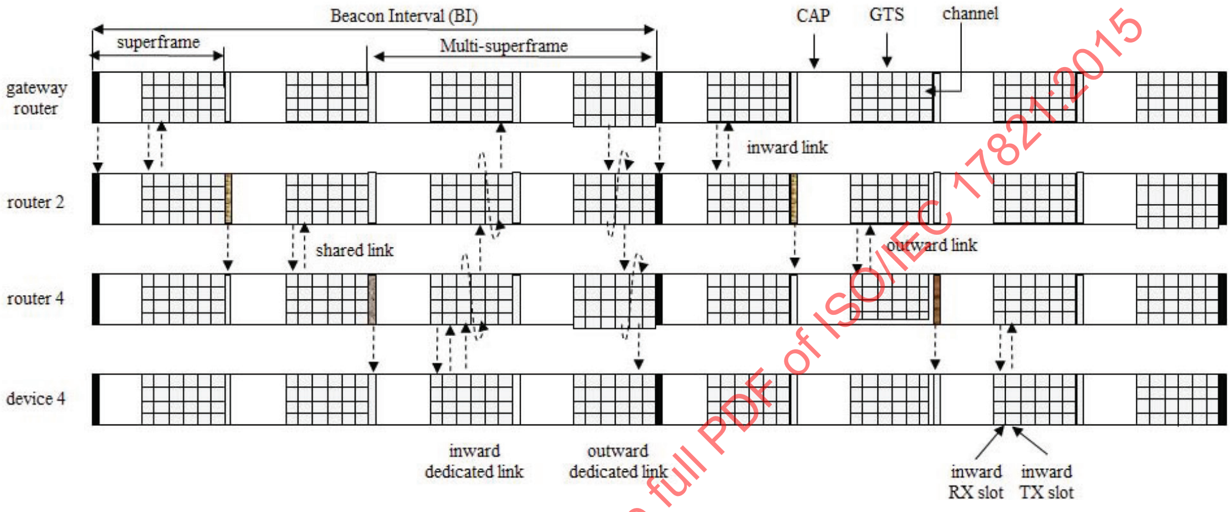

Figure 5.4 — CAP link, shared link, dedicated link of the LPWMN

5.4.3 Unbalanced cluster-tree addressing

To support the fast formation of a network and to provide address based tree routing, the distributed address allocation scheme with an unbalanced cluster-tree structure is designed as shown in Figure 5.5.

The 16-bit addressing space is divided two parts for identifying a cluster (cluster ID) and a device in the cluster (locator ID). The gateway router manages the cluster identifier space and assigns a cluster identifier when the root router of a cluster joins a link network. The root router of a cluster assigns an identifier to a device by using the distributed address allocation scheme with the maximum depth of the cluster (L), the maximum number of devices connected to a router (D), and the maximum number of routers among devices connected to a router (R) values. The locator identifier of the root router of a cluster is 0 and the locator identifier of a device connected to a router h-hopped from the root router of a cluster shall be assigned as follows. If the device is a router, the locator identifier is calculated as follows: *device identifier of a parent router + 1 + (sequential order of a router at cluster depth h - 1)*size of address block at cluster depth h*. If the device is an end device, the identifier is calculated as follows: *device identifier of a parent router + 1 + maximum number of a router at cluster depth h*size of address block at cluster depth h + sequential order of an end device at cluster depth h*. The sequential order of a device at each cluster depth is assigned from 1. The size of address block at cluster depth h, $B(h)$, is calculated as follows: If $R = 1$, $B(h) = 1+ D*(L- h-1)$. If $R \neq 1$, $B(h) = (1+D-R-D*R^{L-h-1})/(1-R)$.

In case of running out of address on the cluster tree or enlarging the network topology dynamically, a router could request assignment of a cluster to the gateway router as a root router of the cluster. The gateway router maintains the connectivity matrix among the clusters. The router provides the cluster-tree routing table based on the cluster connectivity matrix.

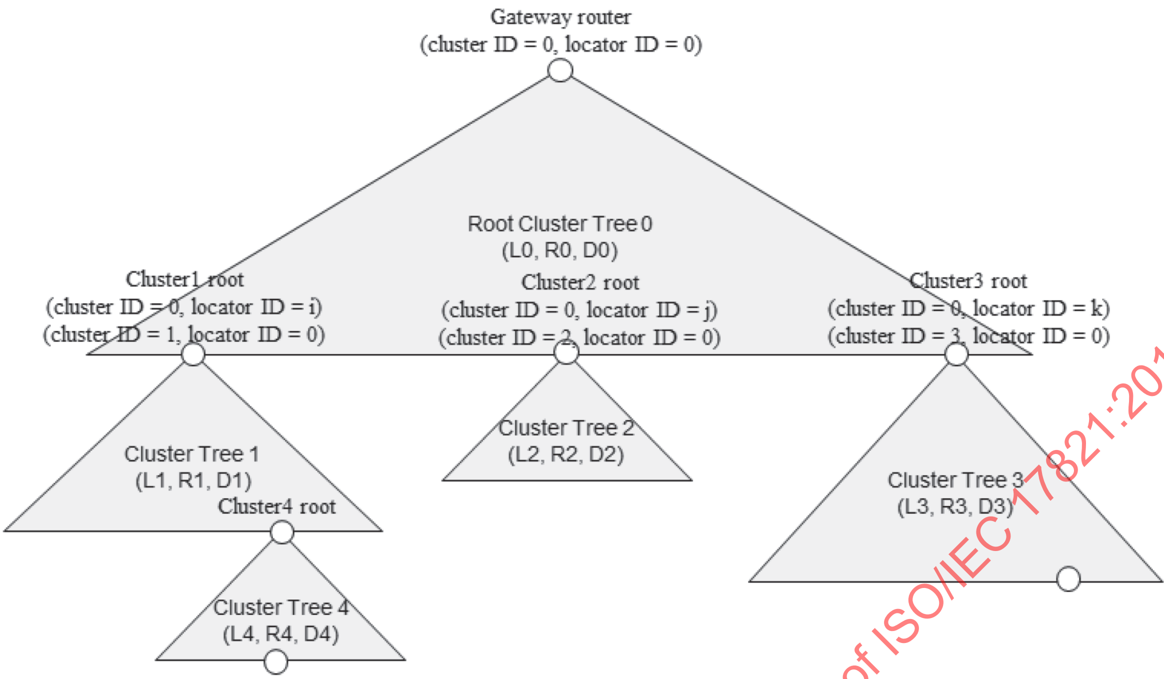

Figure 5.5 — Address assignment based on asymmetrical tree structure

5.4.4 Routing

The DSME MAC offers TDMA slots with two types of channel diversity function. The routing function concerns with not the only matter of connectivity, but considering a series of the time slot and channel number pairs which are allocated on the radio links in a path. To utilize the TDMA MAC optimally, the link network protocol has to offer the light-weight network protocol for reserving the time slots on a link from one end to the other end. The routing protocol has to support the fair load balancing among devices contended to reserve the time slots and channels.

The channels occupied by nodes within 2-hops range have to be excluded for avoiding the collision by hidden nodes. Based upon the neighbour device information, possible links to the inner router are searched by selecting time slot and channel number pairs.

When a device joins a link network, the cluster connectivity matrix is provided from the gateway router or the root router. The cluster connectivity matrix provides relations among the root routers of the clusters and inter-cluster mesh links. Selection of a route is performed based on the cluster identifier of reachable devices. The device of which the cluster level or cluster-tree level is closer to the destination is selected for forwarding a frame, as shown in Figure 5.6. A data packet is forwarded to the next hop on a reserved GTS link or a shared link, which is specified by a certain time slot and a hopping channel.

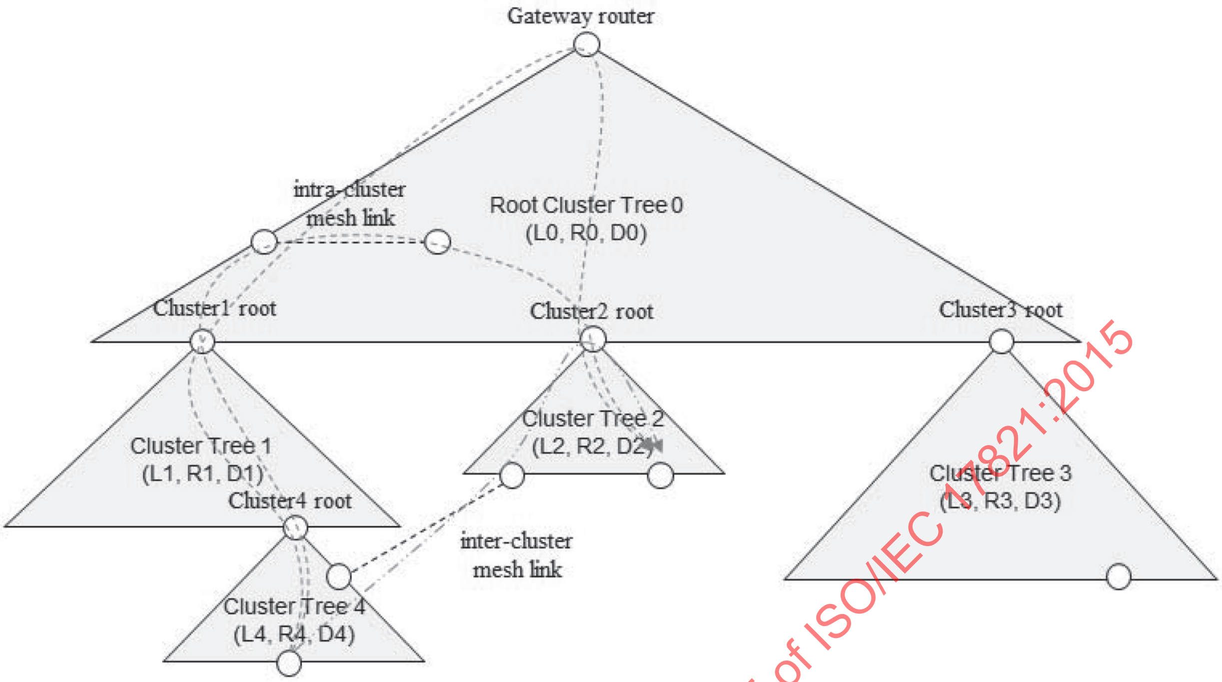

Figure 5.6 — Cluster-tree address based mesh routing

6 Functional description

6.1 Starting the LPWMN

6.1.1 Starting a link network

The gateway router which is not currently joined to a link network shall attempt to establish a new LPWMN. When the next higher layer of the device issues the DLN-START-NETWORK.request primitive, the procedure to establish a new LPWMN is initiated. If the device is not the gateway router or is joined to a link network, the DLNE shall terminate the procedure and notify the next higher layer of the illegal request by issuing the DLN-START-NETWORK.confirm primitive with the Status parameter set to INVALID_ REQUEST.

The DLNE shall reset the MAC sublayer by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE. The DLNE shall collect the data on the wireless environments and neighbour networks by requesting that the MAC sublayer performs an energy detection scan and an active scan in succession. The DLNE issues the MLME-SCAN.request primitive with the ScanType parameter set to ED and the ScanChannels parameter set to the ScanChannels parameter of the DLN-START-NETWORK.request primitive. On receipt of the results via MLME-SCAN.confirm primitive, the DLNE shall select the channels, whose energy levels are beyond an acceptable level, at most the number of channels specified in the HoppingSequenceLength parameter of the DLN-START-NETWORK.request primitive. The DLNE issues the MLME-SCAN.request primitive with the ScanType parameter set to ACTIVE and the ScanChannels parameter set to the ScanChannels parameter of the DLN-START-NETWORK.request primitive. On receipt of the results via MLME-SCAN.confirm primitive, the DLNE shall review the PANDescriptorList parameter. The DLNE collects the information on the neighbour IEEE 802.15.4-2011 PANs: the PAN identifier, channel number and channel page, superframe specification, and the time at which the beacon frame was received. The DLNE collects the information on the neighbour IEEE 802.15.4e-2012 PANs: the PAN identifier, superframe specification, DSME superframe specification, channel hopping specification, time synchronization specification, and beacon bitmap.

The DLNE may infer whether the channel is occupied or not by superimposing all the superframes of the neighbour networks at a specific time. The DLNE shall select a link network start time and channels, and determine the order of hopping channels. The length of hopping sequence and the hopping sequence are written to the *dlHoppingSequenceLength* attribute and the *dlHoppingSequenceList* attribute of the DSME MAC Link network Information Base (DLIB), respectively. If no suitable channel is found, the DLNE shall terminate the procedure and notify the next higher layer of the startup failure by issuing the DLN-START-NETWORK.confirm primitive with the Status parameter set to STARTUP_FAILURE.

The DLNE shall select a PAN identifier for the new LPWMN by choosing a random PAN identifier less than 0xffff, if the PANId parameter of the DLN-START-NETWORK.request primitive is already in use. The selected PAN identifier is set to the *macPANID* attribute of the MAC sublayer PIB by issuing the MLME-SET.request primitive. Once a PAN identifier is selected, the DLNE shall select a 16-bit link network address equal to 0x0000 and set the *dlLinkNetworkAddress* attribute of the DLIB equal to the selected link network address. If no unique PAN identifier can be chosen, the DLNE shall terminate the procedure and notify the next higher layer of the startup failure by issuing the DLN-START-NETWORK.confirm primitive with the Status parameter set to STARTUP_FAILURE.

At a specific selected time, the DLNE shall begin operation of the new PAN by issuing the MLME-START.request primitive with the PANCoordinator parameter set to TRUE and the CoordRealigment parameter set to FALSE to the MAC sublayer. The parameters of the MLME-START.request primitive shall be set according to those passed in the DLN-START-NETWORK.request, the chosen channel hopping sequence, and the chosen PAN identifier. The Multi-superframeOrder, ChannelDiversityMode, CAPEnable, GACKEnable, and Deferred BeaconEnable are assembled into the parameter DSMESuperframeSpecificaiton. The HoppingSequenceID with set to one, ChannelOffset with set to zero, and value of *dlHoppingSequenceLength* attribute and the *dlHoppingSequenceList* attribute of DLIB are assembled into the parameter HoppingDescriptor.

The status of the PAN startup is communicated back via the MLME-START.confirm primitive. On receipt of the status of the PAN startup, the DLNE shall inform the next higher layer of the status of its request by issuing the DLN-START-NETWORK.confirm primitive with the Status parameter set to the primitive returned in the MLME-START.confirm from the MAC sublayer.

6.1.2 Starting a router

When the next higher layer of the device issues the DLN-START-ROUTER.request primitive, the procedure to join the LPWMN as a new router is initiated, as shown in Figure 6.1. If the device is not the FFD or is joined to a link network, the DLNE shall terminate the procedure and notify the next higher layer of the illegal request by issuing the DLN-START-ROUTER.confirm primitive with the Status parameter set to INVALID_ REQUEST.

The DLNE shall reset the MAC sublayer by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE and reset the DLN sublayer by issuing the DLN-RESET.request primitive. The DLNE shall collect the data on neighbour networks by requesting that the MAC sublayer performs an active scan. The DLNE issues the MLME-SCAN.request primitive with the ScanType parameter set to ACTIVE and the ScanChannels parameter set to the ScanChannels parameter of the DLN-START-ROUTER.request primitive. On receipt of the results via MLME-SCAN.confirm primitive, the DLNE shall review the PANDescriptorList parameter to collect the information on the neighbour IEEE 802.15.4e-2012 PANs: the PAN identifier, superframe specification, DSME superframe specification, channel hopping specification, time synchronization specification, and beacon bitmap. The information on the neighbour router of which a PAN identifier is the PANId parameter of the DLN-START-ROUTER. request primitive is captured in the *dlNeighborDeviceTable* attribute of the DLIB. If no neighbour router can be found, the DLNE shall terminate the procedure and notify the next higher layer of the startup failure by issuing the DLN-START-ROUTER.confirm primitive with the Status parameter set to STARTUP_FAILURE.

The DLNE shall search its neighbour device table for selecting a suitable primary inner router. The DLNE selects a device that the depth to the gateway router from this device is the smallest among devices which the link quality for frames received from this device are beyond an acceptable level.

The DLNE shall associate the PAN through the primary inner router, as described in 6.2.

After completing the cluster connectivity matrix, the DLNE shall select the superframe duration to transmit the beacon by superimposing all beacon bitmaps of the neighbour routers.

The DLNE shall begin the router by issuing the MLME-START.request primitive with the router parameter set to the MAC sublayer. Upon receiving the MLME-START.confirm primitive, the DLNE shall notify the next higher layer by issuing the DLN-START-ROUTER.confirm primitive with the Status parameter. The MAC sublayer enters the operating mode to transmit its beacons and is ready to allow new devices to join the LPWMN.

If no superframe duration for the beacon can be chosen, the DLNE shall terminate the procedure and notify the next higher layer of the startup failure by issuing the DLN-START-ROUTER.confirm primitive with the Status parameter set to STARTUP_FAILURE.

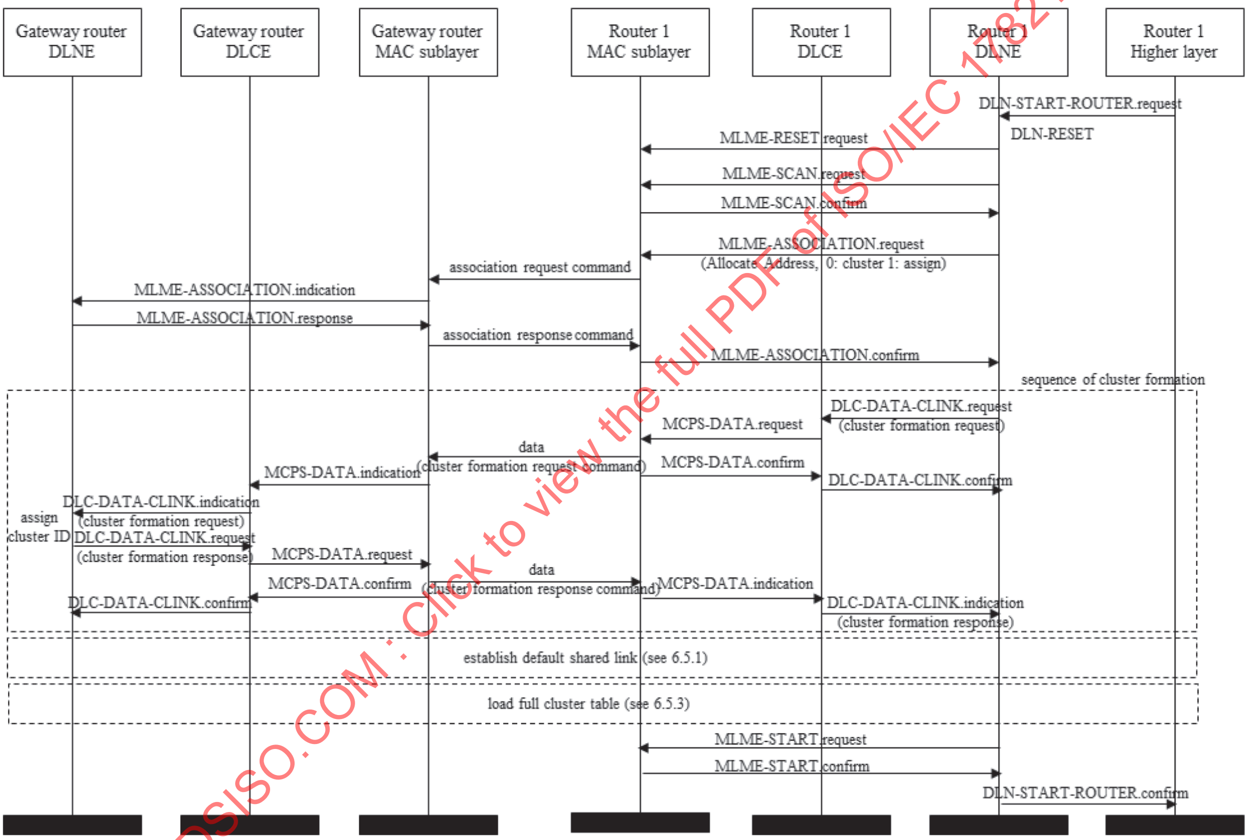

Figure 6.1 — Sequence of the starting a router

6.1.3 Starting a device

When the next higher layer of the device issues the DLN-START-DEVICE.request primitive, the procedure to join the LPWMN as a device is initiated. If the device is joined to a link network, the DLNE shall terminate the procedure and notify the next higher layer of the illegal request by issuing the DLN-START-DEVICE.confirm primitive with the Status parameter set to INVALID_ REQUEST.

The DLNE shall reset the MAC sublayer by issuing the MLME-RESET.request primitive with the SetDefaultPIB parameter set to TRUE and reset the DLN sublayer by issuing the DLN-RESET.request primitive. The DLNE shall collect the data on neighbour networks by requesting that the MAC sublayer performs an active scan. The DLNE issues the MLME-SCAN.request primitive with the ScanType parameter set to ACTIVE and the ScanChannels parameter set to the ScanChannels parameter of the DLN-START-DEVICE.request primitive. On receipt of the results via MLME-SCAN.confirm primitive, the DLNE shall review the PANDescriptorList parameter to collect the information on the neighbour IEEE 802.15.4e-2012 PANs.

The DLNE shall search its neighbour device table for a suitable inner router and select a device which the depth to the gateway router from this device is the smallest.

The DLNE shall associate the PAN through the primary inner router, as described in 6.2.

After completing the cluster connectivity matrix, the DLNE shall notify the next higher layer by issuing the DLN-START-DEVICE.confirm primitive with the Status parameter.

6.2 Joining a link network

For joining a link network, a device associates with the PAN through the MAC layer association procedure, requests a cluster formation, and establishes a shared link to the gateway router. The joining procedure is specified on the view of a device and on the view of the inner router.

6.2.1 Device procedure

When the device is designated as a root router of a cluster, the Allocate Address bit-field of the CapabilityInformation parameter set to zero. The capability information shall be stored as the value of the *dlCapabilityInformation* attribute of the DLIB.

The DLNE shall issue an MLME-ASSOCIATE.request primitive to the MAC sublayer. The CoordAddrMode is set to SHORT_ADDRESS, CoordPANId is set to the value of *dlPANId* of the DLIB, and CoordAddress is set to the address of the selected inner router. The bit-fields of the CapabilityInformation parameter shall have the values shown in Table 9.3. The Allocate Address bit-field of the CapabilityInformation parameter is always set to one.

The MLME-ASSOCIATE.confirm primitive with a status value of SUCESS received shall contain a 16-bit logical address unique to that network. The DLCE shall then update the value of *dlLinkNetworkAddress*, *dlHoppingSequenceLength*, *dlHoppingSequence*, *dlChannelOffset* attributes of the DLIB with the parameter of the MLME-ASSOCIATE.confirm primitive, and set the Device Type field of the corresponding neighbour device table entry to indicate that the neighbour is its parent.

When the device is designated as a root router of a cluster, on receipt of the MLME-ASSOCIATE.confirm primitive with a status value of SUCCESS, the DLNE shall request a cluster allocation by transmitting the link network management command frame containing the cluster formation request command payload by issuing the DLC-DATA-CLINK.request primitive, as shown in Figure 6.1.

Upon receiving the DLC-DATA-CLINK.confirm primitive, if the Length of Cluster Identifier Space field of the cluster formation response command payload is not zero, the cluster identifier is successfully assigned.

After allocating the link network address of the device, the DLNE shall request to establish default shared link to the inner router by issuing the DLC-LINK-SETUP.request primitive, as described in 6.5.1.

Upon receiving the DLC-LINK-SETUP.confirm primitive, the DLNE shall request to get the cluster table to the gateway by transmitting the link network management command frame containing the route update request command payload with the Route Update Request Type field set to full cluster connectivity matrix (0x00) by issuing the DLC-DATA-SLINK.request primitive.

When the DLNE receives the link network management command frame containing the route update response command payload by issuing the DLC-DATA-SLINK.indication primitive, the DLNE shall complete the cluster table.

6.2.2 Router procedure

On receipt of the MLME-ASSOCIATE.indication primitive, the DLNE shall assign a link network address as described in 6.4. The DLNE shall select the channel offset for the device requesting association. The DLNE shall notify the results by issuing the MLME-ASSOCIATE.response primitive to the MAC sublayer.

If no address or channel offset can be chosen, the DLNE shall notify the joining a link network failure by issuing the MLME-ASSOCIATE.response primitive with the Status parameter set to OUT_OF_RESOURCE.

When the router receives the link network management command frame containing the cluster formation request command payload by issuing the DLC-DATA-CLINK.indication primitive, the router shall forward the link network management command to the gateway router through the DLC-DATA-SLINK.request primitive by changing the destination address of the link network management command to the link network address of the gateway router.

The gateway router assigns a cluster identifier to the device, and updates the cluster connectivity matrix. If there is run out of the cluster ID space, the gateway router shall send the link network management response command with the Length of Cluster Identifier Space field of the cluster formation response command payload set to zero.

Upon receiving the link network management command frame containing the cluster formation response command payload by issuing the DLC-DATA-SLINK.indication primitive, the DLNE shall update the cluster table and forward the link network management command frame by issuing the DLC-DATA-CLINK.request primitive.

When the router receives the link network management command frame containing the route update request command payload by issuing the DLC-DATA-SLINK.indication primitive, the router shall forward the link network management command to the root router of the cluster. Upon receiving the link network management command frame containing the route update response command payload by issuing the DLC-DATA-SLINK.indication primitive, the DLNE shall update the cluster table and forward the link network management command frame by issuing the DLC-DATA-SLINK.request primitive.

6.3 Leaving a link network

When the DLN sublayer receives the DLN-MANAGEMENT.request primitive with the Type parameter set to LEAVE, the procedure to remove a device from the link network is initiated.

When the device to be removed is an end device of a cluster, the device shall request to remove from the link network by sending a link network management command frame with the Link Network Management Command Type field set to LEAVE_REQ through the DLC-DATA-CLINK.request primitive to the inner router. The inner router shall reply by sending the link network management command frame with the Link Network Management Command Type field set to LEAVE_RESP through the DLC-DATA-CLINK.request primitive to the end device. The inner router shall release the dedicated links for the end device by deallocating the DSME-GTSs and updating link table, update the route table, and send the link network management command frame containing the route update request command payload by issuing the DLC-DATA-SLINK.request primitive to the root router of the cluster to update the route table.

When the device to be removed is a router of a cluster, the device shall send a link network management command frame with the Link Network Management Command Type field set to LEAVE_REQ through the DLC-DATA-SLINK.request primitive to each end devices of the router. The router shall release the shared links and the dedicated links for end devices by deallocating the DSME-GTSs and updating link table. After completing link release, the router shall send the link network management command frame with the Link Network Management Command Type field set to LEAVE_REQ to the inner router, as shown in Figure 6.2. The inner router shall send the link network management command frame containing the route update request command payload by issuing the DLC-DATA-SLINK.request primitive to the root router of the cluster. The inner router shall reply by sending link network management command frame with the Link Network Management Command Type field set to LEAVE_RESP through the DLC-DATA-CLINK.request primitive to the router. Upon receiving the link network management command frame with the Link Network Management Command Type field set to LEAVE_RESP, the router shall request to release the shared links and the dedicated links to the inner router as described in 6.5.2. If the inner router will not receive the link release request command within *dlMaxResponseTimeout*, the inner router shall release the shared links and the dedicated links for the router.

When the device to be removed is a root router of a cluster, the device shall send the link network management command frame with the Link Network Management Command Type field set to LEAVE_REQ through the DLC-DATA-SLINK.request primitive to each routers except the inner router. Upon receiving the link release request command, the neighbour router shall initiate the procedure to remove the router from the link network. After completing the link release to the neighbour routers, the root router shall send the link network management command frame with the Link Network Management Command Type field set to LEAVE_REQ to the inner router. The inner router shall send the link network management command frame containing the route update request command payload by issuing the DLC-DATA-SLINK.request primitive to the gateway router. The inner router shall reply by sending link network management command frame with the Link Network Management Command Type field set to LEAVE_RESP through the DLC-DATA-CLINK.request primitive to the root router. Upon receiving the link release response command, the root router shall start to release the shared link and the dedicated links to the inner router.

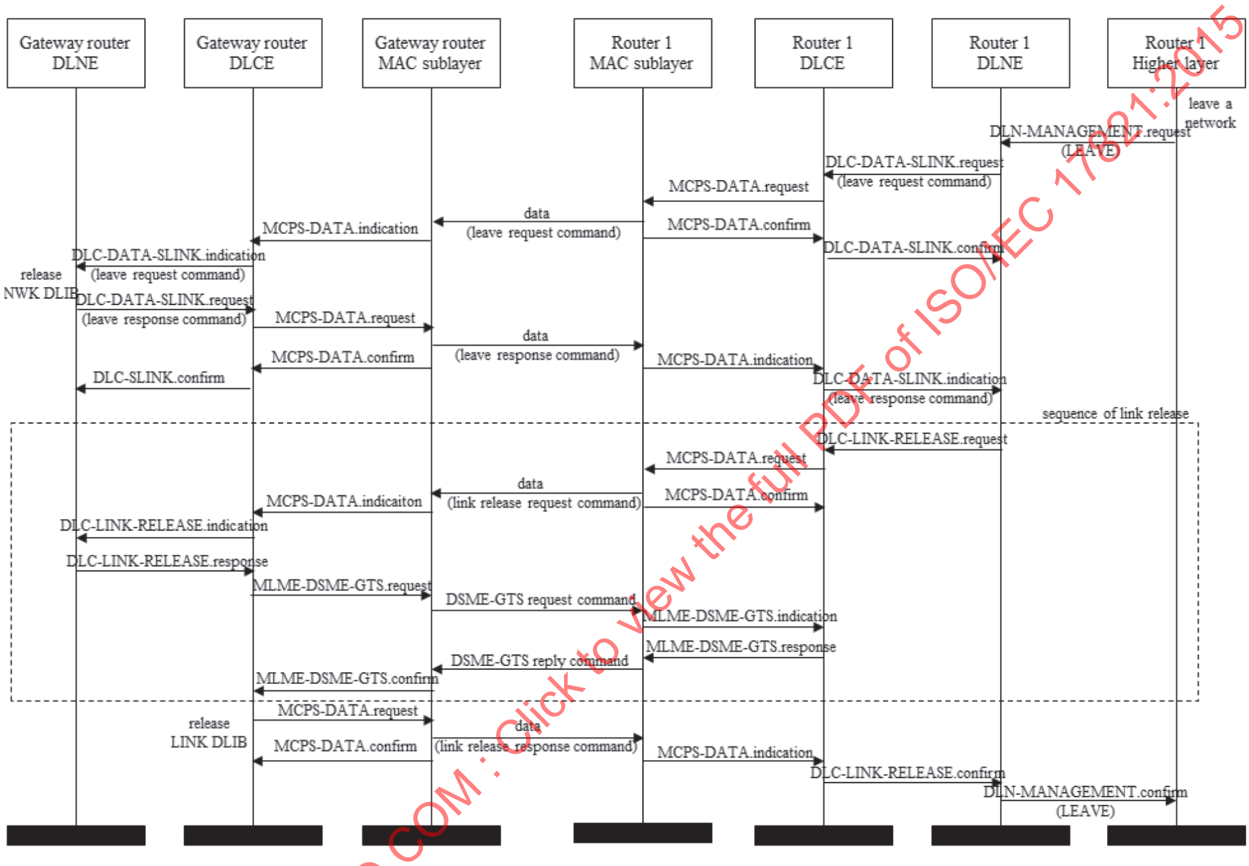

Figure 6.2 — Sequence of the leaving a network at a router

6.4 Address assignment with an unbalanced cluster-tree structure

The link network address of a device in the LPWMN is assigned during the startup procedure. The addressing of the LPWMN has the unbalanced cluster-tree structure. The link network address consists of the cluster identifier and the device locator in a cluster. The 16-bit addressing space is divided into *dlClusterAddrSpace* bits and (16-*dlClusterAddrSpace*) bits. The root of the cluster assigns a device locator by using the distributed address allocation scheme, as described in 5.4.3. The (16-*dlClusterAddrSpace*) bits of link network address shall be assigned to end devices in a sequential manner.

In a LPWMN the link network address can show the cluster in which the device is resided and where the device is located in the cluster. If the cluster is the root cluster, the cluster ID is set to 0. If the device is the root router of a cluster, the device locator is set to 0. The gateway router is the root router of the root cluster. Therefore, the link network address of the gateway router is 0x0000. If the length of cluster identifier is 8-bit and a router 1 in the root cluster becomes the root router of the cluster 1, the router has two link network addresses, the Router Address and Router Reflector Address of Table 9.6: the link network address in the root cluster, 0x0001, and the link network address in the cluster 1, 0x0100.

The gateway router shall maintain the cluster table and provide the information on the cluster. The cluster identifier is assigned by the gateway router. A new cluster can be formed, when a router joins a link network as a root of a cluster or when the address space is run out of. When receiving the DLN-START-ROUTER.request primitive with the non-zero maxDepth parameter, the DLNE shall request the cluster formation to the gateway router, after completing the MAC sublayer association procedure. The gateway router shall reply a cluster formation response command payload with the Cluster Identifier field.

6.5 DSME MAC link and link-path

6.5.1 Establishing a link-path

The LPWMN provides the capabilities to setup a link or link-path to the designated device. When the next higher layer of the DLC sublayer issues the DLC-LINK-SETUP.request primitive, the procedure to establish a shared link or a dedicated link to the designated device is initiated, as shown in Figure 6.3.

When the next higher layer of the DLC sublayer issues the DLC-LINK-SETUP.request primitive with the LinkType parameter set to IN-{SHARED/DEDICATED/PEER} or OUT-{SHARED /DEDICATED/PEER}, the procedure to establish a directional link is initiated. The DLCE shall search the neighbour device table or the route table for establishing a link or a link-path to the destined device. If a next hop to the destined device is found, the DLCE shall transmit the link setup request command to the next hop by issuing MCPS-DATA.request primitive on a CAP link or a default shared link to allocate time slots on a link to the next hop. If the DLCE fails to find a next hop to the destined device, the DLCE requests the DLNE to update the link-path routing information. If no router can be found to the destined device, the DLCE shall terminate the procedure and notify the next higher layer of the link setup failure by issuing the DLC-LINK-SETUP.confirm primitive with the Status parameter set to NOT_REACHABLE.

Upon receiving the MCPS-DATA.indication primitive at the router on the link-path to the destined device, the DLCE shall check the availability of the DSME-GTSs for the directional link. If the DSME-GTS is available, the DLCE shall allocate the DSME-GTSs and request to transmit DSME-GTS request command frame by issuing the MLME-DSME-GTS.request primitive with the PreferredSuperframID and PreferredSlotID parameter. When the DLCE receives the MLME-DSME-GTS.confirm primitive, the DLCE shall update the link table and forward the link setup request command to the next hop by issuing MCPS-DATA.request primitive. If the DSME-GTS is not available or the MLME-DSME-GTS.confirm primitive is not received, the DLCE shall discard the link setup request command and transmit the link setup response command with the Status field set to RESOURCE_FULL to the device requesting link setup.

When the destined device receives the link setup request command, the DLCE shall allocate the DSME-GTSs and request to transmit DSME-GTS request command frame to the previous hop device. When the DLCE receives the MLME-DSME-GTS.confirm primitive, the DLCE shall update the link table and notify the reception of the link setup request by issuing the DLC-LINK-SETUP.indication primitive. When the next higher layer of the DLC sublayer issues the DLC-LINK-SETUP.response primitive, the DLCE of the destined device shall transmit the link setup response command to the device requesting link setup by issuing MCPS-DATA.request primitive on a CAP link or a default shared link.

If no DSME-GTS is available or the MLME-DSME-GTS.confirm primitive is not received, the DLCE at the destined device shall discard the link setup request command and transmit the link setup response command with the Status field set to RESOURCE_FULL to the device requesting link setup. Upon receiving the link setup response command with the Status field set to RESOURCE_FULL at the router on the link-path to the device requesting link setup, the DLCE shall deallocate DSME-GTSs, update the link table by deleting the link, and forward the link setup response command to the device requesting link setup.

To establish a bi-directional dedicated link-path or a bi-directional shared link-path, the DLCE at the destined device shall start to establish a link-path to the device requesting link setup by transmitting the link setup request command, after completing the allocation of DSME-GTS on the link-path from the device requesting link setup to the destined device. When the device requesting link setup receives the MLME-DSME-GTS.confirm primitive, the DLCE shall transmit the link setup response command to the destined device. The link setup response command shall be relayed and the DLCE at the destined device shall notify the completion of bi-directional DSME-GTSs allocation by issuing the DLC-LINK-SETUP.indication primitive. When the next higher layer of the DLC sublayer issues the DLC-LINK-SETUP.response primitive, the DLCE of the destined device shall transmit the link setup response command to the device requesting link setup by issuing MCPS-DATA.request primitive on a CAP link or a default shared link.

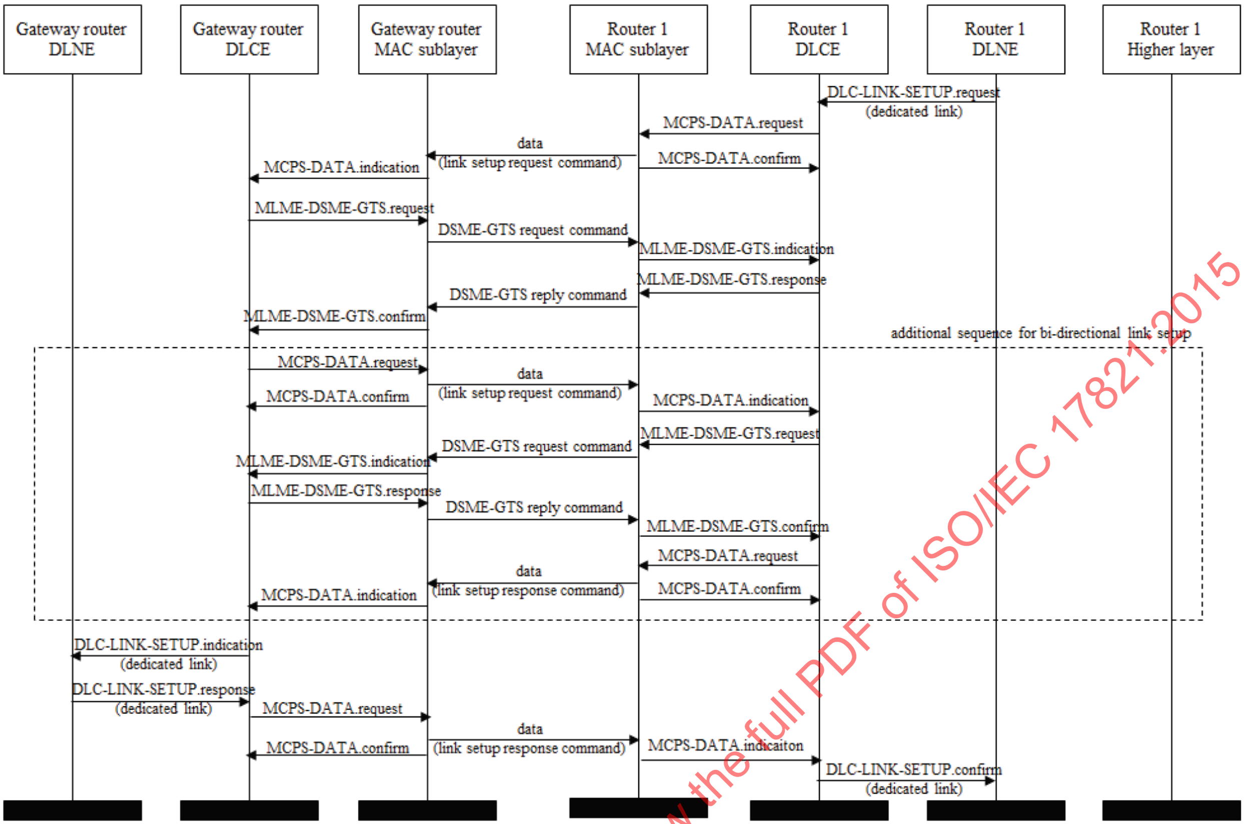

Figure 6.3 — Sequence of the establishing a link-path

6.5.2 Maintaining a link-path

The DLCE shall maintain the link status by listening to the beacon frames from the neighbour routers and by collecting the frame count transmitted on the links. The DLCE maintains the value of the Beacon Timestamp parameter of the Neighbour Device Table and the value of the Frame Count parameter of the Neighbour Link Table.

When the diagnosis of a link is required, the DLCE shall issue the MLME-DSME-LINKSTATUSRPT.request primitive for collecting link status. When the diagnosis of a link-path is required, the DLCE shall transmit the link management command containing the link hello request command payload by issuing the MCPS-DATA.request primitive.

Upon receiving the link hello request command at the end device of the link-path, the DLCE shall report the link-path status by transmitting the link management command containing the link hello response command payload to the device requesting the diagnosis of the link-path.

If the DLCE determines to release the link or link-path, the DLCE shall transmit the link management command containing the link release request command payload to the next hop device by issuing MCPS-DATA.request primitive, as shown in Figure 6.2. When the DLCE of the end device of the link-path receives the link release request command, the DLCE shall start to deallocate the DSME-GTSs on the link-path to the device requesting link release by issuing the MLME-DSME-GTS.request primitive and transmit the link release response command to the next hop on the link-path to the device requesting link release. The deallocation of the DSME-GTSs on the link-path and relaying of the link release response command are continued to the device requesting link release.

6.5.3 Routing a link-path

The selection of a route relies on the cluster table maintained by the gateway router and the route table maintained by the router.

The cluster table maintains the root router information of a cluster and the router information of an inter-cluster mesh link, as described in Table 9.4. An inter-cluster mesh link is separated from the cluster information by checking that at least one of the Router Address and Router Reflector has not the root router address. The cluster connectivity matrix is consisted of the Router Address, Router Reflector Address, and Child Cluster List. A cluster table entry is added when the gateway router receives the cluster formation request or the route update request to report the establishment of an inter-cluster mesh link. A cluster table entry is deleted when the gateway router receives the route update request to report leaving of a root router or the release of an inter-cluster mesh link. The gateway router provides the full cluster table to the root router when joins a link network. The cluster table of a certain cluster is provided on request base. The gateway router provides the updated entries of the cluster connectivity matrix every *dlRouteUpdatePeriod* number of beacons.

The route table maintains the reachable destinations through the current established links, which are one-hop CAP link or shared link to the neighbour device, multi-hop shared link-path, and dedicated link-path that the router is the source, as described in Table 9.7. The link path cost is calculated with the number of hops to the destination, and link quality and frame count of the Table 9.6. The route table entry is added when the link is established or when the first data forwarding to a destination is requested. The route table entry is deleted when the link is released or when the data forwarding is not active for *dlRouteUpdatePeriod* number of beacons. The route table is updated periodically or upon receiving the route update request from the root router.

The route update is requested when a router searches a route to transmit a data or to forward a data, as shown in Figure 6.4. When the DLNE finds no route from the cluster connectivity matrix, the DLNE shall send a link network management command frame containing the route update request command payload of the Route Update Request Type field set to updated cluster connectivity matrix. If the gateway router finds a route, the gateway router replies the route update response command payload with the cluster connectivity matrix entry for the route. If not, the gateway router requests to update the cluster connectivity matrix to the root routers of the cluster and defers to reply to the root router requesting a route until receiving link network management frames containing the route update response command payload from the root routers of the clusters.

When a device joins a link network or needs to update the entire routing table, the router shall request the full cluster table to the gateway router by sending the link network management command with the Route Update Request Type field of the route update request command payload set to 0x00 and the Router Address field set to 0x0000. When a router needs certain cluster information, the Router Address field of the route update request command payload is set to the specific root router address. The gateway router shall provide the cluster table entries by the link network management command with the Routing Information Base field set to the cluster table entry. When the last entry of cluster table is contained in the link network management command frame, the Route Update Response Type field of the route update response command payload is set to 0xf0.

When a device needs to update the routing table partially, the router shall request the updated cluster connectivity matrix to the gateway router by sending the link network management command with the Route Update Request Type field of the route update request command payload set to 0x01 (updated cluster connectivity matrix). The gateway router shall provide the selected cluster connectivity matrix entries, which are changed after the last periodical route update. When the last entry of cluster connectivity matrix is contained in the link network management command frame, the Route Update Response Type field of the route update response command payload is set to 0xf1. When the gateway router sends periodical updated cluster connectivity matrix, the route update response command with the updated cluster connectivity matrix type is used.

When a device needs to update the cluster connectivity matrix of the inner routers above a specified device, the router shall request the updated cluster connectivity matrix to the gateway router by sending the link network management command with the Route Update Request Type field of the route update request command payload set to 0x02 and the link network address of the specified device.

When a device needs to update the cluster connectivity matrix of the outer routers beyond a specified device, the router shall request the updated cluster connectivity matrix to the gateway router by sending the link network management command with the Route Update Request Type field of the route update request command payload set to 0x03 and the link network address of the specified device.

When a device completes to establish or release a link and updates the routing table, the router shall request the gateway router to update the cluster connectivity matrix to by sending the link network management command with the Route Update Response Type field of the route update response command payload set to 0x84 and the updated entry of the cluster connectivity matrix. When a root router needs to check the cluster connectivity matrix of a specified router, the root router shall request to report the cluster connectivity matrix to by sending the link network management command with the Route Update Request Type field of the route update request command payload set to 0x04 and the specified router address.

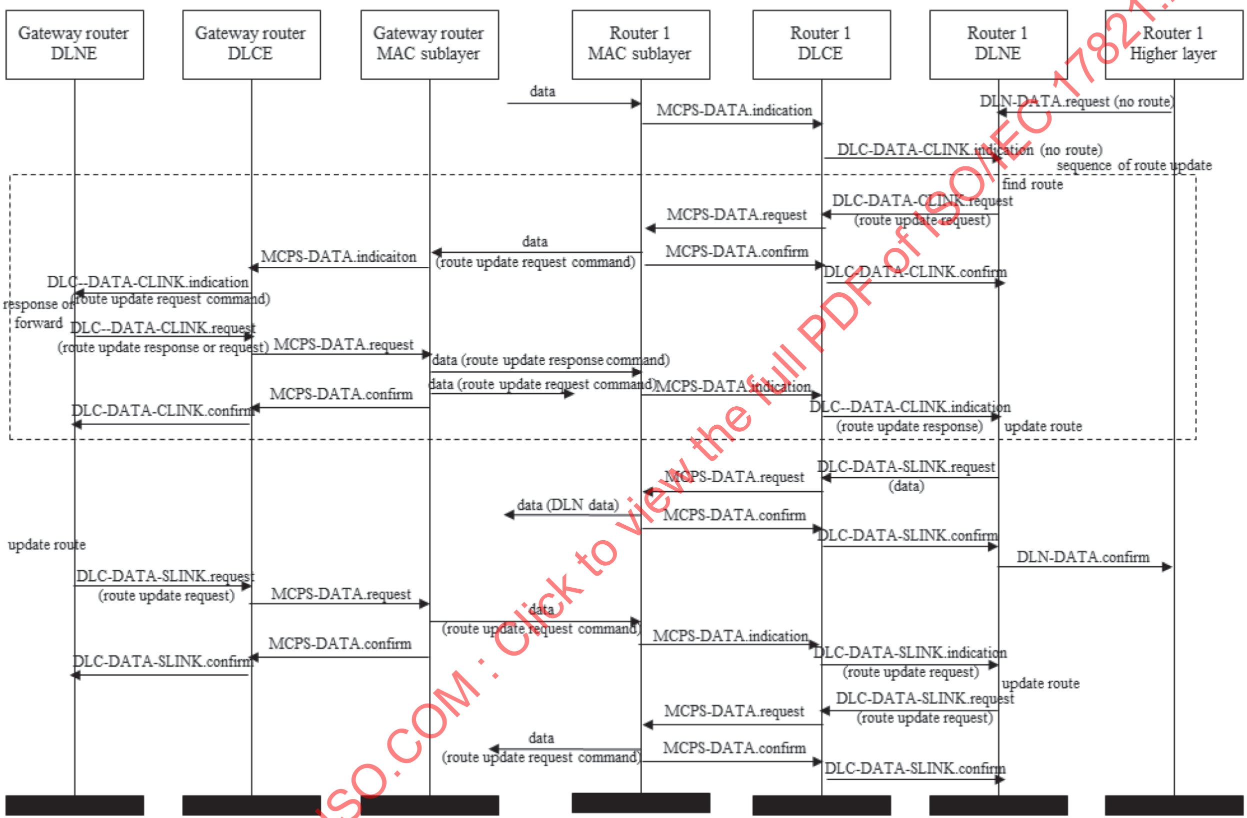

Figure 6.4 — Sequence of the route update and data forwarding

When a router needs to search a mesh link from a specified router, the router shall request to find a mesh link and to report the updated cluster connectivity matrix by sending the link network management request command with the Route Update Request Type field of the route update request command payload set to 0x05 and the link network address of the specified router.

When a router or root router needs to gather route cost information within the cluster, the router or root router shall request the specified router to report the route table by sending the link network management command frame with the Route Update Request Type field of the route update request command payload set to 0x06 and the link network address of the specified router.

6.6 Data services

6.6.1 General

The LPWMN provides six types of the data communication between two end DLCEs.

- Type 1: The DLPDUs shall be exchanged during the contention access period, CAP link. The DLPDUs shall not be acknowledged, and shall be transmitted without flow control and error recovery.
- Type 2: The DLPDUs shall be exchanged during the contention access period, CAP link. The DLPDUs shall be acknowledged, but shall be transmitted without flow control and error recovery.
- Type 3: The DLPDUs shall be exchanged in the shared guaranteed time slots, shared link, which are allocated for connecting two neighbour devices. The DLPDUs shall not be acknowledged, and shall be transmitted without flow control and error recovery.
- Type 4: The DLPDUs shall be exchanged in the shared guaranteed time slots, shared link. The DLPDUs shall be acknowledged, but shall be transmitted without any flow control and error recovery.
- Type 5: The DLPDUs shall be exchanged on the dedicated path which is established by allocating the dedicated guaranteed time slots on each hop link between two devices, dedicated link, prior to exchange. The DLPDUs shall not be acknowledged, and shall be transmitted without flow control and error recovery.
- Type 6: The DLPDUs shall be exchanged on the dedicated path. The DLPDUs shall be acknowledged, and shall be transmitted with flow control between two end DLCEs.

The primitives of the data service are provided by the DLN sublayer or DLC sublayer. The next higher layer of the DLC sublayer may select the data service by choosing the type of a link.

For secured transmission of a data payload, a higher next layer of the DLC sublayer or the DLN sublayer indicates by setting the SecurityEnable parameter of primitives for data service to TRUE. The MSDU is passed to the MAC sublayer and encrypted with the cryptographic mechanism of IEEE 802.15.4. The establishment and maintenance of keys are outside the scope of this standard.

6.6.2 DLN sublayer data service

When the DLN sublayer receives the DLN-DATA.request primitive, the procedure to send a data frame is initiated. If the DLN sublayer receives a request to transmit a data frame before completing the association, it shall discard the frame and notify the next higher layer of the error by issuing the DLN-DATA.confirm primitive with a status of INVALID_REQUEST.

If a route to the destination device is not found in the route table, the DLNE shall initiate the route discovery procedure. If the DLNE fails to discover a route, it shall discard the frame and notify the next higher layer of the error by issuing the DLN-DATA.confirm primitive with a status of NOT_REACHABLE.

If the TxMode parameter is set to TYPE_1, the DLNE shall issue the DLC-DATA-CLINK.request with the ACKEnable parameter set to FALSE. If the TxMode parameter is set to TYPE_2, the DLNE shall issue the DLC-DATA-CLINK.request with the ACKEnable parameter set to TRUE. If the TxMode parameter is set to TYPE_3, the DLNE shall issue the DLC-DATA-SLINK.request with the ACKEnable parameter set to FALSE. If the TxMode parameter is set to TYPE_4, the DLNE shall issue the DLC-DATA-SLINK.request with the ACKEnable parameter set to TRUE. If the TxMode parameter is set to TYPE_5, the DLNE shall issue the DLC-DATA-DLINK.request with the ACKEnable parameter set to FALSE and the LinkId parameter set to a dedicated link selected from the route table. If the TxMode parameter is set to TYPE_6, the DLNE shall issue the DLC-DATA-DLINK.request with the ACKEnable parameter set to TRUE and the LinkId parameter set to a bi-directional dedicated link selected from the route table. The value supplied with the DstAddr, DlsduLength, Dlsdu, DlsduHandle, and SecurityEnable parameter of the DLN-DATA.request primitive is set to the value of the corresponding parameter of the DLC sublayer data request primitives.

When the DLNE receives the DLC data confirm primitive from the DLC sublayer, the DLNE shall notify the next higher layer of the result by issuing a DLN-DATA.confirm primitive with the value of the DlsduHandle and Status parameter of the data confirm primitive from the DLC sublayer.

When the DLNE receives the DLC-DATA-CLINK.indication primitive, the DLC-DATA-SLINK.indication primitive, or the DLC-DATA-DLINK.indication primitive from the DLC sublayer, the DLNE shall check the type of the DLSDU. If it is the Data Payload, the DLNE shall notify the next higher layer of the reception of data frame by issuing the DLN-DATA.indication primitive. The TxMode parameter is set to TYPE_1, if receiving the DLC-DATA-CLINK.indication primitive with the ACKEnable parameter set to FALSE. The TxMode parameter is set to TYPE_2, if receiving the DLC-DATA-CLINK.indication primitive with the ACKEnable parameter set to TRUE. The TxMode parameter is set to TYPE_3, if receiving the DLC-DATA-SLINK.indication primitive with the ACKEnable parameter set to FALSE. The TxMode parameter is set to TYPE_4, if receiving the DLC-DATA-SLINK.indication primitive with the ACKEnable parameter set to TRUE. The TxMode parameter is set to TYPE_5, if receiving the DLC-DATA-DLINK.indication primitive with the ACKEnable parameter set to FALSE. The TxMode parameter is set to TYPE_6, if receiving the DLC-DATA-DLINK.indication primitive with the ACKEnable parameter set to TRUE.

6.6.3 DLC sublayer data service

When the DLC sublayer receives a data request primitive from the next higher layer, the procedure of constructing the DLPDU and passing a frame to the MAC data service is initiated.

When the DLCE receives a DLC-DATA-CLINK.request primitive or a DLC-DATA-SLINK.request primitive, the DLCE shall determine the transmission direction of the data frame by looking up the route table. If the DLCE fails to find a next hop for sending a data frame to the destination, it shall notify the next higher layer of the error by issuing a data confirm primitive with a status of NOT_REACHABLE.

When the DLCE receives a DLC-DATA-DLINK.request primitive, the DLCE shall search an established link to the destination. If the DLCE fails to find an established link to the destination, it shall notify the next higher layer of the error by issuing the DLC-DATA-DLINK.confirm primitive with a status of NOT_REACHABLE.

The Frame Operation Type subfield of the Frame Control field is determined by the type of received data request primitive and the ACKEnable parameter: DLC-DATA-CLINK.request primitive with the ACKEnable parameter set to FALSE (0b000), DLC-DATA-CLINK.request primitive with the ACKEnable parameter set to TRUE (0b001), DLC-DATA-SLINK.request primitive with the ACKEnable parameter set to FALSE (0b010), DLC-DATA-SLINK.request primitive with the ACKEnable parameter set to TRUE (0b011), DLC-DATA-DLINK.request primitive with the ACKEnable parameter set to FALSE (0b100), and DLC-DATA-DLINK.request primitive with the ACKEnable parameter set to TRUE (0b101).

If the address of the next hop is same as the DstAddr parameter, the Destination Address Flag subfield and the Source Address Flag subfield of the Frame Control field are set to zero. Otherwise, the Destination Address Flag subfield and the Source Address Flag subfield are set to one.

The SrcAddr parameter shall be set to the value of *macShortAddress* from the MAC PIB of the router, and the DstAddr parameter shall be the value provided by the next-hop address field of the routing table entry corresponding to the routing destination. The GTSTX parameter is set to zero, if the Frame Operation Type subfield of a frame is TYPE_1 or TYPE_2. Otherwise, the GTSTX parameter is set to one. The AckTx parameter is set to zero, if the Frame Operation Type subfield of a frame is TYPE_1, TYPE_3, or TYPE_5. Otherwise, the AckTx parameter is set to one. When the data frame is constructed, the DLCE shall pass the frame by issuing the MCPS-DATA.request primitive.

When the DLCE receives the MCPS-DATA.confirm primitive from the MAC sublayer, the DLCE shall notify the next higher layer of the result by issuing a DLC data confirm primitive with the value of the DlsduHandle and Status parameter of the MCPS-DATA.confirm primitive.

6.6.4 Data reception

The DLCE is notified the reception of a frame by receiving the MCPS-DATA.indication primitive. The frame is passed to the next higher layer or forwarded to the next hop, or broadcast onward.

The frame for which the destination address matches the device's address, the Link Management subframe shall be processed by the DLCE, and the Link Network Management subframe may be processed at the DLC sublayer or the DLN sublayer.

— Data frame shall be passed to the next higher layer. The primitive to notify the reception of a data frame is selected by the Frame Operation Type subfield of the Frame Control field. If the Frame Operation Type subfield is TYPE_1 operation or TYPE_2 operation, the DLC-DATA-CLINK.indication primitive is issued. If the Frame Operation Type subfield is TYPE_3 operation or TYPE_4 operation, the DLC-DATA-SLINK.indication primitive is issued. If the Frame Operation Type subfield is TYPE_5 operation or TYPE_6 operation, the DLC-DATA-DLINK.indication primitive is issued. The ACKEnable parameter is set to one, only if the Frame Operation Type subfield is TYPE_2 operation or TYPE_4 operation, or TYPE_6 operation.

— If the Frame Operation Type subfield is TYPE_6 operation, the sender and receiver DLC sublayer perform the flow control on the dedicated link. Congestion on the dedicated link is detected by receiving a flow control request command or flow control response command with the Frame Control Command Type field set to receive not ready or when the value of Sender Receive Sequence Number is not equal to the value of the Sender Send Sequence Number. When congestion is indicated, the transmission window control is invoked.

— If the receiver DLC subayer is not ready to receive the data frame, the DLCE may transmit the flow control response command with the Frame Control Command Type field set to receive not ready.

The frame for which the destination address does not match the device's address shall be forwarded. The direction of forwarding is informed by the destination address and the link receiving a frame. If the Frame Operation Type subfield of a frame is TYPE_1 or TYPE_2, the DLCE shall select the next hop by searching the route table. If the route to the destination device is not found in the route table or the cluster connectivity matrix, the DLCE shall initiate the route discovery procedure. If the DLNE fails to discover a route, it shall discard the frame and notify the next higher layer of the error by issuing a DLN-DATA.confirm primitive with a status of NOT_REACHABLE.

If the Frame Operation Type subfield of a frame is TYPE_3, TYPE_4, TYPE_5, or TYPE_6, the direction of the GTS receiving a frame shows whether the frame is forwarded inward or outward. The GTS assigned for the established link onward is listed in the link table. If the established link to the destination device is not found in the Route table, the DLCE shall discard the frame.

The frame with the broadcast address shall be processed at the DLC sublayer and be forwarded onward.

7 LPWMN services

7.1 LPWMN management services

The DLN-SAP allows the transport of management commands between the next higher layer and the DLN sublayer. Table 7.1 lists the primitives supported by the DLN sublayer through the DLN-SAP interface and the sub-clauses containing details on each of these primitives.

Table 7.1 — Summary of the primitives accessed the DLN-SAP

Name	Sub-Clause Number in This Specification			
	Request	Indication	Response	Confirm
DLN-START-NETWORK	7.1.1.1			7.1.1.2
DLN-START-ROUTER	7.1.2.1			7.1.2.2
DLN-START-DEVICE	7.1.3.1			7.1.3.2
DLN-RESET	7.1.4.1			7.1.4.2
DLN-GET	7.1.4.3			7.1.4.4
DLN-SET	7.1.4.5			7.1.4.6

Table 7.1 *(continued)*

Name	Sub-Clause Number in This Specification			
	Request	Indication	Response	Confirm
DLN-MANAGEMENT	7.1.5.1			7.1.5.2

7.1.1 Primitives for link network formation

These primitives are used by a gateway router to initiate a LPWMN.

7.1.1.1 DLN-START-NETWORK.request

This primitive allows the next higher layer to request that the device starts to perform as a gateway router and establishes a LPWMN.

The semantics of this primitive are:

```
DLN-START-NETWORK.request      {
                               PANId,
                               ScanChannels,
                               ScanDuration,
                               HoppingSequenceLength,
                               BeaconOrder,
                               SuperframeOrder,
                               Multi-superframeOrder,
                               ChannelDiversityMode,
                               CAPEnable,
                               GACKEnable,
                               DeferredBeaconEnable,
                               maxDepth,
                               maxRouters,
                               maxChildren,
                               PowerSourceEnable,
                               SecurityCapabilityEnable,
                               ReceiverOnEnable
                               }
```

The primitive parameters are defined in Table 7.2.

Table 7.2 — DLN-START-NETWORK.request parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000 – 0xffff	The PAN identifier to be used by the device.
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned in preparation for starting a network (1=scan, 0=do not scan) for each of the 27 valid channels.
ScanDuration	Integer	0x00 – 0x0e	A value used to calculate the length of time to spend scanning each channel for ED, active, and passive scans. This parameter is ignored for orphan scans. The time spent scanning each channel is [*aBaseSuperframeDuration* × (2n +1)], where n is the value of the ScanDuration parameter.
HoppingSequenceLength	Integer	0x0000 – 0xffff	The number of channels in the Hopping Sequence.

Table 7.2 *(continued)*

Name	Type	Valid range	Description
BeaconOrder	Integer	0 – 15	Indicates the frequency with which the beacon is transmitted.
SuperframeOrder	Integer	0-BeaconOrder or 15	The length of the active portion of the superframe, including the beacon frame.
Multi-superframeOrder	Integer	0-(BeaconOrder-SuperframeOrder)	The length of during which a group of superframes that is considered as one multi-superframe.
ChannelDiversityMode	Boolean	TRUE or FALSE	The type of channel diversity. If this value is zero, operates on channel adaptation mode. If this value is one, operates on channel hopping mode.
CAPEnable	Boolean	TRUE or FALSE	Indicates whether the CAP reduction is enabled. If set to TRUE, it is enabled.
GACKEnable	Boolean	TRUE or FALSE	Indicates whether the group ACK is enabled. If set to TRUE, it is enabled.
DeferredBeaconEnable	Boolean	TRUE or FALSE	Indicates whether the device uses CCA before transmitting beacon frame. If set to TRUE, it is enabled.
maxDepth	Integer	0x00 – 0xff	The depth a device can have in a cluster.
maxRouters	Integer	0x00 – 0xff	The number of routers any one device is allowed to have as children.
maxChildren	Integer	0x00 – 0xff	The number of device allowed connecting to a router of the cluster as a child.
PowerSourceEnable	Boolean	TRUE or FALSE	Indicates whether the device is receiving power from the mains. If set to TRUE, it is enabled.
SecurityCapabilityEnable	Boolean	TRUE or FALSE	Indicates whether the device is capable of sending and receiving cryptographically protected MAC frames. If set to TRUE, it is enabled.
ReceiverOnEnable	Boolean	TRUE or FALSE	Indicates whether the device disables its receiver to conserve power during idle periods. If set to TRUE, it is enabled.

7.1.1.2 DLN-START-NETWORK.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to start a LPWMN.

The semantics of this primitive are:

```
DLN-START-NETWORK.confirm        {
                                 Status
                                 }
```

The primitive parameters are defined in Table 7.3.

Table 7.3 — DLN-START-NETWORK.confirm parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, STARTUP_FAILURE or any status value returned from the MLME-START.confirm primitive	The result of the attempt to initialize a LPWMN gateway router.

7.1.2 Primitives for link network joining as a router

These primitives are used by a router to join a LPWMN.

7.1.2.1 DLN-START-ROUTER.request

This primitive allows the next higher layer to request that the device starts to perform as a LPWMN router.

The semantics of this primitive are:

```
DLN-START-ROUTER.request      {
                              PANId,
                              ScanChannels,
                              ScanDuration,
                              maxDepth,
                              maxRouters,
                              maxChildren,
                              PowerSourceEnable,
                              SecurityCapabilityEnable,
                              ReceiverOnEnable
                              }
```

The primitive parameters are defined in Table 7.4.

Table 7.4 — DLN-START-ROUTER.request parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000 – 0xffff	The PAN identifier to be used by the device.
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned in preparation for starting a network (1=scan, 0=do not scan) for each of the 27 valid channels.
ScanDuration	Integer	0x00 – 0x0e	A value used to calculate the length of time to spend scanning each channel. The time spent scanning each channel is (*aBaseSuperframeDuration* * (2n + 1)) symbols, where n is the value of the ScanDuration parameter.
maxDepth	Integer	0x00 – 0xff	The depth a device can have in a cluster.
maxRouters	Integer	0x00 – 0xff	The number of routers any one device is allowed to have as children.
maxChildren	Integer	0x00 – 0xff	The number of device allowed connecting to a router of the cluster as a child.

Table 7.4 *(continued)*

Name	Type	Valid range	Description
PowerSourceEnable	Boolean	TRUE or FALSE	Indicates whether the device is receiving power from the mains. If set to TRUE, it is enabled.
SecurityCapabilityEnable	Boolean	TRUE or FALSE	Indicates whether the device is capable of sending and receiving cryptographically protected MAC frames. If set to TRUE, it is enabled.
ReceiverOnEnable	Boolean	TRUE or FALSE	Indicates whether the device disables its receiver to conserve power during idle periods. If set to TRUE, it is enabled.

7.1.2.2 DLN-START-ROUTER.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to join a LPWMN.

The semantics of this primitive are:

```
DLN-START-ROUTER.confirm        {
                                Status
                                }
```

The primitive parameters are defined in Table 7.5.

Table 7.5 — DLN-START-ROUTER.confirm parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, STARTUP_FAILURE or any status value returned from the MLME-START.confirm primitive	The result of the attempt to initialize a LPWMN router.

7.1.3 Primitives for link network joining as a device

These primitives are used by a device to join a LPWMN.

7.1.3.1 DLN-START-DEVICE.request

This primitive allows the next higher layer to request that the device starts to perform as a device and establishes a link to a LPWMN.

The semantics of this primitive are:

```
DLN-START-DEVICE.request        {
                                PANId,
                                ScanChannels,
                                ScanDuration,
                                PowerSourceEnable,
                                SecurityCapabilityEnable,
                                ReceiverOnEnable
                                }
```

The primitive parameters are defined in Table 7.6.

Table 7.6 — DLN-START-DEVICE.request parameters

Name	Type	Valid range	Description
PANId	Integer	0x0000 – 0xffff	The PAN identifier to be used by the device.
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned in preparation for starting a network (1=scan, 0=do not scan) for each of the 27 valid channels.
ScanDuration	Integer	0x00 – 0x0e	A value used to calculate the length of time to spend scanning each channel. The time spent scanning each channel is (*aBaseSuperframeDuration* * (2n + 1)) symbols, where n is the value of the ScanDuration parameter.
PowerSourceEnable	Boolean	TRUE or FALSE	Indicates whether the device is receiving power from the mains. If set to TRUE, it is enabled.
SecurityCapabilityEnable	Boolean	TRUE or FALSE	Indicates whether the device is capable of sending and receiving cryptographically protected MAC frames. If set to TRUE, it is enabled.
ReceiverOnEnable	Boolean	TRUE or FALSE	Indicates whether the device disables its receiver to conserve power during idle periods. If set to TRUE, it is enabled.

7.1.3.2 DLN-START- DEVICE.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to join a LPWMN.

The semantics of this primitive are:

```
DLN-START-DEVICE.confirm        {
                                Status
                                }
```

The primitive parameters are defined in Table 7.7.

Table 7.7 — DLN-START-DEVICE.confirm parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, STARTUP_FAILURE or any status value returned from the MLME-START.confirm primitive	The result of the attempt to initialize a LPWMN device.

7.1.4 Primitives for the DLIB management

These primitives are used to manage the DLIB.

7.1.4.1 DLN-RESET.request

This primitive allows the next higher layer to request that the DLNE performs a reset operation.

The semantics of this primitive are:

```
DLN-RESET.request            {
                             SetDefaultDLIB
                             }
```

The primitive parameters are defined in Table 7.8.

Table 7.8 — DLN-RESET.request parameters

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

7.1.4.2 DLN-RESET.confirm

This primitive allows the next higher layer to be notified of the result of the reset operation.

The semantics of this primitive are:

```
DLN-RESET.confirm            {
                             Status
                             }
```

The primitive parameters are defined in Table 7.9.

Table 7.9 — DLN-RESET.confirm parameters

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

7.1.4.3 DLN-GET.request

This primitive allows the next higher layer to read the value of an attribute from the DLIB.

The semantics of this primitive are:

```
DLN-GET.request               {
                              DLIBAttribute
                              }
```

The primitive parameters are defined in Table 7.10.

Table 7.10 — DLN-GET.request parameters

Name	Type	Valid range	Description
DLIBAttribute	Integer	As defined in Table 9.2	The name of the DLIB attribute to read.

7.1.4.4 DLN-GET.confirm

This primitive reports the results of an attempt to read the value of an attribute from the DLIB.

The semantics of this primitive are:

```
DLN-GET.confirm              {
                             Status,
```

```
                              DLIBAttribute,
                              DLIBAttributeLength,
                              DLIBAttributeValue
                              }
```

The primitive parameters are defined in Table 7.11.

Table 7.11 — DLN-GET.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, UNSUPPORTED_ATTRIBUTE	The result of the reset operation.
DLIBAttribute	Integer	As defined in Table 9.2	The name of the DLIB attribute that was read.
DLIBAttributeLength	Integer	0x0000 – 0xffff	The length, in octets, of the attribute value being returned.
DLIBAttributeValue	Various	Attribute specific; as defined in Table 9.2	The value of the indicated DLIB attribute that was read.

7.1.4.5 DLN-SET.request

This primitive allows the next higher layer to write the value of an attribute in the DLIB.

The semantics of this primitive are:

```
DLN-SET.request               {
                              DLIBAttribute,
                              DLIBAttributeLength,
                              DLIBAttributeValue
                              }
```

The primitive parameters are defined in Table 7.12.

Table 7.12 — DLN-SET.request parameters

Name	Type	Valid range	Description
DLIBAttribute	Integer	As defined in Table 9.2	The name of the DLIB attribute to write.
DLIBAttributeLength	Integer	0x0000 – 0xffff	The length, in octets, of the attribute value being set.
DLIBAttributeValue	Various	Attribute specific; as defined in Table 9.2	The value of the indicated DLIB attribute to write.

7.1.4.6 DLN-SET.confirm

This primitive reports the results of an attempt to write the value of an attribute from the DLIB.

The semantics of this primitive are:

```
DLN-SET.confirm               {
                              Status,
                              DLIBAttribute
                              }
```

The primitive parameters are defined in Table 7.13.

Table 7.13 — DLN-SET.confirm parameters

Name	Type	Valid range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER , UNSUPPORTED_ATTRIBUTE	The result of the reset operation.

Table 7.13 *(continued)*

Name	Type	Valid range	Description
DLIBAttribute	Integer	As defined in Table 9.2	The name of the DLIB attribute that was written.

7.1.5 Primitives for link network resources management

These primitives are used by a gateway router or router to manage the link network resources.

7.1.5.1 DLN-MANAGEMENT.request

This primitive allows the next higher layer to request that the DLNE performs a management of a LPWMN.

The semantics of this primitive are:

```
DLN-MANAGEMENT.request          {
                                ManagementType,
                                PANId,
                                DeviceAddress,
                                RemoveChildren
                                }
```

The primitive parameters are defined in Table 7.14.

Table 7.14 — DLN-MANAGEMENT.request parameters

Name	Type	Valid range	Description
ManagementType	Enumeration	REJOIN, LEAVE, UPDATE	The type of LPWMN management to be performed.
PANId	Integer	0x0000 – 0xffff	The PAN identifier to be used by the device.
DeviceAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the managed device.
RemoveChildren	Boolean	TRUE or FALSE	This parameter has a value of TRUE if the device being asked to leave the network is also being asked to remove its child devices, if any. Otherwise, it has a value of FALSE.

When the ManagementType parameter is set to REJOIN, all parameters except PANId shall be ignored and the DLNE shall attempt to join a LPWMN.

When the ManagementType parameter is set to LEAVE, all parameters except DeviceAddress and RemoveChildren shall be ignored and the DLNE shall attempt to leave a LPWMN.

When the ManagementType parameter is set to UPDATE, all parameters except DeviceAddress shall be ignored and the DLNE shall attempt to update a route table.

7.1.5.2 DLN-MANAGEMENT.confirm

This primitive allows the next higher layer to be notified of the result of the management operation.

The semantics of this primitive are:

```
DLN- MANAGEMENT.confirm         {
                                ManagementType,
                                Status
                                }
```

The primitive parameters are defined in Table 7.15.

Table 7.15 — DLN-MANAGEMENT.confirm parameters

Name	Type	Valid range	Description
ManagementType	Enumeration	REJOIN, LEAVE, UPDATE	The type of LPWMN management to be performed.
Status	Enumeration	SUCCESS, INVALID_REQUEST, UNKNOWN_DEVICE or any status returned by the MCPS-DATA.confirm primitive	The result of the management of a LPWMN.

7.2 LPWMN link management services

The DLC-SAP allows the transport of management commands between the next higher layers and the DLC sublayer. Table 7.16 summarizes the primitives supported by the DLC sublayer through the DLC-SAP interface and the sub-clauses containing details on each of these primitives.

Table 7.16 — Summary of the primitives accessed the DLC-SAP

Name	Sub-Clause Number in This Specification			
	Request	Indication	Response	Confirm
DLC-LINK-SETUP	7.2.1.1	7.2.1.2	7.2.1.3	7.2.1.4
DLC-LINK-RELEASE	7.2.2.1	7.2.2.2	7.2.2.3	7.2.2.4
DLC-MANAGEMENT	7.2.3.1			7.2.3.2

7.2.1 Primitives for link establishment

These primitives are used by the next higher layer to establish a shared link between the device requesting this primitive and neighbour inner router, or to establish a dedicated link-path from the device requesting this primitive to the designated device.

7.2.1.1 DLC-LINK-SETUP.request

This primitive allows the next higher layer to request that the device starts to establish a shared link to the gateway router or a router, or to establish a dedicated link-path from the source device to the destination device.

The semantics of this primitive are:

```
DLC-LINK-SETUP.request          {
                                LinkType,
                                DstAddress,
                                NumberSlot
                                }
```

The primitive parameters are defined in Table 7.17.

Table 7.17 — DLC-LINK-SETUP.request parameters

Name	Type	Valid range	Description
LinkType	Enumeration	IN-SHARED, OUT-SHARED, BI-SHARED, IN-DEDICATED, OUT-DEDICATED, BI-DEDICATED, IN-PEER, OUT-PEER, BI-PEER	The type of link.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the destined device.
NumberSlot	Integer	0x00 – 0xff	The number of slots allocated sequentially to this link.

7.2.1.2 DLC-LINK-SETUP.indication

This primitive allows the next higher layer to be notified the reception of a link setup request.

The semantics of this primitive are:

```
DLC-LINK-SETUP.indication       {
                                LinkType,
                                SrcAddress,
                                LinkId,
                                NumberSlot
                                }
```

The primitive parameters are defined in Table 7.18.

Table 7.18 — DLC-LINK-SETUP.indication parameters

Name	Type	Valid range	Description
LinkType	Enumeration	IN-SHARED, OUT-SHARED, BI-SHARED, IN-DEDICATED, OUT-DEDICATED, BI-DEDICATED, IN-PEER, OUT-PEER, BI-PEER	The type of link.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the device requesting link setup.
LinkId	Integer	0x00 – 0xff	The link identifier.
NumberSlot	Integer	0x00 – 0xff	The number of slots allocated sequentially to this link.

7.2.1.3 DLC-LINK-SETUP.response

This primitive allows the next higher layer to initiate a response to a link setup request.

The semantics of this primitive are:

```
DLC-LINK-SETUP.response      {
                             Status,
                             SrcAddress,
                             LinkId
                             }
```

The primitive parameters are defined in Table 7.19.

Table 7.19 — DLC-LINK-SETUP. response parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, OUT_OF_RESOURCE, or any status value returned from the status value returned from the MLME-DSME-GTS.confirm primitive	The result of the attempt to setup a link.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the device requesting link setup.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.1.4 DLC-LINK-SETUP.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to setup a link.

The semantics of this primitive are:

```
DLC-LINK-SETUP.confirm             {
                                   Status,
                                   LinkId
                                   }
```

The primitive parameters are defined in Table 7.20.

Table 7.20 — DLC-LINK-SETUP.confirm parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, OUT_OF_RESOURCE, or any status value returned from the status value returned from the MLME-DSME-GTS.confirm primitive	The result of the attempt to setup a link.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.2 Primitives for link release

These primitives are used by the next higher layer to release a dedicated link connection.

7.2.2.1 DLC-LINK-RELEASE.request

This primitive allows the next higher layer to request that the device attempts to release a dedicated link.

The semantics of this primitive are:


```
DLC-LINK-RELEASE.request        {
                                SrcAddress,
                                DstAddress,
                                LinkId
                                }
```

The primitive parameters are defined in Table 7.21.

Table 7.21 — DLC-LINK-RELEASE.request parameters

Name	Type	Valid range	Description
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the device requesting link release.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the opposite end device of the link to be released.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.2.2 DLC-LINK-RELEASE.indication

This primitive allows the next higher layer to be notified the reception of a link release request.

The semantics of this primitive are:

```
DLC-LINK- RELEASE.indication    {
                                SrcAddress,
                                DstAddress,
                                LinkId
                                }
```

The primitive parameters are defined in Table 7.22.

Table 7.22 — DLC-LINK-RELEASE.indication parameters

Name	Type	Valid range	Description
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the device requesting link release.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the opposite end device of the link to be released.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.2.3 DLC-LINK-RELEASE.response

This primitive allows the next higher layer to initiate a response to a link release request.

The semantics of this primitive are:

```
DLC-LINK-RELEASE.response       {
                                Status,
                                SrcAddress,
                                DstAddress,
                                LinkId
                                }
```

The primitive parameters are defined in Table 7.23.

Table 7.23 — DLC-LINK-RELEASE.response parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, or any status value returned from the MLME-DSME-GTS. confirm primitive	The result of the attempt to release a link.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the device requesting link release.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the opposite end device of the link to be released.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.2.4 DLC-LINK-RELEASE.confirm

This primitive allows the next higher layer to be notified of the result of the link release request.

The semantics of this primitive are:

```
DLC-LINK- RELEASE.confirm          {
                                   Status,
                                   DstAddress,
                                   LinkId
                                   }
```

The primitive parameters are defined in Table 7.24.

Table 7.24 — DLC-LINK-RELEASE.confirm parameters

Name	Type	Valid range	Description
Status	Status	INVALID_REQUEST, or any status value returned from the MLME-DSME-GTS.confirm primitive	The result of the attempt to release a link.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the opposite end device of the link to be released.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.3 Primitives for link management

These primitives are used by the next higher layer to manage a link or a link-path.

7.2.3.1 DLC-MANAGEMENT.request

This primitive allows the next higher layer to request that the DLCE performs a management of a link.

The semantics of this primitive are:

```
DLC-MANAGEMENT.request             {
                                   ManagementType,
                                   LinkId
                                   }
```

The primitive parameters are defined in Table 7.25.

Table 7.25 — DLC-LINK-MANAGEMENT.request parameters

Name	Type	Valid range	Description
ManagementType	Enumeration	HELLO, RESET	The type of link management to be performed.
LinkId	Integer	0x00 – 0xff	The link identifier.

7.2.3.2 DLC-MANAGEMENT.confirm

This primitive allows the next higher layer to be notified of the result of the management operation.

The semantics of this primitive are:

```
DLC-MANAGEMENT.confirm          {
                                ManagementType,
                                LinkId,
                                Status
                                }
```

The primitive parameters are defined in Table 7.26.

Table 7.26 — DLC-MANAGEMENT.confirm parameters

Name	Type	Valid range	Description
ManagementType	Enumeration	HELLO, RESET	The type of link management to be performed.
LinkId	Integer	0x00 – 0xff	The link identifier.
Status	Enumeration	SUCCESS, FAIL, INVALID_REQUEST, UNKNOWN_DEVICE or any status returned by the MCPS-DATA.confirm primitive	The result of the link management.

7.3 LPWMN data services

The LPWMN provides four data transport modes. Table 7.27 summarizes the primitives supported by the DLC sublayer and DLN sublayer, and the sub-clauses containing details on each of these primitives.

Table 7.27 — Summary of the primitives for data services

Name	Sub-Clause Number in This Specification			
	Request	Indication	Response	Confirm
DLC-DATA-CLINK	7.3.1.1	7.3.1.2		7.3.1.3
DLC-DATA-SLINK	7.3.2.1	7.3.2.2		7.3.2.3
DLC-DATA-DLINK	7.3.3.1	7.3.3.2	7.3.3.3	7.3.3.4
DLN-DATA	7.3.4.1	7.3.4.2		7.3.4.3

7.3.1 Primitives for data service on a CAP link

These primitives are used by the next higher layer for sending a data frame during a contention access period.

7.3.1.1 DLC-DATA-CLINK.request

This primitive allows the next higher layer to request that the device attempts to transmit a data frame on a CAP link.

The semantics of this primitive are:

```
DLC- DATA-CLINK.request         {
                                DstAddr,
```

```
                                        DlsduLength,
                                        Dlsdu,
                                        DlsduHandle,
                                        SecurityEnable,
                                        ACKEnable
                                        }
```

The primitive parameters are defined in Table 7.28.

Table 7.28 — DLC-DATA-CLINK.request parameters

Name	Type	Valid range	Description
DstAddr	Integer	0x0000 – 0xffff	The 16-bit address of the device to which the DLSDU is sent.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The set of octets comprising the DLSDU to be transferred.
DlsduHandle	Integer	0x00 – 0xff	The handle of the DLSDU to be transmitted by the DLC sublayer entity.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.

7.3.1.2 DLC-DATA-CLINK.indication

This primitive allows the next higher layer to be notified that the device receives a data frame (DLSDU) on a CAP link.

The semantics of this primitive are:

```
DLC-DATA-CLINK.indication          {
                                   SrcAddr,
                                   DstAddr,
                                   DlsduLength,
                                   Dlsdu,
                                   RxTime,
                                   ACKEnable
                                   }
```

The primitive parameters are defined in Table 7.29.

Table 7.29 — DLC-DATA-CLINK.indication parameters

Name	Type	Valid range	Description
SrcAddr	Integer	0x0000 – 0xfffd	The 16-bit address of the device from which the DLSDU is originated.
DstAddr	Integer	0x0000 – 0xffff	The 16-bit address of the device to which the DLSDU is sent.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The set of octets comprising the DLSDU to be transferred.
RxTime	Integer	0x000000 – 0xffffff	The time, in symbols, at which the data were received, as described in 5.1.4.1 of the IEEE 802.15.4.

Table 7.29 *(continued)*

Name	Type	Valid range	Description
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.

7.3.1.3 DLC-DATA-CLINK.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to transmit a data frame (DLSDU) on a CAP link.

The semantics of this primitive are:

```
DLC-DATA-CLINK.confirm            {
                                  DlsduHandle,
                                  Status
                                  }
```

The primitive parameters are defined in Table 7.30.

Table 7.30 — DLC-DATA-CLINK.confirm parameters

Name	Type	Valid range	Description
DlsduHandle	Integer	0x00 – 0xff	The handle associated with the DLSDU being confirmed.
Status	Status	INVALID_REQUEST, or any status value returned from the MCPS-DATA.confirm primitive of the IEEE 802.15.4	The result of the attempt to transmit a data frame.

7.3.2 Primitives for data service on a shared GTS link

These primitives are used by the next higher layer to transport data frame on a shared GTS link.

7.3.2.1 DLC-DATA-SLINK.request

This primitive allows the next higher layer to request that the device attempts to transmit a data frame on a shared link.

The semantics of this primitive are:

```
DLC-DATA-SLINK.request         {
                               DstAddress,
                               DlsduLength,
                               Dlsdu,
                               DlsduHandle,
                               ACKEnable,
                               SecurityEnable
                               }
```

The primitive parameters are defined in Table 7.31.

Table 7.31 — DLC-DATA-SLINK.request parameters

Name	Type	Valid range	Description
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the destination device.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.

Table 7.31 *(continued)*

Name	Type	Valid range	Description
Dlsdu	Set of Octets		The number of octets comprising the DLSDU to be transferred.
DlsduHandle	Integer	0x00 – 0xff	The handle of the DLSDU to be transmitted by the DLC sublayer entity.
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.

7.3.2.2 DLC-DATA-SLINK.indication

This primitive allows the next higher layer to be notified that the device receives a data frame on a shared link.

The semantics of this primitive are:

```
DLC-DATA-SLINK.indication          {
                                   DstAddress,
                                   SrcAddress,
                                   DlsduLength,
                                   Dlsdu,
                                   RxTime,
                                   SecurityEnable
                                   }
```

The primitive parameters are defined in Table 7.32.

Table 7.32 — DLC-DATA-SLINK.indication parameters

Name	Type	Valid range	Description
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the destination device.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the source device.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The number of octets comprising the DLSDU to be transferred.
RxTime	Integer	0x000000 – 0xffffff	The time, in symbols, at which the data were received, as described in 5.1.4.1 of the IEEE 802.15.4.
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.

7.3.2.3 DLC-DATA-SLINK.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to transmit a data frame on a shared link.

The semantics of this primitive are:


```
DLC- DATA-SLINK.confirm          {
                                 DlsduHandle,
                                 Status
                                 }
```

The primitive parameters are defined in Table 7.33.

Table 7.33 — DLC-DATA-SLINK.confirm parameters

Name	Type	Valid range	Description
DlsduHandle	Integer	0x00 – 0xff	The handle associated with the DLSDU being confirmed.
Status	Status	INVALID_REQUEST, or any status value returned from the MCPS-DATA.confirm primitive	The result of the attempt to transmit a data frame on a shared link in acknowledged manner.

7.3.3 Primitives for data service on a dedicated link path

These primitives are used by the next higher layer to transport data frame on a dedicated link path to the destination device. It is required to setup a link path before using this primitive.

7.3.3.1 DLC-DATA-DLINK.request

This primitive allows the next higher layer to request that the device attempts to transmit a data frame on a dedicated link path.

The semantics of this primitive are:

```
DLC-DATA-DLINK.request         {
                               LinkId,
                               DstAddress,
                               DlsduLength,
                               Dlsdu,
                               DlsduHandle,
                               ACKEnable,
                               SecurityEnable
                               }
```

The primitive parameters are defined in Table 7.34.

Table 7.34 — DLC-DATA-DLINK.request parameters

Name	Type	Valid range	Description
LinkId	Integer	0x00 – 0xff	The link identifier.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the destination device.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The number of octets comprising the DLSDU to be transferred.
DlsduHandle	Integer	0x00 – 0xff	The handle of the DLSDU to be transmitted by the DLC sublayer entity.
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.

7.3.3.2 DLC-DATA-DLINK.indication

This primitive allows the next higher layer to be notified that the device receives a data frame on a dedicated link path.

The semantics of this primitive are:

```
DLC-DATA-DLINK.indication          {
                                   LinkId,
                                   DstAddress,
                                   SrcAddress,
                                   DlsduLength,
                                   Dlsdu,
                                   RxTime,
                                   ACKEnable,
                                   SecurityEnable
                                   }
```

The primitive parameters are defined in Table 7.35.

Table 7.35 — DLC-DATA-DLINK.indication parameters

Name	Type	Valid range	Description
LinkId	Integer	0x00 – 0xff	The link identifier.
DstAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the destination device.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the source device.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMin-HeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The number of octets comprising the DLSDU to be transferred.
RxTime	Integer	0x000000 – 0xffffff	The time, in symbols, at which the data were received, as described in 5.1.4.1 of the IEEE 802.15.4.
ACKEnable	Boolean	TRUE or FALSE	Indicates whether the ACK is enabled. If set to TRUE, it is enabled.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.

7.3.3.3 DLC-DATA-DLINK.response

This primitive allows the next higher layer to initiate a response to the device to transmit a data frame on dedicated link path with flow control. This primitive is valid for the TYPE_6 data operation and is required to setup a bi-directional link path before using this primitive.

The semantics of this primitive are:

```
DLC-DATA-DLINK.response          {
                                 LinkId,
                                 SrcAddress,
                                 Status
                                 }
```

The primitive parameters are defined in Table 7.36.

Table 7.36 — DLC-DATA-DLINK.response parameters

Name	Type	Valid range	Description
LinkId	Integer	0x00 – 0xff	The link identifier.
SrcAddress	Integer	0x0000 – 0xfffd	The 16-bit address of the source device.
Status	Status	INVALID_REQUEST, or any status value returned from the MCPS-DATA.confirm primitive	The result of the attempt to forward the data frame.

7.3.3.4 DLC-DATA-DLINK.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to transmit a data frame on a dedicated link path.

The semantics of this primitive are:

```
DLC- DATA-DLINK.confirm          {
                                 LinkId,
                                 Status
                                 }
```

The primitive parameters are defined in Table 7.37.

Table 7.37 — DLC-DATA-DLINK.confirm parameters

Name	Type	Valid range	Description
LinkId	Integer	0x00 – 0xff	The link identifier.
Status	Status	INVALID_REQUEST, or any status value returned from the MCPS-DATA.confirm primitive	The result of the attempt to transmit a data frame on a dedicated link path.

7.3.4 Primitives for data service through DLN sublayer

These primitives are used by the application layer of a device or a router to transport data.

7.3.4.1 DLN-DATA.request

This primitive allows the next higher layer of the DLN sublayer to request that the device attempts to transmit a data frame.

The semantics of this primitive are:

```
DLN-DATA.request                 {
                                 TxMode,
                                 DstAddr,
                                 DlsduLength,
                                 Dlsdu,
                                 DlsduHandle,
                                 SecurityEnable
                                 }
```

The primitive parameters are defined in Table 7.38.

Table 7.38 — DLN-DATA.request parameters

Name	Type	Valid range	Description
TxMode	Enumeration	TYPE_1, TYPE_2, TYPE_3, TYPE_4, TYPE_5, TYPE_6	The type of operation for data communication between DLC-SAPs. When the TxMode parameter is set to TYPE_1, a data frame shall be transmitted or received with the Type 1 operation (see clause 6.6.2).
DstAddr	Integer	0x0000 – 0xfffd	The 16-bit address of the device to which the DLSDU is sent.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The set of octets comprising the DLSDU to be transferred.
DlsduHandle	Integer	0x00 – 0xff	The handle of the DLSDU to be transmitted by the DLN sublayer entity.
SecurityEnable	Boolean	TRUE or FALSE	Indicates whether the MAC sublayer security is enabled. If set to TRUE, the security level set at the MAC layer is enabled.

7.3.4.2 DLN-DATA.indication

This primitive allows the next higher layer to be notified that the device receives a data frame.

The semantics of this primitive are:

```
DLN-DATA.indication          {
                             TxMode,
                             SrcAddr,
                             DstAddr,
                             DlsduLength,
                             Dlsdu,
                             RxTime
                             }
```

The primitive parameters are defined in Table 7.39.

Table 7.39 — DLN-DATA.indication parameters

Name	Type	Valid range	Description
TxMode	Enumeration	TYPE_1, TYPE_2, TYPE_3, TYPE_4, TYPE_5, TYPE_6	The type of operation for data communication between DLC-SAPs. When the TxMode parameter is set to TYPE_1, a data frame shall be transmitted or received with the Type 1 operation (see clause 6.6.2).
SrcAddr	Integer	0x0000 – 0xfffd	The 16-bit address of the device from which the DLSDU is originated.
DstAddr	Integer	0x0000 – 0xfffd	The 16-bit address of the device to which the DLSDU is sent.
DlsduLength	Integer	*aMaxMACPayloadSize – cdlMinHeaderOverhead*	The number of octets comprising the DLSDU to be transferred.
Dlsdu	Set of Octets		The set of octets comprising the DLSDU to be transferred.

Table 7.39 *(continued)*

Name	Type	Valid range	Description
RxTime	Integer	0x000000 – 0xffffff	The time, in symbols, at which the data were received, as described in 5.1.4.1 of the IEEE 802.15.4.

7.3.4.3 DLN-DATA.confirm

This primitive allows the next higher layer to be notified of the result of the attempt to transmit a data frame.

The semantics of this primitive are:

```
DLN-DATA.confirm                {
                                DlsduHandle,
                                Status
                                }
```

The primitive parameters are defined in Table 7.40.

Table 7.40 — DLN-DATA.confirm parameters

Name	Type	Valid range	Description
DlsduHandle	Integer	0x00–0xff	The handle associated with the DLSDU being confirmed.
Status	Status	INVALID_REQUEST, or any status value returned from the MCPS-DATA.confirm primitive of the IEEE 802.15.4	The result of the attempt to transmit a data frame.

8 LPWMN frame formats

8.1 General link network frame format

This subclause specifies the format of the DSME MAC link network protocol data unit (DLPDU).

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

Unless otherwise specified in this Clause, all reserved bits shall be set to zero upon transmission and may be ignored upon receipt.

The general link network frame shall be formatted as illustrated in Figure 8.1.

Octets: 2	0/2	0/2	0/8	0/8	0/variable	0/variable	variable
Frame Control	Destination Address	Source Address	Destination IEEE Address	Source IEEE Address	Link Management subframe	Link Network Management subframe	Frame Payload
	Link Network Addressing fields						
Link Network Header							Link Network Payload

Figure 8.1 — General DSME MAC link network frame format

8.1.1 Frame Control field

The Frame Control field contains information defining the type of frame operation, addressing fields, and other control flags. The Frame Control field shall be formatted as illustrated in Figure 8.2.

Bits: 0-1	Bits: 2-4	5	6	7	8	9	10	11-15
Protocol Version	Frame Operation Type	Destination Address Flag	Source Address Flag	Destination Address Mode	Source Address Mode	Link Flag	Link Network Flag	Reserved

Figure 8.2 — Format of the Frame Control field

The Protocol Version subfield is 2 bits in length and is invariant in size and placement across all revisions of this standard. For this standard, the value of the protocol version is set to the value of the network constant *cdlProtocolVersion*.

The Frame Operation Type sub-field is 3 bits in length and shall be set to one of the non-reserved values listed in Table 8.1.

Table 8.1 — Values of the Frame Operation Type subfield

Frame Operation Type value	Frame Operation Type name
000	TYPE_1 operation
001	TYPE_2 operation
010	TYPE_3 operation
011	TYPE_4 operation
100	TYPE_5 operation
101	TYPE_6 operation
110	Reserved
111	Reserved

Destination and Source Address Flag subfields indicate whether the address subfield is used. The subfields should be set to zero when the address subfield is not used and should be set to one when the address subfield is used.

Destination and Source Address Mode subfields indicate whether the 64-bit extended addresses or the 16-bit short addresses are used in the Destination and Source Address fields. The subfields should be set to zero when 64-bit extended addresses are used and should be set to one when 16-bit short addresses are used.

Link flag subfield indicates whether Link Management subframe is used. The subfield should be set to one when Link Management subframe is used.

Link Network flag subfield indicates whether Link Network Management subframe is used. The subfield should be set to one when Link Network Management subframe is used.

8.1.2 Destination Address field

The Destination Address field, when present, specifies the address of the intended recipient of the frame. A value of 0xffff in this field shall represent the broadcast short address, which shall be accepted as a valid address by all devices currently listening to the channel.

This field shall be included in the link network frame only if the Destination Address Flag field is set to one and the Destination Address Mode field is set to one.

8.1.3 Source Address field

The Destination Address field, when present, specifies the address of the originator of the frame.

This field shall be included in the link network frame only if the Source Address Flag field is set to one and the Source Address Mode field is set to one.

8.1.4 Destination IEEE Address field

The Destination IEEE Address field, when present, specifies the 64-bit IEEE address of the intended recipient of the frame.

This field shall be included in the link network frame only if the Destination Address Flag field is set to one and the Destination Address Mode field is set to zero.

8.1.5 Source IEEE Address field

The Source IEEE Address field, when present, specifies the 64-bit IEEE address of the originator of the frame.

This field shall be included in the link network frame only if the Source Address Flag field is set to one and the Source Address Mode field is set to zero.

8.1.6 Link Management subframe field

The Link Management subframe field, when present, specifies the Link Management command frames (see 8.3).

8.1.7 Link Network Management subframe field

The Link Network Management subframe field, when present, specifies the Link Network Management command frames (see 8.4).

8.1.8 Frame Payload field

The Frame Payload field has a variable length and contains the sequence of octets that the next higher layer has requested to transmit.

8.2 Data frames

The data frame shall be formatted as illustrated in Figure 8.3.

Octets: 2	0/2	0/2	0/8	0/8	0/variable	0/variable	variable
Frame Control	Destination Address	Source Address	Destination IEEE Address	Source IEEE Address	Link Management subframe	Link Network Management subframe	Data Payload
	Link Network Addressing fields						
Link Network Header							Link Network Payload

Figure 8.3 — Data frame format

The Link Management subframe or the Link Network Management subframe can be accompanied by the data frame.

The data payload field shall contain the sequence of octets that the next higher layer has requested to transmit.

8.3 Link management command frames

The Link management command frame shall be formatted as illustrated in Figure 8.4.

Octets: 2	0/2	0/2	0/8	0/8	0/variable	0/variable
Frame Control	Destination Address	Source Address	Destination IEEE Address	Source IEEE Address	Link Management subframe	Link Network Management subframe
	Link Network Addressing fields					
Link Network Header						

Figure 8.4 — Link management frame format

The Link Network Management subframe can be accompanied by the link management command frame.

The Link Management subframe shall be formatted as illustrated in Figure 8.5.

Bits: 0-2	Bits: 3-7	Octets: 1	Octets: Variable
Link Management Command Type	Sequence Number	Length of Link Management Command	Link Management Command Payload

Figure 8.5 — Link Management subframe format

8.3.1 Link Management Command Frame Identifier field

The Link Management Command Frame Identifier field is 3 bits in length and shall be set to one of the non-reserved values listed in Table 8.2.

Table 8.2 — Values of the Link Management Command Type field

Link Management Command Type value	Command Type name
000	SETUP_REQ
001	REL_REQ
010	HELLO_REQ
011	Reserved
100	SETUP_RESP
101	REL_RESP
110	HELLO_RESP
111	Reserved

8.3.2 Sequence Number field

The Sequence Number field specifies the sequence identifier for the frame.

The sequence number value shall be incremented by 1 with each new frame transmitted. The values of the source address, link management command type, and sequence number fields of a frame, taken as a pair, may be used to uniquely identify a frame within the constraints imposed by the sequence number's one-octet range.

8.3.3 Length of Link Management Command field

The Length of Link Management Command field is 8 bits in length and shall specify the length of the Link Management Command Payload.

8.3.4 Link Management Command Payload field

The Link Management Command Frame Payload field shall contain one of the link setup request command payload, link setup response command payload, link release request command payload, link release response command payload, hello request command payload, and hello response command payload.

8.3.4.1 Link setup request command payload

The link setup request command payload shall be formatted as illustrated in Figure 8.6.

Octets: 1	Octets: 2	Octets: 2	Octets: 1
Link Type	Source Address	Destination Address	Number of Slot

Figure 8.6 — Link setup request command payload format

The Link Type field shall be set one of the link type: IN-SHARED (0x00), OUT-SHARED (0x01), BI-SHARED(0x02), IN-DEDICATED (0x03), OUT-DEDICATED (0x04), BI-DEDICATED(0x05), IN-PEER (0x06), OUT-PEER (0x07), BI-PEER (0x08).

The Number of Slot field shall contain the number of slots allocated sequentially to this link.

8.3.4.2 Link setup response command payload

The link setup request command payload shall be formatted as illustrated in Figure 8.7.

Octets: 1	Octets: 2	Octets: 2	Octets: 1	Octets: 1
Link Type	Source Address	Destination Address	Link ID	Status

Figure 8.7 — Link setup request command payload format

The Link Type field shall be set one of the link type: IN-SHARED (0x00), OUT-SHARED (0x01), BI-SHARED(0x02), IN-DEDICATED (0x03), OUT-DEDICATED (0x04), BI-DEDICATED(0x05), IN-PEER (0x06), OUT-PEER (0x07), BI-PEER (0x08).

The Link ID field shall contain the established link identifier.

The Status field shall contain the result of the link setup.

8.3.4.3 Link release request command payload

The link release request command payload shall be formatted as illustrated in Figure 8.8.

Octets: 1	Octets: 2	Octets: 2	Octets: 1
Link Type	Source Address	Destination Address	Link ID

Figure 8.8 — Link release request command payload format

The Link Type field shall be set one of the link type: IN-SHARED (0x00), OUT-SHARED (0x01), BI-SHARED(0x02), IN-DEDICATED (0x03), OUT-DEDICATED (0x04), BI-DEDICATED(0x05), IN-PEER (0x06), OUT-PEER (0x07), BI-PEER (0x08).

The Link ID field shall contain the established link identifier.

8.3.4.4 Link release response command payload

The link release response command payload shall be formatted as illustrated in Figure 8.9.

Octets: 1	Octets: 2	Octets: 2	Octets: 1	Octets: 1
Link Type	Source Address	Destination Address	Link ID	Status

Figure 8.9 — Link release response command payload format

The Link Type field shall be set one of the link type: IN-SHARED (0x00), OUT-SHARED (0x01), BI-SHARED(0x02), IN-DEDICATED (0x03), OUT-DEDICATED (0x04), BI-DEDICATED(0x05), IN-PEER (0x06), OUT-PEER (0x07), BI-PEER (0x08).

The Link ID field shall contain the established link identifier.

The Status field shall contain the result of the link release.

8.3.4.5 Link hello request command payload

The link hello request command payload shall be formatted as illustrated in Figure 8.10.

Octets: 1	Octets: 2	Octets: 2
Link ID	Source Address	Destination Address

Figure 8.10 — Link hello request command payload format

The Link ID field shall contain the established link identifier.

8.3.4.6 Link hello response command payload

The link hello response command payload shall be formatted as illustrated in Figure 8.11.

Octets: 1	Octets: 2	Octets: 2	Octets: 1
Link ID	Source Address	Destination Address	Status

Figure 8.11 — Link hello response command payload format

The Link ID field shall contain the established link identifier.

The Status field shall contain the result of the link hello.

8.4 Link network management command frames

The Link Network management command frame shall be formatted as illustrated in Figure 8.12.

<table>
<tr><td>Octets: 2</td><td>0/2</td><td>0/2</td><td>0/8</td><td>0/8</td><td>0/variable</td><td>0/variable</td></tr>
<tr><td rowspan="2">Frame Control</td><td>Destination Address</td><td>Source Address</td><td>Destination IEEE Address</td><td>Source IEEE Address</td><td rowspan="2">Link Management subframe</td><td rowspan="2">Link Network Management subframe</td></tr>
<tr><td colspan="4">Link Network Addressing fields</td></tr>
<tr><td colspan="7">Link Network Header</td></tr>
</table>

Figure 8.12 — Link network management command frame format

The Link Management subframe can be accompanied by the link network management command frame.

The Link Network Management subframe shall be formatted as illustrated in Figure 8.13.

Bits: 0-2	Bits: 3-7	Octets: 1	Octets: Variable
Link Network Management Command Type	Sequence Number	Length of Link Network Management Command Payload	Link Network Management Command Payload

Figure 8.13 — Link Network Management subframe format

The command payload field shall contain the sequence of octets that the next higher layer has requested the LPWMN layer to transmit.

8.4.1 Link Network Management Command Frame Identifier field

The Link Network Management Command Frame Identifier field is 3 bits in length and shall be set to one of the non-reserved values listed in Table 8.3.

Table 8.3 — Values of the Link Network Management Command Type field

Link Network Management Command Type value	Command Type name
000	CLUSTER_REQ
001	UPDATE_REQ
010	LEAVE_REQ
011	FLOW_REQ
100	CLUSTER_RESP
101	UPDATE_RESP
110	LEAVE_RESP
111	FLOW_RESP

8.4.2 Sequence Number field

The Sequence Number field specifies the sequence identifier for the frame.

The sequence number value shall be incremented by 1 with each new frame transmitted. The values of the source address, link network management command type, and sequence number fields of a frame, taken as a pair, may be used to uniquely identify a frame within the constraints imposed by the sequence number's one-octet range.

8.4.3 Length of Link Network Management Command Frame field

The Length of Link Network Management Command Frame field is 8 bits in length and shall specify the length of the Link Network Management Command Frame Payload.

8.4.4 Link Network Management Command Payload field

The Link Network Management Command Payload field shall contain one of the cluster formation request command payload, cluster formation response command payload, route update request command payload, route update response command payload, leave request command payload, leave response command payload, flow control request command payload, and flow control response command payload.

8.4.4.1 Cluster formation request command payload

The Destination Address Flag subfields of the Frame Control field of the link management command frame is set to one.

The cluster formation request command payload shall be formatted as illustrated in Figure 8.14.

Octets: 1	Octets: 1	Octets: 1
Max Depth	Max Children	Max Router

Figure 8.14 — Cluster formation request command payload format

The Max Depth field shall contain the *dlMaxDepth* attribute of the DLIB. The Max Children field shall contain the *dlMaxChildren* attribute of the DLIB. The Max Router field shall contain the *dlMaxRouter* attribute of the DLIB.

8.4.4.2 Cluster formation response command payload

The Destination Address Flag subfields of the Frame Control field of the link management command frame is set to one.

The cluster formation response command payload shall be formatted as illustrated in Figure 8.15.

Octets: 1	Octets: 2
Length of Cluster Identifier Space	Cluster Identifier

Figure 8.15 — Cluster formation response command payload format

The Length of Cluster Identifier field shall contain the *dlClusterAddrSpcae* attribute of the DLIB.

The Cluster Identifier field shall contain the assigned cluster identifier to the device requesting cluster formation.

8.4.4.3 Route update request command payload

The Destination Address Flag subfields of the Frame Control field of the link management command frame is set to one.

The route update request command payload shall be formatted as illustrated in Figure 8.16.

Octets: 1	Octets: 2
Route Update Request Type	Router Address

Figure 8.16 — Route update request command payload format

The Route Update Request Type field shall be set one of the route update request: full cluster connectivity matrix (0x00), updated cluster connectivity matrix (0x01), inner cluster connectivity matrix (0x02), outer cluster connectivity matrix (0x03), route update report (0x04), search mesh route (0x05), route cost report (0x06).

The Router Address field is valid only when the specified device is designated. If not, it contains 0x0000.

8.4.4.4 Route update response command payload

The Destination Address Flag subfields of the Frame Control field of the link management command frame is set to one.

The route update response command payload shall be formatted as illustrated in Figure 8.17.

Octets: 1	Octets: 2	Octets: 0/1	0/Variable
Route Update Response Type	Router Address	Number of Entry of Routing Information Base	Routing Information Base

Figure 8.17 — Route update response command payload format

The Route Update Response Type field shall be set one of the route update response: full cluster connectivity matrix (0x80), updated cluster connectivity matrix (0x81), inner cluster connectivity matrix (0x82), outer cluster connectivity matrix (0x83), route update report (0x84), search mesh route (0x85), route cost report (0x86), end of full cluster connectivity matrix (0xf0), end of updated cluster connectivity matrix (0xf1), end of inner cluster connectivity matrix (0xf2), end of outer cluster connectivity matrix (0xf3), end of search mesh route (0xf5), end of route cost report (0xf6).

The Router Address field is valid only when the specified device is designated. If not, it contains 0x0000.

The Number of Entry of Routing Information Base field shall contain the number of the cluster table entry or cluster connectivity matrix entry or route table entry depending on the Route Update Response Type field: cluster table entry (0x80, 0xf0), cluster connectivity matrix (0x81, 0xf1, 0x82, 0xf2, 0x83, 0xf3, 0x84, 0x85, 0xf5), route table entry (0x86, 0xf6).

The Routing Information Base field shall contain the cluster table entry or cluster connectivity matrix entry or route table entry depending on the Route Update Response Type field: cluster table entry (0x80, 0xf0), cluster connectivity matrix (0x81, 0xf1, 0x82, 0xf2, 0x83, 0xf3, 0x84, 0x85, 0xf5), route table entry (0x86, 0xf6).

8.4.4.5 Flow control request command payload

The flow control request command payload shall be formatted as illustrated in Figure 8.18.

Octets: 1	Octets: 1	Octets: 1
Flow Control Command Type	Sender Send Sequence Number	Sender Receive Sequence Number

Figure 8.18 — Flow control request command payload format

